OPTIMIZATION ACROSS DISCIPLINES

A Transdisciplinary Field Guide to
Better Choices, Designs, Systems, and Works

OPTIMIZATION ACROSS DISCIPLINES

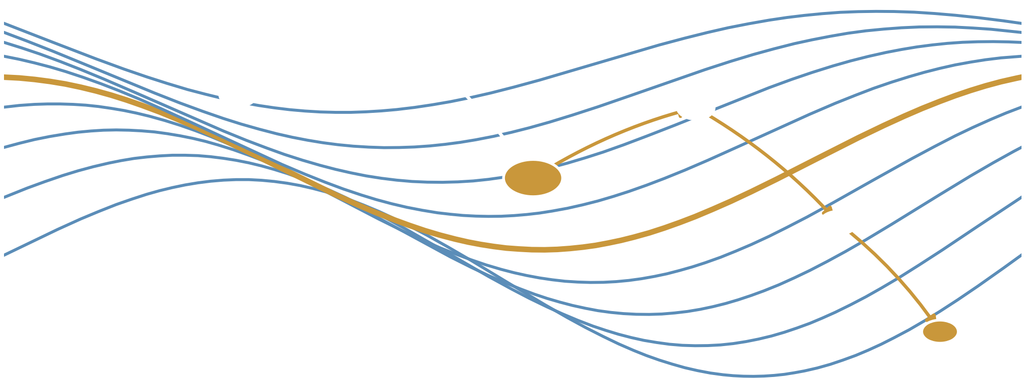
A Transdisciplinary Field Guide to Better Choices, Designs, Systems, and Works

Idea: Dmitry Vostokov

Author: GPT-5.6 Sol Max

Referee: Fable 5 Extra

Version 33 · 2026



Copyright

ISBN-13: 978-1-919135-00-7

Version 33 · 2026

The book provides general educational discussion. It is not a substitute for professional engineering, medical, legal, financial, safety, or regulatory judgment. Models, standards, legislation, guidance, and software practices must be checked in their current operational context.

About this book

Optimization is often introduced as a branch of mathematics: choose variables, write an objective function, impose constraints, and find a minimum or maximum. That description is correct, but it is smaller than the subject. Optimization is also a way of organizing action. Engineers trade mass against strength. Physicians trade benefit against harm under uncertainty. Computer scientists trade time against memory. Societies distribute scarce resources while disputing what fairness means. Artists work within and against constraints. Musicians balance continuity, surprise, playability, and expression. Writers optimize neither a single score nor a fixed endpoint, yet they repeatedly select, revise, compress, expand, and reorder in a structured space of alternatives.

This book develops a common language for those activities without pretending that they are identical. It moves from mathematical foundations to computer science, software, hardware, engineering, medicine, biology, economics, social science, the humanities, art, music, and literature. It also studies the limits of optimization: values that resist measurement, goals that change when measured, objectives that conflict, systems that adapt to the metric, and situations in which restraint, satisficing, care, exploration, or open-ended creation is wiser than maximization.

The result is both a map and a toolkit. The map shows how diverse disciplines instantiate a recurring structure. The toolkit explains how to formulate, solve, test, govern, and revise optimization problems. Equations are used where they clarify structure; prose and illustrations carry the conceptual load where formalism would conceal more than it reveals.

Scope and intellectual stance

No finite volume can literally enumerate every possible discipline. The stronger and more useful aim is generative coverage: to identify a framework from which optimization in a known or future field can be reconstructed. The book therefore combines broad domain chapters with a reusable seven-part schema:

1. the decision or design variables;
2. the space of feasible possibilities;
3. the objective, preference relation, or evaluation process;
4. the constraints and invariants;
5. uncertainty, dynamics, and other agents;
6. evidence, computation, and validation; and
7. values, stakeholders, governance, and revision.

An optimization problem is never only an objective function. It is an objective function embedded in a model, and a model embedded in a world.

Reference policy

The bibliography favors primary papers, standard textbooks, standards bodies, regulators, and authoritative institutional publications. Structural checks apply to every cited record; identifiers for high-use, corrected, and newly added records were checked against publisher, DOI, standards-body, or official catalog metadata. Uncertain or misdirected links are omitted rather than presented as verified. The audit snapshot is 22 August 2026; Appendix D records the method and its limits.

A note on equations

Displayed equations are genuine mathematical statements rather than ornamental notation. Each is introduced in words and interpreted after use. Symbols are local unless a chapter explicitly carries them forward. Readers may follow the conceptual path without deriving every formula, while technically trained readers can use the equations as compact specifications.

Preface: Optimization as a shared language

The word *optimal* is powerful because it compresses a comparison: among some possibilities, under some conditions, according to some criterion, one is no worse than the others. Every omitted phrase matters. Optimal *among what*? Under *which* conditions? According to *whose* criterion? With *what* information? For *how long*? At *what* cost of search? With *which* risks excluded from the model?

In mathematical optimization, the omissions are made explicit. A decision vector x is chosen from a feasible set $\mathcal{X}$ to minimize a function f:

$$x^* \in \arg\min_{x \in \mathcal{X}} f(x). \tag{1}$$

This is the seed from which much of the book grows. Yet the seed does not tell us who chose f, how $\mathcal{X}$ was constructed, whether measurements are reliable, whether other people respond strategically, or whether the problem changes while it is being solved. Those questions connect optimization to statistics, control, economics, ethics, politics, design, and interpretation.

Category theory supplies a useful metaphor. Each discipline presents objects and morphisms of its own: programs and transformations, structures and load paths, patients and interventions, melodies and voice leadings, narratives and revisions. A transdisciplinary theory seeks functors between these domains and a common category of decision problems. But translation is rarely an equivalence. A cost in one field may map only approximately to a cost in another; a feasible transformation may preserve some structure while destroying meaning elsewhere. The common language is valuable precisely when its information loss is kept visible.

Optimization is therefore treated here as a *discipline of explicit preference under constraint*, not as a doctrine that everything should be maximized. The central practical skill is not merely solving the mathematical program. It is deciding what problem is worth formalizing, recognizing what the formalization omits, selecting a method appropriate to the structure, and keeping the resulting system corrigible.

How to read the book

Choose a route through the twenty chapters: **bird's-eye**—Chapters 1–3, 17, 20, then selected cases in Chapters 18–19; **mathematical**—Chapters 1–7, then selected domains; **technical systems**—Chapters 1–3, 5, 7–11, 17–20; **medicine and society**—Chapters 1–3, 6, 12–14, 17–20; **creative**—Chapters 1–3, 6–7, 14–16, 17–20; or category theory—Chapters 1–3, 7, 10, 14, 18–20. For mathematics first, use Appendix F: Units 1–7, 8–22, 23–35, and 36–42.

Each chapter ends with a compact *translation lens* and field checklist. The lens names what can travel; the checklist names what must be re-established locally.

Contents

List of figures

List of tables

Bird's-eye view

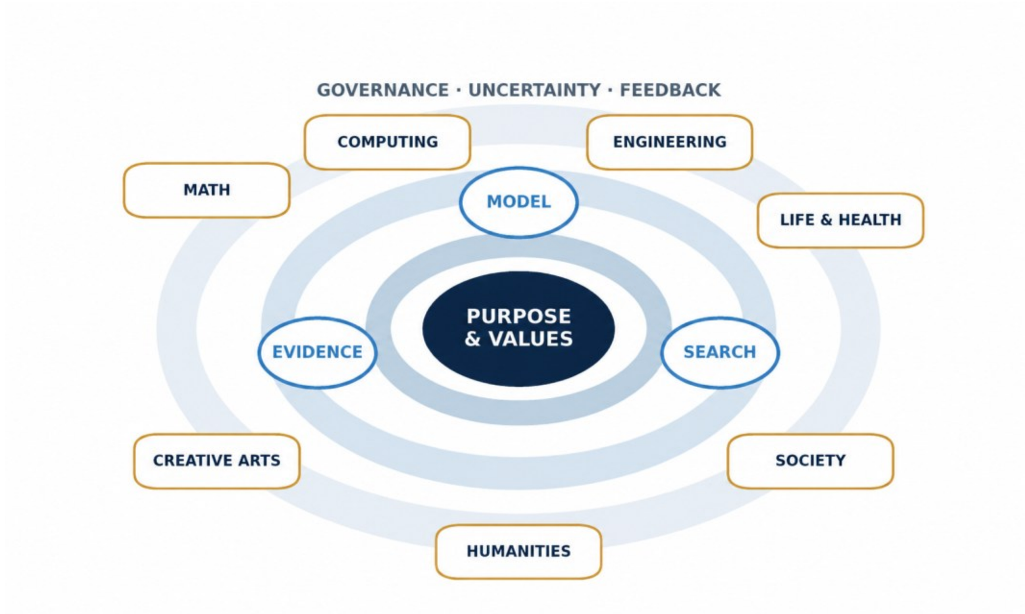


Figure 1. Bird's-eye view: purpose and values at center; model, evidence, search, and seven domains around them; governance, uncertainty, and feedback spanning the map.

Figure 1 separates the invariant grammar from the domain-specific objects around it: purpose and values remain central, while models, evidence, search, governance, and feedback mediate every application.

Optimization begins before a solver and continues after a solution. A situation in the world is interpreted as a decision problem. The model identifies variables, possible actions, objectives, constraints, uncertainties, affected parties, and a time horizon. A search process produces candidates. A selection rule chooses or recommends one. Implementation changes the world, generates evidence, and may alter the preferences and constraints that defined the problem. Evaluation then closes the loop.

The full cycle can be written schematically as

$$W_t \xrightarrow{M} P_t \xrightarrow{A} S_t \xrightarrow{I} W_{t+1} \xrightarrow{E} \widehat{W}_{t+1}, \tag{2}$$

where W_t is the world at time t, M is modeling, P_t is the formalized problem, A is an algorithm or deliberative procedure, S_t is a selected solution, I is implementation, and E is observation or evaluation. The next problem is not necessarily the old problem with new data: implementation may change incentives, institutions, technologies, populations, meanings, and available actions.

The seven questions

The bird's-eye view can be turned into seven questions that apply in every discipline.

1. What can change?

These are the decision variables. They may be continuous quantities such as pressure, dosage, or portfolio weights; discrete choices such as routes, treatments, words, or circuit placements; functions such as a control policy; distributions such as a transport plan; or structured artifacts such as programs, buildings, melodies, and institutions.

2. What counts as admissible?

The feasible set encodes physical laws, budgets, safety margins, logical consistency, contracts, rights, style rules, manufacturability, deadlines, and other invariants. Constraints are not merely obstacles. They define the grammar of possible solutions and frequently make creativity intelligible.

3. What is better?

A scalar objective provides a total ordering only after distinct concerns have been commensurated. Many real problems instead use multiple objectives, partial orders, lexicographic priorities, thresholds, vetoes, or deliberation. The question is not always "What is the maximum?" It may be "Which alternatives are nondominated?", "Which option is acceptable to all?", or "Which design opens the best future possibilities?"

4. What is unknown or changing?

Uncertainty may concern parameters, observations, models, other agents, future conditions, or the objective itself. Robust, stochastic, Bayesian, adaptive, and distributionally robust optimization answer different versions of this question. Dynamics add state and path dependence; feedback makes the optimizer part of the system being optimized.

5. How will candidates be generated and compared?

The search may use proof, calculus, enumeration, relaxation, decomposition, simulation, heuristics, human judgment, evolutionary variation, negotiation, or hybrid methods. Computational cost is part of the problem. An exact answer that arrives after the decision deadline is not operationally optimal.

6. How will success be tested?

Validation distinguishes optimization from wishful scoring. It includes held-out data, sensitivity analysis, optimality certificates, simulation, experiments, stress tests, safety cases, audits, replication, qualitative interpretation, and post-deployment monitoring.

7. Who defines, bears, and may revise the objective?

This is the governance question. Benefits and harms are distributed. Metrics create incentives. People adapt. Rights may constrain trade-offs. Some values should be represented as hard constraints or decision rights rather than prices. An optimization system needs an appeals path, stopping conditions, and a mechanism for changing its own formulation.

The master form

A broad mathematical template is

$$\min_{x,\pi} \rho(F(x,\pi,\xi);\ \theta) \tag{3}$$

subject to

$$g_i(x,\pi,\xi) \le 0, \qquad i = 1, \dots, m, \tag{4}$$

$$h_j(x,\pi,\xi) = 0, \qquad j = 1, \dots, p, \tag{5}$$

$$x \in \mathcal{X}, \qquad \pi \in \Pi, \qquad R_k(x,\pi) \ge r_k. \tag{6}$$

Here x is a design or allocation, π is a policy, ξ represents uncertainty, F is a vector of consequences, ρ aggregates or orders those consequences using parameters θ, g_i and h_j encode inequality and equality constraints, and $R_k \ge r_k$ represents rights, resilience, reliability, or other non-negotiable requirements. Different disciplines instantiate the same form differently; some reject the aggregation ρ and preserve a set of incomparable alternatives.

Domain map

The book organizes disciplines into seven large families, with porous borders. The six parts organize the reading sequence; the seven families classify domains. These are different decompositions of the same field guide.

Table 1. Seven domain families and their characteristic optimization objects.

Family	Typical variables	Typical objectives	Characteristic difficulty
Mathematics	vectors, sets, measures, graphs, functions	optimum, bound, convergence, proof	existence, geometry, complexity
Computing	representations, programs, schedules, policies	correctness, runtime, memory, service	scale, discrete structure, feedback
Engineering	geometry, materials, controls, processes	performance, cost, safety, energy	coupled physics and manufacturability

Family	Typical variables	Typical objectives	Characteristic difficulty
Life and health	interventions, allocations, pathways, protocols	health, survival, function, discovery	heterogeneity, uncertainty, ethics
Economy and society	prices, rules, matches, budgets, policies	welfare, stability, equity, growth	strategic response and contested values
Humanities	corpora, classifications, narratives, arguments	coherence, insight, preservation, reach	meaning resists scalarization
Creative arts	form, sequence, material, gesture, style	expression, novelty, fit, experience	objectives emerge during making

Table 1 is a translation aid: similar mathematical forms sit beside different variables, objectives, and characteristic difficulties.

The families share methods but not identities. A Pareto front in aircraft design and a set of editorial alternatives are formally analogous only to the degree that their orderings and constraints have been made explicit. The analogy is a morphism, not an isomorphism.

The optimization lifecycle

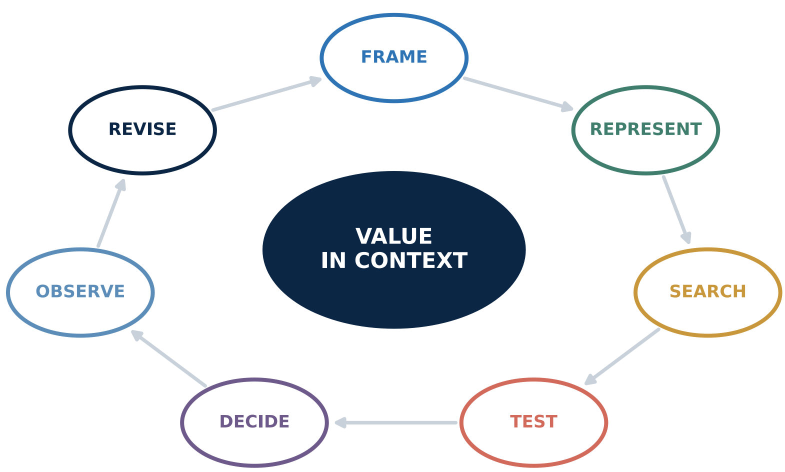


Figure 2. Seven-stage optimization lifecycle—frame, represent, search, test, decide, observe, revise—around value in context.

Figure 2 should be read as a control loop rather than a waterfall: observation can reopen framing, and governance spans every stage rather than appearing only at approval.

The lifecycle is deliberately circular. In high-stakes settings, every stage should expose artifacts that can be inspected: a problem statement, data sheet, objective specification, constraint register, solver configuration, optimality or approximation evidence, sensitivity report, decision rationale, implementation plan, monitoring metrics, and revision trigger. The artifact trail is the operational memory of the optimization process.

Ten recurring tensions

Across disciplines, ten tensions recur: **single objective versus plural values; local improvement versus global performance; short versus long horizon; efficiency versus resilience; average outcome versus tail risk; precision versus robustness; central coordination versus distributed autonomy; exploitation versus exploration; measurability versus meaning;** and **optimization power versus accountability.**

The book develops tools for navigating these tensions rather than declaring one side universally correct.

Part I · The Grammar of Optimization

Chapter 1 · Purpose, alternatives, and the grammar of optimization

Purpose, alternatives, and decision systems

Optimization is disciplined comparison among alternatives. Its minimal form contains a set of candidates $\mathcal{X}$ and a rule for preferring one candidate to another. When the rule is represented by a real-valued function f, minimization means finding the set

$$\arg\min_{x\in\mathcal{X}} f(x) = \{x \in \mathcal{X}: f(x) \leq f(y) \text{ for all } y \in \mathcal{X}\}. \tag{7}$$

The notation distinguishes the *optimal value* $f^* = \inf_{x\in\mathcal{X}} f(x)$ from the *optimizer* x^*. The distinction matters: a best value may be approached but never attained; many different candidates may attain it; or the infimum may be $-\infty$. Existence, uniqueness, and computability are separate questions.

In practice, optimization has at least five layers.

1. **Ontological layer.** What entities and actions exist in the model?
2. **Evaluative layer.** What counts as better, and for whom?
3. **Mathematical layer.** What variables, objectives, constraints, distributions, and dynamics represent the situation?
4. **Computational layer.** What algorithm can produce a useful answer with available time, data, and hardware?
5. **Institutional layer.** Who authorizes the decision, bears its effects, monitors it, and may appeal or revise it?

Mathematical optimization concentrates on the middle layers, but real success depends on all five. A perfectly solved wrong problem is a failure with an optimality certificate.

Improvement, optimum, and satisficing

Optimization is often conflated with improvement. Improvement compares a new candidate with a reference candidate; global optimality compares it with every feasible candidate. Local optimality compares it only with a neighborhood. These distinctions explain why an intervention can be beneficial without being best, and why a solver can converge while the design remains globally poor.

Exact optimization is not always the rational target. Search has a cost. Let $c(a, t)$ be the cost of running algorithm a for time t, and let $L(x)$ be the loss of the returned decision. The operational problem is closer to

$$\min_{a,t} \mathbb{E}\big[L\big(x_{a,t}\big)\big] + c(a, t), \tag{8}$$

than to minimizing L alone. This is why approximation algorithms, anytime methods, heuristics, and satisficing are principled tools rather than merely inferior substitutes. Simon's bounded rationality treats decision procedures as adapted to limited information and computation, not as defective copies of omniscient maximization (Simon 1955, 1956).

Optimization as compression

An objective function compresses many observations and values into a comparison. Compression makes action possible, but it discards information. Weighted sums, scores, rankings, utility functions, and dashboards are lossy encodings. The central discipline is to track what was compressed away. Multiobjective methods retain more structure by returning a Pareto set; robust methods retain uncertainty sets; distributional methods retain probability; qualitative review can retain meanings that resist numeric encoding.

Optimization as search and proof

Some methods primarily search; others primarily certify. Enumeration can prove optimality by exhaustion. Convex duality can produce a lower bound that meets an achieved value. Branch-and-bound alternates search with bounds. Metaheuristics can explore irregular landscapes effectively but usually do not prove global optimality. The correct claim must match the evidence: *best found*, *locally stationary*, *ε-optimal*, *globally optimal under the model*, and *preferred after deliberation* are different predicates.

The compact expression $x^* \in \arg\min_{x \in \mathcal{F}} f(x)$ is a useful object, but it is not yet a complete decision. It omits the origin of the alternatives, the authority to act, the reliability of observations, the cost of implementation, and the rule for reopening the problem. A field guide therefore treats an optimization project as a tuple of connected artifacts: a purpose statement, a decision space, a consequence model, an ordering or deliberative procedure, a search method, an evidence plan, an owner, and a revision rule. The mathematical program is a central morphism in this system, not the whole category.

The distinction can be seen in a simple operational example. Suppose a service team must choose an alert threshold τ. Lowering τ detects more incidents but sends more false alarms; raising it reduces workload but misses some failures. The candidate set is not merely the real line. Operationally admissible thresholds depend on sensor calibration, staffing, escalation time, user impact, and the authority of an on-call engineer to suppress alerts. The consequence of τ is a vector containing detection probability, false-alert rate, response delay, interruption cost, and residual risk. A scalar score may be useful for search, yet the final decision may impose a hard miss-rate limit and ask engineers to choose among the remaining Pareto-efficient thresholds.

This example reveals four distinct comparisons:

1. **Technical comparison:** which threshold performs better on representative data?
2. **Operational comparison:** which threshold is sustainable under actual staffing and incident load?
3. **Risk comparison:** which threshold avoids intolerable tail outcomes?
4. **Institutional comparison:** which threshold is authorized, contestable, and reversible?

Collapsing the four too early creates false precision. Keeping them separate forever can prevent action. Good optimization designs a sequence of lawful compressions: first preserve the consequence vector, then eliminate inadmissible options, then use a transparent preference rule, and finally record the judgment that selected among near-equivalent candidates.

Different projects should deliberately choose different output types.

- A **point** is appropriate when implementation needs one setting and uncertainty is small enough.
- A **set** is appropriate when several alternatives are nondominated or evidence cannot distinguish them.
- A **policy** is appropriate when future observations should change action.
- A **ranking** is appropriate when a human process allocates scarce review, provided the ranking is not silently converted into entitlement.
- A **certificate** is appropriate when the claim is feasibility, a bound, or a verified invariant.
- A **counterexample** is appropriate when search reveals that a proposed requirement set is inconsistent.
- A **deliberation surface** is appropriate when the central uncertainty concerns values or authority rather than consequences.

Naming the output type prevents a common escalation: a model built to screen or explore becomes an automatic decision because its score happens to be sortable. Authorization should be attached to the type. A screening model can invite review; it cannot acquire power through interface convenience.

Objectives, constraints, values, and authority

The decisive optimization move is often made before any equation: drawing the system boundary. A hospital scheduling model may optimize operating-room utilization while excluding clinician fatigue and downstream bed congestion. A compiler may reduce execution time while increasing energy or binary size. A literary recommender may maximize engagement while narrowing discovery. Boundary choices determine which consequences become variables, constraints, externalities, or silence.

From consequences to objectives

Suppose a decision x produces a consequence vector

$$F(x) = \big(f_1(x), f_2(x), \dots, f_k(x)\big). \tag{9}$$

A scalar objective $J(x) = \rho\big(F(x)\big)$ requires an aggregation rule ρ. The familiar weighted sum

$$J(x) = \textstyle\sum_{i=1}^{k} w_i\, f_i(x), \qquad w_i \geq 0, \tag{10}$$

looks neutral but makes substantive commitments. Units must be normalized; weights encode trade-offs; nonlinearities and thresholds may be lost; and some Pareto-efficient points cannot be recovered by weighted sums when the attainable objective set is nonconvex. Alternatives include lexicographic priority, goal programming, reference-point methods, outranking, constrained optimization, and deliberative selection from a Pareto set.

Values can occupy different formal roles.

- A **target** invites maximization or minimization.
- A **constraint** establishes an admissibility boundary.
- A **penalty** permits violation at a price.
- A **veto** removes options regardless of other gains.
- A **right** may restrict which trade-offs are legitimate at all.
- A **process requirement** specifies participation, explanation, or review rather than an outcome score.

Treating every value as a commensurable objective is a modeling choice, not a mathematical necessity. Rawlsian priority, Sen's capability approach, and social-choice theory show why aggregation can conflict with plurality, liberty, and interpersonal incomparability (Rawls 1971; Sen 1985; Arrow 1951).

Why value types are not merely large weights

The strongest reductionist objection is straightforward. If a right outranks ordinary welfare, give it lexicographic priority; if violation is merely very bad, attach a sufficiently large penalty. On this view, targets, constraints, vetoes, rights, and process requirements are convenient labels for one sufficiently expressive preference ordering.

That reduction is not automatic. Let $v(x) \geq 0$ measure violation and M be finite. Without a proved bound or an exact-penalty theorem, the penalized and constrained problems need not select the same decision:

$$\arg\min_{x \in X} f(x) + Mv(x) \not\equiv \arg\min_{x \in X, v(x)=0} f(x)$$

For any finite M, search may accept a violation when the modeled improvement in f exceeds its price. Making M enormous can create ill-conditioning while concealing, rather than eliminating, the substantive choice of how large is large enough. Exact-penalty results are useful precisely because they require hypotheses; they are not a general license to convert every boundary into a weight (Nocedal and Wright 2006).

A penalty is a price, and optimization is unusually effective at discovering arbitrage against prices. A hard constraint removes inadmissible options; a veto identifies an actor or rule that can block them; a right restricts which exchanges may be offered; and a process requirement governs how a decision becomes legitimate. These roles differ even when a numerical encoding happens to reproduce one ranking on one dataset.

Type also carries authority and evidence. A modeling team may tune a performance weight. It may not use the same interface to revise a legal duty, a safety case, a patient's consent, or a community's jurisdiction. Two representations can induce the same ordering today and still differ institutionally because different people may change them, on different evidence, with different notice and appeal.

Reduction is defensible when the same authority controls the ordering, relevant scales are bounded or exactness conditions are proved, lexicographic and tie-breaking rules are explicit, and procedural protections survive the encoding. Otherwise preserve the value type at the interface. The distinction is therefore not metaphysical decoration; it is a claim about admissibility, substitution, authority, and revision.

Constraints create form

Constraints are often described negatively, yet they also make design possible. A sonnet, a bridge, a chip floorplan, and a clinical protocol become recognizable through structured limits. Equality constraints define manifolds or conservation laws. Inequalities define resources and safety margins. Logical constraints encode compatibility. Chance constraints control the probability of violation. Temporal constraints govern order and deadlines.

For a differentiable program

$$\min_x f(x) \quad \text{subject to} \quad g_i(x) \le 0,\; h_j(x) = 0, \tag{11}$$

the Lagrangian is

$$\mathcal{L}(x, \lambda, \nu) = f(x) + \textstyle\sum_{i=1}^{m} \lambda_i\, g_i(x) + \sum_{j=1}^{p} \nu_j\, h_j(x). \tag{12}$$

The multipliers λ_i and ν_j quantify local sensitivity to constraints under regularity conditions. Economists interpret them as shadow prices; engineers interpret them as marginal value or burden; governance should resist the temptation to interpret every multiplier as a legitimate market price.

The feasible set is socially produced

In technical textbooks, $\mathcal{X}$ is given. In institutions, it is negotiated and historically contingent. Laws, standards, infrastructure, procurement rules, professional norms, accessibility requirements, and political power shape what appears feasible. Expanding the feasible set can dominate solver improvement. A new material, diagnostic test, public process, or legal rule can create possibilities that no algorithm could discover inside the earlier formulation.

Reframing as a first-class operation

Optimization practice should maintain a *framing register*: the system boundary, stakeholders, excluded effects, time horizon, baseline, counterfactual, and reasons for choosing each objective and constraint. When evidence contradicts assumptions, the correct response may be to reframe rather than retune. Reframing is a morphism from one problem object to another; it can preserve some structures while deliberately changing others.

Optimization is especially dangerous when it silently converts an end into a metric. Health becomes a utilization score, learning becomes test performance, scholarship becomes citation count, friendship becomes engagement, and art becomes predicted attention. The issue is not that numbers are inherently corrupting. It is that a measurement morphism preserves selected structure and has a kernel: differences in the world that map to the same score. Search pressure tends to discover that kernel and exploit it.

Practical reasoning therefore precedes technical reasoning. Aristotle distinguishes deliberation about means from the formation and cultivation of ends; contemporary capability and justice traditions similarly resist treating every good as a commensurable quantity (Aristotle 2002; Sen 1985; Rawls 1971; Nussbaum 2011). A defensible project asks whether optimization is appropriate at all. It may be appropriate to optimize the scheduling of a care service while refusing to rank the intrinsic worth of its patients. It may be appropriate to optimize acoustic treatment while refusing to optimize a composition to an engagement model. Refusal is not an absence of rigor; it is a boundary judgment about which morphisms are legitimate.

Draw a table with decisions as rows and roles as columns: propose, model, validate, approve, execute, override, investigate, appeal, and retire. Assign each consequential act. Blank cells reveal orphaned responsibility; a single role filling every cell reveals lack of independent challenge.

Authority should match response time. An operator can stop an unsafe action immediately, while permanent objective changes require broader review. A user may correct personal data without owning the global model. A community body may hold consent or veto rights that are not reducible to advisory preference.

Publish the map proportionally to stakes. It turns “the system decided” into named human and institutional acts and makes escalation testable before an incident.

Search cost, baselines, and levels of intervention

Optimization itself consumes resources. Modeling delays action; data collection burdens people; simulation consumes energy; solver tuning uses expertise; exact search can outlive the decision horizon. A more complete operational objective is

$$\min_{a,t,x} \mathbb{E}[L(x;\xi) \mid a,t] + C_{\text{search}}(a,t) + C_{\text{delay}}(t) + C_{\text{change}}(x,x_0), \tag{13}$$

where a is a method, t is the effort or stopping time, x_0 is the current practice, and ξ represents uncertain consequences. This formulation extends the simpler search-cost model in Equation (8) by adding delay and change costs. It explains why the most sophisticated solver is not automatically the best intervention. A fast transparent approximation can dominate an exact but delayed answer; an existing safe practice can dominate a fragile improvement whose transition cost is ignored. This is bounded rationality made explicit rather than treated as an embarrassment (Simon 1955, 1956).

The search cost also changes the appropriate guarantee. When consequences are low and decisions repeat frequently, empirical improvement may be sufficient. When a design is irreversible or safety-critical, feasibility proofs, worst-case bounds, independent verification, and conservative margins become part of the objective system. The required warrant should be specified before method selection, because a heuristic candidate cannot be retrofitted into a certificate after the fact.

An existing practice carries defects and accumulated knowledge. Its performance, workarounds, social meaning, and recovery paths form the baseline. A proposed optimum should be compared with the baseline under transition: migration effort, dual running, training, temporary risk, loss of tacit skill, and the ability to reverse.

The baseline also protects epistemic humility. If a complex method cannot beat a transparent rule under forward validation, the method has not earned deployment. If it beats the rule by less than measurement uncertainty or implementation cost, the correct decision may be to keep the rule and improve evidence. Optimization should make “no change” a real candidate.

Many disputes disappear when the level of optimization is named. **Operational optimization** chooses an action inside an accepted rule: route this vehicle, tune this controller, or assign this shift. **Design optimization** changes the artifact or process that creates operational choices: fleet composition, sensor layout, or workflow. **Constitutional optimization** changes who may decide, which values are admissible, and how the rule can be contested.

The levels are coupled. An operational model can reveal that the design leaves no safe action. A design can make a constitutional promise feasible. A constitutional limit can exclude an apparently efficient design. Yet authority and evidence differ. A dispatcher can reroute a vehicle and cannot redefine the right to service. An engineer can tune a controller and cannot waive a safety duty. A board can approve an objective and cannot make an invalid measurement true.

Write the decision level beside every variable. If a model silently promotes a fixed policy parameter into a decision variable, identify who has authority to change it. If it treats a political or ethical choice as data, restore it to deliberation. If it freezes a design that dominates operational performance, widen the feasible set.

The three levels also imply different cadence. Operations may update each minute, design each quarter, and constitutional purpose each year or after an incident. Faster loops should not overwrite slower commitments without an explicit escalation. Slower loops should receive summarized evidence from faster ones, including overrides and excluded demand.

This hierarchy turns “optimize the system” into a tractable question: which layer can change now, what remains fixed, and what observation justifies moving the boundary?

A decision charter in use

Return to the alert threshold. A team that starts with “minimize false alarms plus missed incidents” has already hidden four choices: what counts as an incident, whose interruption counts as cost, how severe events are weighted, and who accepts residual risk. A decision charter reopens them.

Purpose. Detect service degradation early enough to limit user harm while sustaining an effective on-call practice. **Decision.** Choose threshold, persistence window, routing, suppression, escalation, and review cadence—not threshold alone. **Boundary.** Include sensor, detection logic, paging service, engineer attention, escalation, user impact, remediation, and learning after the event. **Authority.** Reliability leadership approves operational policy; security and safety owners can impose separate alarms; on-call engineers can suppress a faulty signal; affected product teams can challenge repeated burden. **Evidence.** Historical replay, injected incidents, calibration, workload observation, and staged deployment. **Stop and reopen.** Roll back if severe misses, unmanageable load, or degraded response crosses a trigger; reopen when architecture, workload, staffing, or sensor behavior changes.

The charter changes the variables. Persistence and grouping can reduce nuisance alarms without raising the signal threshold. Better sensor calibration can improve both detection and workload. Staffing or automated remediation can change the feasible region. The original one-dimensional trade-off becomes a co-design problem.

It also changes validation. Randomly splitting historical alerts into train and test sets can leak incident families and preserve an obsolete operating regime. Replay should hold out later periods, entire incident types, and major architecture changes. Synthetic injection tests whether true faults propagate through sensing and paging. Shadow operation reveals how often the candidate policy would have paged, while interviews reveal whether pages are interpretable and actionable.

The final record need not claim a universal optimum. It can state: policy P met a hard severe-event detection requirement, reduced median nonactionable load against baseline P_0, remained feasible under staffing stress, and was chosen over a slightly quieter policy because independent review judged its tail miss risk unacceptable. This is a justified decision even when preference uncertainty prevents a single “true” objective.

Field checklist

- Name the decision maker, affected parties, and those with veto or appeal rights.
- Distinguish candidate generation, candidate comparison, and authorization to act.
- State whether the output is a point, a set, a policy, a ranking, or a deliberative aid.
- Price the costs of data, modeling, search, delay, transition, and monitoring.
- Match the claimed guarantee to the evidence actually produced.
- Record what the objective compresses away and which consequences remain non-negotiable.
- Define a stopping condition for search and a reopening condition for the formulation.

Translation lens. Across disciplines, optimization begins with alternatives and a comparison, but a complete translation must also preserve purpose, authority, evidence, and revision. The invariant is not the scalar objective; it is the justified relation between a choice and the world it is meant to improve.

Chapter 2 · Problem structure, taxonomy, and historical foundations

Taxonomy and structural fingerprint

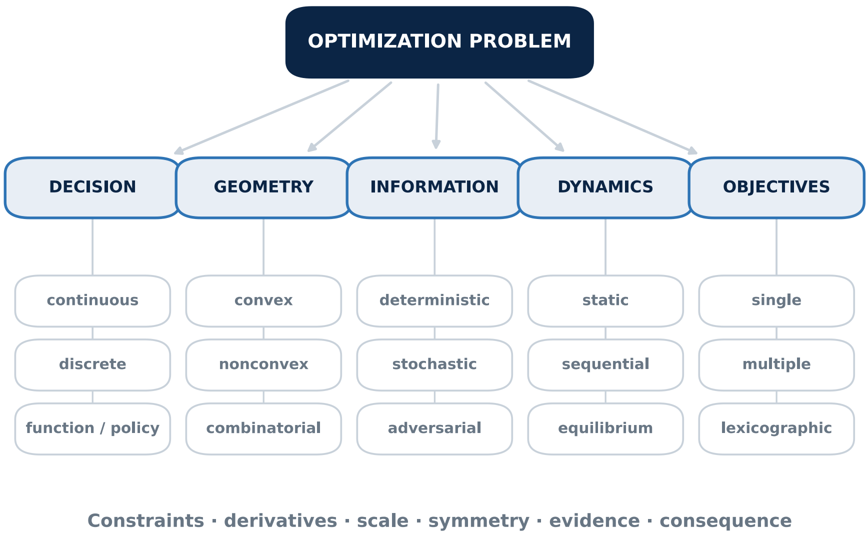


Figure 3. Structural fingerprint of an optimization problem across five axes: decision, geometry, information, dynamics, and objectives.

Figure 3 is a method-selection aid: two problems that share an objective may require different algorithms because their variables, information regimes, uncertainty, or guarantees differ.

Optimization problems can be classified along independent axes. The classification matters because algorithm choice should follow structure.

Variable and feasible-set structure

Variables may be continuous, integer, binary, categorical, graph-valued, sequence-valued, function-valued, or mixed. A continuous relaxation may provide bounds for a discrete problem. A topology or architecture search may require variable-dimensional representations. Function-space optimization leads to the calculus of variations and optimal control.

Objective geometry

A convex objective over a convex feasible set has no non-global local minima. Strong convexity adds uniqueness and quantitative curvature. Smoothness enables gradients and Hessians; nonsmoothness calls for subgradients, proximal operators, bundle methods, or smoothing. Nonconvexity can arise from physics, discrete structure, neural networks, equilibrium constraints, or composition.

Information regime

In white-box optimization, derivatives and algebraic structure are available. In black-box optimization, only evaluations can be requested. Zeroth-order methods estimate useful directions from evaluations. Simulation optimization adds noise and expense. Online optimization receives losses sequentially. Derivative-free methods are not synonymous with heuristics: some have convergence theory, while some gradient methods are heuristic in nonconvex settings.

Uncertainty

- **Deterministic:** parameters are treated as known.
- **Stochastic:** a probability model supports expectations or risk measures.
- **Robust:** decisions are protected against an uncertainty set.
- **Distributionally robust:** the distribution belongs to an ambiguity set.
- **Adaptive:** information arrives and decisions may change.
- **Adversarial:** another process selects difficult inputs.

Time and agency

Static problems choose once. Dynamic problems choose a trajectory or policy. Control problems couple action to evolving state. Games include strategic agents. Bilevel problems place one optimizer inside another. Mechanism design optimizes rules under strategic response. Distributed optimization divides data, variables, or authority among participants.

Guarantee

An algorithm may provide an exact optimum, a certified bound, an approximation ratio, asymptotic convergence, regret bounds, probabilistic guarantees, empirical superiority, or only a plausible candidate. Complexity theory identifies problems for which exact worst-case solution scales intractably; approximation and problem-specific structure then become central (Garey and Johnson 1979; Papadimitriou and Steiglitz 1982; Goemans and Williamson 1995). No-free-lunch results formalize that performance advantages depend on assumptions about the problem class (Wolpert and Macready 1997).

A selection rule

The practical algorithm-selection sequence is:

1. expose variable and constraint types;
2. test convexity, smoothness, separability, sparsity, and symmetry;
3. identify uncertainty and observation cost;
4. state the required guarantee and deadline;
5. select a method family;
6. compare multiple implementations on representative instances; and
7. verify sensitivity to modeling and solver choices.

Before choosing a method, construct a structural fingerprint of the problem. The fingerprint is a compact inventory of decision type, feasible-set geometry, objective regularity, information access, uncertainty, temporal structure, agency, scale, and required guarantee. Two problems described with the same domain noun may have different fingerprints: “routing” can mean a deterministic shortest path, a capacitated vehicle-routing MILP, an online dispatch policy, or a robust multi-agent game. Conversely, a chip floorplanner and a hospital bed allocator may share a decomposable assignment structure even though their semantics are radically different.

A useful fingerprint answers the following questions in order.

Representation. Are the variables continuous, integer, categorical, graph-valued, sequence-valued, function-valued, or mixed? Is the apparent representation canonical, or does it contain symmetries that duplicate the same physical or semantic object?

Geometry. Is the feasible set convex, disconnected, manifold-valued, combinatorial, or accessible only through a simulator? Are objectives smooth, Lipschitz, separable, sparse, submodular, piecewise linear, or discontinuous? Which of these claims are mathematical facts, and which are empirical observations over a limited region?

Information. Can the method query exact values, noisy values, derivatives, adjoints, constraints, samples, or only rankings? How expensive is a query? Does evaluation fail for invalid candidates? Can experiments run in parallel?

Time and agency. Is the choice one-shot, repeated, sequential, or closed-loop? Do other agents adapt strategically? Does the chosen action change the data-generating process or future feasible set?

Warrant. Is the decision served by a feasible candidate, a local stationary point, a certified bound, an approximation ratio, a regret guarantee, a calibrated predictive distribution, or a humanly defensible set of alternatives?

The fingerprint prevents method-first modeling. A familiar solver is a tool, not an ontology. Forcing every problem into a mixed-integer program, differentiable objective, or reinforcement-learning environment can erase the very structure that would make it solvable and valid.

Before accepting an aggregation or surrogate, list invariants that must survive. Common invariants include conservation of material, nonnegative inventory, temporal precedence, legal eligibility, connectedness, mass or energy balance, probability normalization, anonymity, and an accessibility guarantee. Then test the transformation.

Aggregating hourly demand to a day can preserve total quantity and destroy peak capacity. Replacing a network by Euclidean distance can preserve proximity and destroy bridges, slopes, borders, or accessibility. Relaxing integrality preserves a bound and destroys implementability. Embedding text in a vector space preserves selected statistical neighborhoods and destroys exact citation, chronology, and negation unless separately represented.

This is a practical categorical test. A proposed morphism is admissible for a task only if the diagrams expressing required invariants commute within a declared error. When they do not, either enrich the representation, weaken the claim, or reject the transformation.

History as accumulated structure

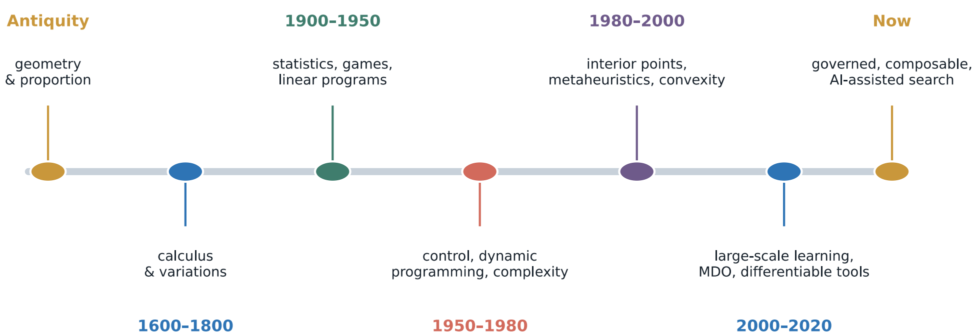


Figure 4. A timeline from ancient reasoning about the good, through calculus, variational principles, industrial operations research, convexity and complexity, to data-driven, compositional, and governed optimization.

Figure 4 makes the historical pattern visible: the field advances by enlarging what can be represented and certified, not by following a single line of ever-faster search.

The history of optimization is not a single line. It is a braid of ethics, geometry, mechanics, economics, computation, statistics, and administration.

Ancient ethics asked what constitutes a good life and how practical judgment selects action; Aristotle’s account of deliberation is not mathematical programming, but it reminds modern optimization that ends are not generated by technique alone (Aristotle 2002). Early geometric extremum problems and later differential calculus made maxima and minima mathematically tractable. Variational principles in mechanics replaced point decisions with optimal curves and functions. Economic theories formalized utility, exchange, efficiency, and equilibrium, while also provoking durable questions about aggregation and welfare.

The twentieth century created optimization as an operational technology. Linear programming grew from production, transportation, military logistics, and resource allocation. Dantzig’s simplex method, Kantorovich’s allocation work, and Koopmans’s activity analysis connected geometry to institutions (Dantzig 1963; Kantorovich 1960; Koopmans 1951). Dynamic programming reframed sequential choice through a principle of optimality (Bellman

1957). Optimal control connected variational reasoning, feedback, and engineering (Pontryagin et al. 1962). Kuhn-Tucker conditions unified constrained nonlinear reasoning (Kuhn and Tucker 1951).

Computational complexity changed the question from “Does an optimum exist?” to “What resources are required to find or approximate it?” Polynomial-time algorithms for linear programming showed that algorithmic paradigms could redraw boundaries of tractability (Karmarkar 1984). Combinatorial optimization connected polyhedral geometry, graphs, approximation, and integer programming (Schrijver 2003).

Uncertainty produced stochastic programming, robust optimization, Bayesian decision methods, and risk measures. Multiobjective optimization replaced the single best point with a frontier of trade-offs (Miettinen 1998). Evolutionary, swarm, annealing, and surrogate-based methods expanded the repertoire for irregular or expensive landscapes (Kirkpatrick, Gelatt, and Vecchi 1983; Kennedy and Eberhart 1995; Jones, Schonlau, and Welch 1998).

Machine learning made optimization ubiquitous and high-dimensional. Stochastic gradients became an industrial primitive (Robbins and Monro 1951; Bottou, Curtis, and Nocedal 2018). Learned models began to optimize not only parameters but architectures, data acquisition, policies, chip layouts, and scientific hypotheses. At the same time, rankings, recommendation systems, and automated allocation revealed that deployed objectives reshape the social worlds they measure (Espeland and Sauder 2007; Sculley et al. 2015).

The contemporary phase adds three themes. First, optimization is increasingly compositional: large systems are assembled from interacting subsystems, motivating category-theoretic, distributed, and multidisciplinary approaches (Fong and Spivak 2019; Martins and Lambe 2013). Second, optimization is increasingly governed: safety, fairness, privacy, sustainability, and rights enter formulation and lifecycle management (Tabassi 2023; European Parliament and Council of the European Union 2024). Third, optimization is increasingly reflexive: the system observes and changes itself, while users and institutions adapt in response.

History therefore teaches humility. Many “new” methods recompose older ideas under new computational conditions. Many old debates about ends, power, and judgment return when an objective is encoded in software.

The history of optimization is best read as an accumulation of representational capacities. Calculus made local differential structure usable. Variational methods admitted functions and trajectories. Linear programming made resource allocation and dual prices computational objects. Dynamic programming made state sufficient statistics explicit. Complexity theory distinguished existence from tractable construction. Robust and stochastic optimization admitted different kinds of uncertainty. Multiobjective methods preserved partial order. Category-theoretic and compositional approaches now ask how guarantees survive connection across interfaces.

This view avoids a progress myth in which each new method supersedes the previous one. Simplex, interior-point, dynamic programming, decomposition, gradient methods, heuristics, simulation, and deliberation occupy different regions of the structural map. Old methods often become newly powerful when hardware, automatic differentiation, data, or modeling languages change. Historical literacy helps recognize a rediscovered idea and its earlier failure conditions (Dantzig 1963; Bellman 1957; Karmarkar 1984; Schrijver 2003).

Representation, complexity, and equivalent decisions

Consider locating p service facilities to cover demand points. A direct binary formulation can use $y_j \in \{0,1\}$ to indicate whether facility j opens and $x_{ij} \in \{0,1\}$ to assign demand point i to facility j. Let $\mathcal{F}_p$ contain the binary assignments satisfying $\sum_{j=1}^{m} x_{ij} = 1$ for every demand point i, $x_{ij} \le y_j$ for every pair i, j, and $\sum_{j=1}^{m} y_j = p$. Here d_i is the demand weight at point i, c_{ij} is the assignment cost, and f_j is the fixed cost of opening facility j. The model is then

$$\min_{(x,y)\in\mathcal{F}_p} \left(\sum_{i=1}^{n}\sum_{j=1}^{m} d_i\, c_{ij} x_{ij} + \sum_{j=1}^{m} f_j\, y_j\right). \tag{14}$$

The algebra exposes fixed costs, assignment costs, and coupling. Yet domain meaning still determines whether distance c_{ij} represents travel time, accessibility, cultural acceptability, network latency, or ecological disruption. The same formulation can support emergency stations, content replicas, warehouses, or clinics, but the validation evidence is different in each case. Equation (14) is therefore a reusable syntax with typed semantics.

Representation can also change computational difficulty. Aggregating demand may shrink the model but hide vulnerable groups. Relaxing integrality gives a lower bound but may create impossible fractional facilities. Decomposing by geography can accelerate solution but mishandle cross-boundary demand. Symmetry-breaking can remove redundant facility labels without changing physical alternatives. Each transformation should state what it preserves: feasible real decisions, objective bounds, uncertainty, distributional information, or merely average behavior.

Worst-case complexity is indispensable: it prevents claims that an expressive formulation is automatically tractable (Garey and Johnson 1979; Papadimitriou and Steiglitz 1982). Operational choice also depends on instance distribution, horizon, warm starts, and repetition. A difficult combinatorial problem can be easy on a stable sparse network and hard after one business rule creates symmetry or dense coupling. Conversely, a polynomial method can be unusable if each evaluation invokes a high-fidelity simulator.

Record both formal complexity and empirical scaling. Vary problem size, density, conditioning, uncertainty, and constraint tightness separately. Inspect failure regimes, not only median runtime. If a learned or heuristic method is used, test shifted instance families that challenge its inductive bias. The goal is not to classify the domain once; it is to understand which structural changes make the chosen method stop being appropriate.

An optimization can contain many numerical points representing the same substantive choice. Facilities may be unlabeled, neural units permutable, rotations physically equivalent, and schedules identical up to workers with the same qualification. These symmetries multiply search without adding alternatives. Symmetry-breaking constraints or quotient representations can improve computation if they preserve every real equivalence class.

The reverse problem is identifiability: distinct world mechanisms can produce the same observations. A calibrated simulation may fit parameters that compensate for one another. A causal model may match outcomes under current policy and disagree under intervention. An inverse design may have many geometries with indistinguishable measured performance and different manufacturability.

Before search, ask two questions. Which numerical differences should be identified as the same object? Which substantively different mechanisms are collapsed by the observation? The first guides quotienting and symmetry breaking; the second guides experiments and uncertainty.

For every claimed parameter estimate, report whether the decision depends on the parameter itself or only on a prediction. If two parameter sets lead to the same admissible action across relevant scenarios, identifiability may not be decision-critical. If they lead to different actions under a foreseeable condition, collect discriminating evidence or choose a policy robust to both.

Symmetry and identifiability meet in translation. A musical chord can be considered equivalent under octave or permutation for one analysis and distinct in voicing and instrument for performance. A transport model can identify people by demand class for capacity planning and must distinguish access needs for service design. Equivalence is always typed by purpose.

Decompose when components have sparse coupling and can exchange enough information to coordinate. Geographic, temporal, scenario, resource, or functional decomposition can shrink computation and align organizational ownership. It fails when the interface omits a strong global constraint, shared uncertainty, or common-cause failure.

Test decomposition against a centralized model on smaller instances. Compare feasibility and objective, inspect interface residuals, and vary coupling strength. If local solutions oscillate, expose richer response functions or use coordination penalties. If organizations strategically report local information, the decomposition is also a mechanism and needs incentive analysis.

Benchmark representations, not only solvers. For a manageable set of real instances, implement two formulations that encode the same declared decision. Compare model size, relaxation bound, solve time, numerical behavior, ease of adding domain constraints, feasibility after export, and explanation to reviewers.

Then perturb the domain: add a fairness floor, correlated failure, new resource, or changed horizon. A representation that was fastest may be brittle to model evolution. Maintenance is part of inferential fitness.

Check semantic equivalence with generated and hand-built examples. Map each solution back to the domain and compare. Exact algebraic reformulations should preserve feasible decisions and values; approximations should satisfy declared error; aggregations should preserve selected invariants. A discrepancy is either a bug or evidence that the two models never represented the same object.

Representation benchmarks reduce method folklore. They also create reusable tests when a model is refactored, decomposed, or moved to a new solver.

Method selection worked end to end

Suppose a regional blood service must route vehicles from collection sites to laboratories under time limits, uncertain volume, road disruption, temperature control, and vehicle failure. The domain noun suggests “routing,” but the fingerprint determines the method.

Decision type. Routes and assignments are discrete; departure times and quantities may be continuous. **Geometry.** The static core is a capacitated network problem with time windows, but vehicle reuse and pickup-delivery coupling break a simple shortest-path representation. **Information.** Travel and volume are forecast, not exact; temperature feasibility is monitored; some disruption is observed during operation. **Time.** The plan is repeated and must be repaired online. **Agency.** Drivers and hospitals respond operationally but are not strategic opponents. **Warrant.** Every dispatched route must be feasible and traceable; a global optimality proof is less valuable than rapid recovery and low tail delay.

This fingerprint suggests a layered method: a mixed-integer planning model for fleet and nominal routes; scenario or robust constraints for critical time windows; a fast insertion-and-repair heuristic online; and a safety rule that refuses any action violating validated temperature and handling requirements. A pure traveling-salesperson solver would preserve distance and discard the typed constraints that give the problem meaning. A reinforcement-learning policy would require a simulator and safety architecture that may cost more than the decision warrants.

Now test representation. A time-expanded network makes time windows explicit and can grow rapidly. A route-column formulation represents complete feasible routes and supports column generation. A vehicle-indexed MILP is direct and can suffer symmetry. An event-based constraint model may propagate time and resource conflicts effectively. Benchmark at least two on instances that preserve real disruption and route structure.

The translation from blood logistics to ordinary parcel routing preserves capacity, time, flow, and recovery. It does not preserve consequence: a late parcel and a compromised medical product are not the same penalty. Temperature and traceability remain hard interface types.

Field checklist

- Build the structural fingerprint before selecting software or an algorithm family.
- Separate facts about the mathematical object from observations about sampled instances.
- Identify symmetries, decompositions, conservation laws, sparsity, and natural interfaces.

- Record every relaxation, aggregation, surrogate, and discretization with its preservation claim.
- Test at least one representation different from the first formulation.
- Specify the warrant required by the decision, not merely the guarantee advertised by a solver.
- Revisit the fingerprint when the boundary, data regime, or implementation cadence changes.

Translation lens. Structural correspondences justify transfer. A formulation may travel from logistics to computing because assignment and capacity are preserved, while distance, harm, and evidence remain domain-typed. Translation is sound when the preserved structure and the discarded structure are both named.

Chapter 3 · Measurement, evidence, governance, and the optimization lifecycle

Measurement and the evidence chain

Optimization requires comparison, and comparison requires evidence. Yet evidence is not identical to a metric. Measurements are generated by instruments, definitions, sampling processes, and institutions. A metric can be reliable but irrelevant, relevant but manipulable, or informative at one scale and misleading at another.

The measurement model

Let z be a latent property of interest and let y be the measured proxy:

$$y = m(z, c) + \varepsilon, \tag{15}$$

where m is a measurement process that depends on context c, and ε represents error. Optimizing y changes incentives around m and may change the relationship between y and z. This is the mechanism beneath Goodhart-like failures: when a proxy becomes a target, selection pressure concentrates precisely on the gap between proxy and purpose (Goodhart 1975; Campbell 1976; Muller 2018).

Evidence for solutions

Evidence should answer several distinct questions.

- **Mathematical validity:** Were the objective and constraints solved as claimed?
- **Numerical validity:** Are tolerances, conditioning, discretization, and stochastic error controlled?
- **Model validity:** Do assumptions and parameters represent the relevant system?
- **Comparative validity:** Is the baseline fair, and were alternatives tuned equally?
- **External validity:** Does performance transfer across settings, populations, and time?
- **Causal validity:** Will intervention produce the predicted consequence, rather than merely correlate with it?
- **Normative validity:** Do stakeholders accept the objectives, constraints, and distribution of effects?

Sensitivity analysis reveals how results change under parameter perturbation. Scenario analysis explores coherent futures. Stress testing searches for failure under extremes. Ablation studies isolate components. Randomized trials can estimate causal effects where ethical and practical. Qualitative inquiry can identify mechanisms and harms absent from numeric traces.

Governance as control over the formulation

Governance is not a final ethics checkbox. It is the allocation of rights over the entire optimization lifecycle. Who may set the objective? Who can add a constraint? Who sees the data and assumptions? Who can stop deployment? Who hears appeals? Who is responsible when the model and world diverge?

A governed optimization system should have:

1. a named owner for the decision and a distinct reviewer;
2. a documented objective and constraint register;
3. traceable data and model versions;
4. thresholds for abstention or human escalation;
5. monitoring for drift, distributional effects, and strategic response;
6. incident and rollback procedures; and
7. scheduled reframing, not only parameter retraining.

Public frameworks increasingly express this lifecycle view. NIST's AI RMF organizes risk work around Govern, Map, Measure, and Manage, while health guidance emphasizes autonomy, safety, transparency, accountability, inclusiveness, and sustainability (Tabassi 2023; World Health Organization 2021). Standards such as ISO/IEC 25010:2023 provide quality models that keep software optimization from collapsing into one performance metric (International Organization for Standardization 2023).

Audit the objective, not only the algorithm

Algorithm audits often inspect bias, robustness, or accuracy while taking the objective as fixed. Objective auditing asks why the target exists, how it was operationalized, whether it displaces other purposes, and what behavior it induces. Constraint auditing asks which interests received hard protection and which were priced as trade-offs. Governance is strongest when affected people can contest these upstream choices.

An optimization result inherits uncertainty from every upstream link: observation, data cleaning, parameter estimation, causal assumptions, model structure, numerical approximation, solver termination, implementation, and behavioral response. Reporting only the objective value at the final link creates an evidence illusion. A field guide instead traces a chain of claims and attaches an appropriate test to each claim.

Suppose a city optimizes school-support funding using a predicted "risk of underperformance." The prediction may be calibrated, yet the allocation question is causal: will a particular support package improve learning for a particular school under actual delivery constraints? Historical labels may reflect unequal resources; measured attainment may omit well-being, inclusion, and long-term opportunity; schools may change behavior when the score controls funding. A technically accurate model can therefore support an invalid intervention.

The evidence chain asks:

1. **Construct validity:** does the measurement represent the intended property?
2. **Statistical validity:** are sampling, dependence, missingness, and uncertainty handled?
3. **Causal validity:** does the proposed action change the outcome as predicted?

4. **Numerical validity:** are discretization, tolerances, and conditioning controlled?
5. **Operational validity:** can the intervention be implemented with actual people and infrastructure?
6. **Distributional validity:** who benefits, who bears errors, and which subgroups are hidden by averages?
7. **Temporal validity:** what drifts, adapts, or becomes obsolete after deployment?
8. **Normative validity:** is the objective legitimate and contestable?

No single method answers all eight. Randomized trials can estimate some causal effects but may have limited transportability. Simulation can expose dynamics but only within its assumptions. Formal verification can prove conformance to a model but not adequacy of the model. Qualitative work can reveal meanings, workarounds, and harms absent from recorded variables. Robust evidence is heterogeneous and deliberately composed.

Every consequential optimization should produce a dossier that another competent reader can inspect without reconstructing the project from code. The dossier includes:

- decision charter and system boundary;
- stakeholder and rights analysis;
- data provenance and measurement definitions;
- model equations, assumptions, and units;
- solver, version, tolerances, random seeds, and computational budget;
- baseline and alternative formulations;
- feasibility, optimality, approximation, or empirical evidence;
- sensitivity, scenarios, and stress tests;
- distributional and accessibility analysis;
- implementation, monitoring, override, and rollback plan; and
- unresolved disagreements and known unknowns.

This is not paperwork surrounding the “real” optimization. It is the externalized state that allows the optimization process to compose across teams and time. Without it, later users receive an untyped output and silently guess the missing interfaces.

A decision dossier becomes more useful when its claims are arranged as rows and evidence sources as columns. For a predictive-maintenance policy, rows might include “the sensor measures the intended condition,” “the model is calibrated before failure,” “the threshold reduces unplanned downtime,” “maintenance capacity can absorb alerts,” “the policy does not systematically defer difficult assets,” and “the fallback is safe.” Columns might include calibration bench tests, historical data, designed inspections, simulation, shadow deployment, operator review, and prospective outcomes.

No row should be justified merely because many columns contain data. Evidence must address the claim. A large historical dataset cannot establish the safe response to a sensor fault never recorded. A successful bench test cannot establish maintenance benefit. Operator acceptance cannot prove calibration, but repeated operator disagreement can expose a missing state or infeasible workflow.

Mark each cell as direct, indirect, contradictory, absent, or not applicable. Record population, conditions, version, uncertainty, and owner. The matrix reveals brittle chains: a high-stakes claim may depend on one indirect study or a model validated only near the current policy. It also prevents evidence laundering, where a citation to a general method is presented as validation of a local implementation.

Post-deployment incidents are unusually informative because they expose a complete chain. Reconstruct the world state, available data, model version, recommendation, interface, human response, system action, and consequence. Then run counterfactual replays with the prior policy, candidate revision, and simple baseline. Ask which differences are supported and which depend on unobserved assumptions.

Do not optimize the retrospective story. Hindsight makes a rare signal look obvious and hides the false alarms a more sensitive rule would have created. Preserve the information available at the time. Use several plausible causal accounts when the mechanism is uncertain. The purpose of replay is not to find an individual to blame; it is to test whether sensing, representation, authority, and recovery composed as designed.

Near misses and overrides belong in the same process. A human correction that prevents harm is evidence that the safeguard worked and that the upstream model may need revision. If overrides become a target, staff may stop recording them. Separate learning logs from punitive performance metrics.

Distribution shift has several types. Covariate shift changes the frequency of observed conditions. Label shift changes outcome prevalence. Concept drift changes the relation between observations and outcome. Policy shift changes data because the decision system acts. Boundary shift changes who or what is included. Normative shift changes the institution’s values or legal duties.

Each type needs a different response. Reweighting can help with some covariate shifts and cannot repair a changed causal mechanism. Recalibration can repair probabilities and cannot legitimate a changed use. New data can reduce estimation uncertainty and cannot supply missing consent. Monitoring should therefore track inputs, residuals, calibration, decisions, outcomes, population, and governance conditions separately.

The reopening rule should specify what can be updated automatically and what requires authority. A temperature-forecast parameter may update routinely; a risk threshold controlling access to care should not drift because service capacity changed. If a constrained resource makes the threshold infeasible, that is a policy conflict to surface, not an invitation for silent normalization.

Independence is a structural property, not a team name. A reviewer is independent when they can access relevant evidence, use a different method or model, report dissent, and are not rewarded for deployment. Challenge can include reimplementation, alternate data cleaning, sensitivity to boundary and metric, adversarial cases, affected-party review, and comparison with a simple baseline.

For the highest-stakes systems, divide evidence production and approval. The model team can explain assumptions; it should not be the sole judge of their adequacy. Publish unresolved disagreement proportionally to the decision. A decision can proceed under disagreement, but the disagreement must not disappear from the record.

Every modeled quantity should have a measurement protocol. Name the construct, operational definition, instrument or data source, unit, sampling frame, timing, aggregation, missing-data rule, uncertainty, and owner. State how the measure can be manipulated and which groups or conditions it represents poorly.

Consider “wait time” in a public service. It can mean time from need to first request, request to appointment, arrival to service, or cumulative time across referrals. Each definition answers a different question. Administrative timestamps may omit people who abandon before registration. Averaging completed cases can improve when the longest-waiting cases remain unresolved. The optimization must choose a clock and include censoring, abandonment, and repeated contact.

Construct validation combines quantitative and qualitative evidence. Compare the measure with related and distinct constructs, inspect known groups and edge cases, and ask participants whether the operational definition captures their experience. A high correlation with another proxy is not enough. The protocol should specify when the construct changes—for example, when virtual service changes what “arrival” means.

Measurement error propagates through optimization nonlinearly. If a decision selects extremes, even unbiased error can create biased selected values. Use repeat measurement, reliability models, shrinkage, and out-of-sample confirmation where appropriate. For thresholds, study classification near the boundary; for constraints, include uncertainty margins rather than pretend the measured value is exact.

A baseline is a policy and operating context, not a single historical average. Document who received what action, how resources were rationed, and what adaptations occurred. If a new optimizer changes eligibility, workload, or reporting, outcomes cannot be compared without accounting for the changed population and mechanism.

Use several baselines: current practice, simple transparent rule, no-action or minimum-service case, and a domain-expert policy. A complex method should show incremental value against the strongest reasonable simple alternative. When randomization is not appropriate, use prospective shadow decisions, natural experiments, matched comparisons, interrupted time, or simulation with explicit causal limits.

Evidence rarely transfers unchanged across site, population, time, or implementation. Write a transport argument: which causal mechanisms are shared, which effect modifiers differ, which resources and workflows differ, and what local observations can test the bridge.

A trial can estimate an effect under participating centers and protocol. Deployment may involve different staffing, adherence, language, baseline risk, or alternatives. A software benchmark can establish performance on one hardware and workload; production adds concurrency, failures, and operators. A material test can establish coupon behavior; a manufactured assembly adds joints and defects.

Separate three claims. **Internal validity** concerns the original estimate. **Transportability** concerns whether relevant mechanisms and modifiers support a new setting. **Implementation validity** concerns whether the intervention delivered there matches the one studied. New local calibration cannot repair a different causal mechanism.

Use a bridge plan: small prospective study, shadow deployment, local measurement of modifiers, mechanism checks, and predefined discrepancy triggers. Preserve the external source and local evidence rather than blending them into one confidence score.

Missingness can arise from failure, burden, access, discretion, avoidance, or because an outcome has not yet occurred. The missing-data model should describe this process. Imputation under a convenient assumption is not neutral.

Map missingness by group, time, state, and decision. Ask whether the optimization changes who is observed. A triage model can reduce follow-up for low-ranked cases, making their outcomes missing and confirming the model. A maintenance policy can inspect only high-risk equipment and bias failure labels.

Use sensitivity analyses for plausible missing-not-at-random mechanisms, collect audit samples, and include missingness indicators only with causal care. Sometimes the correct intervention is to improve measurement access rather than optimize over incomplete records.

Assign each claim an expiration or review condition. Software, law, population, equipment, climate, and practice change at different rates. A study does not become false on a date, but the bridge assumptions can weaken.

Triggers include model or interface change, new population, policy response, incident, material supplier change, drift, or revised scientific guidance. Review should ask whether old evidence still covers the current system, not merely whether the citation remains accessible.

Governance as feedback and memory

Governance can be modeled as a control layer over the optimization lifecycle. It observes model behavior and consequences, compares them with requirements, and changes permissions, constraints, or the formulation. This analogy is useful only if its limits are respected: stakeholders are not disturbances, rights are not tunable penalties, and legitimacy is not a stability margin. Still, the control perspective clarifies why a one-time ethics review is insufficient.

A governed project has at least four loops.

Fast operational loop. Monitors feasibility, service limits, safety alarms, and data quality; can pause or fall back immediately.

Model loop. Re-estimates parameters, checks drift, recalibrates uncertainty, and retests sensitivity on a scheduled cadence.

Decision loop. Reviews whether recommendations improve real outcomes, whether operators override them, and whether affected parties can obtain correction or recourse.

Purpose loop. Reopens the boundary, objective, and institutional mandate. It asks whether the system should continue to exist in its present form.

The loops require distinct authority. The team benefiting from a favorable metric should not be the sole judge of its meaning. A named owner can be accountable without being omnipotent; independent review, incident reporting, and external contestation provide necessary counter-morphisms.

A metric should be tested not only on historical data but under anticipated optimization pressure. If admissions, funding, compensation, or ranking depends on a score, ask how a rational participant could raise the score without improving purpose. Red-team the measurement process: change documentation, shift difficult cases, alter denominators, delay events, relabel outcomes, or move cost outside the boundary. The aim is not cynicism. It is to sample the kernel of the measurement morphism before large incentives do so in production (Campbell 1976; Goodhart 1975; Espeland and Sauder 2007; Muller 2018).

Metrics should also be paired with counter-metrics and qualitative evidence. A throughput objective may be paired with rework, delay distribution, staff workload, and user-reported quality. A model-accuracy objective may be paired with calibration, abstention, distribution shift, energy, and downstream utility. Multiple metrics do not eliminate gaming, but they make a one-dimensional exploit less likely to masquerade as universal improvement.

Monitoring without a response rule creates observation theater. For each metric, name threshold, trend, review cadence, owner, action, and escalation. Include slow outcomes and leading indicators. A system can remain within monthly limits while accumulating a rare hazard or eroding trust.

Institutional memory requires versioned decisions, incidents, appeals, overrides, and model changes. Staff turnover should not erase why a constraint exists. Link each change to evidence and preserve rejected alternatives. A future reviewer should be able to reconstruct not only what the system did but why the institution believed it was justified.

Field checklist

- Write the evidence chain and associate each link with a distinct validation method.

- Test causal claims as causal claims rather than substituting prediction accuracy.
- Define fast operational, model, decision, and purpose-review loops.
- Separate ownership from independent review and provide a real path for contestation.
- Red-team the metric under selection pressure before deployment.
- Preserve baselines, failed alternatives, overrides, incidents, and dissent in the dossier.
- Schedule reframing; do not wait for parameter drift to reveal a purpose failure.

Translation lens. Evidence is the natural transformation connecting modeled consequences with observed consequences across versions of a problem. Governance controls whether that transformation is inspected, challenged, and used to revise the object being optimized.

Part II · Mathematical Foundations

Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates

Calculus, local geometry, and numerical truth

Continuous optimization begins with local change. If $f: \mathbb{R}^n \to \mathbb{R}$ is differentiable, its gradient $\nabla f(x)$ is the direction of steepest increase under the Euclidean norm. The negative gradient therefore supplies a local descent direction:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \tag{16}$$

where the step size α_k may be fixed, scheduled, or chosen by line search. The equation is simple; its behavior depends on geometry. Poor scaling produces narrow valleys and oscillation. Saddle points can slow or redirect progress. Noise turns the path into a stochastic process. Constraints may require projection or a different geometry.

First- and second-order conditions

For an interior local minimizer x^* of a differentiable function, $\nabla f(x^*) = 0$ is necessary. If f is twice differentiable, the Hessian $H(x) = \nabla^2 f(x)$ describes local curvature. A positive-definite Hessian at a stationary point is sufficient for a strict local minimum. Newton's method uses a local quadratic model:

$$x_{k+1} = x_k - H(x_k)^{-1} \nabla f(x_k). \tag{17}$$

Near a well-behaved solution it can converge rapidly, but forming and solving with the Hessian can be expensive. Quasi-Newton methods such as BFGS update an approximation using gradient differences. Conjugate-gradient and Krylov methods exploit large sparse structure. Trust-region methods restrict each local model to a region where it is credible (Nocedal and Wright 2006).

Geometry other than Euclidean geometry

The gradient is not coordinate-free; it depends on an inner product. Preconditioning changes the metric so that equal algorithmic steps better match the problem's scales. Mirror descent replaces Euclidean distance with a Bregman divergence, making simplex, probability, and sparse geometries natural. Natural gradients use an information-geometric metric. On a manifold, a Riemannian gradient lies in the tangent space and a retraction maps the step back to the manifold.

Nonscalar and complex variables

Matrix variables appear in control, covariance estimation, semidefinite programming, and quantum information. Function variables appear when the decision is a curve, field, or policy. The calculus of variations considers a functional such as

$$J[y] = \int_a^b L\left(t, y(t), y'(t)\right) dt. \tag{18}$$

Under regularity and fixed endpoint conditions, an extremal satisfies the Euler-Lagrange equation

$$\frac{\partial L}{\partial y} - \frac{d}{dt}\left(\frac{\partial L}{\partial y\prime}\right) = 0. \tag{19}$$

This passage from optimizing points to optimizing paths underlies geodesics, mechanics, shape design, image processing, and optimal control.

Discretize then optimize, or optimize then discretize?

Scientific and engineering problems often begin in infinite-dimensional form and are discretized. If the discretization is coarse, the computed optimum may exploit numerical artifacts. If gradients are derived after discretization, they may differ from discretized continuous adjoints. Mesh refinement, consistency checks, and adjoint verification are therefore part of optimization, not post-processing.

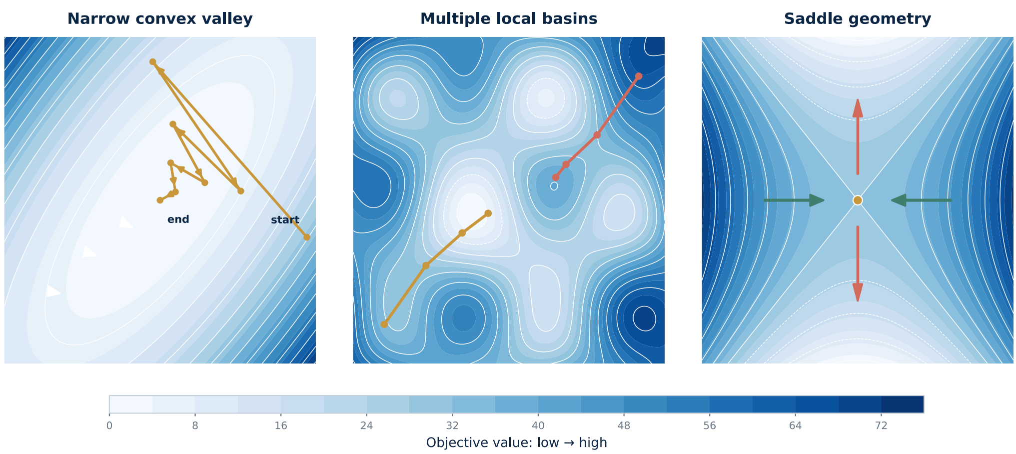


Figure 5. Narrow-valley, multiple-basin, and saddle contour landscapes with distinct search directions.

Figure 5 makes the narrow-valley descent explicit: arrowheads carry a shrinking zig-zag from the darker start to the light minimum. The other panels show several basins and saddle directions; the shared scale runs from low to high.

Continuous optimization succeeds when local information meaningfully constrains the global problem. A gradient is a covector before an inner product identifies it with a direction; a Hessian describes local curvature; convexity turns supporting hyperplanes into global bounds. Confusing these roles produces common errors. A zero gradient can be a minimum, maximum, saddle, or flat point. A positive semidefinite Hessian at one point does not establish global convexity. A small step norm may indicate convergence, bad scaling, or a stalled algorithm.

The practical sequence is to establish the claim one is entitled to make. For an unconstrained differentiable problem, first-order stationarity $\nabla f(x^*) = 0$ is necessary at an interior local optimum. Positive definiteness of the Hessian is a sufficient local condition. For a convex differentiable function, stationarity is sufficient for global optimality. Constraints replace the zero-gradient condition with a normal-cone balance. The KKT system is therefore geometry: the negative objective gradient must lie in the cone generated by active constraint normals.

Constraint qualifications are not decorative technicalities. They ensure that the local geometry of the constraint representation matches the geometry of the feasible set closely enough for multipliers to exist. A convex problem does not automatically make every KKT necessity claim valid; Slater's condition is a standard sufficient hypothesis for convex inequalities, while other settings require other regularity conditions (Boyd and Vandenberghe 2004; Nocedal and Wright 2006). A certificate is only as good as its premises, so these conditions must be stated explicitly.

Real solvers work with floating-point arithmetic, scaled residuals, tolerances, and stopping tests. A status of "optimal" commonly means that primal infeasibility, dual infeasibility, and a scaled gap fall below configured thresholds. Ill-conditioning can make small residuals coexist with large errors in variables or sensitivities. Units that differ by twelve orders of magnitude can distort a method even when the algebra is correct.

A numerical certificate should report at least the primal residual, dual residual, complementarity or gap, scaling, iteration limit, and termination reason. For nonconvex problems it should distinguish local stationarity from global evidence. Multiple starts, lower bounds, interval methods, convex relaxations, or problem-specific arguments can strengthen a claim, but none should be implied by the word "converged."

Gradient verification deserves special care. Automatic differentiation computes derivatives of the implemented program, not necessarily of the intended mathematics. Discontinuous branches, iterative inner solvers, stochastic sampling, and stateful code can create subtle mismatches. Compare adjoint or automatic derivatives with complex-step or carefully scaled finite differences on small instances; test directional derivatives; verify units and signs; and check the gradient at more than one point.

Use first-order methods when dimensions are large, gradients are available, and moderate accuracy suffices. Use accelerated or variance-reduced variants only when their assumptions match. Newton and quasi-Newton methods earn their additional curvature cost near a well-behaved solution. Trust-region methods are valuable when a local model needs explicit credibility. Interior-point methods provide strong primal-dual information for structured constrained problems. Proximal and splitting methods exploit separable nonsmooth terms. Derivative-free methods are appropriate when evaluations, not gradients, are the true information unit.

The method should be chosen with the entire computation graph in view. A nominally cheap iteration can hide an expensive PDE solve or distributed synchronization. A method with fewer iterations can use more energy or wall time. Algorithmic complexity, memory traffic, parallelism, and evaluation fidelity all belong to the cost model.

The mathematical problem and its numerical representation can have very different geometry. If one variable is measured in meters and another in micrometers, or one constraint residual is near 10^6 while another is near 10^{-4}, termination tests and search directions can be dominated by units. Nondimensionalize where possible, choose physically meaningful scales, and report residuals in both scaled and original units.

Conditioning asks how much the solution changes under small perturbations. An ill-conditioned optimum can be computed accurately for the recorded data and remain substantively unreliable. Near-collinear regressors, nearly active constraints, flat valleys, and almost tied alternatives amplify measurement error. Regularization, bounds, or alternative parameterization can stabilize the computation, but they change the problem and must be justified.

Derivative checking should be routine. Compare analytic or automatic derivatives with directional finite differences at representative and boundary points. Complex-step checks can avoid subtraction error for analytic code paths. Nonsmooth operations, conditionals, clipping, and iterative inner solvers require special care. Automatic differentiation returns a derivative of executed code, not proof that the code represents the intended physics.

Many practical objectives are composite: a smooth loss plus a norm, indicator, hinge, maximum, or piecewise tariff. For

$$\min_x F(x) = f(x) + g(x), \tag{20}$$

with smooth f and convex possibly nonsmooth g, proximal methods use the map

$$\operatorname{prox}_{\alpha g}(v) = \arg\min_x \left(g(x) + \frac{1}{2\alpha} \| x - v \|_2^2 \right). \tag{21}$$

The proximal step preserves structure that smoothing can blur: sparsity, constraints, or a sharp penalty. Equation (21) appears in regularized estimation, imaging, signal recovery, and resource allocation (Boyd and Vandenberghe 2004; Duchi, Hazan, and Singer 2011). The translation is sound when the regularizer encodes a defensible prior or cost. It is not sound when sparsity is preferred merely because the algorithm supports it.

For genuinely nonsmooth nonconvex models, stationarity has several definitions and solver messages can be easy to overinterpret. Report the notion used, residual, and local perturbation tests. If discontinuity comes from a design rule or simulation switch, ask whether the discontinuity is real or a modeling artifact.

Numerical trust-region methods restrict steps to where a local approximation is credible (Nocedal and Wright 2006). The same principle should govern deployment. Define a model trust region in terms of input distribution, state, action, environment, and consequence. Inside it, evidence supports interpolation; near its edge, require caution; outside it, abstain, fall back, or collect evidence.

The optimizer tends to push toward boundaries, so distance from the trust region should be an explicit diagnostic. A candidate can be mathematically feasible and evidentially unsupported. Stage tests outward and expand the region only after observations reconcile with prediction.

For irreversible decisions, the trust region may concern consequences rather than variables: even familiar inputs can combine into an unvalidated tail. Stress interactions and common-cause conditions, not just independent parameter ranges.

Convex relaxations can supply lower bounds, feasible guidance, or tractable surrogates. Report which role is used. If a rank, integrality, or physics constraint is relaxed, inspect the relaxed solution's violation and whether rounding preserves hard requirements. A tight objective bound does not ensure an easily recoverable design.

Successive convex approximation solves local convex models of a nonconvex problem. It needs a feasible or recoverable initialization, conservative constraint treatment, and monitoring of actual versus predicted improvement. Report dependence on starts. The method produces a local claim unless global structure supplies more.

Attach physical or semantic units to variables, objectives, gradients, multipliers, and residuals. A gradient component has objective units per variable unit; a multiplier has objective units per constraint unit. Dimensional inconsistency catches modeling and implementation errors before optimization.

For dimensionless learned scores, state their scale and transformation. A one-unit change in a logit, embedding distance, or standardized feature has no direct operational meaning. Translate sensitivity through calibration to consequences before presenting it as importance.

Unit tests should accompany data transformations and exported decisions. Many catastrophic model defects are not advanced mathematics; they are timestamps, currencies, coordinate systems, and factors of a thousand.

Convexity, duality, and certificates

Convexity is the central tractable structure of continuous optimization. A set C is convex when every line segment between two of its points remains inside it. A function f is convex on C when

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta) f(y), \qquad 0 \le \theta \le 1. \tag{22}$$

The inequality says that the graph lies below its chords. Convex problems have a remarkable local-to-global property: every local minimum is global. When f is strictly convex, the minimizer is unique if it exists. Strong convexity and smoothness give quantitative convergence rates and condition numbers (Boyd and Vandenberghe 2004; Bubeck 2015).

Convexity as model knowledge

Recognizing convexity often matters more than inventing a new algorithm. Affine functions are both convex and concave. Norms are convex. Pointwise maxima of convex functions are convex. Convexity is preserved under nonnegative sums and affine composition. Disciplined convex modeling systems exploit such composition rules to certify that a formulation belongs to a solvable class.

Lagrange duality

For the constrained problem of Chapter 1, the dual function is

$$q(\lambda, \nu) = \inf_{x} \mathcal{L}(x, \lambda, \nu), \qquad \lambda \geq 0. \tag{23}$$

Every dual-feasible pair produces a lower bound on a minimization problem. The dual problem maximizes this bound:

$$\max_{\lambda \geq 0, \nu} q(\lambda, \nu). \tag{24}$$

Weak duality always gives $d^* \leq p^*$. For a convex problem satisfying Slater's condition, strong duality gives $d^* = p^*$. More generally, strong duality requires hypotheses that must be stated rather than inferred from convexity alone. A matching primal solution and dual bound form an optimality certificate.

Karush-Kuhn-Tucker conditions

For differentiable convex problems, the KKT conditions are

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \nu_j^* \nabla h_j(x^*) = 0, \tag{25}$$

$$g_i(x^*) \leq 0, \qquad h_j(x^*) = 0, \tag{26}$$

$$\lambda_i^* \geq 0, \qquad \lambda_i^* g_i(x^*) = 0. \tag{27}$$

The last condition is complementary slackness: either an inequality is inactive and its multiplier is zero, or it is active and may carry a nonzero shadow value. Even for convex problems, necessity of the KKT conditions requires an appropriate constraint qualification, such as Slater's condition in the standard differentiable inequality-constrained setting. Under convexity, KKT conditions are sufficient for global optimality when a feasible primal-dual point satisfies them; outside convexity they are generally only local necessary conditions under regularity and do not certify a global optimum (Kuhn and Tucker 1951; Rockafellar 1970; Boyd and Vandenberghe 2004).

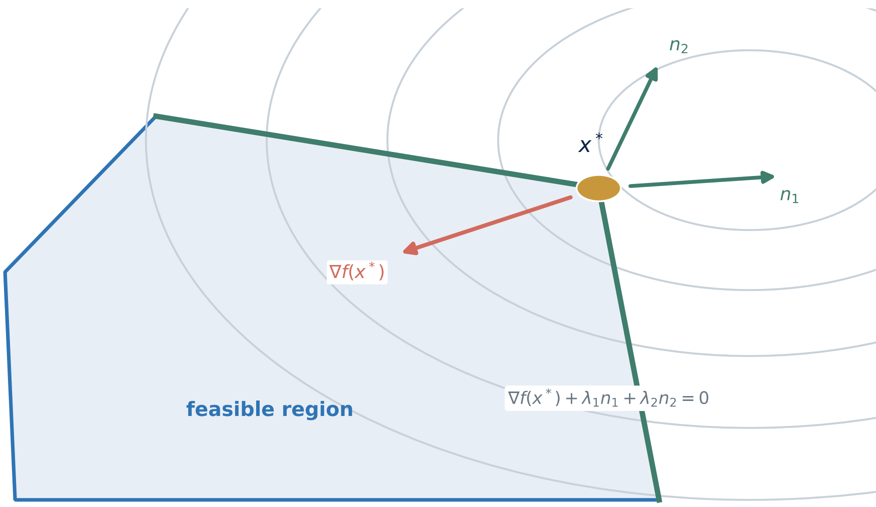


Figure 6. Feasible region, objective contours, active constraints, normal vectors, and the KKT balance at an optimum.

Figure 6 interprets KKT stationarity geometrically: at the constrained optimum, the objective gradient is balanced by normals to the active constraints.

Fenchel duality and proximal structure

For convex f, the conjugate

$$f^*(y) = \sup_x \big(\langle y, x\rangle - f(x)\big) \tag{28}$$

translates between primal and dual representations. Nonsmooth regularizers often have simple proximal maps:

$$\mathrm{prox}_{\gamma f}(v) = \arg\min_x \left(f(x) + \frac{1}{2\gamma} \parallel x - v \parallel^2\right). \tag{29}$$

This restates the proximal map of Equation (21), using f for the generic nonsmooth term and γ for the proximal parameter. Splitting methods exploit this to decompose objectives into parts whose individual proximal operations are easy.

Suppose a workshop produces two products, with contributions 3 and 5, using two limited resources. The continuous relaxation is

$$\max_{x_1, x_2 \geq 0} 3x_1 + 5x_2 \quad \text{subject to} \quad x_1 + 2x_2 \leq 8,\ 3x_1 + 2x_2 \leq 12. \tag{30}$$

Its dual assigns nonnegative shadow values y_1, y_2 to the two resources:

$$\min_{y_1, y_2 \geq 0} 8y_1 + 12y_2 \quad \text{subject to} \quad y_1 + 3y_2 \geq 3,\ 2y_1 + 2y_2 \geq 5. \tag{31}$$

Weak duality says every dual-feasible resource valuation upper-bounds every primal-feasible production value. When primal and dual values coincide, the pair certifies global optimality. The multipliers also give local sensitivity: if a resource limit increases slightly and the active set remains stable, the corresponding dual value estimates marginal benefit. That interpretation is conditional. Large changes can alter the basis; discrete production invalidates marginal divisibility; market, environmental, or labor consequences may not be present in the objective.

The example illustrates a recurring translation. In networks, dual variables become potentials or congestion prices. In mechanics, they become reactions or sensitivities. In control, adjoints propagate the value of state perturbations backward. In machine learning, Lagrange multipliers enforce fairness, privacy, or resource constraints. The mathematical role transfers, but the legitimacy of interpreting a multiplier as a price does not transfer automatically.

Duality is often introduced as an alternate optimization problem. In practice it is also a diagnostic system. A dual certificate of primal infeasibility can show that a primal target set is impossible. A large multiplier can reveal a bottleneck or a bad scale. A duality gap can expose nonconvexity, numerical trouble, or incomplete convergence. Decomposition uses dual messages to coordinate components.

Interpretation must remain conditional. A multiplier is a marginal value for a small relaxation under the model and active set. It may not remain valid for a discrete policy change, large expansion, strategic response, or moral trade. Before calling it a “price,” test perturbations, units, sign, stability, and whether the constraint owner recognizes the proposed trade.

Match the evidence to a ladder of claims: a sampled candidate; a feasible point; a stationary point; a local minimum under stated regularity; a global optimum under convexity or global search; an approximation guarantee; a verified invariant; or an implementation-level safety case. Higher rungs do not automatically dominate lower ones. A globally optimal model with invalid data can support less than a carefully validated feasible baseline.

The report should place mathematical and empirical certificates side by side. “Optimality gap 0.2%” and “outcome uncertainty ±15%” tell the reader where more effort matters. Stop numerical search when its remaining gap is immaterial to the decision.

A derivative summarizes an infinitesimal perturbation under a fixed active structure. Decisions usually face finite change, uncertainty, and regime switching. Build a sensitivity envelope by varying parameters across plausible ranges, resolving the model, and recording objective, decision, active constraints, feasibility margin, and qualitative regime.

Plot where the selected alternative changes. A stable objective value can hide a discontinuous decision; a changing objective can coexist with a stable action. For discrete or nonconvex problems, one-sided and scenario sensitivities are often more informative than a derivative. For learned models, include retraining or policy response where relevant rather than perturbing one prediction in isolation.

Dual multipliers can prioritize this analysis. A large stable multiplier suggests a consequential constraint; a large unstable multiplier signals degeneracy or near-ties. If two active constraints have interchangeable multipliers, report the range of dual solutions rather than one authoritative price.

Field checklist

- State the exact claim and its convexity or regularity hypotheses.
- Verify active-constraint signs and normal geometry.
- Report scaled primal, dual, and complementarity residuals.
- Cross-check derivatives independently at representative points.
- Test finite-range sensitivity before interpreting shadow values.
- Separate a mathematical multiplier from a legitimate social price.

Translation lens. Continuous optimization transfers through geometry: tangent directions, normal cones, curvature, adjoints, and dual bounds. What must be retyped in every discipline is the meaning of a perturbation, the units of sensitivity, and the authority to trade one constraint against another.

Chapter 5 · Structured search: linear, conic, discrete, global, and derivative-free methods

Linear and conic forms as compilation targets

Linear programming chooses $x \in \mathbb{R}^n$ under linear objectives and constraints:

$$\min_x c^T x \quad \text{subject to} \quad Ax = b,\ x \geq 0. \tag{32}$$

Its feasible set is a polyhedron. Extreme points encode basic feasible solutions; the simplex method travels among them, while interior-point methods move through the interior. Linear programming became a general language for production, transport, diet, blending, scheduling relaxations, network flow, and resource allocation (Dantzig 1963; Karmarkar 1984).

The corresponding dual is

$$\max_y b^T y \quad \text{subject to} \quad A^T y \leq c. \tag{33}$$

Primal and dual variables often express two views of one system: quantities versus marginal values, flows versus potentials, or plans versus certificates.

Quadratic programming

A quadratic program has objective

$$\frac{1}{2} x^T Q x + c^T x \tag{34}$$

with linear constraints. If $Q \succcurlyeq 0$, the problem is convex. Quadratic programs appear in least squares, portfolio optimization, support-vector machines, model predictive control, and sequential quadratic programming. Equality-constrained QPs reduce to structured linear systems; active-set and interior-point methods address inequalities.

Conic form

Conic optimization unifies several families:

$$\min_x c^T x \quad \text{subject to} \quad Ax = b,\ x \in K, \tag{35}$$

where K is a convex cone. Choosing the nonnegative orthant yields linear programming. Second-order cones express norm constraints such as $\| Ax + b \|_2 \leq c^T x + d$. The positive-semidefinite cone yields semidefinite programming (SDP), where a symmetric matrix variable X must satisfy $X \succcurlyeq 0$.

SDP supports control synthesis, moment relaxations, covariance problems, geometry, and combinatorial approximation. The Goemans-Williamson Max-Cut algorithm solves an SDP relaxation and rounds its solution to obtain a provable approximation (Goemans and Williamson 1995). Cones thus provide a categorical-looking common interface: different feasible geometries become instances of one algebraic form.

Modeling with relaxations

A relaxation enlarges the feasible set or weakens a constraint to obtain a tractable bound. A tight relaxation guides exact search and can be operationally useful by itself. Relaxations are also diagnostics: a large gap between the relaxed and feasible solution reveals where discrete or nonlinear structure matters. Poorly chosen “big-M” constants, however, can destroy numerical conditioning and bound quality.

Numerical scale and sparsity

Large structured programs are solved by exploiting sparsity, block structure, low rank, separability, and warm starts. Rescaling variables and constraints is not cosmetic; condition numbers govern error amplification and iteration behavior. Solver status must be read with tolerances in mind. “Optimal” means optimal within a numerical model and stopping criterion.

A solver-ready model is compiled through several intermediate representations. Domain concepts become variables and constraints; nonlinear relations may become conic forms or piecewise approximations; discrete logic becomes binaries, clauses, or propagators; uncertainty becomes scenarios or sets; the resulting algebra becomes sparse matrices and expression graphs; presolve produces a reduced instance; the algorithm generates iterates, nodes, cuts, or samples; and machine code turns those operations into memory traffic and communication.

Each compilation stage can preserve, relax, or distort structure. An exact reformulation preserves feasible decisions and objective values, although it may introduce auxiliary variables. A relaxation enlarges the feasible set and usually provides a bound. An approximation changes values within a stated error. A surrogate predicts an expensive response and carries statistical uncertainty. A heuristic transformation may preserve neither feasibility nor bounds but improve search empirically. These are different morphism types and should have different labels in a model audit.

Discrete search as proof construction

Many decisions are indivisible: build or do not build, assign a worker to a shift, select edges in a route, place a block on a grid, include a feature, choose a word. Integer optimization represents such choices directly. A mixed-integer linear program (MILP) has the form

$$\min_x c^T x \quad \text{subject to} \quad Ax \le b, \quad x_i \in \mathbb{Z} \text{ for } i \in I. \tag{36}$$

Integrality transforms geometry. The continuous relaxation may have an attractive fractional solution that has no physical meaning. Exact methods combine several ideas.

- **Branching** divides the feasible set.
- **Bounding** uses relaxations to exclude subtrees.
- **Cutting planes** add valid inequalities that remove fractional solutions.
- **Presolve** simplifies variables and constraints.

- **Primal heuristics** seek feasible incumbents early.
- **Decomposition** separates complicating variables or scenarios.

Branch-and-bound is best understood as a proof tree: every unexplored branch is either bounded, infeasible, or eventually searched. The final optimality gap records the distance between the incumbent and the strongest surviving bound.

Graph and network problems

Shortest paths, flows, matchings, spanning trees, cuts, and colorings reveal how combinatorial structure creates specialized algorithms. Some problems admit integral polyhedra, so a linear relaxation already returns integer solutions. Others are NP-hard and motivate approximation, parameterized algorithms, or problem-specific heuristics (Schrijver 2003).

The stable matching problem illustrates that the objective need not be a scalar. Gale-Shapley deferred acceptance finds a matching with no blocking pair under stated preferences (Gale and Shapley 1962). Stability, fairness, strategy-proofness, and welfare are distinct properties; improving one may worsen another.

Submodularity

A set function $f: 2^V \to \mathbb{R}$ is submodular when it has diminishing returns:

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B) \tag{37}$$

whenever $A \subseteq B$ and $e \notin B$. Submodularity supports strong greedy approximations for coverage, sensor placement, influence, and summarization. It is a discrete analogue of convexity in some respects, though the analogy must be used carefully.

Constraint programming and satisfiability

Constraint programming propagates domains and searches over structured constraints. SAT and SMT solving treat feasibility as logical satisfaction; MaxSAT and pseudo-Boolean optimization add objectives. Planning, verification, configuration, and scheduling often benefit from these representations more than from a generic MILP.

Symmetry and representation

Equivalent representations create redundant search. Symmetry-breaking constraints, canonical forms, and quotient spaces reduce the domain without changing distinct solutions. Representation can alter difficulty dramatically even when the underlying problem is unchanged.

For a mixed-integer program, branch-and-bound builds a proof tree. The incumbent is a feasible witness. Relaxation bounds rule out regions. Cutting planes strengthen the logical description of integer feasibility. Presolve proves that variables can be fixed or constraints removed. When the lower and upper bounds meet within tolerance, the tree is a certificate relative to the formulation and arithmetic.

This perspective improves practice. A solver log is not merely progress animation; it is a trace of the evolving proof. Rapid incumbent improvement with a stubborn bound suggests a weak relaxation. A strong bound with no feasible solution suggests difficult primal structure. Exploding nodes may indicate symmetry, poor branching, or missing formulation strength. Numerical warnings can invalidate the apparent proof. The modeler should read the trace structurally rather than tune parameters blindly.

Alternative representations expose different inference. Constraint programming is strong when domains and global constraints propagate combinatorial consequences. SAT and SMT solvers exploit clauses, conflict analysis, and theory reasoning. Dynamic programming exploits a small sufficient state. Network algorithms exploit total unimodularity and graph structure. Submodular optimization exploits diminishing returns. A generic MILP can express many of these problems, but expressive completeness is not the same as inferential fitness (Papadimitriou and Steiglitz 1982; Schrijver 2003; Nemhauser and Wolsey 1988).

In discrete optimization, an incumbent and a bound answer different questions. The incumbent shows an implementable value if it is truly feasible in the domain. The bound shows how much the formulation could still improve. A small gap can support stopping; a large gap does not mean the incumbent is poor relative to current practice.

Report four comparisons: incumbent versus operational baseline, incumbent versus best known alternative, bound versus incumbent, and real-world sensitivity versus the gap. An incumbent that reduces patient delay materially and is stable under duration uncertainty may justify use despite a loose proof bound. Conversely, a certified optimum that exploits unrealistic no-show estimates should be rejected.

Feasibility deserves independent repair and checking. Solver tolerances can permit small algebraic violations that become meaningful after rounding or unit conversion. Recompute every hard constraint from exported decisions in original units. For combinatorial designs, run a domain checker that does not share indexing code with the model.

Global, black-box, and surrogate search

Real objectives contain kinks, discontinuities, multiple basins, simulator failures, discrete transitions, and unknown derivatives. The method must match the irregularity.

Nonsmooth structure

Convex nonsmooth functions retain global tractability but require generalized derivatives. A subgradient $g \in \partial f(x)$ satisfies

$$f(y) \geq f(x) + g^T(y - x) \quad \text{for all } y. \tag{38}$$

Subgradient methods are robust but can be slow. Proximal-gradient methods are effective for composite objectives $f(x) + r(x)$ with smooth f and simple nonsmooth r:

$$x_{k+1} = \mathrm{prox}_{\alpha r}\big(x_k - \alpha \nabla f(x_k)\big). \tag{39}$$

This pattern underlies sparsity-inducing regularization, constrained projections, and operator splitting.

Nonconvex landscapes

Nonconvexity removes the local-to-global guarantee. Multiple starts, continuation, homotopy, deterministic global methods, relaxations, and problem-specific initialization become important. In large machine-learning problems, the practical target is often a solution with good validation behavior rather than a certified global optimum; training loss and decision utility must not be confused.

Deterministic global optimization uses interval bounds, convex envelopes, spatial branch-and-bound, or algebraic moment relaxations. These methods can certify globality but may scale poorly. When evaluations are expensive, surrogate models approximate the objective and acquisition functions trade exploration against exploitation (Jones, Schonlau, and Welch 1998; Shahriari et al. 2016).

Derivative-free and zeroth-order methods

Direct-search methods compare carefully chosen points without estimating a full derivative. Model-based methods fit local linear or quadratic surrogates. Evolution strategies and simultaneous perturbation estimate directions stochastically. Derivative-free methods are appropriate when gradients are unavailable, unreliable, discontinuous, or more expensive than evaluations.

Metaheuristics

Simulated annealing accepts some uphill moves with a temperature-dependent probability, echoing statistical mechanics (Kirkpatrick, Gelatt, and Vecchi 1983). Evolutionary algorithms vary and select populations (Holland 1975). Particle swarm methods use social and individual memory (Kennedy and Eberhart 1995). Ant-colony optimization reinforces promising paths (Dorigo and Stützle 2004). These methods are flexible, parallelizable, and often effective on mixed or black-box spaces. Their weakness is not that they are "nonmathematical," but that their guarantees are usually weaker and their performance is sensitive to budget, representation, and tuning.

A fair comparison must equalize evaluation budgets, include strong baselines, report variability across seeds, and disclose tuning effort. A sophisticated heuristic compared with an untuned baseline proves little.

Hybrid search

The strongest practical systems often hybridize: global exploration plus local refinement; learned proposals plus exact verification; evolutionary outer loops plus gradient inner loops; constraint propagation plus MILP bounds; human aesthetic selection plus generative variation. Hybridization composes complementary morphisms rather than seeking one universal optimizer.

Nonconvex continuous problems combine local geometry with global uncertainty. Deterministic global methods partition domains and compute valid bounds; they can certify small or specially structured problems but scale poorly. Multi-start local search samples basins without proving coverage. Simulated annealing, evolutionary algorithms, particle or swarm methods, and hybrid metaheuristics explore irregular spaces, but their success depends on representation, variation operators, budgets, and problem structure (Kirkpatrick, Gelatt, and Vecchi 1983; Holland 1975; Kennedy and Eberhart 1995; Dorigo and Stützle 2004).

Black-box optimization adds an information-design problem: where should the next evaluation occur? Bayesian optimization uses a probabilistic surrogate and acquisition rule to trade improvement against uncertainty (Jones, Schonlau, and Welch 1998; Shahriari et al. 2016). It is effective for expensive, moderately dimensional evaluations, but the surrogate can become confidently wrong outside observed regions. Batch evaluation, failures, categorical variables, multi-fidelity simulation, and constraints require explicit treatment.

No-free-lunch results do not say that algorithm choice is futile. Under uniform averaging over an appropriate finite, permutation-closed function class, non-revisiting black-box methods have equal aggregate performance (Wolpert and Macready 1997). Real success comes precisely from nonuniform structure: smoothness, locality, sparsity, invariance, low effective dimension, generative regularity, or a distribution of instances. The correct lesson is to state the inductive bias and test it.

When evaluation is expensive, a surrogate becomes an intermediate model with its own boundary. Record training design, representation, predictive uncertainty, calibration domain, and failure handling. Use acquisition functions that respect constraints and experimental risk. Never treat a high surrogate score as evidence until the candidate is evaluated with the authoritative process.

The optimizer will seek regions favorable to surrogate error. Trust regions and uncertainty penalties reduce exposure; adversarial and space-filling tests challenge the model. Multi-fidelity methods can combine cheap and expensive simulations, but bias between fidelities must be modeled. A large number of cheap runs does not wash away systematic error.

A heuristic is strongest when every move preserves hard invariants by construction. A routing move can preserve vehicle capacity and route continuity; a schedule move can preserve qualification and precedence; a geometric mutation can preserve manufacturable parameterization. This shrinks repair cost and prevents an apparently productive search from spending most evaluations on impossible candidates.

When feasibility cannot be preserved, separate violation by meaning. Safety, law, and physical impossibility should not share a generic penalty with preferences. Use lexicographic repair or a feasibility restoration phase. Report the rate and severity of violations and verify exported candidates independently.

Hybridize with exact structure. Relaxations can provide bounds and guide variables; heuristics can produce incumbents; constraint propagation can reject moves; local search can refine.

State which guarantees survive. If a learned component proposes actions, place a deterministic checker at the interface.

Allocating evaluations across seeds, starts, fidelities, and instances is itself an optimization. A single long run estimates one trajectory. Multiple starts estimate basin variability. Repeated runs estimate stochastic stability. Diverse instances test transfer. High-fidelity confirmation tests substantive validity.

Predeclare the budget split before selecting a champion. Keep a final confirmation budget untouched. Use performance profiles or empirical cumulative distributions across instances rather than a single average. Report time to first feasible, time to target, best value, and failure rate.

Energy and memory can matter as much as runtime. Include them when methods operate at scale or repeat frequently. A slightly better solution can be dominated by its search burden and delayed implementation.

Solver portfolios, comparison, and infeasibility

For consequential problems, compare a small portfolio rather than anointing one algorithm. The portfolio can include a transparent baseline, a structure-exploiting exact or convex method, and one exploratory heuristic. Equalize evaluation budgets and implementation attention. Measure wall time, energy, memory, feasibility rate, best value, bound, variability, and sensitivity to seeds. Use representative and adversarial instances, not only the examples on which the model was developed.

Hybrid methods often dominate ideological purity. A heuristic can generate incumbents for branch-and-bound; a relaxation can guide a population method; a learned policy can propose branches while a conventional solver preserves correctness; a local method can refine Bayesian candidates; a human can repair feasibility after inspecting a near solution. The interfaces should state which guarantees survive the hybrid composition.

A clinic must schedule imaging appointments with different durations, preparation requirements, equipment, staff, urgency, and no-show risk. A naive objective minimizes unused capacity. It will overbook, compress recovery, and transfer uncertainty to patients and staff.

A stronger formulation uses discrete assignment and start variables, precedence and equipment constraints, staff skills, cleaning and safety buffers, accessibility needs, and reserved urgent capacity. Objectives distinguish patient delay, overtime, schedule disruption, and unused capacity. Some requirements—clinical readiness, safe staffing, equipment compatibility, and urgent access—are constraints. The model retains slack because variability creates queues.

Three representations illuminate different structure. A time-indexed MILP gives direct capacity accounting and can be large. A constraint-programming model represents intervals and resource conflicts naturally. A two-stage stochastic model schedules first and repairs after attendance or duration is observed. A simulation evaluates the policy under realistic arrival and recovery behavior. Comparing them is more informative than tuning one solver.

The formulation must include the human workflow. A mathematically available ten-minute slot can be unusable if transport, consent, preparation, or interpretation is omitted. Staff review of rejected and extreme schedules is a validation method. If the optimizer persistently uses a "legal" pattern that staff refuse, the pattern is either an unmodeled burden or a governance conflict.

Algorithm comparisons require equal problem definitions, stopping rules, hardware disclosure, preprocessing, and tuning budgets. Separate author time from machine time. Report variability over seeds and instances. Avoid counting an infeasible solution as a better objective. Preserve solver logs, versions, parameters, and checks.

Benchmarking can itself overfit. Keep a final untouched instance set and include structurally shifted cases. A portfolio chosen after repeated testing on the same benchmark inherits the optimizer's curse from Chapter 17. Report selection procedure and final evaluation separately.

Infeasibility can reveal conflicting requirements, bad data, unit errors, or a genuinely impossible mandate. Use irreducible inconsistent subsets, feasibility relaxation, dual certificates, and domain review. Do not immediately soften every constraint.

Classify constraints by authority and meaning, then ask which change is legitimate. A unit or data correction differs from relaxing a service promise; a design-variable expansion differs from purchasing away a right. The report should show the smallest conflicting sets and alternative ways to restore feasibility.

Sometimes the most valuable output is proof that a requested combination is impossible; surface it for priority and capacity decisions rather than hide it in a penalty.

Field checklist

- Map domain objects to machine operations; label every transformation.
- Use solver traces to assess formulation, feasibility, proof, and representation fit.
- Compare portfolios under equal budgets, seeds, and implementation effort; report bounds and guarantees.
- Challenge inductive bias with shifted, adversarial, and structurally different cases.

Translation lens. Structured optimization transfers as a typed compiler pipeline; record which semantics and guarantees survive each lowering step.

Chapter 6 · Uncertainty, multiple objectives, games, and hierarchy

Uncertainty, risk, and four attitudes to action

Uncertainty changes what "best" means. Minimizing expected loss

$$\min_x \ \mathbb{E}_\xi[f(x,\xi)] \tag{40}$$

is appropriate only when a probability model is credible and average performance captures the decision maker's concern. Rare harms, ambiguity, nonstationarity, and irreversibility require other criteria.

Stochastic programming

Two-stage stochastic programming separates a here-and-now decision x from recourse $y(\xi)$ after uncertainty is observed:

$$\min_x \ c^T x + \mathbb{E}_\xi[Q(x,\xi)], \tag{41}$$

where

$$Q(x,\xi) = \min_y \{q(\xi)^T y : W(\xi)y = h(\xi) - T(\xi)x,\ \ y \geq 0\}. \tag{42}$$

Scenario trees generalize to multiple stages but grow rapidly. Decomposition, sampling, and approximate dynamic programming manage this complexity (Shapiro, Dentcheva, and Ruszczyński 2021).

Chance constraints and risk measures

A chance constraint

$$\Pr(g(x,\xi) \leq 0) \geq 1 - \varepsilon \tag{43}$$

limits violation probability, but can be hard to compute and may hide the severity of violations. Value-at-Risk is a quantile; Conditional Value-at-Risk (CVaR) averages losses beyond that quantile. For loss $L(x,\xi)$ and confidence α,

$$\mathrm{CVaR}_\alpha(L) = \min_\eta \left[\eta + \tfrac{1}{1-\alpha}\mathbb{E}[(L-\eta)_+]\right]. \tag{44}$$

Here $(z)_+ = \max(z,0)$ is the positive part. This representation makes CVaR optimization tractable in many sampled problems (Rockafellar and Uryasev 2000).

Robust optimization

Robust optimization protects against an uncertainty set $\mathcal{U}$:

$$\min_x \ \max_{\xi \in \mathcal{U}} f(x,\xi). \tag{45}$$

The set encodes epistemic judgment. If too small, protection is illusory; if too large, solutions are unusably conservative. Budgeted uncertainty provides a tunable price of robustness (Ben-

Tal, El Ghaoui, and Nemirovski 2009; Bertsimas and Sim 2004). Distributionally robust optimization instead considers a set $\mathcal{P}$ of plausible probability distributions:

$$\min_{x} \sup_{P \in \mathcal{P}} \mathbb{E}_P[f(x,\xi)]. \tag{46}$$

Ambiguity sets may be defined by moments, divergences, or optimal-transport distances.

Bayesian decision and value of information

Bayesian decision theory integrates posterior uncertainty with utility. Information has value when it can change action. The expected value of perfect information compares decisions made before and after revealing uncertainty; the expected value of sample information evaluates a proposed experiment. This turns measurement design into an optimization problem.

Reliability and resilience

Reliability optimizes the probability of performing within specified conditions. Resilience also considers recovery, graceful degradation, adaptability, and performance outside nominal assumptions. Maximum efficiency can remove slack and diversity that resilience needs. A system should often optimize a portfolio of nominal performance, tail risk, recoverability, and option value.

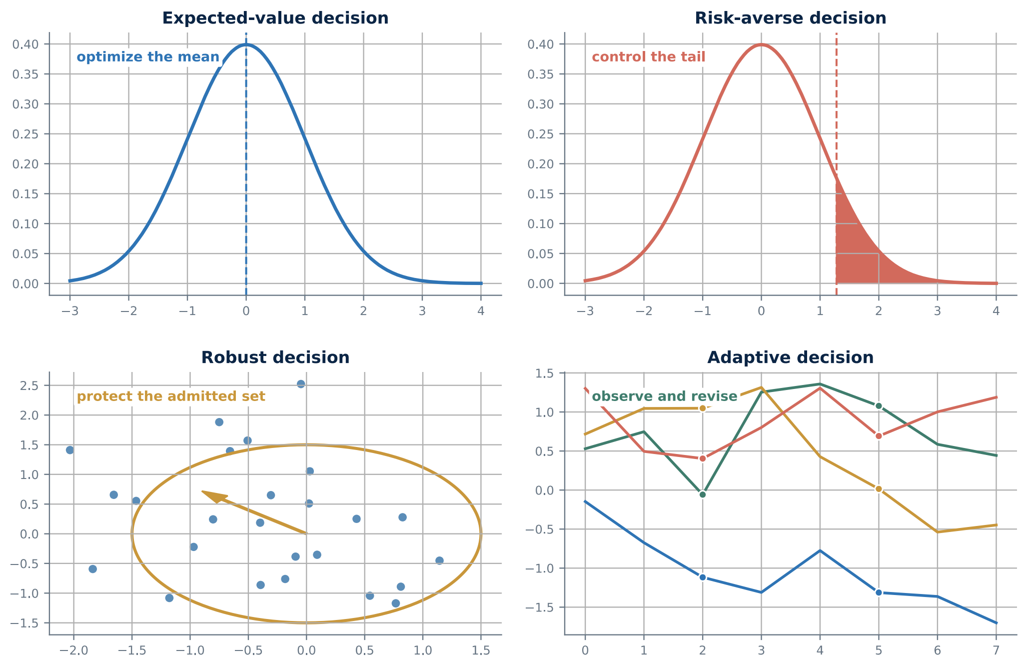


Figure 7. A 2 × 2 comparison of expected-value, risk-averse, robust, and adaptive decisions under uncertainty.

Figure 7 distinguishes four questions often blurred together: what is best on average, what limits tail loss, what survives an uncertainty set, and what can adapt after observation.

Uncertainty is not a single mathematical substance. A probability distribution, a set of plausible parameters, an ambiguity set of distributions, and a stream of future observations encode different knowledge. Choosing among them is an epistemic decision before it is an algorithmic decision.

An expected-value formulation is appropriate when repeated outcomes, calibrated probabilities, and approximately linear consequence aggregation make the mean decision-relevant. A risk measure such as CVaR emphasizes a tail while retaining probabilistic structure. Robust optimization protects against every point in an admitted set, which is valuable for feasibility but can be overconservative. Distributionally robust optimization protects against a family of distributions. Adaptive optimization reserves recourse so that actions can change after information arrives.

The four attitudes can recommend different actions even with the same raw observations. Consider flood defense. An expected-cost model may favor a barrier sized for the center of a climate distribution. A risk-averse model increases protection against severe tails. A robust model protects against an entire scenario envelope. An adaptive pathway builds an initial measure while preserving land, permits, and trigger rules for later expansion. The last option values flexibility, not just terminal performance.

Uncertainty sets and ambiguity sets should be calibrated. A set chosen for algebraic convenience can create meaningless safety. Historical samples may underrepresent structural change. Expert scenarios may omit correlated failures. Robustness should be tested against out-of-set events, because the boundary of admitted uncertainty is itself an assumption.

Begin by asking what can be observed before action, what can be observed afterward, and what can never be known in time. This divides decisions into here-and-now commitments and recourse. A hospital may choose base staffing before demand, call additional staff after a forecast, and divert patients only after local capacity is observed. A static “robust schedule” can be needlessly conservative if recourse is available and unsafe if the recourse is infeasible.

Scenario trees represent staged revelation. Their branches should reflect conditional information, not merely sampled futures. If weather is observed before a transport decision, policies may depend on it. If a supply failure is discovered only after dispatch, they may not. Nonanticipativity constraints enforce this information structure.

Probability is warranted when frequencies or calibrated beliefs support it. Robust sets are warranted when feasibility must survive bounded error. Ambiguity sets are warranted when distributions are estimated. Narratives are warranted when mechanisms are plausible but not meaningfully probabilized. A mature decision can use all four: expected operations, tail-risk budget, robust safety envelope, and narrative stress test.

Robustness should be tested by group and location. A policy with stable total performance can transfer variability to a small population. For each scenario, report group outcomes and constraint violations; then inspect which scenarios and groups drive the robust solution. An uncertainty set built from aggregate error can underrepresent sparse groups.

Resilience adds actions after the set fails: detection, fallback, mutual aid, recovery, compensation, and learning. Robustness is resistance within a model of uncertainty. Resilience is continued function and repair when that model is wrong. Keeping the terms distinct prevents a large safety margin from masquerading as recovery capability.

More information is valuable only if it can change action. For a prospective observation Y, the expected value of sample information is the difference between the best expected consequence after observing Y and the best consequence now, before subtracting study cost. The value is decision-specific: a more precise parameter can have zero value if all plausible estimates select the same policy.

Study design should target uncertainty near decision boundaries, active constraints, or scenario failures. Compare the expected value of information with delay, burden, and ethical cost. In urgent settings, act with a safe provisional policy while collecting information for revision.

Flexibility is related and distinct. A modular design, reserved capacity, reversible contract, staged project, or feedback policy can reduce the need to know now. Its value appears because future action can respond. A static optimization that omits recourse will undervalue flexibility and overvalue precise forecasting.

CVaR summarizes the mean loss in a selected tail and supports convex formulations under common conditions (Rockafellar and Uryasev 2000). The confidence level and loss variable require interpretation. Tail financial cost, deaths, service delay, and ecological threshold crossings are not commensurate merely because each admits CVaR.

Test whether the risk measure hides severity or frequency. A chance constraint limits violation probability and can ignore magnitude. Worst-case loss protects magnitude and can be driven by a dubious extreme. Entropic or variance penalties respond differently to distributions. Use several views and explain which consequence they protect.

For group harms, compute tails by group as well as population. An aggregate tail can be dominated by a large group and conceal systematic severe outcomes in a smaller one. If a harm is unacceptable, use a constraint or veto rather than relying on a risk weight.

Deep uncertainty occurs when parties cannot agree on a probability model, consequences, or future system. For long-lived infrastructure and policy, forcing a precise distribution can make the output look decisive while moving disagreement into hidden assumptions.

Use exploratory modeling. Generate many plausible futures across physical, economic, behavioral, and institutional dimensions. Evaluate policies, identify vulnerability regions, and discover which assumptions cause failure. Seek robust or low-regret policies and adaptive triggers rather than one scenario-optimal plan.

Scenario diversity should include mechanisms, not only parameter grids. Correlated shocks, political delay, technology change, migration, and strategic response can alter structure. Narrative scenarios can expose conditions no smooth distribution represents.

Robustness is not unanimity. A policy can be acceptable across models and contested on distribution or rights. Present vulnerability and trade-offs to the legitimate decision process. Keep monitoring and funding tied to triggers so adaptation is executable.

Risk models often assume independent failures because they are easier to combine. Shared supplier, software, climate, institution, data, or incentive creates common cause. Correlation can rise precisely during stress.

Map dependency explicitly. Use copulas or joint scenarios where data supports them, and stress structural common causes where it does not. Diversification by label is not diversification if assets share hidden exposure. Redundancy should vary technology, location, control, and organization where appropriate.

After an event, update the dependency map rather than only marginal failure rates. A new common cause changes system architecture.

Multiple objectives and uncertain preferences

When consequences conflict, there may be no single best solution. For objectives $F(x) = \big(f_1(x), \dots, f_k(x)\big)$, a feasible point x dominates y for minimization when $f_i(x) \le f_i(y)$ for all i and the inequality is strict for at least one component. A point is Pareto efficient if no feasible point dominates it.

The Pareto front makes trade-offs visible. It does not choose among them. Selection requires preferences, priorities, negotiation, or governance (Miettinen 1998; Deb 2001).

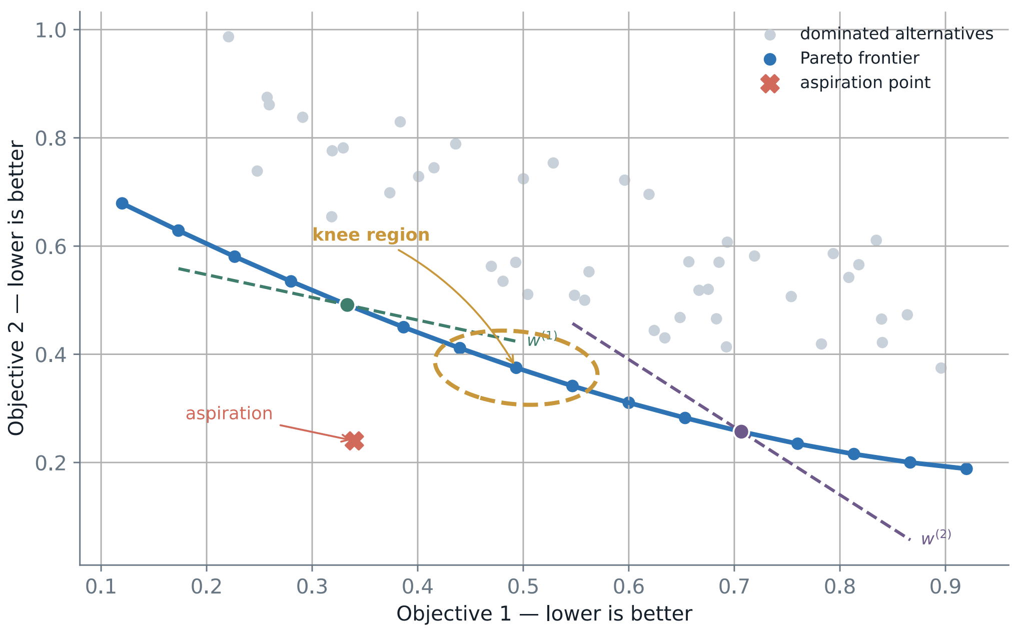


Figure 8. A Pareto front showing dominated designs, nondominated alternatives, a knee region, aspiration point, and the effect of changing scalarization weights.

Figure 8 shows that a Pareto frontier exposes trade-offs but does not choose for the decision maker; knees, aspirations, and scalarization slopes encode additional judgments.

Scalarization and preference articulation

Weighted sums are simple but scale-sensitive and incomplete for nonconvex fronts. The ε-constraint method optimizes one objective while bounding others. Reference-point methods minimize distance to aspirations. Lexicographic optimization protects priority order. Interactive methods alternate optimization with preference feedback. Evolutionary multiobjective algorithms approximate diverse fronts when gradients or convexity are unavailable (Deb et al. 2002).

Games

In a game, each agent's result depends on other agents' choices. A Nash equilibrium x^* satisfies

$$u_i(x_i^*, x_{-i}^*) \geq u_i(x_i, x_{-i}^*) \quad \text{for every player } i \text{ and action } x_i. \tag{47}$$

No player can profit by unilateral deviation. For every finite game, Nash's existence theorem guarantees at least one equilibrium in mixed strategies; a pure-strategy equilibrium need not exist. Equilibrium is not the same as collective optimality, fairness, or stability under learning. Mechanism design reverses the problem: choose the rules so that strategic behavior produces desirable outcomes (Nash 1950; von Neumann and Morgenstern 1944; Myerson 1981).

Bilevel optimization

Bilevel problems contain an optimization problem inside a constraint:

$$\min_{x,y} F(x,y) \quad \text{subject to} \quad y \in \arg\min_{z} \{f(x,z) : g(x,z) \leq 0\}. \tag{48}$$

They model leader-follower games, pricing, hyperparameter tuning, adversarial learning, infrastructure planning, and policy response. Even linear bilevel problems can be hard because the lower-level solution map is discontinuous or set-valued (Dempe 2002).

Social choice and incomplete order

Collective decisions may not admit a consistent scalar social welfare function satisfying attractive axioms. Arrow's theorem is a warning against assuming that individual rankings aggregate without loss or paradox (Arrow 1951). In such cases, optimization must be coupled to constitutional rules, deliberation, or plural criteria. The output may be a justified process and a defensible set of options, not a mathematically unique winner.

Multiobjective optimization separates two tasks: discovering what is achievable and deciding what is preferred. A Pareto frontier performs the first task. It preserves the partial order induced by dominance and reveals opportunity costs. It does not determine weights, declare a knee uniquely best, or transform political disagreement into geometry.

Different selection rules answer different preference questions. Weighted sums assume compensability and are sensitive to scale. The ε-constraint method turns some objectives into limits. Lexicographic optimization protects priority. Reference-point methods measure shortfall from aspirations. Outranking methods can retain vetoes and incomparability. Interactive methods allow preferences to develop while alternatives become visible (Miettinen 1998; Deb 2001).

The frontier itself is uncertain. Estimated objectives, finite simulation, and model error can reorder candidates. Report confidence regions or dominance probabilities where possible. Avoid showing a thin deterministic curve when the evidence supports a band. A robustly nondominated option may be more useful than a nominal knee that disappears under small perturbations.

Preferences can be uncertain because stakeholders disagree, consequences are hard to compare, or alternatives reveal new values. Do not represent every form as a probability distribution over weights. Preserve intervals, vetoes, ordinal comparisons, or incomparability when they match the evidence.

Robust preference analysis asks which alternatives remain nondominated over a set of plausible weights or aspiration levels. A candidate chosen only under a narrow weight range needs a stronger justification than one that performs acceptably across many. Value-of-information analysis can ask whether another study would change the chosen region of the frontier rather than merely narrow estimates.

Deliberation should show the effect of moving a criterion from objective to constraint. This change is often more important than tuning a weight. A minimum service level, carbon budget, or rights condition changes which alternatives are admissible and expresses a different ethical structure.

Games, mechanisms, and strategic stress tests

Optimization with multiple agents cannot assume the world holds still while one actor improves. In games, each agent's objective depends on others' actions. A Nash equilibrium is stable against unilateral deviation under the model; finite games have a mixed-strategy equilibrium, but not necessarily a pure one (Nash 1950; von Neumann and Morgenstern 1944). Equilibrium does not imply efficiency, justice, uniqueness, or convergence of actual learning dynamics.

Mechanism design treats rules as variables. A matching system, auction, congestion charge, admissions policy, or platform ranking changes incentives and information revelation. Desirable properties commonly conflict: efficiency, incentive compatibility, budget balance, privacy, simplicity, stability, and distributional fairness cannot always be achieved simultaneously (Myerson 1981; Gale and Shapley 1962). Simulation with strategic agents can reveal exploits but does not prove absence of new ones.

Bilevel optimization is a mathematical representation of hierarchy. The leader chooses x anticipating a follower solution $y(x)$. The follower map may be set-valued, discontinuous, or strategically misreported. Selecting one favorable follower optimum without specifying tie-breaking can make a model optimistic by construction. Pessimistic variants, equilibrium constraints, and sensitivity analysis expose this ambiguity.

For a policy or platform, list actors, information, actions, payoffs, and adaptation speed. Then ask how each actor could improve their outcome under the rule. Test honest participation, gaming, collusion, withdrawal, and creation of new intermediaries. Include administrators and model owners, not only end users; institutions optimize reports too.

Equilibrium analysis can identify stable responses under strong assumptions. Agent-based or behavioral simulation can explore heterogeneous adaptation. Historical and qualitative evidence can reveal tactics the model omitted. None proves that the mechanism is strategy-proof in the world.

When an equilibrium is undesirable, change the mechanism rather than exhort participants. When multiple equilibria exist, implementation history and coordination matter. A rule can be “efficient” at one equilibrium and predictably settle elsewhere. Publish equilibrium selection assumptions.

A one-shot equilibrium can be a poor description of repeated institutions. Participants learn, build reputations, coordinate, retaliate, and invest in influence. The rule changes the future population by entry and exit. Include repeated incentives, information feedback, and the cost of participation.

Robust mechanisms should be simple enough to understand, monitor, and appeal. Complexity can create strategic advantage for well-resourced actors even if formal rules are symmetric. Test not only equilibrium allocation but administrative error, communication, enforcement, and contestability.

When a mechanism fails, distinguish bad equilibrium, manipulation, infeasible enforcement, and illegitimate objective. Each calls for a different redesign.

A flood-adaptation decision room

A defensible flood-adaptation process can be organized in three stages. First, generate a portfolio of infrastructure, nature-based measures, building codes, warning systems, insurance changes, and retreat options. Second, simulate them across sea-level, rainfall, economic, ecological, and demographic scenarios. Third, present a frontier of expected cost, severe loss, ecological effect, displacement, and option value to an accountable decision room.

Some criteria remain hard constraints: legal rights, minimum safety, protected habitats, and procedural requirements. Others are uncertain objectives. The model can identify dominated portfolios and important sensitivities, while deliberation selects an adaptive pathway. Monitoring triggers are part of the policy: observed sea level, maintenance condition, population exposure, and distributional harm can activate the next stage. Optimization supports the constitutional process; it does not replace it.

Field checklist

- Identify whether uncertainty is probabilistic, set-valued, distributional, adversarial, or sequential.
- Calibrate uncertainty objects and stress events outside their boundaries.
- Preserve a frontier long enough for preferences and disagreement to remain visible.
- Show uncertainty around the frontier, not only uncertainty inside each simulation.
- State whether strategic agents, equilibrium selection, and tie-breaking are modeled.
- Treat recourse, reversibility, and option value as design variables.
- Keep rights and vetoes out of compensatory scalarizations unless their status is explicitly changed.

Translation lens. Probability, dominance, equilibrium, and hierarchy are different structures. They can be composed, but they should not be collapsed. Translation succeeds when uncertainty type, preference incomparability, strategic response, and authority remain visible in the target problem.

Chapter 7 · Dynamics, transport, category theory, algorithms, and verification

Decisions through time

Dynamic optimization selects actions whose consequences unfold through state. Let

$$s_{t+1} = T(s_t, a_t, w_t) \tag{49}$$

describe a system with state s_t, action a_t, and disturbance w_t. A policy π maps observations to actions. The objective may discount future costs:

$$J^\pi(s_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t\, c(s_t, a_t)], 0 \ \le\ \gamma\ <\ 1, |c(s,a)|\ \le\ C\ <\ \infty. \tag{50}$$

Bellman's principle

Under the assumptions stated with Equation (50), the optimal value function satisfies

$$V^*(s) = \min_a [c(s,a) + \gamma\, \mathbb{E}[V^*(s') \mid s, a]]. \tag{51}$$

This recursive decomposition is the foundation of dynamic programming, Markov decision processes, and much reinforcement learning (Bellman 1957; Puterman 1994; Sutton and Barto 2018). Its power comes with the curse of dimensionality: the state space can grow exponentially.

Optimal control

For continuous dynamics $\dot{x} = f(x, u, t)$ with terminal cost Φ and running cost L, define the objective

$$J[u] = \Phi\big(x(T)\big) + \int_0^T L\,(x(t), u(t), t)\, dt, \tag{52}$$

Pontryagin's maximum principle introduces a costate λ and Hamiltonian

$$H(x, u, \lambda, t) = L(x, u, t) + \lambda^T f(x, u, t). \tag{53}$$

Necessary conditions couple state, costate, and pointwise control selection. The Hamilton-Jacobi-Bellman equation provides a dynamic-programming view. Linear-quadratic control yields analytic structure; nonlinear constrained control typically requires numerical methods (Pontryagin et al. 1962; Bryson and Ho 1975).

Model predictive control

Model predictive control repeatedly solves a finite-horizon problem, implements the first action, observes the new state, and solves again. The repeated replanning supplies feedback and handles constraints, while terminal costs and sets help establish stability (Rawlings, Mayne, and Diehl 2017).

Reinforcement learning

Reinforcement learning estimates values or policies from interaction. Model-free methods avoid an explicit transition model but do not avoid assumptions: exploration policy, reward design, observability, stationarity, and data coverage remain decisive. Reward misspecification can produce capable optimization of an unintended behavior. Offline RL adds extrapolation risk because the learned policy may choose actions unsupported by historical data.

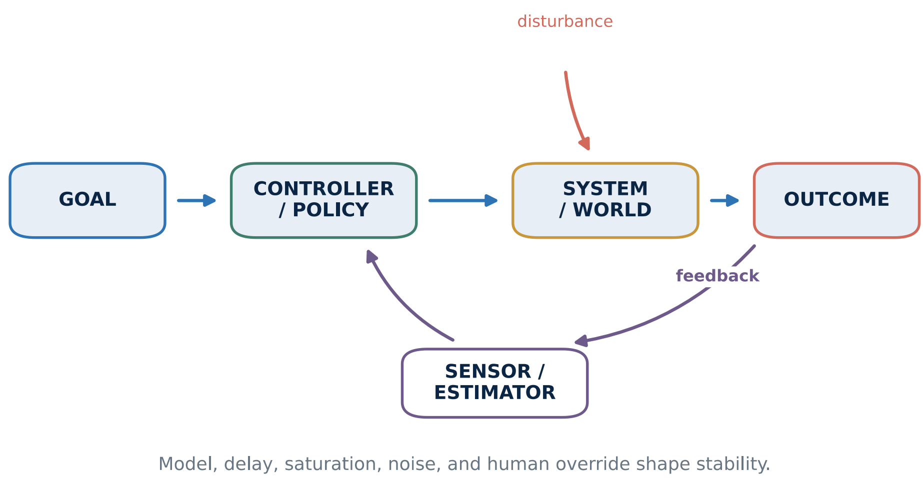


Figure 9. Closed loop from goal through controller or policy, system or world, and outcome, with disturbance and sensor or estimator feedback.

Figure 9 locates optimization inside feedback: the policy acts on a plant, an estimator reconstructs state, and observed consequences update both model and objective.

Partial observability and belief states

When the state is hidden, observations update a belief distribution. The belief is sufficient in theory but often high-dimensional. Filtering, state estimation, memory, and active sensing become coupled optimization problems. Kalman filtering provides an optimal linear-Gaussian estimator under its assumptions (Kalman 1960).

Dynamic optimization replaces a point with a trajectory or policy. The state must contain enough information to evaluate future consequences; the action changes both immediate cost and later opportunity. Bellman's principle expresses this recursively, but the practical challenge is constructing a valid state. If relevant history is omitted, the Markov model is false. If every historical detail is retained, computation becomes impossible.

For a controlled system $s_{t+1} = T(s_t, a_t, \xi_t)$, a finite-horizon value function satisfies

$$V_t(s) = \min_{a \in \mathcal{A}(s)} \mathbb{E}\big[c_t(s, a, \xi_t) + V_{t+1}\big(T(s, a, \xi_t)\big)\big]. \tag{54}$$

Equation (54) is simultaneously an optimization equation and an interface: the future is summarized by V_{t+1}. Dynamic programming, model predictive control, and reinforcement learning differ in how they represent or learn that interface. Exact tables are possible for small states; approximate value functions, policy gradients, tree search, and fitted models trade bias, variance, and computation at larger scales (Bellman 1957; Puterman 1994; Sutton and Barto 2018).

Feedback changes the meaning of robustness. An open-loop plan must survive uncertainty without observing it; a feedback policy can adapt but depends on sensors, estimators, communication, and actuation. Delay, saturation, partial observability, and human override can destabilize a policy that is optimal in an ideal simulator. Model predictive control mitigates some mismatch by repeatedly solving a finite-horizon problem, yet it still requires terminal assumptions, feasible fallback behavior, and computational deadlines (Rawlings, Mayne, and Diehl 2017; Åström and Murray 2008).

A state is sufficient only relative to a prediction and decision horizon. In maintenance, current sensor values may predict immediate failure while maintenance history and environment are needed for long-term planning. In education, a score history may predict a test and omit the relationships and opportunities an intervention changes. State construction is therefore a claim that histories mapped to the same state are interchangeable for the future consequences being optimized.

Test the claim by searching for histories with the same encoded state and different outcomes or optimal actions. Include delayed effects, path dependence, accumulated exposure, and strategic memory. If the representation fails, enrich the state, shorten the claim horizon, use a partially observed model, or retain human context at the interface.

Approximate state can still be useful. The report should state the information loss and how the policy detects when it matters. A compact representation with a safe fallback may be preferable to a high-dimensional state that cannot be validated.

Consider a building-energy system. A weather and occupancy forecaster sends distributions to a scheduler; the scheduler sends temperature and power trajectories to local controllers; controllers act on equipment and return measurements.

The forecast interface specifies units, spatial and temporal resolution, quantiles or scenarios, calibration window, issue time, missing-data behavior, and conditions outside support. The scheduling interface specifies targets, comfort and safety envelopes, flexibility, price or carbon signals, and priority when constraints conflict. The control interface specifies admissible set points, ramp limits, timing, acknowledgment, local safety override, and fallback.

If the scheduler receives only point forecasts, uncertainty is lost. If the controller receives a trajectory without flexibility, it cannot trade local disturbance against system goals. If local controllers hide override events, the scheduler learns from a false account of its own actions. Passing richer objects—scenario sets, value functions, feasible response envelopes, and status—can preserve composability.

This is why a box-and-arrow diagram needs typed arrows. The box names a component; the arrow names the contract on which a system claim depends.

Transport and geometry as semantics

Optimization is shaped by the geometry of its space. Euclidean vectors are only one case. Probabilities, rotations, graphs, shapes, strings, and equivalence classes require metrics and topologies that respect their structure.

Optimal transport

Given probability measures μ and ν and a cost $c(x,y)$, Kantorovich's formulation seeks a coupling γ:

$$\min_{\gamma\in\Pi(\mu,\nu)} \int c(x,y)\, d\gamma(x,y). \tag{55}$$

For $c(x,y) = d(x,y)^p$, the optimum defines the p-Wasserstein distance after taking the pth root. Unlike pointwise divergences, transport respects the ground geometry: nearby mass is cheaper to move than distant mass (Villani 2003; Peyré and Cuturi 2019).

Entropic regularization adds

$$\varepsilon\, \mathrm{KL}(\gamma \parallel \mu \otimes \nu) \tag{56}$$

and enables fast Sinkhorn iterations (Cuturi 2013). The regularization smooths and accelerates but also changes the solution; ε is a modeling as well as numerical parameter.

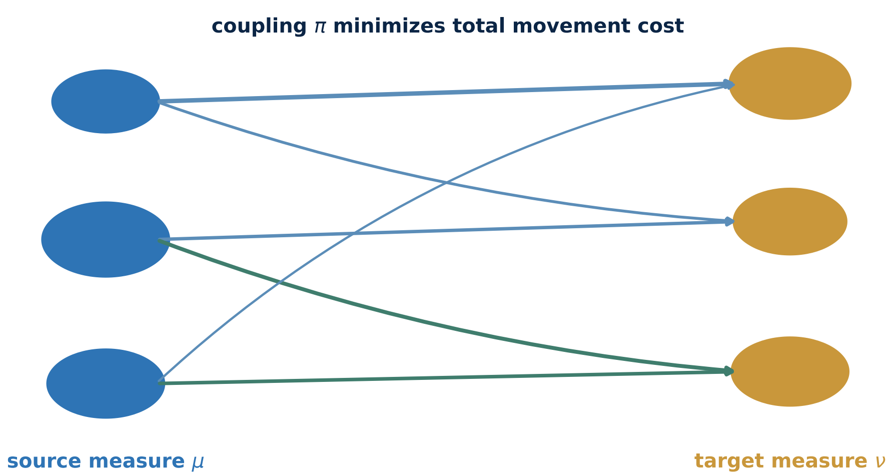


Figure 10. Transport coupling between three source and three target masses; arrow thickness shows moved mass.

Figure 10 makes the ground metric operational: alternative couplings move the same mass, but only the least-cost plan respects the chosen geometry, while entropy spreads transport.

Manifolds and quotient spaces

Rotations lie on $SO(3)$, covariance matrices form a positive-definite cone, and low-rank factorizations have redundant representations. Optimization on manifolds respects constraints intrinsically rather than repeatedly projecting from an ambient space. Quotient geometry identifies parameter values that encode the same object, reducing artificial symmetries.

Topology and shape

Topology distinguishes connectivity, holes, and continuity properties that survive deformation. Topology optimization in engineering asks where material should exist, but topological data analysis also turns persistent features into objectives or constraints. Shape optimization varies boundaries; level-set methods represent them implicitly. These problems often require regularization because unconstrained fine-scale structure exploits the discretization.

Metrics as semantics

A metric says which changes are small. Edit distance, perceptual distance, geodesic distance, Wasserstein distance, and task loss induce different neighborhoods and therefore different local optima. Metric learning and representation learning make the geometry itself a learned object. If the metric fails to preserve the distinctions stakeholders care about, optimization faithfully travels in the wrong space.

Optimal transport compares distributions by the work required to move mass between them. The coupling, not merely a pointwise distance, carries information about how one distribution can become another (Villani 2003; Peyré and Cuturi 2019). Entropic regularization makes large problems computationally tractable and produces diffuse couplings (Cuturi 2013). In machine learning it supports domain comparison; in logistics it represents physical flow; in imaging it aligns shapes; in social analysis it can expose distributional movement. The same metric is not morally neutral: a ground cost encodes which transformations count as nearby.

More generally, geometry is a model of allowable variation. Quotient spaces identify representations considered equivalent; manifolds encode constrained coordinates; graph metrics encode relational paths; information divergences encode statistical distinction. Algorithm performance and substantive meaning both depend on this geometry. A Euclidean norm applied by default can make unlike changes look equal and equivalent changes look far apart.

In optimal transport, the ground cost determines which movement is expensive. For images, Euclidean pixel distance may be physically meaningful; for language, embedding distance may merge antonyms or historical meanings; for social policy, geographic distance can ignore borders, disability, safety, or time. The transport plan is optimal relative to this cost.

Before interpreting a transport distance, test invariance and counterexamples. Which transformations should have zero or small cost? Which distinctions must remain large? Does the cost obey symmetry or triangle inequality, and does the domain require those properties? A nonmetric cost may be correct if direction or irreversible harm matters.

Entropic regularization improves computation and smooths couplings. It also spreads mass, which may be a useful uncertainty model or an artifact. Report the regularization level and test whether substantive conclusions persist as it changes.

Category theory and composition

Category theory asks how structures and transformations compose. This is especially valuable when optimization concerns systems built from subsystems, each with its own variables, constraints, objectives, and interfaces.

A category of open optimization problems

Consider an open component with input interface X, output interface Y, internal decision z, feasibility relation $R \subseteq X \times Z \times Y$, and cost $c: X \times Z \times Y \to \overline{\mathbb{R}}$. It can be represented schematically as a morphism

$$P: X \to Y. \tag{57}$$

Sequential composition connects the output of one component to the input of another. Parallel composition uses a monoidal product:

$$(P_1 \otimes P_2): (X_1 \otimes X_2) \to (Y_1 \otimes Y_2). \tag{58}$$

The total cost might add, take a maximum, or combine in another monoidal way. The choice of monoid is part of the semantics. Decorated cospans and related constructions provide formal languages for open networks whose internal decorations may be circuits, dynamical systems, resource requirements, or optimization data (Fong 2015; Fong and Spivak 2019).

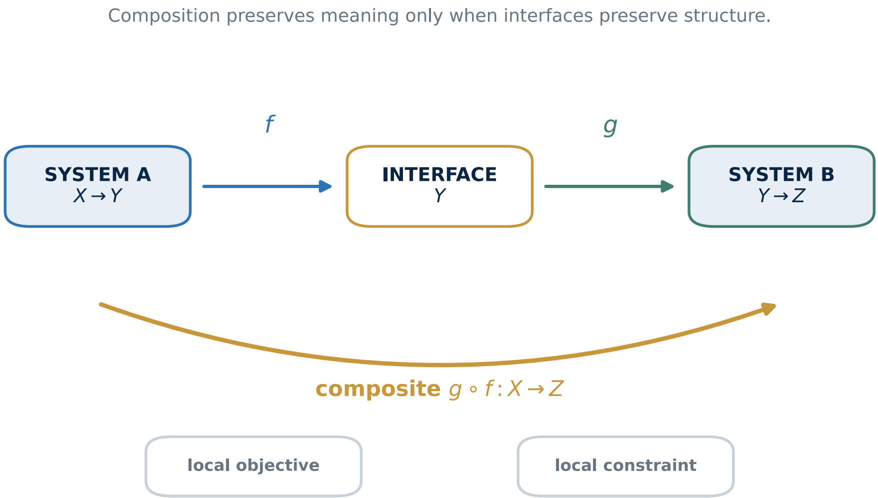


Figure 11. Sequential composition of SYSTEM A and SYSTEM B through an interface, with f, g, g∘f, a local objective, and a constraint.

Figure 11 illustrates the categorical requirement for composition: sequential and parallel components connect safely only when their interfaces, hidden states, costs, and constraints have compatible types.

Why argmin is not generally a functor

It is tempting to define an operation that maps every problem to “its solution” and every problem transformation to a solution transformation. This fails in general. Minimizers may be nonunique; small perturbations may cause discontinuous switches; composition can introduce coupling; and a local optimum of components need not produce a global optimum of the whole. A more faithful object is the set-valued correspondence

$$\mathrm{Argmin}(P) \subseteq \mathcal{X}_P, \tag{59}$$

possibly equipped with order, topology, probability, or approximation structure. Functorial behavior may be recovered only under additional assumptions or by enriching the codomain.

Enrichment and cost

Lawvere observed that generalized metric spaces can be viewed as categories enriched over nonnegative extended reals (Lawvere 1973). Composition corresponds to a triangle inequality; the monoidal operation combines costs. This perspective makes shortest paths, resource accounting, and quantitative semantics categorical without erasing magnitude.

Adjunctions and duality

Adjunctions formalize optimal translations between descriptions. Galois connections link abstraction and concretization; convex conjugacy links primal and dual functions; free and forgetful constructions trade structure against universality. These are not all the same adjunction, but they share a pattern: one transformation is the best approximation compatible with another.

Compositional games and co-design

Open games make strategic interactions composable as morphisms in a symmetric monoidal category (Ghani et al. 2018). Co-design problems represent components by feasibility or resource relations and compose them along interfaces. String diagrams expose information flow, feedback, and parallelism. This can prevent a common systems error: optimizing modules against incompatible local objectives and hoping their optima assemble.

Sheaves and local-to-global consistency

In distributed systems, local decisions may agree on overlaps yet fail to form a global solution. Sheaf-like structures organize local data and compatibility. This is useful for sensor fusion, network constraints, collaborative design, and multi-view evidence. The categorical lesson is that global feasibility is a gluing problem, not merely a sum of local scores.

A practical categorical checklist

1. Identify objects as typed interfaces, not only internal models.
2. Specify sequential and parallel composition.
3. State how costs and constraints compose.
4. Preserve provenance through transformations.
5. Treat feedback explicitly.
6. Check whether local solutions glue to a global one.
7. Record what structure each translation forgets.

Large optimization systems are assembled from components: a forecaster feeds a scheduler, a scheduler feeds a controller, a controller acts on a plant, and measurements return to the forecaster. Each component exposes variables, constraints, costs, and guarantees at an interface. Category theory asks which structures survive sequential and parallel composition (Mac Lane 1998; Spivak 2014; Fong and Spivak 2019).

An object in a category of open optimization problems can represent a typed boundary. A morphism represents a component that relates input and output boundary variables while hiding internal variables. Sequential composition identifies a downstream input with an upstream output; monoidal composition places independent or weakly coupled components side by side. Costs may add, constraints may conjoin, and feasible relations may compose existentially over the hidden interface.

The subtlety is selection. The operation $\arg\min$ is generally set-valued and does not preserve composition. Locally optimal components need not form a globally optimal system; an interface value that is suboptimal for one component can enable a much larger gain downstream. Ties create further trouble: arbitrary local tie-breaking can destroy a feasible or optimal composite. A compositional theory should pass value functions, feasible relations, dual messages, or games across the interface rather than prematurely passing a selected optimizer.

This observation turns category theory from ornament into design discipline. It tells an architect what not to hide. If subsystem A exports only its chosen point, subsystem B cannot price alternatives. If A exports a response curve or value function, the composite retains trade-off information. If privacy or organizational boundaries prevent sharing internal models, dual prices or contracts can serve as compressed interface messages, provided their assumptions are documented.

Suppose a predictor is calibrated, a scheduler is optimal, and a controller is stable. The assembled system is not automatically safe. Calibration may fail after the scheduler changes the input distribution. The controller's stability region may exclude the scheduler's extreme target. Communication delay may violate both models. Human overrides can create an unmodeled hybrid mode.

Write assurance as assume-guarantee contracts. Each component lists assumptions about inputs and environment and guarantees under them. Composition is valid when upstream guarantees satisfy downstream assumptions and feedback does not invalidate upstream conditions. Then test common-cause failures and modes outside every contract.

Category theory supplies the composition grammar; engineering supplies the quantitative contracts; governance supplies authority when a contract fails. None alone is sufficient.

Worked composition: predictor → scheduler → controller

Let a predictor take a history $h \in H$ to a forecast distribution $\mu \in D$. Let a scheduler take μ together with a capacity state $s \in S$ to a target $u \in U$. Let a controller take u and plant state $q \in Q$ to an action $a \in A$. The interfaces are typed: a point forecast cannot silently replace a distribution, a power target cannot enter a temperature port, and a stale state cannot satisfy a freshness contract.

Represent each component by a feasible relation and a value function. For open components $P_1 : X \to Y$ and $P_2 : Y \to Z$ with additive costs, sequential composition hides the shared interface y and retains its best compatible value:

$$V_{P_2 \circ P_1}(x, z) = \inf_y \left[V_{P_1}(x, y) + V_{P_2}(y, z) \right]$$

For the predictor and scheduler, $x = (h, s)$, $y = \mu$, and $z = u$. Composing again with the controller eliminates u only after control cost and feasibility have been included. The composite therefore chooses forecast and schedule interfaces for total downstream value, rather than asking each component to export one locally preferred point.

A two-candidate calculation shows the danger. Suppose forecasts μ_1 and μ_2 have local predictor losses 0 and 1, while their best scheduler-controller costs are 10 and 0. The predictor's isolated argmin exports μ_1 and yields total cost 10; the composed value selects μ_2

and yields total cost 1. The feasible relation and value function compose. The premature selection does not.

Selection-composition criterion. Selection composes when interfaces type-check, feasibility composes exactly, every downstream contribution is represented upstream, the infimum is attained, and minimizers are unique or ties are resolved coherently across components. Otherwise compose feasible relations, value functions, or set-valued argmin correspondences—not isolated winning points.

Proof sketch. With additive cost, exact relational composition, and attained minima, associativity follows by eliminating hidden interfaces in either order: nested infima over μ and u equal one infimum over the feasible pair (μ, u). If the minimizing pair is unique, selection inherits that associativity. With multiple minimizers, independent tie-breaking can choose incompatible representatives; with omitted coupling, even the composed value is wrong.

The hypotheses are design requirements. A failed composition exposes a hidden state, cost, or authority that the interface forgot. At the system boundary, test distribution shift, stability regions, latency, uncertainty, and feedback explicitly.

This is the architectural payoff of categorical language. A component may hide implementation while exporting the structures its neighbors need: types, feasibility, value, uncertainty, provenance, and assume-guarantee conditions. Exporting less is an optimization decision and must itself be justified (Fong and Spivak 2019; Ghani et al. 2018).

Algorithms and verification across the stack

An optimization method is an algorithm only when its inputs, outputs, stopping conditions, resource model, and failure states are specified. The same mathematical iteration can behave differently under finite precision, parallel execution, noisy evaluation, and imperfect derivatives.

Convergence and rates

Convergence asks whether iterates approach an appropriate solution concept. Rates ask how fast. For an L-smooth convex function, gradient descent with suitable step size obtains an objective gap of order $O(1/k)$. Strong convexity yields linear convergence. Accelerated first-order methods can achieve $O(1/k^2)$ for smooth convex problems. Stochastic gradients often yield rates in expectation that trade bias, variance, batch size, and step schedule (Bubeck 2015; Robbins and Monro 1951).

Worst-case rates are not runtime predictions. Constants, conditioning, memory traffic, parallelism, and stopping accuracy matter. Benchmarking should report wall time, evaluations, energy where relevant, and quality as a function of budget.

Complexity and approximation

Complexity classifies scaling with input size and representation. A pseudo-polynomial algorithm is not polynomial in encoded input length. Continuous problems require an accuracy parameter ε. Oracle complexity counts function, gradient, or Hessian queries. Approximation algorithms guarantee a ratio or additive error; randomized algorithms state probabilities. Practical tractability can come from sparsity, low treewidth, convexity, fixed parameters, or benign data distributions even when the general problem is hard.

Sensitivity and conditioning

Sensitivity asks how the solution changes when data change. If $x^*(\theta)$ solves a parameterized problem, derivatives such as $dx^*/d\theta$ support design, inference, and bilevel learning. The envelope theorem often differentiates the optimal value without differentiating the optimizer. Ill-conditioning means small perturbations can create large changes; an unstable optimum may be mathematically correct and operationally unusable.

Automatic differentiation and adjoints

Forward-mode automatic differentiation propagates directional derivatives; reverse mode propagates sensitivities from outputs to many inputs. Reverse mode is the computational basis of backpropagation and adjoint methods. Differentiating through an optimizer can use unrolled iterations or implicit differentiation of optimality conditions. Each choice has memory, accuracy, and stability trade-offs.

Verification

Verification should be layered.

- **Implementation:** derivative checks, unit tests, invariant checks, deterministic seeds.
- **Numerics:** residuals, feasibility tolerances, conditioning, precision sensitivity.
- **Optimality:** primal-dual gaps, KKT residuals, bounds, approximation guarantees.
- **Robustness:** perturbation, scenario, adversarial, and out-of-distribution tests.
- **Model:** calibration, causal evidence, physical conservation, domain review.
- **Decision:** stakeholder review, alternatives, distributional impact, reversibility.

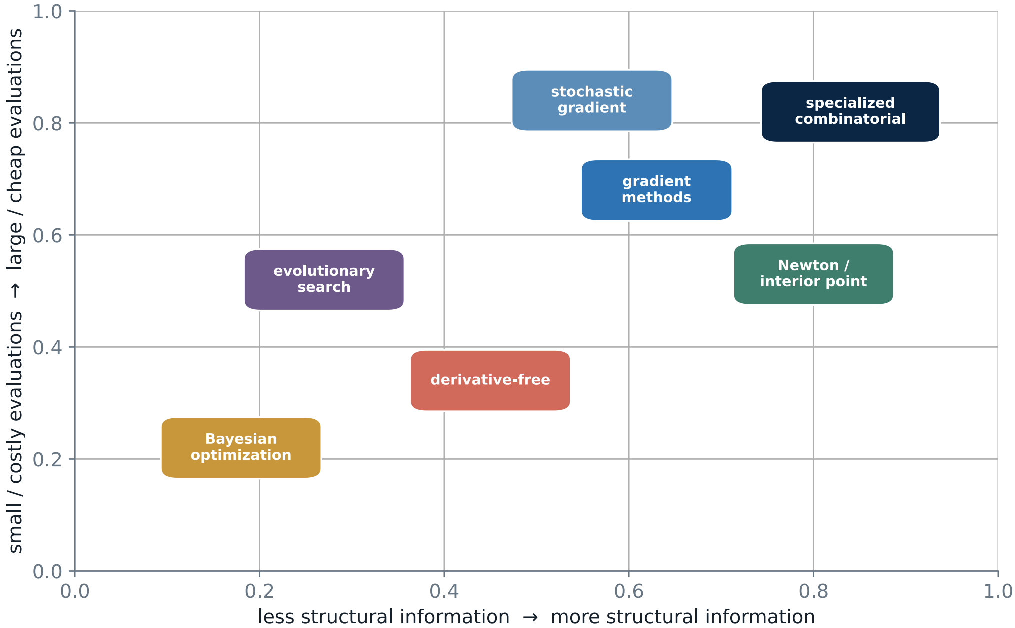


Figure 12. Algorithm families positioned by available structural information and by evaluation scale and cost.

Figure 12 is a claim ladder: problem structure narrows the algorithm family, the algorithm returns a candidate and certificate, and validation determines what may be asserted in the world.

Reproducibility

A reproducible optimization result includes data and preprocessing versions, mathematical formulation, solver and version, parameter settings, hardware, random seeds, stopping criteria, logs, and enough intermediate evidence to reconstruct the claim. For commercial or sensitive systems, reproducibility may occur in a controlled audit environment rather than through public release.

Verification should follow the same composition. At the mathematical layer, check feasibility, residuals, bounds, and convergence. At the implementation layer, test gradients, sparse indexing, units, and solver configuration. At the component interface, test contracts, ranges, latency, and failure behavior. At the system layer, simulate closed-loop scenarios, strategic response, and common-cause failures. At the decision layer, validate consequences and governance.

Reproducibility is part of verification but not identical to it. Re-running the same flawed pipeline reproduces an error. Preserve data versions, code, environment, solver, seeds, tolerances, and hardware; then independently challenge the model, implementation, and interpretation. A useful certificate is small enough to check and rich enough to support the actual claim.

Field checklist

- Define a state that is sufficient for prediction and decision, then test that assumption.
- Distinguish open-loop plans from feedback policies and model delay, saturation, and fallback.
- Treat the metric or ground cost as semantics, not as a neutral default.
- At every interface, state variables, units, admissible ranges, uncertainty, and guarantees.
- Delay local selection when downstream components need alternatives or value functions.
- Verify mathematical, implementation, interface, system, and decision layers separately.
- Reproduce the pipeline and independently challenge the formulation.

Translation lens. Dynamics translates choices into policies; transport translates difference into transformation cost; category theory translates components through typed interfaces. Their shared lesson is that composition should preserve the information required by downstream decisions, not merely the locally selected answer.

Part III · Computing and Engineered Systems

Chapter 8 · Algorithms, computing systems, software, and cloud platforms

Algorithms, representations, and resource models

An algorithm is an optimization argument made executable. It chooses a sequence of operations under budgets for time, memory, communication, energy, precision, and implementation effort. The mathematically shortest description is not always the fastest program; the asymptotically best method is not always best at realistic sizes; and the lowest-latency method may be unacceptable if it is fragile or opaque.

Complexity as a resource model

For input size n, asymptotic analysis studies growth such as $T(n) = O(n\log n)$. It deliberately suppresses constants and machine details, which makes it portable but incomplete. A fuller cost model might be

$$C(A,n) = w_t T(A,n) + w_m M(A,n) + w_i I(A,n) + w_e E(A,n), \tag{60}$$

where T is time, M memory, I input/output, and E energy. The weights depend on the deployment environment. External-memory algorithms optimize transfers between storage levels; cache-oblivious algorithms seek good locality without hard-coding cache parameters; parallel algorithms trade total work against span, synchronization, and communication (Cormen et al. 2022; Knuth 1997).

Worst-case, average-case, amortized, smoothed, and competitive analyses answer different questions. A hash table's expected constant-time lookup does not remove adversarial collision risk. A dynamic array's occasional expensive resize is compatible with low amortized cost. An online algorithm may be judged by a competitive ratio against an omniscient offline optimum. Picking the analysis is itself a modeling choice.

Data structures as compiled assumptions

A data structure makes some operations cheap by making others expensive. Balanced trees favor ordered updates and queries; hash tables favor equality lookup; heaps favor repeated extremum extraction; union-find favors incremental connectivity; sketches exchange exactness for compactness. The right structure depends on the operation distribution, not merely the data type.

Representation also controls search. Dynamic programming turns repeated subproblems into reusable state, but its state space may grow exponentially. Greedy algorithms commit early and are exact only when exchange properties or matroid-like structure justify them. Divide-and-conquer exposes parallelism; branch-and-bound pays for certificates; randomized algorithms can simplify structure or evade pathological inputs (Kleinberg and Tardos 2005).

Approximation, streaming, and learning-augmented methods

When exact computation is too costly, one can optimize the guarantee. An approximation algorithm for minimization may promise

$$f(\hat{x}) \leq \rho f(x^*) \tag{61}$$

for ratio $\rho \geq 1$. Streaming algorithms control memory and passes. Fixed-parameter algorithms isolate a small structural parameter. Learned indexes and learned heuristics exploit observed distributions, but need fallback guarantees when the distribution shifts.

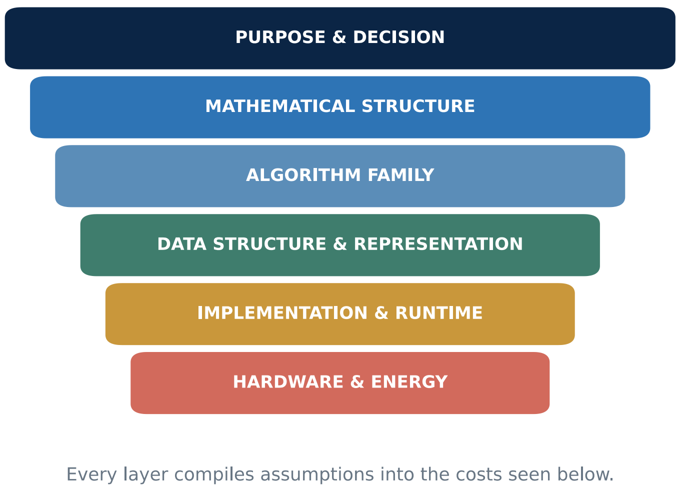


Figure 13. Six-layer stack from purpose and decision to hardware and energy.

Figure 13 prevents an abstract complexity claim from being mistaken for machine performance by tracing structure through algorithms to memory, communication, parallelism, and energy.

Engineering the crossover point

Algorithm choice should be benchmarked across the expected input distribution. A theoretically inferior method may win below a crossover size because of vectorization, locality, or setup cost. Conversely, a microbenchmark can hide garbage collection, allocation, cold caches, data skew, or tail latency. Report the workload, hardware, compiler, repetitions, uncertainty, and stopping rule. Optimization without a reproducible measurement protocol is performance folklore.

Computing optimization spans at least four scales: the abstract algorithm, the implementation, the machine, and the service. Each scale has a different cost model. Big-O notation suppresses constants and hardware details to describe growth; a profiler measures one implementation and workload; a hardware counter exposes cache misses, branches, or vector utilization; a service trace exposes queues, retries, fan-out, tail latency, and failure. A claim that an algorithm is “faster” is incomplete until the scale and resource model are named.

The layers do not compose monotonically. An algorithm with fewer arithmetic operations can move more memory and run slower. Compression can save network bandwidth while consuming CPU and latency. Parallelism can reduce mean completion time while increasing coordination, cost, and tail variance. Caching can improve common requests while creating invalidation complexity. The correct object is a resource vector rather than a single runtime:

$$R(x) = \big(T_{50}(x), T_{99}(x), M(x), B(x), E(x), C(x), A(x)\big), \tag{62}$$

where the components may represent median and tail latency, memory, bandwidth, energy, monetary cost, and availability. Equation (62) invites Pareto analysis and explicit service constraints. It also prevents a local microbenchmark from silently standing in for user experience.

Data structures compile assumptions about future operations. A hash table privileges expected lookup; a balanced tree preserves order and worst-case bounds; a columnar layout privileges scans and compression; an index accelerates reads while increasing storage and write cost; a sketch gives bounded approximate answers under limited memory. There is no context-free “optimal data structure.”

The same holds for distributed state. Strong consistency, causal consistency, eventual convergence, partition tolerance, availability, durability, and latency form a design space. A database schema embeds anticipated joins and invariants. A partition key embeds an expected access pattern. When the workload changes, a previously good representation becomes technical debt. Optimization should therefore include migration and observability, not only initial performance.

Systems, queues, and cloud capacity

A computer system is a queueing network governed by feedback. Requests compete for processors, memory, storage, accelerators, network links, locks, and human operators. Local utilization gains can worsen global latency, and average performance can conceal catastrophic tails.

Throughput, latency, and queues

Little’s law relates average number in a stable system L, arrival rate λ, and average time W:

$$L = \lambda W. \tag{63}$$

It is a conservation law, not a scheduling policy (Little 1961). As utilization approaches capacity, latency often rises nonlinearly. Batching improves throughput but adds waiting. Replication reduces read latency and increases availability while creating consistency and coordination costs. Backpressure protects downstream components but can propagate upstream.

Tail latency matters because a fan-out request waits for slow children. If n independent components each meet a latency target with probability p, the probability that all do so is p^n. Hedged requests, timeouts, load balancing, admission control, and redundancy trade extra work for more predictable service (Dean and Barroso 2013).

Parallelism and the memory wall

Amdahl's law estimates speedup when fraction s is serial and $1 - s$ is accelerated on p processors:

$$S(p) = \frac{1}{s+(1-s)/p}. \tag{64}$$

Gustafson's law instead asks how problem size can scale with available processors (Amdahl 1967; Gustafson 1988). Both are idealizations. Real scaling is limited by memory bandwidth, communication, load imbalance, synchronization, and energy. Roofline-style reasoning compares arithmetic intensity with compute and bandwidth ceilings.

Databases and distributed state

Database optimization chooses physical plans for a declarative query. Join order, access paths, cardinality estimates, partitioning, caching, and materialization interact combinatorially. A wrong selectivity estimate can dominate an otherwise sophisticated cost model. Adaptive execution treats the plan as a policy that can change after observing runtime evidence.

Distributed data introduces consistency choices. Stronger ordering simplifies application reasoning but may increase coordination and reduce availability under partition. Eventual consistency reduces coordination but shifts reconciliation to data types and applications. The correct point is workload- and harm-dependent, not a universal slogan.

Cloud scheduling and reliability

Cloud platforms place workloads across heterogeneous machines while controlling cost, carbon, locality, deadlines, and failure domains. A simplified scheduler may solve

$$\min_x \ C_{\text{money}}(x) + \lambda C_{\text{latency}}(x) + \mu C_{\text{carbon}}(x) \tag{65}$$

subject to capacity, affinity, security, and service-level constraints. Autoscaling is a feedback controller; if delayed or overly aggressive, it oscillates. Reliability engineering uses redundancy, graceful degradation, fault containment, and recovery objectives rather than assuming failure can be optimized away.

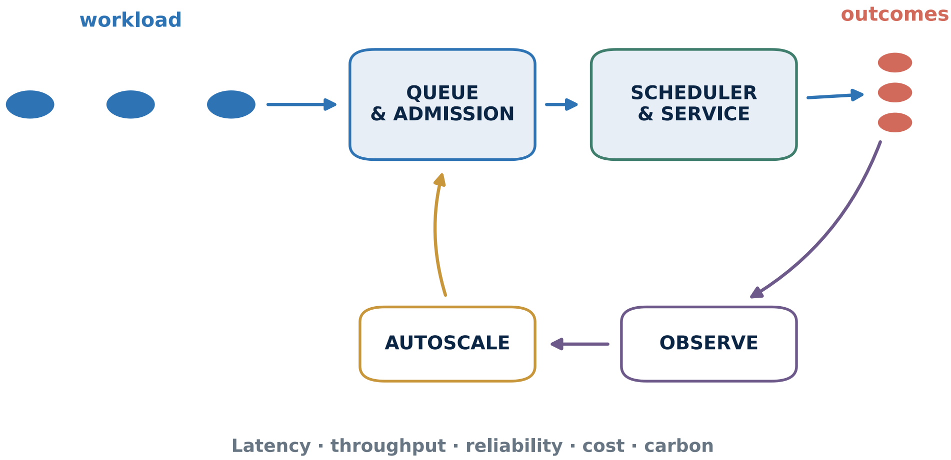


Figure 14. Service flow from workload to outcomes, with observation and autoscaling feedback.

Figure 14 shows why service optimization cannot stop at a scheduler: workload, queues, hardware, feedback, and service objectives form one coupled stack.

Computing services are queueing systems. Little's law $L = \lambda W$ relates long-run average number in system, throughput, and time in system under stable conditions (Little 1961). The relation is simple; its implications are not. As utilization approaches capacity, small arrival or service-time changes can cause large delay changes. Variability and correlated bursts matter. A system optimized to high average utilization can have intolerable tails and weak recovery.

Admission control, batching, replication, hedged requests, load balancing, autoscaling, and scheduling all transform the queue. Each technique moves cost. Hedging reduces a tail by issuing extra work; replication improves availability while creating consistency and write amplification; batching improves throughput while adding waiting; autoscaling responds after measurement and provisioning delay. The model should include these feedback loops, not just steady-state capacity (Dean and Barroso 2013; Barroso, Hölzle, and Ranganathan 2018).

A useful performance experiment therefore specifies the workload distribution, concurrency, warm-up, cache state, failure conditions, request dependency graph, and measurement interval. It reports distributions, not only averages. It separates service time from queueing time and client-perceived time. It checks coordinated omission, in which a load generator stops sending work while a service is slow and thereby under-samples the worst delays.

Suppose a service allocates replicas r_{sz} by service s and zone z. Let $\mathcal{R}$ be the set of nonnegative integer allocations satisfying $Ar \le b$, the latency-risk bound $\Pr(L_s(r,\xi) > \ell_s) \le \varepsilon_s$, and the survivability requirement $\sum_z r_{sz}\,\mathbf{1}\{z \notin F\} \ge k_s$ for every service s and admitted failure domain F. A resource model can then minimize cost and energy compactly:

$$\min_{r\in\mathcal{R}}\left(\sum_{s,z} c_{sz}\, r_{sz} + \lambda_E E(r)\right). \tag{66}$$

The feasible-set definition paired with Equation (66) makes three structures visible: integer capacity, chance or risk constraints for latency, and survivability under failure sets. The model should include startup delay, state replication, data locality, deployment surge, and recovery workload. A capacity plan that is feasible only in steady state can fail during the change that implements it.

Failure domains must be physical and organizational. Zones can share power, control planes, software versions, credentials, or operators. Correlated software failure can defeat geographic redundancy. Stress tests should remove common dependencies, delay telemetry, and corrupt coordination—not only simulate independent host loss.

Software quality, architecture, and delivery

Software is optimized twice: first as a product that must remain understandable and changeable, then as a running artifact that consumes resources. Focusing only on runtime speed creates brittle systems; focusing only on elegance can ignore users, cost, and physical limits.

Quality is a vector

Software quality includes functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, flexibility, and safety-related properties. ISO/IEC 25010 formalizes a contemporary quality model, but every project must operationalize its own thresholds (International Organization for Standardization 2023). Keeping these as named dimensions prevents a convenient metric from masquerading as total quality.

Technical debt is not merely bad code; it is a liability whose future change cost is uncertain. An expedient shortcut can be rational if the option value of speed exceeds expected remediation cost. It becomes dangerous when the debt is hidden, compounding, or coupled to safety-critical behavior (Brooks 1995; Sculley et al. 2015).

Profile before transforming

Performance work follows a causal loop: define a user-visible target, measure representative workloads, locate a bottleneck, change one mechanism, and remeasure. CPU profiles, allocation traces, lock contention, I/O traces, database plans, and distributed traces observe different layers. Optimizing the hottest function may not improve wall-clock time if the critical path lies elsewhere.

Compiler optimization preserves a chosen semantics while changing representation. Constant folding, common-subexpression elimination, dead-code elimination, inlining, loop transformation, vectorization, instruction scheduling, register allocation, and code generation solve interacting problems (Aho et al. 2006; Muchnick 1997). Whole-program and profile-guided optimization use more context but raise build time, reproducibility, and debugging costs.

Delivery as flow optimization

Build and deployment pipelines optimize lead time subject to correctness and recovery constraints. Small batches shorten feedback; automated tests reduce verification cost; canary releases and feature flags expose a controlled fraction of users; rollback and observability limit downside (Humble and Farley 2010). Common metrics—deployment frequency, lead time, change failure rate, and recovery time—should be used diagnostically, not converted into isolated quotas (Forsgren, Humble, and Kim 2018).

Figure 15. A radar chart comparing runtime, reliability, security, maintainability, delivery, and usability.

Figure 15 treats software quality as a vector; improving runtime can move reliability, security, maintainability, delivery speed, or developer cognition in the wrong direction.

The optimization boundary

Some concerns belong in code, some in architecture, some in operations, and some in product policy. Caching can reduce latency but create staleness. Retries can improve success rates but amplify overload. Compression saves bandwidth but costs compute. A locally reasonable library choice may fragment the platform. Architecture makes these couplings explicit.

Secure software development integrates threat modeling, dependency control, least privilege, code review, reproducible builds, vulnerability response, and provenance throughout the lifecycle (Souppaya, Scarfone, and Dodson 2022). Security added after performance tuning is rarely compositional.

Software engineering often optimizes transformations of transformations. A compiler changes code to improve runtime while preserving semantics. A refactoring changes structure while preserving behavior. A deployment pipeline changes delivery cadence while preserving safety. A scheduler changes execution order while preserving dependencies. In each case, the preservation claim is more important than the local gain.

Quality is plural. ISO/IEC 25010 includes performance efficiency, reliability, security, maintainability, compatibility, interaction capability, flexibility, and safety-related concerns within a broader product-quality model (International Organization for Standardization 2023). Optimizing one property can move cost into another: aggressive inlining grows binaries and instruction-cache pressure; abstraction improves maintainability but may allocate; a fast release process without observability increases recovery time; strict controls without usable workflows create bypasses.

Profile-guided work should start with a decision-relevant baseline. Measure the user journey or service objective, identify the critical path, form a causal hypothesis, change one mechanism, and test end to end. A local speedup matters only if its component contributes enough to total time; Amdahl's law makes this explicit (Amdahl 1967). Gustafson's perspective shows that increased capacity may be used to solve larger problems rather than shorten a fixed one (Gustafson 1988). Both remind us that the workload definition is part of the claim.

Software delivery is also an operational optimization problem. Batch size, work in progress, test selection, review latency, deployment frequency, recovery, and architecture interact. Smaller changes often reduce integration risk and feedback delay, but excessive fragmentation can increase coordination and review overhead. Automated tests accelerate feedback only when they are trustworthy and maintained. Metrics such as deployment frequency or lead time should be interpreted as system signals, not individual targets (Humble and Farley 2010; Forsgren, Humble, and Kim 2018).

Organizational boundaries become software interfaces. A service contract that is unstable, undocumented, or slow creates queues across teams. Local team utilization can worsen global flow. Limiting work in progress, clarifying ownership, and making dependencies visible can outperform a tool-level optimization because they enlarge the feasible set of coordinated action.

Computing systems rarely expose a stable objective function. Caches warm, compilers change code shape, garbage collectors move work across time, networks share capacity, cloud hosts vary, and measurement itself consumes resources. Performance optimization is therefore an experimental discipline. A benchmark result is a conditional observation, not a property of a program without context.

Begin with a service-level consequence rather than an internal counter. Users may care about tail response, throughput under a fixed cost, energy per completed task, or time to recovery. CPU utilization, allocation rate, cache misses, and request concurrency are explanatory measurements. Optimizing an internal counter can move work elsewhere or improve a microbenchmark while degrading the end-to-end path.

A reproducible performance experiment records hardware and topology, firmware, operating system, runtime, compiler and flags, container limits, power state, dataset, request mix, warm-up, affinity, concurrency, sample count, clock source, and background load. Separate initialization, steady state, and recovery. Report distributions and tails, not only a mean.

For an optimization with parameters x, measure paired alternatives under randomized or interleaved order where feasible. Use common workloads to reduce comparison variance. Hold out workloads and machines for final evaluation. If search repeatedly probes the same benchmark, the reported best inherits selection bias; confirm it on an untouched run.

Architectural optimization chooses boundaries, interfaces, consistency, caching, batching, and deployment structure. These decisions determine which later resource allocations are possible. A stateful synchronous dependency creates latency coupling; an asynchronous queue creates buffering and eventual consistency; a cache changes data freshness and invalidation risk. “Optimize the cloud bill” after architecture is fixed may miss the dominant variables.

Use architecture fitness functions carefully. Automated tests can enforce dependency rules, latency budgets, schema compatibility, security controls, and deployment invariants. They protect declared qualities and can ossify a poor boundary. Review whether the rule still represents purpose and whether teams are creating workarounds.

Technical debt belongs in the objective through future change cost and failure exposure, but a guessed scalar debt score is weak. Preserve an evidence register: known coupling, unsupported versions, missing tests, incident history, manual work, cognitive load, and change lead time. Some debt is a rational option taken under uncertainty; some is an unowned hazard.

Optimizing every team for utilization increases queues between teams. Work waits for review, environment, security, release, or another specialization. Limit work in progress, reduce batch size, automate safe handoffs, and measure lead time and recovery (Forsgren, Humble, and Kim 2018). The unit of value is a completed, reliable change, not a busy person.

A release system should optimize for small reversible changes within safety constraints. Feature flags, canaries, compatibility, observability, and automated rollback create recourse. They can also increase state space and operational complexity. Include cleanup and ownership; a permanent forest of flags is not free flexibility.

Field checklist

- Name the scale of every performance claim: algorithm, implementation, machine, service, or user journey.
- Use a resource vector and explicit service constraints rather than a single benchmark score.
- Model queues, variability, bursts, retries, and feedback near capacity.
- Treat data structures, schemas, partitioning, and indexes as compiled workload assumptions.
- Require a preservation claim for compiler, refactoring, scheduling, and deployment transformations.
- Benchmark distributions under representative and failure workloads; guard against measurement artifacts.
- Include migration, observability, rollback, and organizational flow in the optimization boundary.

Translation lens. Computing translates abstract operations into physical resource use through several compiler-like layers. Valid optimization preserves semantics and service obligations while making the cost movement across time, memory, bandwidth, energy, and human work observable.

Chapter 9 · Machine learning, artificial intelligence, security, privacy, and assurance

Learning as nested, decision-focused optimization

Machine learning optimizes models from data, but the training loss is only a surrogate for real-world value. The pipeline includes problem selection, data generation, labeling, representation, optimization, evaluation, deployment, monitoring, and governance. Error can enter at every boundary.

Empirical risk and generalization

Given data $(x_i, y_i)_{i=1}^{n}$, empirical risk minimization chooses

$$\hat{\theta} \in \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell\left(f_\theta(x_i), y_i\right) + \lambda R(\theta). \quad (67)$$

The loss ℓ encodes an error model; the regularizer R encodes preference or capacity control. Training error is not deployment error. Generalization depends on hypothesis class, data dependence, sampling, optimization, and shift (Hastie, Tibshirani, and Friedman 2009; Goodfellow, Bengio, and Courville 2016). Cross-validation estimates out-of-sample performance only when splits respect time, groups, leakage paths, and intended use.

Stochastic gradient methods estimate a full gradient from minibatches. Momentum, adaptive methods, schedules, normalization, and preconditioning reshape the trajectory (Robbins and Monro 1951; Duchi, Hazan, and Singer 2011; Kingma and Ba 2015; Bottou, Curtis, and Nocedal 2018). In deep learning, reaching zero training loss does not identify a unique solution; initialization and implicit bias help select among many interpolating models.

Metrics, calibration, and decisions

Accuracy can be meaningless under imbalance or unequal costs. Precision, recall, specificity, ranking loss, calibration, time-to-event error, robustness, latency, and abstention address different uses. A calibrated probability model is not yet a decision rule. With action a, state y, loss $L(a, y)$, and features x, Bayes decision theory chooses

$$a^*(x) = \arg\min_{a} \mathbb{E}[L(a, Y) \mid X = x]. \quad (68)$$

Thresholds therefore depend on consequences and capacity, not merely model discrimination.

Scaling and search

Architecture search, hyperparameter tuning, data selection, and prompt or policy optimization form outer loops around training (Elsken, Metzen, and Hutter 2019). Bayesian optimization is useful when evaluations are expensive. Distributed training trades elapsed time against communication, synchronization, reproducibility, and energy. Larger models can improve broad capability while increasing cost, concentration of access, and evaluation difficulty (Kaplan et al. 2020).

Deployment as a controlled intervention

A deployed model changes the environment that generates future data. Ranking influences clicks; moderation influences language; pricing influences demand; clinical advice influences treatment. Off-policy evaluation, randomized trials where ethical, staged rollout, monitoring, and causal analysis are needed to distinguish model quality from system effect.

Federated learning can keep raw data decentralized, but it does not by itself ensure privacy, fairness, or robustness (Kairouz et al. 2021). Differential privacy supplies a formal stability guarantee: a mechanism M is (ε, δ)-differentially private if adjacent datasets D, D' satisfy

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \Pr[M(D') \in S] + \delta \tag{69}$$

for all measurable S (Dwork and Roth 2014). Privacy budget, accuracy, and participation remain coupled.

Machine learning contains several nested loops. An inner loop fits parameters; an outer loop chooses architecture, data, regularization, and hyperparameters; a product loop chooses thresholds and actions; an operational loop monitors consequences and retraining; a governance loop may change the objective or stop the system. Treating only parameter fitting as “the optimization” hides the decisions with the largest real-world effects.

Empirical risk minimization assumes that an average over training observations is informative about a target distribution. Generalization theory and validation methods qualify this step, but deployment adds further shifts: users adapt, upstream systems change, selection policies alter who receives labels, and interventions change outcomes. A model can continue minimizing its training loss while its decision value falls.

The deployment decision should therefore be represented explicitly. If a score $s_\theta(x)$ supports action a, a threshold is chosen by consequences, not accuracy alone:

$$a^*(x) \in \arg\max_{a \in \mathcal{A}} \textstyle\sum_{y \in \mathcal{Y}} P_\theta\,(y \mid x)\, U(a, y; x). \tag{70}$$

Equation (70) separates probabilistic estimation from utility and permits abstention, referral, or information gathering. Calibration matters because predicted probabilities enter the consequence calculation. Class balance and a confusion matrix are insufficient when harms differ, capacity is constrained, or the action changes future labels.

Data selection is itself optimization. Sampling, labeling, augmentation, deduplication, active learning, and curriculum determine which distinctions a model can learn. More data can worsen a system if it reinforces a biased measurement, leaks evaluation cases, duplicates common examples, or increases compute without adding decision-relevant information. Data quality should be described along provenance, coverage, label process, dependence, recency, and consent.

Hyperparameter search can overfit the validation set. Repeated evaluation selects favorable noise just as any optimizer does. Nested validation, sealed test sets, preregistered metrics, and limited access reduce this optimizer's curse. When teams compare many architectures, datasets, and seeds, they should report the selection process and variability rather than only the winning run.

Scaling changes more than loss. Larger models consume energy, hardware, data-engineering capacity, and evaluation effort; they can create latency, privacy, and supply-chain constraints. A scaling decision is multiobjective. Compute-optimal training under a narrow loss may not be deployment-optimal once serving, maintenance, and governance are included (Kaplan et al. 2020; Hoffmann et al. 2022).

A model maps inputs to predictions or generated outputs. An AI system also chooses data, prompts or context, retrieval, tools, thresholds, abstention, user interface, logging, escalation, and update. Evaluation must follow this full decision path. A better benchmark model can produce a worse system if latency removes review time, confidence is miscommunicated, retrieval leaks data, or automation changes who receives service.

Start with a use-case contract: intended users and affected nonusers, decision and authority, unacceptable actions, required evidence, operating conditions, and safe alternative. The contract constrains data and model choice. A system intended to summarize documents needs source fidelity and citation behavior; a ranking system needs exposure and appeal analysis; a control agent needs permissions and recovery.

Organize evaluation by claim and condition. Rows can include task performance, calibration, robustness, safety, security, privacy, fairness, human usability, resource cost, and recovery. Columns can include ordinary data, temporal shift, subgroup or language, adversarial input, missing context, tool failure, stale retrieval, overload, and operator disagreement.

Each cell requires an appropriate test. Accuracy does not validate calibration. Red-team success does not estimate frequency. A fairness metric does not establish that protected groups were defined adequately. A privacy mechanism has a formal budget and an implementation boundary. Human preference for an interface does not prove correct reliance.

Generative evaluation needs reference-grounded and open-ended tasks. Test factual consistency, source entailment, uncertainty expression, refusal, instruction hierarchy, harmful transformation, and recovery after error. Preserve examples and rationales, because aggregate scores can improve while a severe capability regresses. For open-ended quality, use multiple trained reviewers, calibrate disagreement, and avoid presenting a preference model as universal taste.

Training and evaluation data define which distinctions the system learns. Sampling, filtering, labeling, deduplication, augmentation, and retention are optimized processes with distributional consequences. A "clean" dataset can remove dialect, rare events, disability access patterns, or contested cases. A large web corpus can contain private, copyrighted, false, or harmful material.

Write a data constitution: permitted sources and uses, consent or legal basis, provenance, exclusions, retention, correction, deletion, access, and audit. Document whose labor labels and moderates data and what ambiguity guidelines suppress. When model outputs create new training data, mark synthetic provenance and avoid feedback that narrows diversity or amplifies errors.

Data minimization and utility should be co-designed. Differential privacy provides a formal bound on how much one record can change output distributions (Dwork and Roth 2014), but the guarantee composes across queries and depends on the mechanism and neighboring relation. Security, access control, and purpose limitation remain necessary. A private model can still produce an illegitimate decision.

Predictive error and decision loss need not align. Errors near an action threshold or on a constrained subgroup can matter more than large errors where the same action remains optimal. Decision-focused training differentiates or approximates downstream optimization so a model learns features useful for the decision. This can improve end-to-end performance and can hide model defects behind one policy.

Retain independent predictive and causal evaluation. If the downstream objective is misspecified, decision-focused learning will exploit it more efficiently. If constraints or prices change, a model specialized to yesterday's decision may transfer poorly. Compare modular and end-to-end approaches under policy change.

Uncertainty should enter the decision, not merely decorate prediction. Calibrate probabilities or intervals where action changes, evaluate abstention, and propagate model uncertainty through the optimizer. For high stakes, use conservative constraints and professional review rather than treating posterior confidence as authority.

Model selection consumes compute, energy, hardware, data labor, and time. Report training and inference resources at the service boundary, including repeated experiments and utilization. A smaller model can be dominated if its poor accuracy causes costly downstream action; a larger model can be dominated if marginal benefit is negligible.

Optimize total service consequence: hardware manufacture, electricity, latency, throughput, retrieval, networking, cooling, and human review. Test demand rebound. A faster or cheaper model can increase use enough to raise absolute burden.

Resource access also shapes who can reproduce research or compete. Use end-to-end baselines and disclose budgets. Automated architecture or hyperparameter search should count all trials, not only the final run (Hutter, Kotthoff, and Vanschoren 2019; Elsken, Metzen, and Hutter 2019).

Security, privacy, and adversarial optimization

Security is optimization against an adaptive opponent. The defender allocates finite resources across prevention, detection, response, and recovery while the attacker searches for leverage. Historical incident rates are endogenous: successful defense hides attacks, and observed attacks reflect earlier choices.

Threat models and attack surfaces

A threat model identifies assets, adversaries, capabilities, entry points, trust boundaries, and harms. A stylized defense problem is

$$\min_{x \in X} \mathbb{E}_{a \sim P(a|x)}[H(x,a)] + C(x), \tag{71}$$

where investment x changes both harm H and the distribution of attacks. Static risk scores miss strategic adaptation. Game theory, robust optimization, and red-team exercises supply complementary views.

Least privilege, segmentation, secure defaults, diversity, rate limits, and layered controls reduce blast radius. Yet every control adds friction and complexity. Optimizing user convenience without adversarial testing creates exploitable shortcuts; optimizing lockdown without workflow study encourages bypass.

Privacy is not secrecy alone

Privacy concerns appropriate information flow, autonomy, inference, and power. Encryption protects data in transit or storage but not necessarily data in use. Access control limits authorized operations; minimization reduces retained attack surface; differential privacy bounds contribution leakage; secure multiparty computation and trusted execution change who can observe intermediate state. These mechanisms solve different problems.

Resilience over perfect prevention

Expected loss alone can underweight rare systemic failure. Resilience asks whether the system can continue essential functions, degrade safely, restore state, and learn. Objectives may combine expected loss with tail risk:

$$\min_{x} \mathbb{E}[L(x,\xi)] + \lambda \mathrm{CVaR}_{\alpha}\big(L(x,\xi)\big). \tag{72}$$

Backups must be isolated and restored in exercises; incident response must be rehearsed; recovery objectives must reflect dependency chains. Diversity can reduce common-mode failure but increases maintenance burden.

Formal methods and certificates

Testing samples behaviors. Formal verification reasons over a model of all admitted behaviors. Model checking explores state transitions; theorem proving derives claims from axioms; abstract interpretation computes safe approximations; type systems exclude classes of programs. Optimization enters synthesis, counterexample search, invariant discovery, solver-backed verification, and minimal repair.

A certificate is valuable only relative to assumptions. A proof about source semantics says little if the compiler, hardware, configuration, or specification is wrong. Assurance therefore links each claim backward through its evidence, model, and assumptions.

NIST's AI Risk Management Framework similarly treats governance, mapping, measurement, and management as an iterative system rather than a single score (Tabassi 2023).

Security begins with a threat model: assets, adversaries, capabilities, knowledge, access, incentives, and acceptable residual risk. Without it, "maximize security" is meaningless. Defenders allocate limited attention across prevention, detection, response, and recovery. Attackers search for low-cost paths through the actual system, including people, suppliers, credentials, dependencies, and operational exceptions.

Adversarial machine learning makes the nested structure explicit. A defender chooses model parameters while an attacker chooses a perturbation:

$$\min_{\theta} \mathbb{E}_{(x,y)}\left[\max_{\delta\in\Delta(x)} \ell(f_\theta(x+\delta), y)\right]. \tag{73}$$

The set $\Delta(x)$ is a threat model, not a natural fact. A small norm ball may be mathematically convenient yet fail to represent semantic manipulation, data poisoning, model extraction, prompt injection, or supply-chain compromise. Robustness inside Δ should not be advertised as general security.

Formal assurance can prove properties of software or models under assumptions. Type systems, model checking, theorem proving, abstract interpretation, and SMT solving produce powerful certificates. They do not establish that the specification captures the intended policy, that deployment matches the verified artifact, or that an attacker respects the modeled interface. Assurance is a chain of preservation claims.

Privacy is not simply another accuracy penalty. It includes legal purpose limitation, consent, contextual integrity, data minimization, access control, retention, and protection against inference. Differential privacy gives a composable mathematical bound on how much an output distribution changes when one individual's record changes (Dwork and Roth 2014). Its parameter is meaningful only with the adjacency definition, composition, threat model, and implementation.

Federated learning moves some computation to data holders but does not automatically provide privacy or fairness. Updates can leak information; client participation can be unequal; communication and device energy can dominate; non-identical data complicates convergence (Kairouz et al. 2021). Secure aggregation, differential privacy, robust aggregation, and governance address different failure modes.

Treating privacy as a hard design constraint can stimulate better architecture: local computation, aggregation, limited retention, synthetic or sampled data, and purpose-specific models. A penalty formulation that trades privacy against marginal accuracy may silently change the normative status of the requirement.

Threat modeling identifies assets, actors, capabilities, entry points, and consequence. Adversarial examples are one layer. Others include training-data poisoning, prompt or instruction injection, retrieval manipulation, tool abuse, model extraction, privacy inference, credential compromise, dependency attack, and strategic user adaptation.

Robust training solves a minimax approximation under a threat set. The threat set is a governance artifact: it says what the defender anticipates and what perturbations are considered equivalent. Robustness inside it can trade against ordinary performance and does not protect against a different mechanism (Goodfellow, Bengio, and Courville 2016). Red teams should therefore challenge the boundary, permissions, and recovery as well as model inputs.

Agents require least privilege, scoped credentials, confirmation for consequential actions, rate and spend limits, sandboxing, idempotence where possible, transaction logs, and a tested stop. Tool output is untrusted input. A planning system should not be able to grant itself broader authority because the objective is difficult.

Assurance for adaptive systems and change

Adaptive systems require assurance cases rather than a single metric. The case links claims, evidence, assumptions, monitors, and response. It should cover data quality, performance by relevant group and condition, calibration, robustness, security, privacy, energy, human factors, fallback behavior, and impact. NIST's AI RMF organizes risk work around Govern, Map, Measure, and Manage; the EU AI Act adds legal duties differentiated by system role and risk (Tabassi 2023; European Parliament and Council of the European Union 2024). These frameworks do not replace domain regulation or professional judgment.

Monitoring must be tied to action. A drift statistic without a threshold, owner, investigation path, and rollback rule is observability theater. Human oversight must be feasible under workload: a nominal reviewer who cannot understand, delay, or reverse the recommendation is not a meaningful control. Override data should be treated as evidence, not as operator failure by default.

Adaptive systems change through data refresh, model update, configuration, retrieval corpus, prompt, tool, interface, or user population. Define change classes and proportional evidence. A retrieval update can introduce new legal or factual risk without changing weights. A threshold change can alter distribution. A model update can change calibration even when average performance rises.

Maintain a system card with versions, intended use, evidence, limitations, incidents, and owners. Monitor input and outcome drift, calibration, abstention, overrides, subgroup effects, resource use, security events, and appeal. Use the NIST AI Risk Management Framework as an institutional vocabulary, not a certification shortcut (Tabassi 2023). Applicable law and sector regulation determine duties; technical evaluation supports rather than replaces them (European Parliament and Council of the European Union 2024).

A safety case is a structured argument from claims to evidence, assumptions, and defeaters. It can state that a system remains within a defined operational envelope, detects specified faults, refuses unsupported cases, and hands off safely. Evidence includes analysis, tests, formal checks, simulation, human-factors work, and monitored use.

Learning systems require runtime evidence because behavior can be difficult to exhaustively enumerate. Monitor distribution, confidence, constraint residuals, tool actions, and outcome. Runtime monitors should be independent enough that a common model error does not defeat both actor and guard.

The case should include negative claims it cannot support. “No harmful output exists” is usually indefensible; “the system blocks these enumerated actions, was challenged under these conditions, and routes uncertain cases through this process” is testable. Update the case with every material system change.

Field checklist

- Map the parameter, hyperparameter, data, decision, operational, and governance loops separately.
- Convert predictive scores into actions through calibrated consequences, capacity, and abstention.
- Protect the test process from repeated selection and report seeds, search budget, and variability.
- State the adversary and justify the perturbation or attack set semantically.
- Treat privacy, security, rights, and safety according to their logical status, not as default penalties.
- Build an assurance case connecting claims to heterogeneous evidence and response actions.
- Verify that human oversight has time, information, authority, and a safe fallback.

Translation lens. Learning, attack, privacy protection, and assurance are coupled optimizations with different agents and constraints. The transferable structure is nesting; the domain-specific obligation is to type each loop’s data, authority, threat, and consequence.

Chapter 10 · Systems engineering, structures, signals, control, hardware, and co-design

Systems engineering, co-design, and exchanged envelopes

Engineering optimization begins with a purpose and ends with an artifact embedded in a larger system. Between them lie requirements, architectures, physical models, uncertainty, manufacturing, verification, operations, maintenance, and retirement. The “best component” is often a poor system choice because interfaces carry the dominant trade-offs.

From needs to a design space

Requirements convert stakeholder needs into testable constraints and preferences. Good requirements state the operational context, units, tolerance, and verification method. Premature numerical targets can freeze one architecture; vague aspirations cannot support trade studies. A design structure matrix or dependency graph helps identify coupled variables, analyses, and teams.

Multidisciplinary design optimization can be written schematically as

$$\min_{x,y_1,\dots,y_m} f(x, y_1, \dots, y_m) \tag{74}$$

subject to discipline equations

$$r_i(x, y_1, \dots, y_m) = 0, \qquad i = 1, \dots, m, \tag{75}$$

and system constraints. The shared design x might contain geometry and material choices; y_i might represent aerodynamic, structural, thermal, control, or economic states. Architectures differ in whether they solve discipline consistency inside each evaluation, distribute targets, or optimize all states simultaneously (Martins and Lambe 2013).

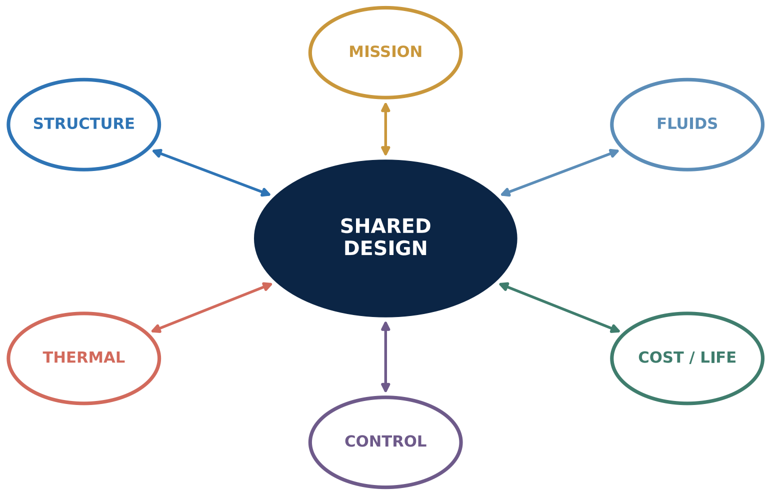


Figure 16. Coupled engineering disciplines exchanging shared design variables, states, sensitivities, and requirements.

Figure 16 visualizes co-design as an exchange of shared variables, states, sensitivities, and requirements rather than a sequence of isolated component optima.

Trade studies and value models

Weighted scoring tables are useful for organizing evidence, but their arithmetic can suggest false precision. Scores should expose dominance, sensitivity to weights, and uncertainty. Hard safety constraints should not become small penalties merely because a spreadsheet needs a total.

Surrogate models and reduced-order models accelerate expensive analyses. Their trust region is both mathematical and epistemic: an optimizer can find precisely the region where a surrogate is least supported. Adaptive sampling should therefore follow predicted value and model uncertainty. Verification checks whether equations were solved correctly; validation checks whether the equations represent the intended reality.

Lifecycle optimization

Acquisition price is rarely lifecycle cost. Reliability, maintainability, spares, training, energy, upgrades, disposal, and opportunity cost matter. Design for modularity can raise initial mass or price while lowering future adaptation cost. Real options value the ability to defer, expand, switch, or abandon.

Robustness and resilience are distinct. A robust design performs acceptably across anticipated variation. A resilient system can detect, absorb, recover, and adapt after disruption. Both require margins whose apparent “inefficiency” is functional.

Engineering optimization rarely concerns an isolated component. A structure changes loads on actuators; a controller changes required stiffness; a chip architecture changes power density and cooling; a manufacturing process changes achievable geometry; software changes hardware utilization. Systems engineering makes these couplings explicit through requirements, interfaces, budgets, models, and verification plans.

Multidisciplinary design optimization (MDO) coordinates coupled analyses and shared variables (Martins and Lambe 2013). A monolithic formulation exposes all coupling but can be organizationally and computationally unwieldy. Decomposition assigns subsystem problems and coordinates them through consistency constraints, targets, or prices. The choice is not purely mathematical: it must match ownership, confidentiality, model cadence, and the ability to exchange sensitivities.

An interface contract should specify variables, units, coordinate frames, valid ranges, uncertainty, fidelity, update cadence, and responsibility. A component that exports only a nominal value conceals sensitivity and makes downstream robustness impossible. Exporting a response surface, bounds, or an adjoint may preserve more useful structure without revealing every internal detail.

Complex engineering teams often pass nominal points: a structure supplies one mass, electronics one power number, software one workload, and controls one trajectory. Nominal points hide flexibility and cause serial redesign. A better interface exchanges feasible or performance envelopes.

A thermal team can expose allowable heat versus ambient and flow. A compute team can expose throughput versus power, precision, and latency. A battery team can expose energy, peak power, temperature, degradation, and reserve. A controller can expose reachable performance under actuator and timing limits. System optimization then composes envelopes before selecting component points.

Envelopes have versions, uncertainty, and validity regions. Overly conservative envelopes can make the system infeasible; optimistic ones create late integration failure. Joint tests should update them. If confidentiality prevents full models, response surfaces, dual sensitivities, or contract bounds can communicate enough structure.

Engineering margin absorbs model error, manufacturing tolerance, degradation, load growth, and rare events. It should not be scattered invisibly through every component: stacked conservative margins can produce excess mass or cost, while assumed shared margin can leave a gap.

Create a margin register with source, owner, uncertainty covered, confidence, consumption, and release condition. Separate deterministic allowance, statistical variation, and contingency. Allocate system margin where it most effectively protects consequence, then verify that components do not spend the same reserve twice.

Optimization can price margin, but the price is conditional. A tiny objective improvement that consumes all thermal or schedule reserve is fragile. Report distance to constraints and response after one component misses its promise. Robustness requires scenario-specific margin; resilience requires detection and recovery after margin is exhausted.

Structures and physical feasibility

Mechanical and civil optimization shape matter to carry loads, manage motion and heat, endure fatigue, resist hazards, and remain buildable. The governing equations are physical; the design variables may be dimensions, topology, materials, controls, or construction sequences.

Size, shape, and topology

A structural sizing problem may minimize mass using a vector of fixed element mass coefficients m

$$\min_x m^T x \tag{76}$$

subject to equilibrium $K(x)u = f$, stress limits, displacement limits, buckling, vibration, fatigue, and manufacturability. Shape optimization moves boundaries. Topology optimization decides where material exists, often through a density field $0 \le \rho_e \le 1$. A common compliance formulation is

$$\min_\rho f^T u(\rho) \quad \text{subject to} \quad K(\rho)u = f, \quad \textstyle\sum_e v_e \rho_e \le V. \tag{77}$$

Penalization encourages near-binary densities; filters and length-scale constraints suppress mesh-dependent checkerboards (Bendsøe and Sigmund 2004; Sigmund and Maute 2013). The mathematically optimal geometry may still need support removal, minimum radii, standardized stock, inspection access, and tolerances.

Mechanics across scales

Material optimization spans microstructure, component, assembly, and infrastructure. Homogenization links microstructure to effective properties; multiscale methods trade fidelity against cost. Metamaterials invert the usual process by designing geometry to obtain stiffness, wave, thermal, or acoustic behavior.

Contact, fracture, plasticity, turbulence, and large deformation create nonsmoothness or expensive simulation. Adjoint methods compute gradients of a scalar objective with cost weakly dependent on the number of design variables, but deriving and verifying them is substantial work. Finite differences remain useful as a gradient check, not as a universal method.

Civil systems and codes

Buildings, bridges, dams, and geotechnical works face uncertain loads, long lifetimes, changing climate, and public consequences. Code compliance is a floor, not an objective. Reliability-based design uses failure probability or reliability index, while performance-based design evaluates multiple hazard levels and consequences.

Construction scheduling couples design with temporary states. A final structure may be safe although an erection stage is not. Whole-life carbon can conflict with cost and schedule; reuse and adaptability can dominate material efficiency over decades. Equity enters infrastructure through who receives service, who bears disruption, and who is exposed to residual risk.

The density formulation earlier in this section already distinguishes size, shape, and topology decisions. Its filters, length scales, and near-binary densities are only the mathematical beginning: draw directions, supports, tolerances, inspection, repair, and standardized stock define the production-feasible set.

Uncertainty comes from loads, material properties, geometry, degradation, model discrepancy, and future use. Reliability-based design treats failure probability; robust design protects against sets; safety factors compress several uncertainties into a traditional rule. Combining them without understanding can double-count conservatism or leave gaps. Codes define minimum admissibility, not the complete objective.

Civil infrastructure adds long horizons and public distribution. Construction sequence, temporary works, maintenance access, climate change, adaptability, and end-of-life matter. A low-mass design can have high embodied carbon; a low-cost route can divide a community; a locally resilient structure can transfer flood risk. The lifecycle and geography are part of the feasible system.

Validate material coupon, component, subsystem, integrated system, and operation at their relevant scales. Model correlation across levels rather than assuming a coupon property transfers unchanged to a manufactured assembly. Tolerances, joints, interfaces, software timing, operators, and environment enter progressively.

For control, prove or test invariant regions under actuator limits and delay, then challenge estimator error and mode transitions. For structures, combine stress and deformation with fatigue, stability, damage tolerance, and manufacturability. For hardware, verify logical function, timing, power integrity, thermal behavior, packaging, and workload performance. A system certificate is a composition of conditional claims, not a stack of component pass marks.

Signals, estimation, sensing, and control

Electrical systems optimize energy, information, dynamics, and uncertainty. Circuit design, estimation, filtering, communication, power conversion, and control share mathematical objects—quadratic forms, spectra, state spaces, norms, and feedback—while differing in physical interpretation.

Signals and estimators

Least squares estimates x from measurements $y = Ax + \varepsilon$ by minimizing

$$\| Ax - y \|_2^2. \tag{78}$$

Weighted least squares reflects unequal noise; regularization stabilizes ill-posed inversion; sparsity penalties support compressed representations. Fourier and wavelet transforms choose bases in which signals or operators become simple. Wiener and Kalman filters optimize mean-square estimation under stated stochastic models (Oppenheim and Schafer 2010; Kalman 1960). Model mismatch, non-Gaussian tails, and sensor bias require robust or nonlinear alternatives.

Information and communication

For a discrete source X, entropy is

$$H(X) = -\sum_x p(x)\log_2 p(x). \tag{79}$$

Channel capacity is the maximum mutual information over input distributions,

$$C = \max_{p(x)} I(X;Y), \tag{80}$$

under the channel model and constraints (Shannon 1948). Coding approaches these limits with delay, complexity, and energy costs. Network design adds routing, congestion, spectrum allocation, interference, fairness, and resilience.

Feedback control

For a linear system

$$x_{t+1} = Ax_t + Bu_t, \qquad y_t = Cx_t, \tag{81}$$

linear-quadratic regulation minimizes

$$J = \sum_{t=0}^{\infty}(x_t^T Q x_t + u_t^T R u_t). \tag{82}$$

Optimality relies on stabilizability, detectability, and the model. Robust control bounds uncertainty; adaptive control updates parameters; model predictive control repeatedly solves a finite-horizon constrained problem using new state estimates (Åström and Murray 2008; Rawlings, Mayne, and Diehl 2017). Delay and saturation can destabilize a controller that looks optimal in an instantaneous model.

Circuit and power design

Analog and digital circuits balance gain, bandwidth, noise, linearity, power, area, yield, and temperature. Power electronics balance switching loss, passive size, electromagnetic compatibility, and control. Electric grids coordinate generation, storage, transmission, demand response, and reserves under network physics. Security-constrained optimal power flow must survive credible contingencies, not only minimize nominal generation cost.

Signal processing chooses representations and estimators under noise, bandwidth, latency, and compute. A filter optimized for mean-square error may suppress rare but important transients. Compression optimized for perceptual quality may damage downstream diagnosis. The intended consumer of the signal determines the loss.

Control closes the loop. A controller designed for a nominal plant must tolerate uncertainty, delay, saturation, and sensor failure. State estimation and control are coupled because actions depend on inferred state and actions affect future observability. Model predictive control enforces constraints by repeated optimization, while classical feedback, robust control, and safety filters can provide complementary guarantees (Kalman 1960; Åström and Murray 2008; Rawlings, Mayne, and Diehl 2017).

Human operators are part of the control architecture. Automation changes skill, attention, mode awareness, and workload. An intervention should specify when the human receives information, what action remains possible, how rapidly control can be transferred, and whether the interface supports diagnosis under stress.

Sensors are often chosen before controllers and models. Yet sensing determines observability, fault detection, wiring, energy, latency, maintenance, privacy, and cost. Co-design sensor placement, precision, sampling, redundancy, and calibration with the decisions the system must make.

Value-of-information analysis can show which measurement changes action. Observability analysis shows which states can be reconstructed under a model. Neither captures maintenance access, cyberattack, consent, or the trust people place in a measurement. Add those constraints explicitly.

Redundancy should consider common cause. Two identical sensors sharing power, location, calibration software, or environmental exposure are not independent. Diverse sensing and analytical consistency checks can improve diagnosis. When sensors disagree, preserve the disagreement and enter a safe mode rather than averaging away a fault.

Hardware beyond PPA

Hardware optimization works under unusually hard constraints: geometry is discrete, fabrication is expensive, verification is unforgiving, and decisions persist for years. The familiar design triangle is power, performance, and area (PPA), but yield, reliability, security, thermal behavior, verification time, and software compatibility also matter.

Architecture and the energy cost of movement

Processor design allocates transistors and power across execution units, caches, predictors, interconnects, accelerators, and memory controllers. Dennard scaling once allowed smaller transistors to switch faster at roughly constant power density; its breakdown made energy and parallelism central (Dennard et al. 1974; Hennessy and Patterson 2019). Data movement often costs more energy than arithmetic, motivating locality, compression, near-memory computing, and specialized accelerators (Horowitz 2014).

Speedup from an accelerator is limited by the fraction it accelerates, conversion overhead, and utilization. A tensor unit with high peak throughput may be idle because of memory bandwidth, sparsity, small batches, or control flow. Hardware-software co-design shapes algorithms, numerical formats, compilers, and silicon together (Jouppi et al. 2017).

From logic to layout

Electronic design automation transforms a specification through synthesis, technology mapping, floorplanning, placement, clock-tree synthesis, routing, timing closure, power integrity, and sign-off. Each stage changes the feasible region of the next. Placement may minimize a surrogate such as weighted wirelength,

$$\min_{(x_i, y_i)} \sum_{e \in E} w_e \ \mathrm{HPWL}(e), \tag{83}$$

subject to density and block constraints. True performance depends on routed parasitics, congestion, timing, voltage drop, and thermal coupling.

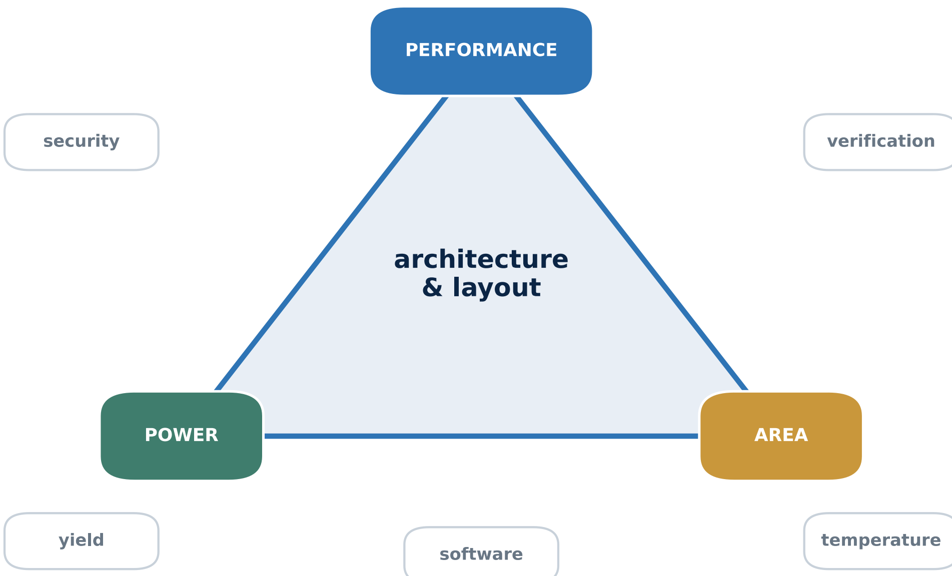


Figure 17. Hardware trade space around performance, power, area, and architecture or layout, with security, verification, yield, software, and temperature.

Figure 17 extends the familiar PPA triangle: temperature, yield, and verification can determine whether a nominally efficient hardware point is buildable and dependable.

Learned placement can search large design spaces, but comparisons require fixed technology files, constraints, downstream tools, and sign-off metrics (Mirhoseini et al. 2021). Human floorplanners encode tacit constraints that a nominal objective may omit.

Variation, yield, and verification

Manufacturing variation makes timing and power distributions rather than constants. Statistical static timing, corner analysis, guard bands, redundancy, and error correction manage tails. Over-margining wastes area and energy; under-margining harms yield and field reliability.

Functional verification commonly dominates project effort. Simulation, emulation, formal equivalence, property checking, coverage, and post-silicon validation are complementary. A faster design that cannot be verified before the market window is not optimal.

Power, performance, and area are a useful triangle, not a complete objective. Temperature, reliability, yield, variation, verification, security, programmability, development schedule, software ecosystem, and embodied cost can dominate. Moving data often consumes more energy than arithmetic, so memory hierarchy and placement are architectural decisions (Horowitz 2014; Hennessy and Patterson 2019). Specialized accelerators improve selected workloads while narrowing flexibility (Jouppi et al. 2017).

Electronic design automation compiles intent through synthesis, placement, routing, timing, power, verification, and sign-off. Each stage optimizes a surrogate for downstream reality. Early wirelength estimates guide placement but do not equal routed delay. Static timing is a conservative abstraction, not a physical waveform. Design-rule compliance does not guarantee yield. Locally improving one stage can worsen another; iterative closure is therefore a coupled fixed-point process.

Machine learning can propose floorplans or tune tool parameters (Mirhoseini et al. 2021), but its reward must reflect sign-off and reproducibility. A policy that performs well on training designs may fail on a new process, hierarchy, or macro configuration. Warm starts and learned guidance are safest when conventional analysis retains feasibility and sign-off authority.

Consider an edge accelerator for a medical imaging device. Variables include model architecture, arithmetic precision, memory size, dataflow, clock, voltage, floorplan, thermal path, and software batching. Objectives include inference accuracy, worst-case latency, energy per study, cost, reliability, and updateability. Hard constraints include diagnostic performance, maximum temperature, regulatory configuration control, memory, timing, and safe fallback.

Optimizing the neural model alone may select operators inefficient on the target hardware. Optimizing hardware against a frozen model can create a brittle specialization. Co-design alternates or jointly searches model and architecture, but evaluation must include compilation, calibration, corner timing, thermal transients, and clinical workflow. Quantization error interacts with subgroup performance; batching interacts with latency; thermal throttling interacts with sustained throughput. Verification must cross all interfaces.

The design review should present a Pareto set and a traceable reason for selecting one point. It should retain a margin budget, not spend every reserve. It should specify which future model updates remain feasible. Option value—room for new algorithms, security patches, or sensor changes—is a legitimate engineering objective.

Software assumes instruction behavior, memory, timing, precision, concurrency, and fault modes. Hardware assumes workload, locality, duty cycle, thermal environment, and acceptable approximation. Co-design should exchange distributions and envelopes, not a benchmark point.

Approximate arithmetic can improve energy or throughput and alter numerical stability, fairness, or safety. Validate at application and consequence level. A small average error can concentrate near a threshold. Compiler transformations and quantization can change models differently across hardware; regression tests should span targets.

The interface contract includes update. A hardware accelerator can freeze assumptions for years while software and threats evolve. Preserve a fallback path, programmability where justified, and a versioned compatibility matrix.

Field checklist

- Define interface contracts with units, ranges, uncertainty, fidelity, and ownership.
- Expose coupling variables and decide deliberately between monolithic and decomposed coordination.
- Distinguish mathematical, physical, manufacturing, code, and lifecycle feasibility.
- Model state estimation, control, delay, saturation, fallback, and human transfer together.
- Expand PPA to temperature, yield, reliability, verification, security, software, and option value.
- Validate every EDA or design surrogate against the downstream sign-off quantity it represents.
- Preserve margins and future feasible updates rather than optimizing all slack away.

Translation lens. Engineering transfers through conservation laws, interfaces, sensitivities, and margins. Co-design is categorical in spirit: subsystem models compose only when boundary types and preservation claims are explicit.

Chapter 11 · Processes, materials, manufacturing, energy, climate, robotics, and transport

Processes, manufacturing, and safe intensification

Chemical and manufacturing optimization governs transformations of matter under conservation laws, kinetics, thermodynamics, equipment limits, uncertainty, and safety. The central question is rarely maximum yield alone; selectivity, purity, energy, throughput, operability, emissions, and hazard interact.

Process design and operation

A steady-state process model may satisfy

$$F(z, x) = 0, \tag{84}$$

where z contains temperatures, pressures, flows, and compositions, and x contains design or control choices. The optimizer minimizes cost or environmental impact subject to these equations and inequalities. Heat integration, reactor-separator-recycle structure, utilities, and equipment sizing create nonconvex mixed-integer nonlinear programs (Biegler 2010; Edgar, Himmelblau, and Lasdon 2001).

Thermodynamic phase equilibrium and reaction kinetics can generate multiple steady states. A solution that is mathematically feasible may be dynamically unstable or hard to start up. Operability analysis, controllability, hazard review, and relief design must surround economic optimization. Process intensification can remove unit operations but may concentrate couplings and risk.

Materials by design

Materials optimization connects composition and processing to microstructure, properties, performance, and lifecycle. High-dimensional composition spaces motivate design of experiments, surrogate models, and Bayesian optimization. Physics-based models improve extrapolation; data-driven models exploit accumulated measurements; active learning chooses the next experiment by balancing information and potential improvement.

The inverse problem—find a material with target properties—is often many-to-one and constrained by synthesizability. A candidate is not a material technology until it can be made reproducibly, characterized, scaled, joined, certified, and recycled.

Manufacturing systems

Manufacturing optimization includes product geometry, process parameters, toolpaths, nesting, line balance, inventory, inspection, maintenance, and supply networks. Additive manufacturing expands feasible geometry while imposing anisotropy, support, distortion, surface, and qualification constraints. Subtractive manufacturing rewards accessible features and standard tooling. Design for manufacture and assembly brings these realities upstream.

Quality control distinguishes common-cause from special-cause variation. Tightening tolerances everywhere can increase scrap and cost without improving function. Robust parameter design seeks settings that reduce sensitivity to noise. Predictive maintenance should optimize intervention decisions, not merely predict failure scores.

Process systems choose equipment, flows, temperatures, pressures, recipes, schedules, and controls under thermodynamics, kinetics, safety, and product specifications (Biegler 2010; Edgar, Himmelblau, and Lasdon 2001). Steady-state economic optimization can conflict with dynamic controllability. A design near a thermodynamic or safety boundary may be efficient nominally and operationally fragile. Hazard constraints, relief, startup, shutdown, cleaning, and abnormal operation belong to the feasible set.

Manufacturing optimization spans facility layout, job sequencing, lot size, maintenance, quality, inventory, and supply. High utilization lengthens queues and reduces maintenance windows. Large batches improve setup efficiency while increasing delay, work in progress, and defect exposure. Statistical process control and feedback should be integrated with scheduling, because the process producing the data changes over tool life and product mix.

Digital manufacturing links geometry to process planning. Additive designs need support, orientation, scan strategy, residual-stress control, and post-processing. Subtractive designs need tool access, fixturing, and tolerances. A material-efficient shape can be production-inefficient. Optimization should compile through the actual manufacturing route.

Consider a plant with batch processes, storage, renewable electricity, time-varying prices, and delivery commitments. Decisions choose batch start, recipe, inventory, energy purchase, and flexible load. A mixed-integer model can exploit low-price or low-carbon periods, but thermal and material states carry across time. Storage converts temporal variability into capacity and loss.

Include startup, shutdown, cleaning, off-specification risk, minimum run, operator coverage, demand charges, and emergency states. Distinguish average grid carbon from marginal response. A plan that moves every batch to the same “green” hour can create a new peak and change the marginal generator.

Use robust constraints for critical delivery and process safety, stochastic scenarios for price and renewable variation, and receding-horizon updates for operation. Reserve a simple feasible schedule for forecast or solver failure. Compare absolute energy and material throughput so efficiency gains do not hide rebound.

Process intensification can reduce equipment size, inventory, energy, or waste by increasing coupling or operating closer to limits. The improvement may reduce buffer and complicate control. Evaluate hazard, controllability, maintainability, startup, shutdown, and fault propagation alongside steady-state efficiency.

Inherently safer design removes or reduces hazard rather than adding control layers. Prefer less hazardous material, lower inventory, moderated conditions, or simpler pathways where feasible. Optimization can compare alternatives but should not price a catastrophic hazard as an ordinary expected cost.

Lifecycle sustainability includes feedstock, land, water, energy, labor, emissions, toxicity, durability, repair, and end-of-life. Report burden shifting by stage and region. A low operational footprint can depend on scarce or harmful upstream material.

Materials by inverse design and evidence

Materials design maps composition, microstructure, processing, and history to properties and performance. The inverse problem is ill-posed: many structures can realize similar properties, simulations span scales, and data are sparse or biased toward successful experiments. Surrogates and active learning can guide experiments, but uncertainty should control when the model extrapolates.

A closed-loop materials campaign selects a candidate, synthesizes it, measures properties, updates the model, and selects again. Failed synthesis is information about feasibility, not missing data to discard. Multi-fidelity models combine inexpensive calculations, high-fidelity simulation, and laboratory evidence. The acquisition objective can include novelty, uncertainty reduction, cost, hazard, and expected improvement. Human scientific judgment remains essential for mechanisms and anomalous results.

Engineering models move between electronic, material, component, process, plant, network, and lifecycle scales. Fine-scale behavior becomes effective parameters at the next scale. This closure is a model, not a guaranteed averaging. Microstructure can create anisotropy; rare defects can dominate fatigue; local congestion can create network instability.

Validate closure where optimization pushes the design. A surrogate material law calibrated on conventional geometry may fail in topology-optimized thin members. A grid model with aggregated loads may fail when every controller responds to the same price. Carry uncertainty and trigger higher-fidelity analysis for extreme candidates.

Use multi-fidelity optimization with a promotion rule. Cheap models screen broadly; intermediate models refine; authoritative experiments confirm. Preserve candidates that disagree across fidelities because disagreement reveals model structure.

Inverse design proposes composition or microstructure for target properties. The loop must include synthesizability, process window, characterization, uncertainty, durability, scale-up, toxicity, supply, and end-of-life. A predicted material that cannot be made reproducibly is not feasible.

Use active learning to choose experiments that improve decision-relevant knowledge, preserve negative and failed results, and challenge surrogate extrapolation. Diversity among candidates protects against correlated model error and supply risk. Confirm performance with independent methods and conditions.

The object transferred from computation to manufacturing is not a formula alone. It is a recipe, tolerance, provenance, and evidence envelope.

Energy, climate, conservation, and circularity

Energy and environmental optimization spans seconds of grid balancing, decades of infrastructure, and centuries of climate consequence. It is marked by uncertainty, irreversibility, spatial inequity, and multiple decision makers.

Energy-system planning

Capacity expansion chooses generation, storage, transmission, and demand-side resources. A stylized model minimizes discounted system cost

$$\min_{x} \sum_{t=0}^{T} \frac{C_t(x_t)}{(1+r)^t} \tag{85}$$

subject to demand balance, reliability, build rates, network limits, emissions, and technology constraints. Hourly or subhourly operations must be represented well enough that a capacity plan remains feasible. Representative periods reduce computation but can erase rare stress events and long storage cycles.

Unit commitment and dispatch coordinate discrete starts, ramping, reserves, network flow, and uncertain renewable output. Storage arbitrage alone understates capacity, flexibility, congestion, and resilience value. Demand is partly a design variable shaped by efficiency, tariffs, automation, and social practice.

Carbon and ecological boundaries

Lifecycle assessment accounts for impacts across extraction, production, use, and end of life. Results depend on system boundary, allocation rule, counterfactual, time horizon, and data quality. Optimizing one indicator can shift burden to land, water, toxicity, biodiversity, or another region.

Climate mitigation pathways use integrated models to explore technology, land, economy, and policy under carbon budgets (Intergovernmental Panel on Climate Change 2022b). They are scenarios conditioned on assumptions, not forecasts. Planetary-boundary research emphasizes interacting Earth-system limits (Rockström et al. 2009). A safe operating space is more naturally represented by multiple constraints than by a single monetized objective.

Adaptation, resilience, and justice

Flood defenses, heat plans, water systems, ecosystems, and emergency capacity must perform across uncertain futures. Robust decision-making seeks actions that are acceptable in many plausible worlds; adaptive pathways stage decisions and preserve options. Discounting strongly affects intergenerational comparisons. Distribution matters because aggregate benefit can conceal concentrated displacement, pollution, or energy poverty.

Circular-economy design optimizes loops of durability, repair, reuse, remanufacture, and recycling. Yet circularity is not automatically low impact: extra transport, processing, or rebound can offset gains. Absolute resource and emission outcomes remain the test.

Chemical processes, manufacturing lines, energy systems, transport networks, and robots all obey balances. Mass, atoms, charge, energy, momentum, inventory, vehicles, or tasks cannot disappear merely because a model boundary is narrow. Conservation equations are among the strongest structures available for optimization because they prune impossible solutions and provide diagnostic residuals.

For a process network with flow f_{ij} from node i to node j, a steady balance has the form

$$\sum_{j:(j,i)\in E} f_{ji} + s_i = \sum_{j:(i,j)\in E} f_{ij} + d_i, \qquad i \in V, \tag{86}$$

where s_i and d_i represent supply and demand. Equation (86) can describe molecules, electricity under an approximation, vehicles, data, or money, but each domain adds different physics and loss. The equation is a structural correspondence, not an assertion that the entities are interchangeable.

Energy planning couples investment and operation across decades. Variables include generation, storage, transmission, demand response, efficiency, fuel, and retirement. Objectives include cost, emissions, reliability, resilience, land, water, and distribution. Weather and demand are correlated; technology costs and policy change; extreme events dominate adequacy. A single representative year can misprice storage and reliability.

Dispatch models optimize hourly or subhourly operation under network and unit constraints. Planning models choose capacity with simplified operations. Linking them requires chronology, transmission, reserves, maintenance, and rare events. Decomposition and scenario reduction manage scale but can erase the exact periods that drive adequacy. Validate a plan in an independent chronological simulation and stress compound events.

Climate mitigation uses lifecycle boundaries. Electrification can move emissions upstream; efficiency can induce rebound; mineral extraction and land use create new burdens. Adaptation values robustness, redundancy, and option value under deep uncertainty. Justice requires examining who pays, who receives reliable service, whose land is used, and who is exposed during transition (Intergovernmental Panel on Climate Change 2022b; Rockström et al. 2009; Raworth 2017).

Whole-life optimization extends the lifecycle to maintenance, repair, reuse, disassembly, recycling, and disposal. A product optimized for manufacturing cost can be difficult to repair; a lightweight composite can be hard to recycle; a low-operating-energy device can have high embodied impact. Design variables should include modularity, standard fasteners, material separation, diagnostic access, software support, and residual value.

Circularity is not achieved by adding a recycling percentage to an objective. Material quality degrades, reverse logistics consumes energy, markets fluctuate, and informal labor may bear risk. Mass-balance and provenance make claims auditable. The preferred intervention may be longer life or reduced demand rather than more efficient throughput.

Conservation equations are powerful because they travel across domains, but every implementation should close a balance. For material, energy, charge, inventory, people, or financial flow, compute input minus output minus accumulation and compare the residual with measurement and numerical error. A persistent residual indicates missing flow, inconsistent units, sensor bias, or a solver defect.

Balances should be checked at nested boundaries. A factory can close mass onsite while transferring waste upstream; a grid model can balance electrical energy and omit fuel or embodied infrastructure; a transport model can conserve vehicles and omit driver time. The algebra closes only for the chosen ontology.

End-of-life is shaped upstream by material choice, joining, modularity, information, and ownership. A circular system is a reverse logistics network with uncertain timing, quality, and demand. Decisions cover collection, inspection, repair, remanufacture, reuse, recycling, and disposal.

The objective should respect a hierarchy: prevent unnecessary material, extend useful life, preserve high-value function, recover material where appropriate, and manage residual harm. A recycling-rate target can reward heavy low-value material and ignore toxicity or downcycling. Track mass, function, quality, energy, and displacement of virgin production.

Design for disassembly can conflict with durability or safety; modularity can add material; local repair can require training and parts. Pareto analysis should expose these trades. Policy, warranty, right-to-repair, and producer responsibility change the feasible network. Circularity is institutional co-design, not a material property.

Robotics and transportation in open environments

Aerospace, robotics, and transportation combine geometry, dynamics, control, perception, networks, human behavior, and safety. Optimization must bridge design-time models and real-time decisions.

Trajectory and vehicle design

An optimal-control problem chooses state $x(t)$ and control $u(t)$ to minimize the following objective, where Φ denotes the terminal cost and L the running cost:

$$J = \Phi\big(x(T)\big) + \int_0^T L\,(x(t), u(t), t)\,dt \tag{87}$$

subject to dynamics $\dot{x} = f(x, u, t)$, boundary conditions, and path constraints. Direct transcription discretizes the trajectory into a nonlinear program; indirect methods use necessary conditions from optimal control (Pontryagin et al. 1962; Bryson and Ho 1975). Fuel, time, heating, noise, comfort, risk, and control effort may conflict.

Aircraft and spacecraft design couple aerodynamics, structures, propulsion, thermal protection, guidance, manufacturability, and mission operations. A mass reduction in one subsystem may demand greater cooling or certification effort elsewhere. Margins protect against model error and late growth.

Robotics

Robot optimization spans mechanism design, calibration, state estimation, motion planning, manipulation, and policy learning. A motion planner searches configuration space while avoiding obstacles; trajectory optimization refines a smooth feasible path; feedback control rejects disturbances. Perception uncertainty should alter the plan, not remain an annotation.

For humans and robots sharing space, the objective must include legibility, comfort, recoverability, and conservative response to ambiguity. A nominally shortest path that surprises a person is not behaviorally efficient.

Transportation networks

Wardrop's first principle describes user equilibrium: used routes have equal and no greater travel cost than unused alternatives (Wardrop 1952). A system optimum minimizes total travel time. These differ because each traveler ignores congestion imposed on others. Pricing, information, capacity, and service design can move the equilibrium, but demand adapts.

Traffic assignment, transit scheduling, fleet routing, crew planning, charging, and last-mile logistics operate at different levels (Sheffi 1985). Accessibility—what opportunities people can reach—is often a better goal than vehicle speed. Safety, emissions, affordability, disability access, land use, and community severance are core outcomes.

Robotics composes perception, estimation, planning, control, and actuation under real-time constraints. A collision-free planned trajectory is not necessarily safe under localization error, moving agents, delay, or actuator saturation. Hierarchical architectures trade global foresight against fast reactive safety. Formal safety filters can constrain a learned or optimizing policy, but only within modeled dynamics and sensing.

Transportation optimization includes routes, schedules, fleet, pricing, infrastructure, and access. User equilibrium differs from system optimum: individually shortest routes can create congestion that a coordinated assignment would avoid (Wardrop 1952; Sheffi 1985). Pricing or routing advice changes behavior and raises distributional questions. Travel time is not the only objective; reliability, safety, emissions, disability access, affordability, and neighborhood effects matter.

Autonomous and human-driven participants form a mixed system. A policy optimized in a compliant simulator may provoke or confuse real drivers. Field trials should expand gradually, preserve a safe operating domain, and record near misses and interventions. Learning from rare events requires careful exposure denominators and counterfactual analysis.

A robot policy composes perception, state estimation, planning, control, human interaction, and mechanical limits. Simulation enables repeated search and can omit contact, lighting, rare obstacles, social norms, or human improvisation. Test sim-to-real transfer by mechanism, not only aggregate success.

Define safe reachable sets, speed and force limits, stop and recovery, uncertainty-aware planning, and human communication. Operators need mode awareness and authority. In shared spaces, legibility and predictability are performance qualities.

Optimization should include wear, energy, maintenance, and the distribution of work. A robot that raises nominal throughput by creating unplanned human recovery tasks has moved cost across the boundary.

Travel demand responds to routes, price, land use, reliability, and information. A system-optimal flow minimizes a declared total consequence; individual users choose perceived routes and times, producing an equilibrium such as Wardrop's (Wardrop 1952; Sheffi 1985). The two need not coincide.

Infrastructure expansion can induce demand and reshape development. Evaluate short- and long-horizon response, mode shift, access, emissions, safety, and land use. A travel-time improvement is not automatically an accessibility improvement if destinations or affordable housing move.

Scheduling and routing should include transfer reliability, walking and waiting, disability access, freight, maintenance, and disruption. A network with high average performance can isolate a community after one bridge or line fails. Critical connectivity and recovery deserve constraints and stress tests.

Pricing can internalize congestion under a model and can burden people without alternatives. Pair it with distributional analysis, revenue use, and service investment. A shadow price from a network model is not a legitimate toll until policy and justice questions are answered.

Field checklist

- Write conservation and inventory balances before adding economic objectives.
- Include startup, shutdown, cleaning, maintenance, hazard, and abnormal operation.
- Compile designs through the actual manufacturing and inspection route.
- Treat failed experiments as feasibility evidence in inverse design.
- Validate energy plans with independent chronology and compound-event stress tests.
- Distinguish user equilibrium, system optimum, and legitimate transport policy.
- Extend lifecycle boundaries through repair, reuse, disassembly, and disposal.

Translation lens. Physical and infrastructural optimization transfers through balance, flow, dynamics, and lifecycle. The conserved quantity may change, but the discipline of accounting for sources, sinks, losses, and boundary movement remains invariant.

Part IV · Living Systems, Institutions, and Society

Chapter 12 · Biology, clinical medicine, public health, and biomedical discovery

Evolution and contextual biological optimality

Fitness landscapes and frequency dependence

A genotype or phenotype x may be assigned fitness $w(x)$ in a fixed environment, producing a landscape. Mutation and recombination determine which moves are accessible; drift matters in finite populations; environments change. If fitness depends on population composition p, then

$$w_i = w_i(p), \tag{88}$$

and the landscape moves as the population moves. Replicator dynamics model frequency change:

$$\dot{p}_i = p_i\big(w_i(p) - \bar{w}(p)\big), \qquad \bar{w}(p) = \textstyle\sum_{j=1}^{m} p_j\, w_j(p). \tag{89}$$

Evolutionarily stable strategies describe resistance to invasion under a game, not moral or social desirability (Maynard Smith 1982). Traits can be by-products, exaptations, or consequences of constraints; adaptationist stories need alternatives and evidence (Gould and Lewontin 1979).

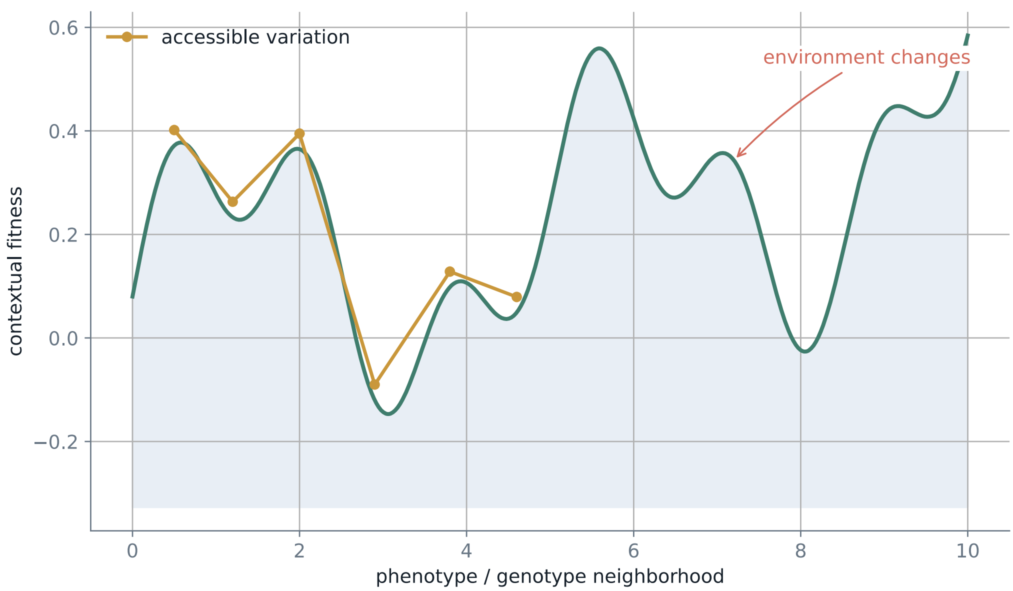


Figure 18. Contextual-fitness landscape with accessible variation and environmental change.

Figure 18 should not be read as a fixed hill-climbing objective: mutation, drift, environmental change, and moving niches continually reshape the adaptive landscape.

Metabolism and cellular allocation

Cells allocate limited energy, ribosomes, membrane, and elements across maintenance, growth, defense, and reproduction. Flux balance analysis represents steady-state metabolism with stoichiometric matrix S:

$$Sv = 0, \qquad \ell \le v \le u, \tag{90}$$

then optimizes a proxy such as biomass production $c^T v$ (Orth, Thiele, and Palsson 2010). The model is powerful because conservation constrains possibilities, but its objective and steady-state assumptions must be experimentally checked. Multiobjective and regulatory extensions help explain trade-offs among growth, yield, stress, and by-products.

Ecology and conservation

Ecological optimization concerns populations, food webs, spatial connectivity, harvesting, reserves, invasive control, and restoration. The system may have tipping points, delayed feedback, and alternative stable states (Levin 1999). Maximum sustainable yield can fail when recruitment is uncertain or ecosystem interactions are omitted. Precautionary harvest control and protected-area design manage uncertainty and option value.

Conservation planning often seeks a minimum-cost set of sites meeting representation and connectivity targets. But ecological value is not fully captured by species counts: evolutionary history, function, cultural relationships, and future adaptation matter. Community consent and land tenure are constraints on legitimate conservation, not implementation details.

Biological design versus human design

Synthetic biology and bioengineering intentionally optimize genetic circuits, metabolic pathways, proteins, and ecosystems. Their design space inherits evolution: engineered organisms mutate, compete, and move. Containment, reversibility, monitoring, and evolutionary stability must enter the objective.

Evolutionary language often sounds optimizing, but natural selection has no external designer, fixed global objective, or guaranteed progress. Fitness is relative to an environment, population, and time; frequency dependence makes the landscape move as the population changes. Mutation, recombination, drift, developmental constraint, and historical contingency shape accessible paths. An observed trait may be adaptive, neutral, constrained, or a by-product (Darwin 1859; Fisher 1930; Gould and Lewontin 1979).

Optimality models in biology are conditional tools. A foraging model can predict behavior under energy, risk, and time assumptions; a game-theoretic model can identify evolutionarily stable strategies (Maynard Smith 1982; Stephens and Krebs 1986). Agreement supports the assumptions but does not prove that organisms solve an explicit program. Disagreement can reveal missing constraints or a wrong scale of selection.

Metabolic and ecological models exploit conservation. Flux balance analysis constrains steady-state reaction flows and optimizes a proxy such as biomass (Orth, Thiele, and Palsson 2010).

The proxy can be effective in a laboratory condition and wrong under stress, community interaction, or regulation. Conservation planning similarly chooses reserves or interventions under connectivity, uncertainty, and multiple species; persistence, not static coverage, is the deeper objective.

Optimality models in biology are explanatory hypotheses under an environment, inheritance, trade-offs, and history. They can predict foraging, allocation, morphology, or behavior when constraints and fitness consequences are well specified (Parker and Maynard Smith 1990; Stephens and Krebs 1986). They should be compared with neutral, developmental, phylogenetic, and historical explanations.

An apparent optimum does not prove direct adaptation. Multiple mechanisms can produce the same phenotype; environments vary; evolution works with inherited material; selection acts at context-dependent levels; and measured reproductive success is noisy. The right question is often whether an optimality model generates discriminating predictions, not whether nature "solved" a program (Gould and Lewontin 1979).

Clinical decisions and treatment policies

Clinical optimization chooses among imperfect actions for a particular person under uncertain diagnosis, prognosis, treatment response, side effects, preferences, time, and resources. It is decision support, not a replacement for a patient's values or a clinician's responsibility.

From probabilities to actions

Suppose disease state $D \in \{0,1\}$ and action a is test, treat, watch, or refer. Expected utility is

$$EU(a \mid x) = \sum_{d \in \mathcal{D}} P\,(D = d \mid x)\, U(a, d; x). \tag{91}$$

The preferred action maximizes expected utility, subject to consent and clinical constraints. If treatment benefit increases with disease probability while treatment harm does not, a threshold probability can separate "do not treat" from "treat" (Pauker and Kassirer 1980). A threshold is not universal: it changes with disease severity, treatment burden, comorbidity, alternatives, and the person's preferences.

Bayes' theorem updates odds with evidence:

$$\frac{P(D|T)}{1-P(D|T)} = \frac{P(T|D)}{P(T|\neg D)} \frac{P(D)}{1-P(D)}. \tag{92}$$

Likelihood ratios characterize a test, while pretest odds place it in context. Testing has value only if results can change an action or meaningfully inform the person.

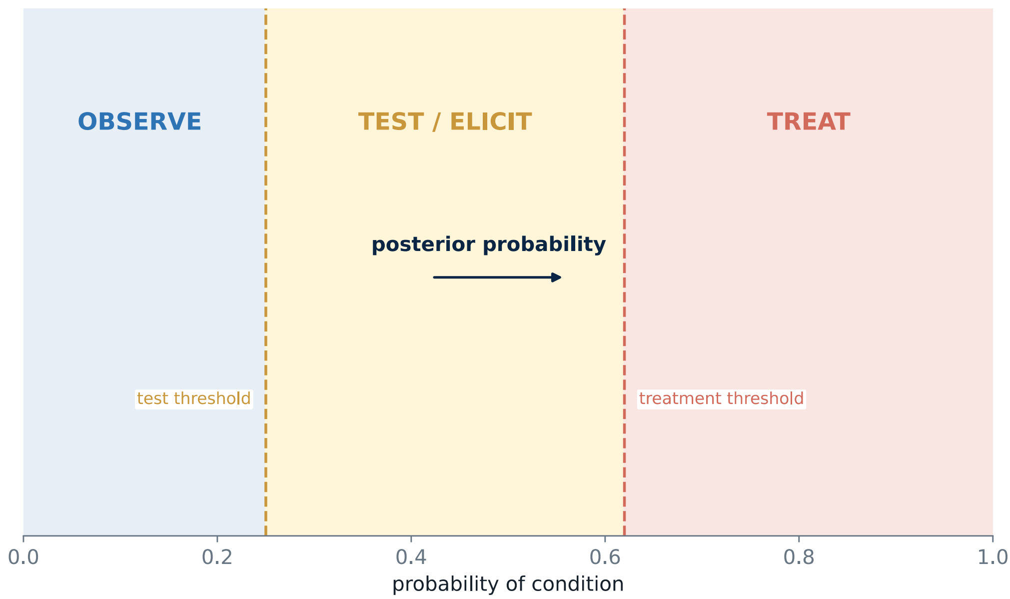


Figure 19. Two clinical thresholds divide posterior probability into OBSERVE, TEST / ELICIT, and TREAT regions.

Figure 19 turns a predictive score into an action policy by adding test, treat, and observe regions together with uncertainty and patient preference.

Diagnostic and predictive models

Sensitivity and specificity condition on disease status; positive and negative predictive values depend on prevalence. Discrimination measures ranking; calibration measures whether predicted probabilities match frequencies. Decision-curve analysis evaluates net benefit across plausible thresholds (Vickers and Elkin 2006). Dataset shift, missingness, workflow effects, subgroup performance, and alert fatigue determine clinical value.

An AI system may optimize an intermediate endpoint while worsening care through delay, automation bias, or fragmented responsibility. Prospective evaluation should compare workflows, not only algorithms (Rajkomar, Dean, and Kohane 2019; Topol 2019). Human override is not a safety mechanism unless users have time, information, authority, and feedback.

Treatment choice and preference sensitivity

Randomized trials estimate average causal effects under eligibility and protocol conditions. Individual choices may depend on heterogeneous effects, absolute baseline risk, interactions, competing risks, and treatment burden. Shared decision-making is especially important when options have similar survival but different quality-of-life profiles. Framing can change choices even when outcomes are numerically equivalent (McNeil et al. 1982).

Quality-adjusted life years combine duration and health-state utility,

$$Q = \int_0^T u(t)\, dt, \tag{93}$$

but utilities vary across people and contexts. Disability, age, and chronic illness raise ethical questions that a scalar cannot settle. Clinical constraints, rights, and individual preference should remain visible.

Sequential care

Care is a partially observed sequence: investigate, act, observe, and revise. Dynamic treatment regimes and adaptive trials formalize this sequence. Real care also includes adherence, access, caregiver burden, and changing goals. The most accurate recommendation has no benefit if it cannot be reached, afforded, understood, or sustained.

Clinical prediction estimates a probability or outcome. Clinical decision-making combines that estimate with treatment benefit, harm, patient preference, alternatives, capacity, and uncertainty. Threshold analysis makes the distinction explicit (Pauker and Kassirer 1980; Vickers and Elkin 2006). Below a test threshold, observation may dominate; between thresholds, more information has value; above a treatment threshold, treatment may dominate. The thresholds vary with consequences and patient context.

Shared decision-making is not a final user-interface step. It elicits values that determine the utility function and sometimes the feasible set. One patient may prioritize survival probability; another may prioritize cognitive function, independence, fertility, pain, or avoiding hospitalization. A population-average optimum can be preference-incompatible for an individual.

Calibration must be evaluated in the intended population and decision range. A model can rank well and give systematically wrong probabilities. Dataset shift, verification bias, missing follow-up, treatment-confounder feedback, and unequal measurement affect validity. Medical AI should support an evidence chain from intended use through clinical workflow and monitored outcomes, not merely an external accuracy score (Rajkomar, Dean, and Kohane 2019; Topol 2019; World Health Organization 2021).

Many treatments are sequential. Dose affects future state; response reveals latent biology; adverse effects constrain later options. A policy maps the observed history or belief state to action. Dynamic treatment regimes, adaptive trials, and closed-loop devices use this structure. The state must include clinically relevant memory, not only the latest measurement.

In an insulin-delivery system, the controller balances time in target range, hypoglycemia, hyperglycemia, variability, burden, and device constraints. Meals, exercise, illness, sensor delay, and physiology create uncertainty. Safety layers, alarms, manual override, and fallback basal delivery are part of the controller. An objective optimized in simulation must be validated across representative people and exceptional conditions.

Stopping is a clinical action. A policy should specify when to withhold, refer, obtain information, or defer to a human. Models trained only on cases where treatment proceeded can be weakest exactly at abstention boundaries.

A risk estimate becomes useful only through an action threshold. Treat when expected benefit exceeds expected harm under the person's condition and preferences (Pauker and Kassirer 1980). The threshold depends on treatment effect, adverse effect, alternatives, delay, and what happens after a false positive or false negative.

Calibration is essential: among cases assigned probability p, outcome frequency should be approximately p in the relevant population and period. Discrimination alone cannot support a threshold. Decision-curve analysis compares net consequence across threshold preferences and can reveal that a model adds no value over treat-all or treat-none strategies (Vickers and Elkin 2006).

The action policy must include abstention, additional testing, clinician review, follow-up, and appeal. A prediction model does not own a clinical decision. Validate workflow, capacity, subgroup performance, and patient understanding in addition to statistical accuracy.

Public health and allocation

Public health allocates attention and resources across prevention, surveillance, treatment, workforce, infrastructure, communication, and social determinants. Outcomes emerge through populations and networks, so individual and collective optima may diverge.

Allocation under scarcity

A health system might maximize total expected health gain

$$\max_{x} \sum_{i=1}^{n} B_i\,(x_i) \tag{94}$$

subject to budget, staff, beds, geography, eligibility, and minimum-service constraints. The objective immediately raises questions: Are benefits measured as deaths averted, quality-adjusted life years, unmet need, financial protection, or capability? Are gains weighted by severity, disadvantage, or reciprocity? Cost-effectiveness supplies evidence, not a complete allocation rule (Weinstein et al. 1996; Hunink et al. 2014).

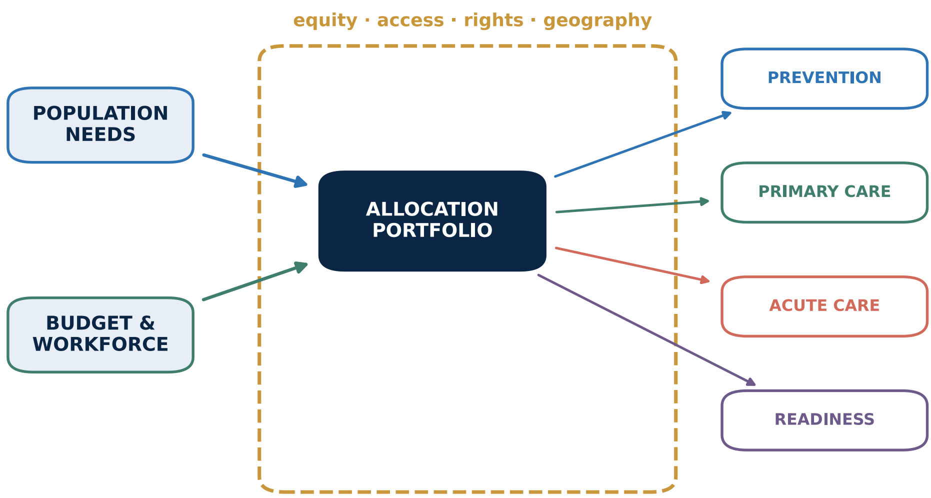


Figure 20. Health-system allocation from needs, budget, and workforce to prevention, primary care, acute care, and readiness, governed by equity, access, rights, and geography.

Figure 20 makes health allocation systemic: prevention, primary and acute care, workforce, geography, and equity safeguards compete for resources while jointly producing outcomes.

Queueing models help size clinics, operating rooms, emergency departments, and laboratories. Minimizing average wait may worsen the longest waits or displace complex cases. Triage rules must be auditable and clinically grounded. Capacity buffers that appear idle are insurance against outbreaks, seasonal peaks, and correlated failures.

Screening and prevention

Screening can advance diagnosis without improving outcomes because of lead-time, length, and overdiagnosis biases. Program evaluation must include downstream confirmation, treatment, false positives, missed disease, participation, and unequal access. A test optimized in a case-control sample may perform poorly in population use.

Vaccination and infection control create externalities: one person’s action changes others’ risk. Compartmental epidemic models approximate transitions among states, while contact networks capture heterogeneity. Optimal intervention timing depends on growth, delay, behavior, health capacity, and uncertainty. Robust policies and transparent communication can be more valuable than a brittle point forecast.

Operations and policy

Hospital optimization covers rostering, bed assignment, theatre scheduling, inventory, referral, and discharge. These interact: maximizing theatre utilization can create recovery-bed congestion; minimizing length of stay can increase readmission or caregiver burden. System-wide objectives and causal evaluation are essential.

Health policy should be monitored for distributional effect. A digital-first service may lower average transaction cost while excluding people without devices, language support, stable housing, or trust. WHO guidance for AI in health emphasizes autonomy, well-being, transparency, accountability, inclusiveness, and sustainability (World Health Organization 2021). Regulation and post-market surveillance add lifecycle obligations (U.S. Food and Drug Administration and Health Canada and Medicines and Healthcare products Regulatory Agency 2021).

Preparedness

Preparedness optimizes before probabilities are known. Stockpiles, flexible contracts, interoperable data, surge staffing, local production, and exercises preserve response options. Their benefit is counterfactual and easy to cut after quiet years. Resilience metrics should therefore track readiness and recovery capacity, not only observed utilization.

Public health optimizes across people and time under budgets, logistics, behavior, and externalities. Vaccination, screening, workforce, stockpiles, and facility location create network and allocation problems. Population benefit can conflict with equity, geographic access, and priority for the worst off. A QALY can support comparison but is ethically incomplete as a sole allocation rule (Weinstein et al. 1996; Neumann et al. 2016; Hunink et al. 2014).

Screening illustrates nested optimization. A test threshold determines referrals; capacity determines delay; delay changes outcome; outreach changes who is observed; false positives create procedures and anxiety; false negatives delay care. Optimizing sensitivity without capacity can degrade the entire pathway. The model boundary should follow the patient through confirmation, treatment, and follow-up.

Preparedness values options that appear idle in normal operation: reserve staff, stockpiles, redundant laboratories, surveillance, mutual aid, and adaptable facilities. A narrow utilization metric prices them as waste. A resilience objective values essential function, safe degradation, recovery, and learning.

Biomedical discovery and translation

Biomedical optimization connects molecules, devices, data, trials, regulation, manufacturing, and care. Its search spaces are vast and its measurements expensive. Biological variability and high consequence make validation central.

Drug discovery and development

A candidate drug must balance potency, selectivity, exposure, toxicity, stability, formulation, manufacturability, and clinical differentiation. Multi-parameter optimization often uses desirability functions or Pareto analysis, but correlated assay error and selection bias can produce “winner’s curse.” Active learning should choose experiments that reduce decision uncertainty, not merely maximize a predicted score.

Dose selection can be modeled through exposure-response and toxicity. A simplistic utility is

$$U(d) = B(d) - \lambda H(d), \tag{95}$$

where both benefit B and harm H are uncertain and heterogeneous. Phase transitions in development are real-option decisions: continue, redesign, partner, or stop after new evidence. Portfolio optimization must resist incentives to hide negative information.

Adaptive trial designs may alter allocation, sample size, arms, or population under prespecified rules. They can improve information and participant benefit but require careful control of error, operational bias, and interpretation. Surrogate endpoints shorten trials only when they reliably predict outcomes that matter.

Medical devices and prostheses

Devices optimize physical performance together with biocompatibility, usability, sterilization, battery life, cybersecurity, alarm behavior, and service. A prosthesis or implant is part of a person's body and routine; comfort, identity, maintenance, and learning can dominate laboratory efficiency. Human factors validation must include realistic environments and foreseeable misuse.

Closed-loop devices—pacemakers, insulin delivery, neurostimulation—combine sensing, estimation, control, and failsafe behavior. Personalization can improve control but creates verification challenges. Safe fallback modes and bounded adaptation are design variables.

Bioinformatics and systems biology

Sequence alignment, assembly, phylogenetics, variant calling, protein structure, network inference, and single-cell analysis optimize likelihoods, scores, or posterior probabilities. The objective embeds a biological error model. For sequence alignment, gap penalties express hypotheses about insertion and deletion; for phylogeny, substitution models and priors shape the tree.

High-dimensional molecular data invite multiple testing and leakage. Family-wise error, false discovery rate, replication, batch correction, and preregistered analysis separate signal from selection. Mechanistic and statistical models can be composed: physics-informed learning constrains prediction with differential equations (Raissi, Perdikaris, and Karniadakis 2019).

Translational evidence

Performance often falls between bench, animal model, trial, and routine care because populations, endpoints, operators, and incentives change. Translation should be treated as a sequence of domain shifts with explicit go/no-go criteria. Negative results and uncertainty are reusable assets when captured.

Drug discovery searches chemical and biological spaces under potency, selectivity, toxicity, pharmacokinetics, manufacturability, novelty, and evidence. Surrogate models and generative systems can propose candidates, but prediction uncertainty and synthesizability constrain value. Active learning should choose experiments that improve decisions, not merely model accuracy.

Preclinical optimization can overfit assays and animal models. Translational validity requires mechanism, exposure, endpoints, population, and clinical context. Multiple testing and repeated candidate selection create optimism. Prospective validation, preregistration, independent replication, and transparent attrition are counterweights.

Bioinformatics optimizes alignments, assemblies, phylogenies, networks, and designs. The objective often encodes an evolutionary or measurement model. A best alignment under one scoring scheme is not an observed historical truth. Report ambiguity and alternative solutions when downstream conclusions depend on them.

Biomedical discovery is a sequence of filters: target identification, assay, hit selection, optimization, preclinical study, trials, review, manufacturing, and monitored use. Optimizing an early proxy can reduce downstream diversity and exploit assay error. Keep orthogonal assays, mechanistic checks, negative results, and replication.

Portfolio optimization should account for correlation among candidates, platform dependencies, stage-specific cost, and value of information. Killing a candidate can be a successful decision if it prevents later harm or releases capacity. Publication and incentive systems should not reward only positive progression.

Translation to populations requires external validation, causal evidence, benefit-harm assessment, manufacturability, access, and monitoring. Statistical normality is not health (Canguilhem 1991). Patient goals and capability remain part of the objective system.

Field checklist

- Treat biological fitness as contextual and historical, not as moral worth or global purpose.
- Separate prediction, utility, capacity, and action in clinical models.
- Calibrate probabilities in the intended population and decision-relevant range.
- Include abstention, referral, override, and safe fallback in sequential policies.
- Follow screening and allocation through the full care pathway and distribution of effects.
- Value preparedness options that nominal utilization metrics call idle.
- Require prospective and translational evidence after computational candidate selection.

Translation lens. Living and health systems transfer optimization structure through adaptation, conservation, thresholds, policies, and allocation. What must remain local is biological mechanism, patient preference, clinical evidence, consent, and the ethics of distribution.

Chapter 13 · Economics, operations research, public policy, law, and education

Markets, finance, and endogenous organizations

Economics studies allocation under scarcity, but its optimization models are descriptions and design instruments, not proofs that one social arrangement is best. Preferences may be incomplete, context-dependent, socially formed, or unequal in purchasing power. Institutions determine which choices appear in the model.

Consumers, firms, and equilibrium

A textbook consumer chooses bundle x to maximize utility under a budget:

$$\max_{x\geq 0} u(x) \quad \text{subject to} \quad p^T x \leq m. \tag{96}$$

A firm chooses inputs and outputs to maximize profit. Prices coordinate these decisions in competitive models. General equilibrium asks whether individually optimizing choices and market clearing can coexist. Welfare theorems require strong assumptions and do not select a distribution of endowments. Arrow's result shows that no social-welfare aggregation rule can satisfy a set of plausible conditions simultaneously when there are at least three alternatives (Arrow 1951).

Pareto efficiency means no one can be made better off without making someone worse off (Pareto 1896). It is silent about inequality: both egalitarian and extremely unequal allocations can be Pareto efficient. Distribution, rights, public goods, externalities, information, and market power require additional institutions.

Strategic behavior and mechanism design

Chapter 6 defines Nash equilibrium and its mixed-strategy existence guarantee. Here the institutional point is that equilibrium is a consistency condition under specified rules, not necessarily an efficient, fair, or dynamically reachable outcome (Nash 1950). Mechanism design works backward from desired properties to rules under strategic reporting. Incentive compatibility, efficiency, budget balance, participation, privacy, and simplicity often conflict (Myerson 1981). Stable matching demonstrates how institutional rules shape who can profitably deviate (Gale and Shapley 1962).

Behavioral and institutional realities

Expected-utility models are useful baselines, while prospect theory models reference dependence, loss aversion, and nonlinear probability weighting (Kahneman and Tversky 1979). Bounded rationality emphasizes limited information and computation; satisficing may be more realistic than global maximization (Simon 1955). Rules, norms, common-property arrangements, and trust can govern resources without a simple state-versus-market dichotomy (Ostrom 1990).

Finance and risk

Mean-variance portfolio theory chooses weights w by trading expected return $\mu^T w$ against variance $w^T \Sigma w$:

$$\min_{w} \ 1/2\, w^T \Sigma w - \lambda \mu^T w, \qquad \mathbf{1}^T w = 1. \tag{97}$$

It established diversification as an optimization problem (Markowitz 1952), but estimated returns are unstable and covariance changes during stress. Robust allocation, scenario analysis, liquidity limits, transaction costs, leverage, taxes, and tail risk matter. Conditional value at risk averages losses beyond a quantile and supports convex formulations (Rockafellar and Uryasev 2000).

Economic optimization models agents choosing under constraints, but preferences, endowments, information, institutions, and power are not exogenous in every application. Prices coordinate and ration; they also distribute access according to purchasing power. A market equilibrium is a consistency condition, not a complete social evaluation. Pareto efficiency excludes some avoidable waste but permits many distributions, including severely unequal ones (Pareto 1896; Arrow 1951; Sen 1970).

Finance makes estimation risk visible. Mean-variance optimization is highly sensitive to expected returns; the optimizer can amplify small errors into extreme portfolios (Markowitz 1952). Constraints, shrinkage, robust uncertainty, transaction costs, liquidity, and stress scenarios often improve out-of-sample decisions. The optimizer's curse is not a footnote: selecting the best estimated strategy preferentially selects estimation error (Smith and Winkler 2006).

Mechanism design recognizes endogeneity. A rule changes reporting and behavior. Auctions, matching, platform ranking, taxes, and service eligibility should be evaluated for incentives, participation, budget, privacy, simplicity, fairness, and strategic adaptation. A mechanism that is optimal for truthful agents can fail when truthfulness is not an equilibrium or when participants learn unintended workarounds.

Predictive targeting estimates who is likely to experience an outcome under historical policy. Allocation needs the effect of an intervention and its opportunity cost. A high-risk person may be hard to help; a moderate-risk person may benefit greatly. Ranking by risk is not ranking by treatment effect.

Policy also changes behavior. Schools teach to measures, firms change reporting, landlords adjust, and agencies ration informally. Campbell's and Goodhart's warnings concern this endogenous response (Campbell 1976; Goodhart 1975). Build mechanisms and enforcement into scenarios. Use pilots and qualitative observation to find adaptations before scaling.

When agents have private information, mechanism design can improve matching or allocation. Incentive compatibility is one value among efficiency, privacy, simplicity, access, and justice (Myerson 1981). A formally strategy-proof mechanism can remain inaccessible or politically illegitimate.

Market prices coordinate scarcity under institutions of ownership, income, information, and power. They are useful signals and incomplete measures of social value. A benefit-cost model that uses willingness to pay can weight the preferences of wealthier people more heavily. Report distribution and capability, and keep rights and public duties outside monetary substitution (Sen 1985; Rawls 1971).

Shadow prices from an allocation model are likewise conditional. They show marginal value under the objective, constraints, and data. They do not establish a fee, wage, compensation, or moral priority. Translating a multiplier into policy requires legal and institutional reasoning.

Use market mechanisms where participants can understand and act, externalities are governed, and concentration or strategic power is addressed. Otherwise the mechanism may optimize exchange within a distorted boundary.

Operations research as institutional engineering

Operations research turns recurring organizational decisions into models: what to produce, stock, route, schedule, staff, locate, or maintain. Its distinctive strength is connecting mathematical structure to operational evidence and implementation.

Flow, inventory, and service

Network-flow models represent supply, demand, capacity, and conservation. Facility-location models trade fixed opening costs against transport and service. Vehicle-routing models add sequence, capacity, time windows, driver rules, and uncertain travel. Inventory control balances holding, ordering, shortage, spoilage, and service; multi-echelon systems coordinate stock across a network.

A simple newsvendor chooses quantity q before random demand D:

$$\min_{q} c_o \, \mathbb{E}[(q-D)_+] + c_u \, \mathbb{E}[(D-q)_+]. \tag{98}$$

Under continuity, the optimal quantile satisfies

$$F_D(q^*) = \frac{c_u}{c_u + c_o}. \tag{99}$$

This compact result shows how costs translate into a service level—and how a wrong cost model creates the wrong stock policy.

Scheduling and projects

Scheduling allocates machines, people, rooms, vehicles, or computing resources over time. Objectives include makespan, lateness, throughput, fairness, setup, and robustness. Exact models may be MILP or constraint programs; dispatching rules and metaheuristics offer speed. A schedule should include recovery mechanisms because processing times, absences, and arrivals change.

Project optimization uses precedence networks, critical paths, resource constraints, and optional crashing. Optimizing the baseline plan is less important than managing uncertainty and feedback. Excessively tight utilization removes the capacity needed to resolve surprises.

From model to decision

An operations model has a data pipeline, ownership structure, and cadence. Master data errors can dominate solver gaps. Planners may hold tacit constraints absent from the model. Good deployment captures overrides and reasons, updates forecasts, and makes infeasibility informative.

Scenario optimization and simulation address uncertainty. Digital experiments can compare policies, but validation must cover arrival patterns, service times, correlations, behavioral responses, and rare disruptions. A solution should be stress-tested for parameter sensitivity, not merely replayed on average history.

Organizational incentives

Key performance indicators allocate attention. If a purchasing team minimizes unit price, it may increase defects, lead time, and working capital. If a call center minimizes handling time, it may increase repeat calls. End-to-end metrics and balanced constraints help, but governance must prevent measurement from becoming an adversarial game.

Operations research connects mathematical models with recurring organizational choices: flow, inventory, routing, scheduling, staffing, maintenance, and projects. Its practical strength is not only algorithms but disciplined problem structuring, data analysis, implementation, and feedback.

Inventory illustrates boundary choice. The classical trade-off balances holding cost against ordering or shortage cost. In practice, stock also carries resilience, obsolescence, working capital, service differentiation, supplier risk, and option value. A just-in-time policy can be efficient under ordinary variation and fragile under common-cause disruption. Segment items by consequence and lead-time uncertainty rather than applying one turnover target.

Scheduling illustrates hidden human constraints. A mathematically feasible roster may violate preferences, recovery, skill development, predictability, or informal coordination. Some constraints are legal or safety-critical; others should be elicited and prioritized. Publishing a schedule changes swapping and absence behavior. Pilot implementation and operator review are part of the solution.

Project optimization similarly requires managing uncertainty. A deterministic critical path can create false precision; correlated delays, rework, resource contention, and learning matter. Buffers should be located according to risk and coordination, not simply removed as waste. A late change can alter scope and the precedence graph itself.

Policy models often assume implementation capacity. Staff, procurement, data, local partners, legal authority, communication, maintenance, and trust determine whether an intervention exists in practice. Include capacity constraints and investments that expand them.

A portfolio that funds programs without administration can be infeasible. A program that relies on local discretion can need training and audit. A policy with a rapid benefit can be dominated by a slower one if the first cannot be delivered equitably.

Capacity is dynamic. Emergency response can consume routine service; repeated reforms can create change fatigue; outsourcing can erode public knowledge. Preserve slack and learning. Measure delivery quality and staff burden, not only funds allocated.

Policy, law, delivery, and accountability

Social optimization intervenes in systems whose participants interpret, resist, learn from, and reshape the intervention. Unlike inert design variables, people have rights and agency. Predictions and rankings can become causes.

Causal goals, not predictive shortcuts

Policy asks what would happen under an action. In potential-outcome notation, the average treatment effect is

$$\mathrm{ATE} = \mathbb{E}[Y(1) - Y(0)]. \tag{100}$$

Only one potential outcome is observed for each unit. Randomization, natural experiments, adjustment, and structural models identify effects under different assumptions. A model that accurately predicts unemployment does not necessarily reveal which intervention reduces it.

Optimization on observational data can exploit confounding. Policy should state the estimand, intervention, population, time horizon, spillovers, and identification assumptions. Distributional effects and qualitative evidence matter alongside averages.

Policy design and administration

Policy portfolios combine taxes, subsidies, standards, information, public provision, and institutional reform. Compliance cost, enforcement, administrative capacity, legitimacy, and political durability affect results. Adaptive policy uses monitoring and predetermined review points without surrendering long-term commitments.

Rankings change organizations. When professional schools and universities are compared on a public ranking, they redirect effort, manage reported categories, and alter admissions and organizational practice (Espeland and Sauder 2007). Campbell’s and Goodhart’s laws warn that a measure used as a target becomes vulnerable to corruption (Campbell 1976; Goodhart 1975). Auditing must look for displaced work and excluded populations.

Law as constraint and objective

Law defines rights, duties, procedures, and remedies; it is not reducible to aggregate efficiency. Proportionality, due process, equality, privacy, and reason-giving constrain administrative optimization. Automated systems used in benefits, policing, hiring, credit, or migration must support notice, contestation, records, and accountable authority.

The European Union's AI Act takes a risk-based lifecycle approach to regulated AI uses (European Parliament and Council of the European Union 2024). Compliance is not identical to ethical adequacy, but legal requirements are hard constraints. A policy cannot purchase a right violation with sufficient predicted benefit.

Education

Education has plural aims: knowledge, skill, curiosity, citizenship, belonging, creativity, and opportunity. Test scores sample some outcomes and can crowd out others. Personalized learning may adapt sequence and difficulty, but must avoid narrowing the curriculum to what is easily measured.

School assignment and admissions combine capacity, preferences, proximity, diversity, siblings, access, and strategic behavior. Stable matching can improve process properties, while the rule still expresses contested policy choices. Teachers and learners need intelligible recommendations and the ability to challenge data.

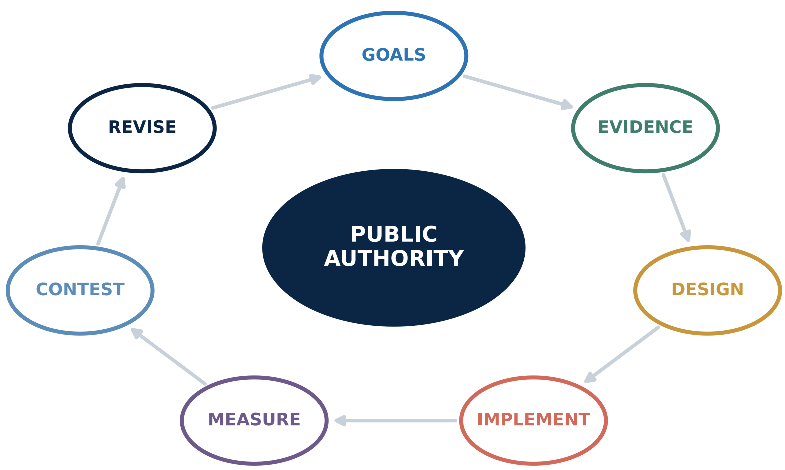


Figure 21. A governance loop linking public goals, causal evidence, policy design, implementation, measurement, contestation, and revision.

Figure 21 interprets governance as learning control: public goals and causal evidence shape policy, implementation changes measurement, and contestation can trigger revision.

Public policy chooses interventions, rules, eligibility, enforcement, and resources. Prediction identifies who is associated with an outcome; policy needs a causal estimand: what outcome would change under an intervention for a defined population and implementation. Historical data reflect previous policies and access. Optimizing on correlations can allocate help away from people whose outcomes appear good precisely because they already received support.

Law shapes both constraints and authority. Rights, due process, equal treatment, proportionality, jurisdiction, and reasons cannot automatically be converted to costs. A legal threshold may be categorical even when evidence is probabilistic. A model can support consistent application while remaining subordinate to interpretation and appeal.

Procedural optimization matters. Notice, explanation, participation, hearing, and review consume time, but that time is not mere inefficiency. It produces legitimacy, error correction, and information. A model that minimizes processing time by suppressing contestation optimizes away a safeguard.

A policy is not a parameter applied directly to society. It travels through legislation, budget, guidance, procurement, organizations, frontline discretion, communication, take-up, and enforcement. Each link can transform the intervention. A model of the statutory rule without delivery is incomplete.

For a proposed benefit, regulation, placement system, or public service, map the chain and assign evidence. Eligibility rules require administrative data and legal review. Take-up requires access, trust, time, language, and documentation. Provider response requires capacity and incentives. Outcomes require causal and distributional analysis. Appeal and correction require operations of their own.

Optimization can allocate budgets or choose parameters after this chain is modeled. It should include administrative burden, waiting, non-take-up, error, discretion, and enforcement—not merely program spending. A low-cost policy that shifts paperwork and uncertainty to vulnerable people is not efficient from the social boundary.

Law changes the type signature. Eligibility, nondiscrimination, notice, reason-giving, hearing, privacy, proportionality, and judicial or administrative review can be requirements. They should appear as constraints, processes, and records, not an ethics coefficient.

Contestability needs capacity: people must know a decision occurred, understand enough to challenge it, obtain relevant data, reach a human with authority, and receive timely remedy. An explanation without correction is not due process. Measure appeal accessibility, resolution, reversal, delay, and repeated error.

Automated systems can standardize and expose inconsistency; they can also turn a defeasible rule into rigid scale. Preserve discretion where law or justice requires context, then monitor its distribution. Removing all discretion can hide policy choices in the model rather than eliminate judgment.

Rankings compress multi-dimensional need or merit into order. They can allocate review efficiently and create false discontinuities around cutoffs. Publish criteria and uncertainty, inspect near-ties, and avoid implying that rank is intrinsic worth.

Triage should include urgency, potential benefit, alternatives, waiting harm, and equity under a clinically or institutionally valid policy. It must support reassessment as state changes. A queue position is not permanent identity.

People need notice and correction where ranking affects material opportunity. Audit missing data, representation, subgroup error, and strategic behavior. When capacity shortfall drives harm, do not let the ranking model absorb responsibility for the shortage.

Education and measurement response

Education involves learners, teachers, families, institutions, and long horizons. Objectives include knowledge, capability, curiosity, inclusion, well-being, citizenship, and opportunity. Test scores measure selected outcomes and can support diagnosis; when made high-stakes targets, they change teaching, enrollment, classification, and exclusion (Campbell 1976).

Adaptive learning systems can sequence material using estimated mastery, but the learner is not a stationary state vector. Motivation, identity, social interaction, fatigue, and misconception matter. A short-term answer-rate objective can prefer easier content and reduce durable learning. Evaluation should include transfer, retention, subgroup experience, teacher workload, and the possibility of independent learning.

Resource allocation should distinguish need from predicted return. Maximizing aggregate measured gain can favor those easiest to improve and abandon students requiring more support. Equity may require floors, priority weights, or rights to provision rather than a compensatory objective.

Education supports knowledge, skill, agency, belonging, creativity, citizenship, and future opportunity. Test scores and completion are partial indicators. If tied tightly to accountability, they change curriculum, selection, and reporting. Evaluate both measured attainment and the conditions that produce it.

An allocation model should include service capacity, student and family burden, peer and neighborhood effects, accessibility, continuity, and uncertainty. Randomized or quasi-experimental evidence can estimate some interventions; implementation and local meaning affect transportability. Avoid ranking students or schools as if predicted outcome were intrinsic worth.

Use optimization to construct feasible resource alternatives and expose trade-offs, then retain public authority over educational purpose. A school is not a production unit whose only output is a scalar gain.

Worked policy formulation: heat-health response

A city heat-health program chooses cooling-center locations, opening hours, transport, outreach, home visits, tree and shade investment, energy assistance, and emergency staffing. The objective system includes mortality and morbidity, access, cost, emissions, trust, and long-term resilience. Constraints include staff, buildings, electricity, transport, disability access, data protection, and legal duties.

Exposure and vulnerability are uncertain and unequally measured. Mobile-phone data can estimate movement while excluding or surveilling vulnerable people. Historical emergency calls may underrepresent those unable to seek help. Participatory mapping and local organizations provide evidence that remote sensing cannot.

The operational model can optimize daily activation under forecasts; the planning model can optimize permanent infrastructure; the governance process determines priorities and data use. Monitoring triggers should include indoor temperature, service uptake, response time, neighborhood coverage, and adverse events. A post-event review updates both parameters and the intervention portfolio.

Field checklist

- Treat prices, preferences, rules, and measured behavior as potentially endogenous.
- Include estimation uncertainty and selection bias in financial or policy optimization.
- Evaluate mechanisms under strategic response, participation, and distribution.
- Elicit human scheduling, workload, predictability, and recovery constraints.
- State the causal estimand and implementation pathway for every policy claim.
- Preserve due process, explanation, and appeal as safeguards, not default delay penalties.
- In education, distinguish measured gain from capability, inclusion, and durable learning.

Translation lens. Economic, organizational, and policy systems translate optimization through allocation, flow, incentives, and causal intervention. Their decisive local types are institutions, rights, endogeneity, and procedural legitimacy.

Chapter 14 · Psychology, ethics, sustainability, philosophy, history, and language

Human factors and function allocation

Human performance is adaptive, bounded, and situated. People distribute attention, switch strategies, learn, fatigue, and coordinate. A design optimized for an abstract “average user” can fail precisely when workload and consequence are highest.

Bounded cognition

Humans often use heuristics because exhaustive search is costly and the world is uncertain. Some heuristics are biases in one environment and efficient adaptations in another (Gigerenzer and Selten 2002). Choice architecture influences decisions through defaults, ordering, framing, and friction. Ethical design makes important consequences visible and preserves meaningful choice.

A simple speed-accuracy trade-off treats decision threshold a as a variable: higher a gathers more evidence and usually improves accuracy while increasing response time. In safety-critical work, the optimal threshold changes with urgency and loss. Training should develop flexible calibration rather than one fixed response style.

Human-machine systems

Automation changes work instead of merely removing it. It can reduce routine load while leaving rare, ambiguous, high-stakes exceptions. Poor function allocation creates distinct failure mechanisms. Mode confusion concerns hidden or misunderstood automation states (Sarter and Woods 1995); out-of-the-loop performance combines loss of situation awareness and skill under passive control (Endsley and Kiris 1995); overreliance and alarm-driven disuse require separate analysis of automation use (Parasuraman and Riley 1997). Displays should support situation awareness—perception of relevant state, comprehension, and projection—without treating that model as evidence for every automation failure (Endsley 1995).

Human reliability analysis estimates error under task, environment, and organizational conditions. “Human error” should trigger analysis of interface, procedure, staffing, incentives, fatigue, and recovery opportunities. In dynamic sociotechnical systems, safe performance can depend on adaptation across operating, managerial, and regulatory levels; that claim belongs to Rasmussen’s later migration-to-the-boundary analysis (Rasmussen 1997). The skill–rule–knowledge taxonomy remains useful for analyzing cognitive control modes (Rasmussen 1983).

Ergonomics and inclusive design

Physical ergonomics optimizes posture, force, repetition, reach, vibration, lighting, and environment. Cognitive ergonomics addresses memory, attention, mental models, alarms, and coordination. Organizational ergonomics addresses schedules, teamwork, and culture. Improvements at one level can move strain elsewhere.

Inclusive design treats variation in bodies, senses, language, cognition, and technology as a design input. Accessibility is not a niche objective; captions, contrast, keyboard control, clear language, and adjustable pacing often help broad populations. Optimization should consider the tails of capability and context.

Interfaces and interaction cost

Keystroke-level and pointing models estimate routine task time (Card, Moran, and Newell 1983), but experienced time also includes uncertainty, interruption, trust, and recovery. A one-click action may be fast and dangerous; a well-placed confirmation can be valuable friction. Usability testing observes real strategies rather than assuming designers' workflows (Norman 2013).

Human factors optimization studies a joint cognitive, social, and technical system. A design that assumes perfect attention, memory, reaction, or compliance is not human-centered; it is misspecified. People adapt, develop expertise, create workarounds, coordinate informally, and reinterpret goals. These capacities can stabilize a brittle system and can also conceal its risk until conditions change.

Bounded rationality treats search and information costs as constitutive (Simon 1955; Gigerenzer and Selten 2002). Heuristics can be ecologically rational when they exploit regularities of an environment. A more complex model is not necessarily a better decision aid if it demands unavailable attention or produces unactionable precision.

Interface optimization should include time to detect, understand, decide, act, and recover. The keystroke and pointing account above supports motor details; situation-awareness and control-mode analyses address larger dynamics (Endsley 1995; Rasmussen 1983). Accessibility changes the feasible set: color, motion, hearing, language, dexterity, cognition, and assistive technology must be tested, not treated as an average-user penalty.

The question is not simply which agent performs a task faster. Automation changes the remaining human task. If routine cases disappear, the human sees a harder, rarer distribution while losing practice. If the system asks for approval after making a recommendation, anchoring and workload can make review nominal. If it acts autonomously until confidence falls, the handover may occur exactly when context is most confusing.

A function-allocation design should specify authority, information, timing, skill maintenance, override, responsibility, and fallback. It should simulate degradation and disagreement. Operators should be able to inspect why a constraint is binding, which data are uncertain, and what alternatives exist. Explanations should support action, not merely satisfy a generic transparency metric.

Participatory design enlarges the model. Users and affected groups reveal hidden work, values, and failure modes. Their contribution is not reducible to preference weights collected after architecture is fixed. Participation can change variables, boundary, ownership, and whether the system is built.

People decide under limited time, attention, memory, computation, and information (Simon 1955; Payne, Bettman, and Johnson 1993). A system should not compare human action with an omniscient optimum and label the difference error. It should design information, defaults, timing, and choice sets that support real reasoning without manipulation.

Measure cognitive work at the point of use. How many signals must be integrated, how often, under what interruption and consequence? A dashboard that exposes every variable can reduce understanding. Progressive disclosure, stable terminology, external memory, and rehearsal can improve control. Automation should reduce low-value load while preserving situation awareness and skill for degraded modes (Endsley 1995).

Heuristics can be ecologically rational when they exploit environmental structure (Gigerenzer and Selten 2002). Compare them with complex models under realistic data and cost. A transparent rule can be more robust, teachable, and contestable.

Ethics, fairness, and moral uncertainty

Ethics asks which objectives and constraints are worthy, who has standing, what procedures are legitimate, and which values must not be traded away. It surrounds optimization; it is not one more term with a tunable weight.

Consequences, duties, virtues, and capabilities

Consequentialist reasoning compares outcomes; deontological reasoning emphasizes duties and persons; virtue ethics emphasizes character and practical judgment; capability approaches ask what people are substantively able to do and be (Mill 1863; Kant 1785; Aristotle 2002; Nussbaum 2011). These perspectives reveal different failure modes. A policy with high aggregate benefit may violate a duty; a formally equal rule may leave capabilities unequal.

Rawlsian reasoning considers basic liberties, fair opportunity, and the position of the least advantaged (Rawls 1971). A maximin objective,

$$\max_{x} \min_{i} u_i(x), \tag{101}$$

captures one protective intuition but not the full theory. Ethical frameworks should guide questions and institutions rather than be caricatured as formulas.

Fairness in algorithms

Statistical fairness criteria include demographic parity, equalized odds, equal opportunity, calibration, and individual or causal notions. They condition on different variables and express different social views. When base rates differ, calibration and equalized error rates can be mutually incompatible except in special cases (Chouldechova 2017; Hardt, Price, and Srebro 2016). No metric resolves whether labels, groups, interventions, or institutions are just.

Abstraction can hide the social system in which a model acts (Selbst et al. 2019). A lending model may equalize error while the credit product remains harmful. A hiring model may remove a protected attribute while proxies and past exclusion remain. Fairness work therefore includes problem formulation, data history, stakeholder participation, recourse, monitoring, and institutional alternatives (Barocas, Hardt, and Narayanan 2023).

Procedural safeguards

Responsible optimization documents purpose, affected groups, evidence, uncertainty, alternatives, constraints, residual risk, ownership, appeal, and retirement. Independent review is important when builders benefit from deployment. High-impact decisions need understandable reasons and a route for correction.

Sustainability

Sustainability joins environmental boundaries, social foundations, and intergenerational justice (Raworth 2017). Efficiency can induce rebound: cheaper resource use increases total demand. Absolute budgets and sufficiency may be needed alongside per-unit improvement. Lifecycle accounting prevents emissions or labor harm from being shifted upstream.

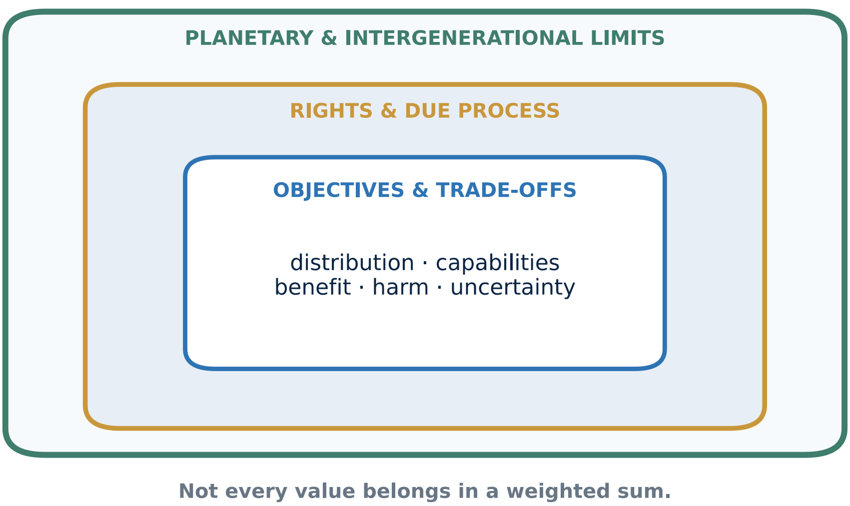


Figure 22. A responsible-optimization frame with rights as boundaries, stakeholder capabilities, distributional outcomes, and planetary constraints.

Figure 22 places rights and planetary limits outside ordinary compensation, while stakeholder capabilities and distributional outcomes remain visible inside the admissible region.

Ethical analysis supplies several non-equivalent structures. Consequential reasoning compares outcomes; duties and rights restrict admissible actions; virtue ethics asks about character and practice; capability approaches ask what people are substantively able to do; care ethics emphasizes relationship and dependency. A single weighted sum cannot faithfully represent all of them.

Fairness metrics illustrate incompatibility. Calibration, error-rate parity, demographic parity, individual similarity, and equal utility answer different questions and can conflict under unequal base rates (Hardt, Price, and Srebro 2016; Chouldechova 2017; Barocas, Hardt, and Narayanan 2023). Choosing a metric requires a theory of harm, a decision context, and institutional accountability. Fairness is not obtained by selecting the metric with the smallest disparity after searching many definitions.

Abstraction creates traps when social categories and institutions are treated as fixed technical inputs (Selbst et al. 2019). A loan, risk score, hiring funnel, or benefits rule operates in a history of access and power. Expanding the boundary can reveal interventions upstream of prediction: changing documentation, offering assistance, redesigning eligibility, or removing the decision altogether.

Statistical fairness criteria encode different conditional independences and commonly conflict when base rates differ (Chouldechova 2017; Hardt, Price, and Srebro 2016). Selecting one requires a theory of the decision, harm, remedy, and institutional duty. Equal error rates, calibration, equal selection, and individual similarity answer different questions.

Define who is affected, which outcome and error matter, whether the model allocates opportunity or burden, and what alternative exists. Examine label quality, selection, measurement, exposure, and downstream response. Aggregate parity can hide intersectional or local harm; very small groups create uncertain estimates.

Fairness is not completed by metric adjustment. Access, notice, explanation, appeal, representation in design, and the distribution of model benefit matter. Some uses remain unacceptable even if parity is achieved (Selbst et al. 2019; Barocas, Hardt, and Narayanan 2023).

Participation is not a survey used to fit weights. Affected people should help define boundary, alternatives, harms, evidence, and governance. Different modes—interviews, workshops, community research, deliberative panels, accessibility testing, and representative institutions—support different claims.

Document who participated, who could not, compensation, language and access, information supplied, disagreement, and how input changed the model. Do not claim community consent from a small advisory process. Some groups have legal or sovereign authority beyond stakeholder preference.

Model outputs can support participation by making constraints and trade-offs visible. Use understandable scenarios and concrete alternatives. Preserve minority reports and allow participants to challenge the categories themselves.

Parameter uncertainty concerns facts inside a model. Moral uncertainty concerns which principles or reasons should govern. Assigning a probability distribution to ethical theories can be one analytic exercise and does not settle legitimacy.

Use robust ethical reasoning where possible: identify actions unacceptable across several relevant frameworks, duties that remain regardless of aggregate benefit, and alternatives that preserve capabilities and reversibility. Where principles conflict, expose the conflict to accountable deliberation.

Ethics review should trace who benefits, who bears risk, whose agency changes, what rights apply, and whether repair is possible. It should examine process as well as outcome. A fair distribution produced through coercion or deception is not therefore justified.

Sustainability across generations

Sustainability introduces long horizons, irreversibility, thresholds, distribution, and deep uncertainty. Discounting converts future effects to present values, but the rate combines empirical opportunity cost with ethical judgment. A single rate can conceal intergenerational disagreement. Report results across rates, physical thresholds, and noncompensatory constraints.

Planetary boundaries and safety limits should not be treated as ordinary weighted objectives when crossing them creates unacceptable or poorly reversible harm (Rockström et al. 2009; Intergovernmental Panel on Climate Change 2022b). Robustness, precaution, adaptive pathways, and option preservation may dominate expected-value efficiency. Lifecycle analysis should include rebound, leakage, embodied effects, and distribution across places and workers.

Discounting translates future consequences into present comparison. The rate combines time preference, growth, uncertainty, and ethical judgment; it is not a neutral market input for public harm. Report results across rates and include physical thresholds and distribution (Intergovernmental Panel on Climate Change 2022b).

Long-horizon decisions should preserve options and avoid irreversible boundary crossings. Expected net benefit can conceal catastrophic tail, ecological loss, and unequal exposure. Use constraints for safety and planetary limits, adaptive pathways for uncertainty, and scenarios for institutional capacity (Rockström et al. 2009; Raworth 2017).

Future people cannot participate directly. Institutions can represent them through durable standards, independent review, conservation duties, and option preservation. The representation is imperfect and should be visible.

Humanities, interpretation, and historical path dependence

The humanities study meaning, value, interpretation, memory, and form. Optimization can illuminate patterns, but it becomes reductive when it treats ambiguity as noise or assumes that cultural significance is a score waiting to be maximized.

Philosophy of optimization

Aristotle distinguished productive making, theoretical understanding, and practical judgment. Modern optimization is strongest in production under stable ends; practical reason is needed when ends are contested. Means can also transform ends: a metric, platform, or institution changes what people notice and desire.

The concept of the normal has both descriptive and normative force. Medicine and society can mistake statistical typicality for desirable life (Canguilhem 1991). Technologies are not neutral containers; their arrangements distribute authority and possibility (Winner 1980). Philosophy therefore asks who authored the objective, what ontology it presumes, and which counterfactuals it excludes.

Historical explanation

Historical actors optimize under beliefs, institutions, technologies, and moral worlds unlike ours. Retrospective claims that an outcome was “inevitable” confuse realized paths with feasible alternatives. Path dependence, contingency, archival silence, and changing categories limit simple objective functions.

Quantitative history can analyze prices, networks, texts, demography, and spatial patterns. Its models should be triangulated with sources and historiography. An optimized fit to surviving records may reproduce preservation bias. The absence of a surviving document is not by itself evidence that an event or practice did not occur.

Language and translation

Language technologies often maximize likelihood:

$$\max_{\theta} \sum_{t=1}^{T} \log p_\theta(w_t \mid w_{<t}), \tag{102}$$

but probable continuation is not the same as truth, relevance, or literary value. Translation trades semantic content, tone, rhythm, cultural reference, and audience. No single alignment score captures all of them.

Lexicography, corpus linguistics, authorship analysis, and distant reading reveal large-scale patterns (Moretti 2005). Close reading detects ambiguity, irony, voice, and form. The methods are complementary at different scales.

Interpretation as constrained search

An interpretation must attend to the work, context, evidence, and alternative readings. It is neither arbitrary nor generally unique. Umberto Eco distinguishes openness from unlimited interpretive license (Eco 1990). One can model interpretation loosely as abductive search for an explanation with coherence and evidential fit, while preserving the fact that new contexts generate new questions.

Category-theoretic caution

Category theory invites a useful metaphor: each discipline has objects, morphisms, and composition laws. Translation between disciplines resembles a functor only when it preserves relevant structure. A mapping that sends “fitness,” “utility,” “beauty,” and “loss” to one scalar may fail to preserve meaning. Sometimes the correct result is not a universal optimizer but a carefully bounded correspondence.

Humanistic inquiry selects archives, categories, narratives, and arguments, but its aim is not always an extremum. Interpretation is a constrained search among readings supported by evidence, language, genre, context, and scholarly dialogue. Multiple readings can remain defensible; disagreement can be productive rather than a failure to converge.

Quantitative text analysis optimizes models over corpora, yet corpus construction and feature choice determine the world made visible. A likelihood objective rewards predictive regularity, not historical truth or literary significance. Distant reading can reveal large-scale patterns (Moretti 2005), while close reading examines ambiguity, voice, form, and exception. The methods are complementary when their preservation claims are explicit.

Historical explanation should resist hindsight optimization. Outcomes do not prove that actors chose the globally best path or that alternatives were infeasible. Archives are selected and incomplete; categories change; institutions have path dependence. Models can test mechanisms and counterfactual coherence without reducing history to a solved control problem.

Language translation offers a categorical caution. A mapping can preserve reference while changing connotation, rhythm, register, or social force. Two translations may commute on factual propositions and diverge on style. Calling the mapping a functor is useful only after the relevant objects, morphisms, and invariants are specified. Otherwise category theory becomes a complicated way of saying “things are related.”

Humanistic models choose corpus, unit, category, similarity, and outcome. Each choice creates a reading. A topic model can reveal distributional patterns and cannot establish what a theme means. A network can show recorded relations and cannot recover unrecorded influence. Quantification should return to close reading and historical context (Moretti 2005).

Interpretation often supports multiple defensible accounts. Optimization can help search evidence, test consistency, or compare explanatory coverage. It should not force convergence when ambiguity or disagreement is constitutive. Preserve excerpts, provenance, excluded material, and counterreadings.

Category theory is useful when it states what a translation preserves: chronological order, citation relation, syntactic structure, or narrative function. Calling translation a functor without these objects and morphisms adds prestige rather than understanding. The test is whether a commutative diagram yields a new checkable question.

Present feasible sets are produced by past investment, exclusion, standards, and institutions. Optimizing within them can reproduce injustice while appearing neutral. Include alternatives that repair or expand capacity, not only allocate current scarcity.

Historical data encodes previous policies. A model can accurately predict outcomes generated by unequal access and then justify the same allocation. Causal and historical inquiry asks how the pattern was made and which intervention can change it.

Path dependence also creates option value and lock-in. Standards, infrastructure, and categories become difficult to reverse. Long-horizon optimization should price transition, preserve interoperability, and ask whose knowledge or livelihood is displaced.

Field checklist

- Model human attention, expertise, adaptation, workload, and coordination as system variables.
- Test automation handover, disagreement, skill retention, and degraded modes.
- Include accessibility in feasibility and involve affected people before the objective is fixed.
- Distinguish consequences, rights, duties, capabilities, care, and procedural requirements.
- Justify fairness metrics through a harm model and institutional context.
- Test long-horizon decisions across discount rates, thresholds, and adaptive pathways.
- In humanistic work, preserve ambiguity and make corpus, category, and interpretation choices inspectable.

Translation lens. Human and interpretive disciplines transfer the ideas of constrained search, evidence, and revision while refusing premature total order. The crucial invariant is meaningful agency: people remain participants in defining, interpreting, contesting, and changing the optimization.

Part V · Design, Art, Music, Literature, and Play

Chapter 15 · Architecture, design, visual art, music, and performance

Creative practice: form, criteria, critique, and accessibility

Creative work is purposeful without always having a fixed objective. Artists and designers discover criteria through making. Optimization is valuable when it expands exploration, resolves technical constraints, or clarifies trade-offs; it is harmful when a proxy for preference becomes the authority on meaning.

Design as reflection and search

Herbert Simon described design as changing existing situations into preferred ones (Simon 1996). Yet “preferred” evolves as sketches, prototypes, users, sites, and materials answer back. Reflective practice alternates framing and action rather than executing a complete specification (Schön 1983).

A design state can be discussed through a vector of concerns—function, cost, carbon, delight, access, and risk—but several components resist reliable quantification. The vector is a conversation surface. Pareto alternatives let a team see what must be sacrificed and where an overlooked concept changes the frontier.

Architecture

Architecture coordinates site, program, structure, envelope, energy, light, acoustics, circulation, code, cost, construction, and civic presence. Parametric design connects geometry to rules; performance simulation evaluates daylight, heat, airflow, views, structure, and egress; generative design searches variants (Woodbury 2010).

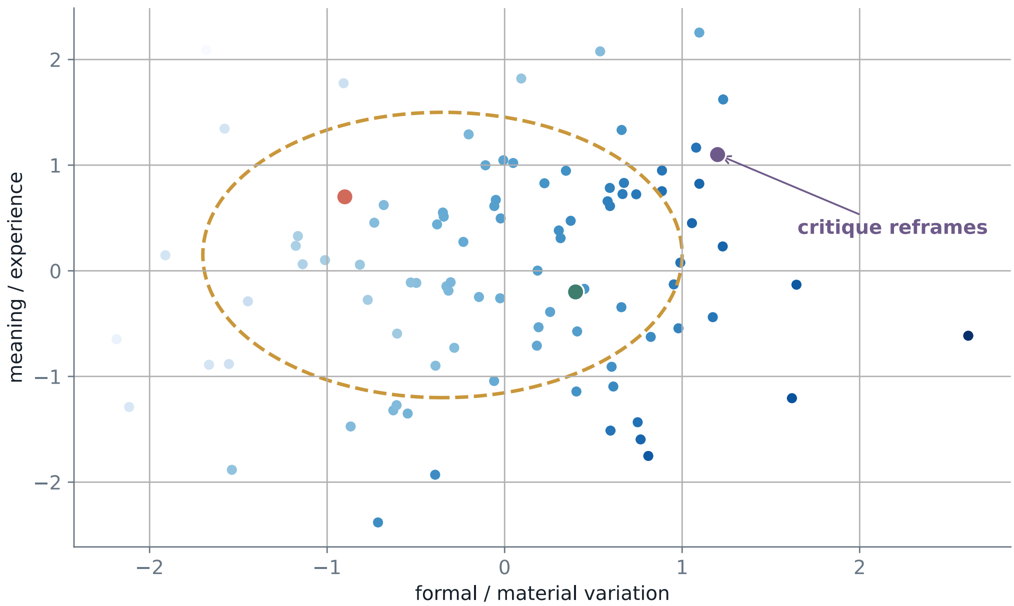


Figure 23. A creative design space in which constraints shape a field of possibilities and critique redirects the search.

Figure 23 depicts creative search as landscape-making: constraints define possibilities, and critique redirects both the work and the criteria rather than merely scoring finished alternatives.

The objective must remain human-scale. Maximizing rentable area can reduce daylight, adaptability, or public realm. Minimizing operational energy can increase embodied carbon. Shortest circulation can erase encounter and ceremony. Building codes are necessary constraints, while culture and place require interpretation.

Christopher Alexander's early work treated design as decomposing interacting misfits (Alexander 1964). Decomposition helps computation but interfaces matter: structure shapes space, envelope shapes comfort, and circulation shapes social relation. Integrated design sessions restore those couplings.

Visual art and computational creativity

Constraint has long structured art: perspective, proportion, palette, serial procedures, chance operations, and material limits. Evolutionary art and generative systems search image spaces through mutation, rules, or learned models (Sims 1991; Elgammal et al. 2017). Aesthetic measures can filter or provoke, but they reflect training data and taste communities.

Computational creativity may be evaluated by novelty, value, surprise, intentionality, and process (Boden 2004; Wiggins 2006). These criteria are relational. A novel image may be trivial; a derivative form may be culturally transformative. Authorship, provenance, consent, labor, and environmental cost belong in the system boundary.

Critique as an optimizer

Studio critique samples how a work is perceived, identifies latent constraints, and proposes transformations. Its aim is not to average opinions. The maker decides which feedback reveals the work's internal necessity and which belongs to another project. This resembles interactive multiobjective optimization with a preference model that is being learned and revised.

The creative disciplines expose a fact that technical accounts often hide: an objective can be produced by the search rather than supplied before it. An architect discovers that a circulation problem is really a question about encounter. A painter discovers that a technically unwanted edge is the work's center. A composer hears that the passage being optimized for smoothness needs resistance. These are not failures to specify the problem. They are changes in the problem's ontology, caused by contact with sketches, materials, performers, sites, and audiences.

That does not make formal methods irrelevant. It changes their role. In creative work, an optimization system is most useful as a generator of contrast, a keeper of hard constraints, a detector of hidden coupling, or a way to make trade-offs discussable. It is least trustworthy when a provisional measure of taste is treated as a final authority. Chapter 6's Pareto frontier is therefore more appropriate than a single score: it displays loss without claiming that the weights have already been settled. Chapter 3's governance loop is equally important, because the maker must be able to revise the criteria after seeing what the criteria produce.

Three loops, not one

A mature creative process contains at least three coupled loops.

1. The **artifact loop** changes a drawing, building, performance, text, image, or sound.
2. The **evaluation loop** changes how alternatives are compared: a criterion is added, dropped, reinterpreted, or moved from objective to constraint.
3. The **framing loop** changes what kind of project is being attempted, who it is for, and what evidence can count.

An ordinary optimizer acts within the first loop. Interactive multiobjective methods can partly support the second. The third remains a matter of authorship, dialogue, institutional authority, and ethical responsibility. Treating all three as parameter tuning hides the most important transition: sometimes a project changes type. A proposal for a faster transit interchange becomes a proposal for a civic room; a prosthetic controller becomes an instrument; a visualization becomes an argument about what is absent from the data.

The categorical language is precise here. Within a fixed framing, transformations between design states can be represented as morphisms. Reframing changes the category: the objects, admissible transformations, and invariants are no longer the same. There may be a translation between the old and new descriptions, but it will forget information. Chapter 7's warning that argmin is not generally functorial becomes practical: the best component under one framing need not remain best after composition, and a minimizer need not survive a change in what counts as equivalent.

Architecture: objectives with different legal status

Consider an urban infill library. The team can simulate daylight, annual energy, structural demand, egress time, acoustic separation, embodied carbon, construction cost, and spatial visibility. These quantities should not all enter a weighted sum. Code compliance and structural safety are feasibility conditions. An accessibility commitment is a right and process requirement, not a benefit that high rentable area can purchase away. Carbon may be governed by a hard project budget and then improved within it. Cost is both a limit and a value requiring lifecycle interpretation. Civic presence, belonging, and the dignity of ordinary use require situated judgment.

The formulation can therefore be layered:

- first exclude illegal, unsafe, inaccessible, or materially infeasible schemes;
- then identify nondominated alternatives on operational energy, embodied carbon, cost, adaptability, and selected measures of comfort;
- then review those alternatives through spatial experience, community use, cultural context, maintenance, and future conversion;
- finally record why one alternative was chosen and what evidence would justify revision.

This hierarchy is lexicographic in part and deliberative in part. It is not reducible to a single scalar without losing the distinction among rights, limits, preferences, and meanings introduced in Chapter 1.

A parametric model helps when its parameters correspond to controllable and buildable decisions. It misleads when it produces geometries whose construction logic, tolerances, maintenance access, or procurement path are absent from the representation. The simulation boundary must include the assemblies and interfaces through which the building will actually exist. Chapter 10’s co-design lesson applies directly: the envelope cannot be optimized independently of structure, environmental systems, controls, and occupant behavior.

Critique as preference learning—and more

Interactive optimization often asks a user to compare alternatives, then infers a latent utility function. Studio critique can look similar: proposals are shown, responses gathered, and the next move chosen. The analogy transfers three useful structures.

- **Sequential information.** Each artifact is both a candidate and an experiment that reveals a preference or problem.
- **Active sampling.** A deliberately extreme proposal can be valuable because it clarifies a boundary, not because it is likely to be selected.
- **Model revision.** Contradictory feedback can reveal missing criteria, context dependence, or multiple audiences.

The analogy also loses essential information. A critic does not merely report a fixed preference. Critique teaches a vocabulary, contests premises, interprets historical references, and changes the maker’s attention. The response may concern the process or ethical position rather than the artifact. Power matters: the same sentence from a teacher, patron, community member, or algorithm does not have the same force. A computational preference model can summarize comparisons; it cannot inherit the critic's responsibility or the maker's authorship.

For this reason, disagreement should be preserved as structured data. Record who evaluated an alternative, in what context, against which concern, and with what confidence. Do not average "more accessible," "historically false," and "personally uninteresting" into 6.7 out of 10. The categories differ, and their consequences differ.

Visual art: search space, provenance, and refusal

Generative systems can vary composition, palette, texture, camera, iconography, or style. Evolutionary art and learned generators enlarge the number of reachable candidates (Sims 1991; Elgammal et al. 2017). The bottleneck then moves from production to selection and interpretation. If the ranking model is trained on past popularity, it will reproduce the visibility structure of the archive. If novelty is computed as distance in a learned embedding, it measures distance under that representation, not cultural originality. If "aesthetic quality" is learned from anonymous preferences, the model erases the communities and purposes that made those judgments meaningful.

Provenance is therefore a first-class constraint. A responsible creative system should disclose source permissions, transformations, attribution requirements, human contributions, and material or energy costs. Consent cannot be approximated by stylistic distance. A work can be statistically unlike a training example and still depend on an exploitative acquisition process. Likewise, an image can be legally usable and ethically inappropriate in a given cultural setting.

The refusal list matters. A museum may refuse to optimize visitor flow if the proposed route destroys contemplative time. An artist may refuse an engagement objective that rewards immediate recognizability. A community archive may refuse automated similarity labels for sacred or contested material. These decisions preserve values that the available objective cannot represent.

Music: local costs and global reasons

Voice-leading cost, tuning error, fingering difficulty, rhythmic alignment, spectral masking, and room response admit meaningful formal models. They are local descriptions of musical problems. The danger is letting the convenience of a local cost dictate the global reason for the work.

Suppose a harmonization system minimizes voice motion, forbidden parallels, range violations, and tonal surprise. The resulting solution may be idiomatic and dramatically inert. Increasing surprise can create surface novelty while leaving form unchanged. The evaluation needs multiple time scales: event-to-event motion, phrase trajectory, sectional contrast, and whole-work memory. A penalty that is sensible at one scale can be destructive at another. Repetition, for example, can be redundant locally and indispensable formally.

Performance adds a body and a room. The target is not reproduction of a symbolic score but communication through timing, touch, breath, balance, and risk. A technically optimal fingering may not support the desired articulation. A click-aligned ensemble may lose collective elasticity. A loudness-normalized master may satisfy a platform specification while losing transient shape. Each translation from score to gesture, gesture to sound, sound to recording, and recording to listener preserves some relations and changes others.

Figure 24 maps these interacting spaces. Its arrows should be read as interfaces, not as a claim that musical meaning is computable from acoustics. Pitch geometry preserves interval relations under chosen equivalences; it does not preserve historical function, embodied difficulty, or emotional association.

Performance and rehearsal as adaptive control

Rehearsal is a closed loop with delayed and noisy observation. The performer chooses a practice action, observes errors and sensations, updates an internal model, and selects the next task. Chapter 17's experimental discipline applies: isolate a difficulty when possible, vary conditions to test robustness, and distinguish a temporary performance gain from learning that persists.

The Banister fitness-fatigue model in Chapter 16 is a useful caution. Load can create both adaptation and fatigue on different time constants. The same is true of cognitive and motor practice. More repetitions can improve the current run while degrading attention or increasing injury risk. A robust rehearsal plan alternates consolidation, contextual variation, rest, and simulated performance. It also defines a stopping rule: continuing after reliable success may yield less than switching to another weakness.

Ensemble rehearsal introduces coordination. Local perfection can harm the whole if a player becomes inflexible. The relevant invariant is not identical execution but a shared capacity to respond. Conductors and directors manage attention as a scarce resource: stopping for every local error can destroy the ensemble's model of the whole. Optimization here is partly the scheduling of feedback.

A protocol for creative optimization

Before introducing a generator, score, or search algorithm, write four short statements.

The preservation statement names what must survive translation: a rhythmic identity, an accessible route, a material practice, a community's authority, a performer's agency, or the ambiguity on which a work depends.

The exploration statement names what the system is allowed to vary and why variation is valuable. Search without a reason merely increases selection burden.

The critique statement names who can reject or reframe a proposal, what evidence they will see, and how disagreement will be recorded.

The provenance statement records sources, permissions, transformations, labor, and environmental cost.

Then run the optimization as a bounded episode. Retain diverse alternatives, including an unoptimized baseline. Ask what the strongest candidate exploits, what it makes newly visible, and what it systematically omits. If the answer changes the project's purpose, return to framing rather than adjusting another weight.

When a creative system proposes alternatives, review them in four passes. The **constraint pass** checks safety, access, provenance, buildability, and other nonnegotiable conditions. The **difference pass** asks what each alternative makes possible that the baseline does not. The **meaning pass** asks how form, material, reference, audience, and context change interpretation. The **process pass** asks what accepting the proposal does to authorship, labor, skill, and future exploration.

Retain rejected candidates and reasons. A pattern of rejection can reveal a missing constraint or an inappropriate representation. Include at least one deliberately nonoptimal or manually made baseline; otherwise the system defines the visual or musical vocabulary within which it is judged.

Critique should permit "these are not the right alternatives." That response changes the feasible set rather than assigning a low score. Record it as reframing. The optimizer can then generate a contrasting family, or the team can stop automation and return to direct making.

Accessibility should enter before candidate generation. In architecture it changes routes, thresholds, acoustics, signage, sensory conditions, and participation. In visual art it changes description, scale, contrast, touch, sound, and display. In performance it changes instruments, pacing, seating, communication, and rehearsal.

Treating access as a late compliance check removes creative possibility. Co-design with disabled participants can produce forms valuable to wider audiences without claiming that all needs are identical. Preserve conflicts and provide alternatives where one solution cannot serve every mode.

Music: geometry, composition, performance, and listening

Music organizes pitch, rhythm, timbre, dynamics, space, expectation, gesture, and social practice. Optimization can generate material, solve voice leading, tune instruments, design rooms, and personalize tools. Musical value, however, is not equivalent to minimizing rule violations.

Pitch spaces and tuning

Equal temperament divides an octave into twelve equal logarithmic steps. A pitch n semitones above frequency f_0 is

$$f_n = f_0\, 2^{n/12}. \tag{103}$$

This makes transposition uniform while approximating just intervals. Other temperaments optimize different compromises among interval purity, key color, modulation, and instrument construction. Tuning is therefore an historical multiobjective design, not a solved natural constant.

Tymoczko represents chords and voice leadings geometrically, revealing equivalence under octave, permutation, and transposition (Tymoczko 2006). A simple voice-leading cost between ordered pitch vectors p and q is

$$C(p,q) = \min_{\pi,k} \sum_{i=1}^{m} w_i \left|p_i - q_{\pi(i)} - 12k_i\right|, \tag{104}$$

where m is the number of voices, $w_i \geq 0$ weights the relative cost of moving voice i, π is a permutation matching voices, and each integer k_i represents an octave shift. The cost captures motion, not expressive function by itself.

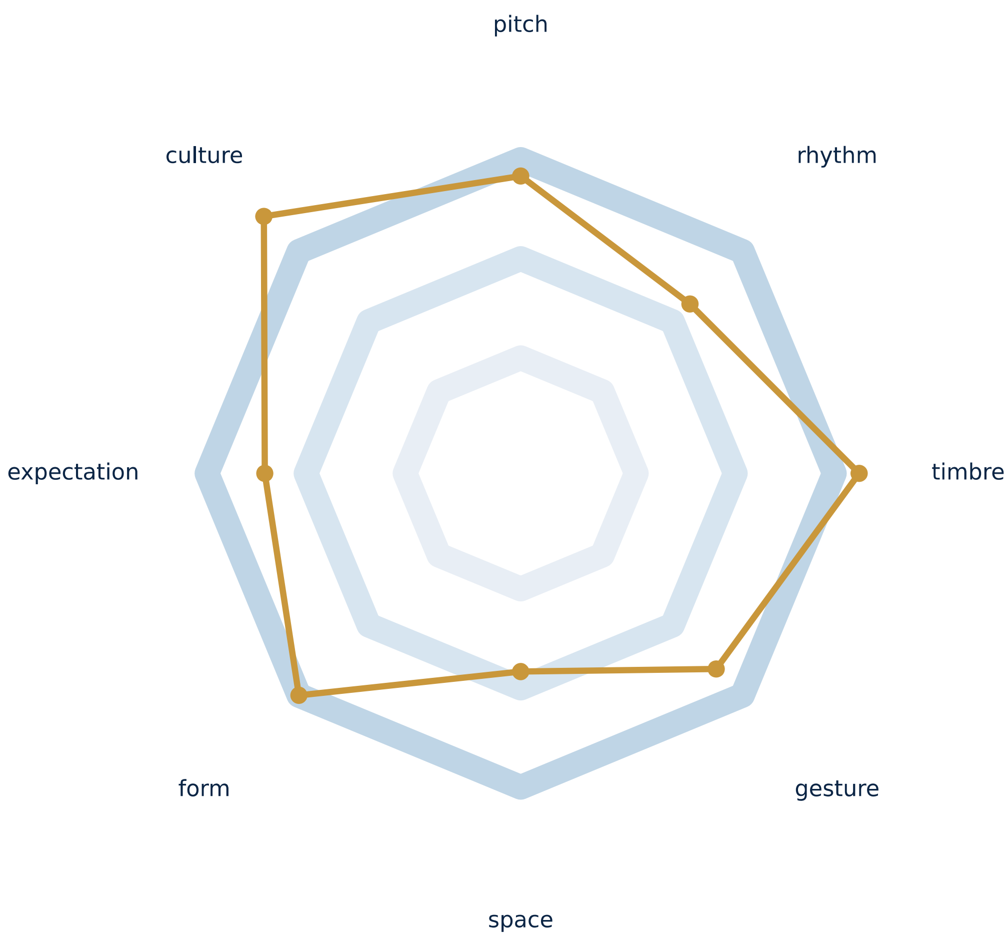


Figure 24. An eight-axis musical design radar spanning pitch, rhythm, timbre, gesture, space, form, expectation, and culture.

Composition and constraint

Counterpoint, harmonic syntax, rhythmic cycles, serialism, stochastic music, and algorithmic composition define different feasible spaces; Xenakis develops stochastic, Markov, game-theoretic, and sieve-based composition rather than evolutionary search (Xenakis 1992). Constraint satisfaction can avoid parallels, respect ranges, or realize a rhythmic pattern. Dynamic programming and shortest paths can optimize voice leading; probabilistic models can generate stylistically likely sequences; evolutionary search can expose surprising combinations (Sims 1991).

Lerdahl and Jackendoff model aspects of tonal cognition through grouping, meter, time-span, and prolongational structures (Lerdahl and Jackendoff 1983). Such theories explain expectations for particular listening traditions, not universal musical law.

Performance

A score underdetermines timing, articulation, balance, phrasing, and physical gesture. Performance optimizes communicative intention under instrument, room, ensemble, fatigue, and rehearsal time. Strict temporal accuracy can sound lifeless; expressive microtiming deliberately deviates from a grid.

Practice itself is a resource-allocation problem. Slow practice, chunking, interleaving, mental rehearsal, recording, and performance simulation target different weaknesses. The shortest route to a technically correct passage may not build resilient recall on stage.

Acoustics and production

Room design balances reverberation, clarity, envelopment, intimacy, bass response, isolation, and variability across seats. Sound reinforcement and mixing manage headroom, masking, loudness, localization, and translation across playback systems. An optimizer can flatten a frequency response while degrading phase, transient behavior, or musical intent.

Recommendation systems optimize engagement and can narrow exposure or reward homogeneity. Diversity, creator sustainability, user agency, and serendipity deserve explicit controls.

Field checklist

- Separate hard obligations, trade-offs, and interpretation; use alternatives for critique, not outsourced authorship.
- Evaluate across spatial, temporal, and social scales; record provenance, consent, labor, and environmental burden.
- Preserve disagreement across value types; test buildability, performance, maintenance, and accessibility.
- Name who may reframe or refuse the optimization.

Translation lens. Creative practice transfers search, constraint, feedback, and Pareto comparison, but preserves accountable authorship rather than a fixed objective.

Chapter 16 · Literature, narrative, media, games, sport, and competition

Narrative systems, language models, and media

Literature optimizes no universal score. A work may seek clarity, ambiguity, suspense, density, voice, estrangement, persuasion, memory, or play—and can change those aims as it unfolds. Formal models help analyze and generate structure when they remain subordinate to reading.

Plot and narrative structure

Narratives arrange events through order, duration, frequency, perspective, and voice (Genette 1980). A story is not identical to its discourse: the same events can be told in different sequences and from different positions. Propp's morphology identifies recurring functions in a corpus of folktales (Propp 1968); it is a descriptive grammar, not a recipe for all stories.

A planning model may define states s, actions a, transitions $P(s' \mid s, a)$, and narrative rewards such as causal coherence or character goal progress. The difficulty is that an action's value depends on interpretation and later revelation. Surprise requires balancing improbability with retrospective intelligibility.

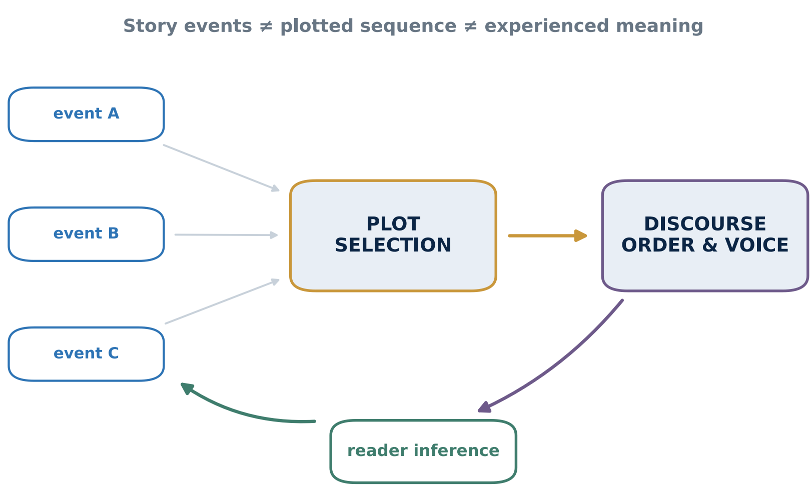


Figure 25. A narrative search diagram connecting world events, plot selection, discourse order, voice, reader inference, and revision.

Style and language models

Stylometry measures word frequencies, syntax, rhythm, or semantic patterns. Compression and probabilistic models can estimate regularity. Generated text commonly maximizes or samples from conditional likelihood, with decoding choices such as beam width, temperature, and nucleus threshold. Higher likelihood tends toward conventional phrasing; unconstrained sampling can lose coherence.

Editing is multiobjective. Shortening can improve pace and remove necessary texture. Simplifying can improve access and erase voice. A practical editorial objective is lexicographic: preserve meaning and ethical commitments first, then improve structure, clarity, rhythm, and correctness within that boundary.

Media systems

Headlines, feeds, search engines, advertising, and streaming platforms optimize attention and conversion. These metrics are behavioral responses shaped by the platform. Click-through can reward outrage, deception, or repetition. Long-term trust, informed agency, diversity, creator welfare, and civic effect require slower measurements and governance.

Communication campaigns optimize reach, comprehension, recall, and action across audiences. Testing variants can improve wording, but effects are contextual and can decay. Persuasion becomes manipulation when it exploits vulnerability, hides intent, or removes meaningful choice.

Interactive narrative

Games and hypertext create branching paths. Aarseth's ergodic literature requires nontrivial reader action (Aarseth 1997). Branching multiplies content, while procedural systems recombine rules. Optimization can allocate authoring effort to high-impact nodes, but rare paths may carry disproportionate artistic or ethical meaning.

Narrative and play sit at opposite ends of a useful spectrum. A finished novel offers a controlled sequence whose effects depend on interpretation; a competitive game offers explicit rules whose outcomes depend on action. Between them lie interactive fiction, recommendation feeds, live performance, sport, role-playing, and procedural media. All can use optimization. None can be understood by its score alone.

The common structure is temporal. Decisions alter what can happen next, and their value depends on later consequences. Chapter 7's Bellman recursion therefore transfers naturally, but only after the state is defined. In chess the state is largely specified by the board and rules. In a novel, "state" includes what the reader believes, remembers, suspects, and values—quantities that cannot be observed directly and need not be shared across readers. In a live sport, fatigue, injury, morale, weather, officiating, and opponent adaptation make the state partially observed. The mathematical structure transfers; the semantics do not.

Narrative as constrained revelation

A narrative orders information, not just events. Let $e_1, \dots, e_n$ be events in the story world and let a discourse choose an ordering, duration, frequency, focalization, and degree of disclosure (Genette 1980). A suspense problem can be represented as a schedule of clues under consistency constraints, yet "suspense" is not a scalar property of a clue. It arises from the relation among reader models: what is possible, what is feared, what is desired, and what appears retrospectively fair.

Planning systems for stories often assign rewards to causal progress, character goals, novelty, or dramatic tension. This is productive when the representation helps an author inspect gaps. It becomes prescriptive when corpus frequency is mistaken for narrative necessity. Propp's functions describe patterns in a particular folktale corpus (Propp 1968); they do not define the feasible set of literature.

An editorial optimization can be layered. Preserve factual commitments, ethical boundaries, and the work's governing ambiguity. Within those constraints, improve causal legibility, pacing, continuity, rhythm, and sentence-level precision. Then reread globally, because a local improvement may remove a motif needed later. The process resembles coordinate descent only superficially: revising a late scene can change the meaning of an early image, so the coordinates are not independent.

Figure 25 distinguishes world events, plot selection, discourse order, voice, reader inference, and revision. The figure is also a warning about compression. A plot graph preserves causal or temporal relations selected by the model. It does not preserve prose texture, unreliable narration, cultural reference, or the experience of rereading.

Language models and decoding objectives

A generative language model assigns conditional probabilities to token sequences. Decoding then turns a distribution into a particular text. Greedy selection, beam search, temperature, nucleus sampling, repetition penalties, and reranking embody different trade-offs among likelihood, variation, consistency, and computational effort. More exhaustive likelihood search can produce dull, repetitive, or even empty output because model score, adequacy, and open-ended literary quality are not monotone (Holtzman et al. 2020; Stahlberg and Byrne 2019).

For a writing system, the objective should be evaluated at several levels. Token likelihood measures statistical fit. Factuality requires sources and entailment checks. Coherence concerns entities, causal relations, themes, and discourse. Style concerns rhythm, voice, register, and departure from cliché. Ethical review concerns representation, privacy, attribution, and foreseeable use. User control concerns whether the writer can understand, edit, and reject the system's contribution. No single automatic metric validates all levels.

Optimization also changes labor. If a platform rewards speed and volume, authors may be pushed toward template production even when the tool was introduced as assistance. Provenance records should make human and machine transformations inspectable without reducing authorship to a percentage. A sentence may be machine-proposed and human-made through rejection, relocation, rewriting, or contextual responsibility.

Media platforms as endogenous optimization

In a recommendation system, the objective changes the data-generating process. Showing an item affects exposure, learning, creator incentives, and future preferences. Click-through, watch time, completion, subscription, and retention are not passive measures. They are interventions in a coupled social system.

The Goodhart mechanism from Chapter 17 appears in at least three forms. Creators adapt thumbnails and pacing to the metric. Users learn habits shaped by the interface. The platform's training data increasingly reflects its own past policy. A model can therefore become accurate about a world it helped narrow. Randomized exploration provides causal information, but experimentation on attention needs limits: vulnerable users, political content, grief, health information, and compulsive patterns cannot be treated as ordinary engagement opportunities.

A governed media objective should include long-horizon and distributional measures: user control, diversity of exposure, creator concentration, satisfaction after use, trust, misinformation burden, and the ability to leave. Some should be constraints or audit triggers rather than weighted rewards. The counterfactual baseline matters. "More time on platform" can mean displaced sleep, work, care, or offline community.

Game strategy and the ecology of rules

Within a fixed game, optimization can be exacting. Search, dynamic programming, equilibrium computation, reinforcement learning, and simulation can identify strong policies. Yet the environment includes other optimizing agents. A policy that is optimal against a population can change that population, provoking counters and changing the value landscape. Chapter 6's game theory supplies equilibrium language; Chapter 3 supplies governance of the rules themselves.

Game design operates one level higher. Designers choose the action space, information structure, reward schedule, matchmaking, resource economy, and enforcement. The aim is not to maximize one player's payoff but to sustain meaningful play for a population. Balance can mean viable strategic diversity, counterplay, accessibility, recoverability, and resistance to exploit. Equal win rates can conceal a strategy that is frustrating, cognitively exclusionary, or dominant only at expert levels.

Telemetry should be stratified. Aggregate success rates can hide platform, region, disability, skill, or party-composition effects. Patch evaluation should distinguish immediate novelty from stable adaptation. A balance change is a sequential experiment with interference: players share information and change opponents' experiences. Confidence intervals derived as if matches were independent can be falsely narrow.

Monetization creates a second objective system. Retention and spending can conflict with player welfare, parental expectations, and the integrity of progression. Dark patterns are not a harmless local optimum; they are a governance failure. Constraints should cover disclosure, age suitability, purchase friction, self-exclusion, and the separation of competitive advantage from spending where appropriate.

Sport: prediction, prescription, and consent

Sports models estimate expected points, win probability, player contribution, injury risk, or tactical value. The distinction between prediction and prescription is crucial. A model may predict that an athlete is likely to succeed under current selection practices without showing that those practices are fair or causally effective. Observational data is selected by coaches, injuries, contracts, and prior models.

Tactical decisions are coupled. A shot's value depends on defender response, clock, score, fatigue, lineup, and what the attempt teaches the opponent. Repeated use changes future opportunity. Robust play therefore balances exploitation with strategic variation. The best immediate action need not be the best policy over a series.

Training planning adds delayed physiology. The readiness model in the sports-analytics subsection expresses performance potential as the difference between adaptation and fatigue on distinct time constants (Banister et al. 1975). It is a conceptual model, not a universal dosing formula. Individual parameters vary, measurement is noisy, and injury mechanisms are not captured by a single performance state. A model may recommend load; medical professionals and athletes decide whether the load is acceptable.

Athlete data raises special governance issues. Location, sleep, menstrual cycles, biometrics, mental health, and injury history can improve planning and enable surveillance or coercion. Consent inside an employment or scholarship relationship may not be freely given. Collect only data tied to a defined decision, limit access and retention, provide correction and appeal, and prohibit secondary uses that were not agreed. An athlete must be able to challenge both the measurement and the inference.

Competition beyond the scoreboard

Rules create the object being optimized. Equipment limits, classifications, salary caps, drafts, anti-doping systems, judging criteria, and tie-breakers distribute opportunity and shape style. Mechanism design asks how rules transform private incentives into collective outcomes. Legitimacy asks a further question: who had standing to choose those outcomes?

Every rule change should name its theory of harm or benefit, anticipated strategic response, distributional effects, evidence window, and reversal condition. A faster game can be less safe. A more "objective" judging system can reward conformity to measurable technique. A cost cap can create enforcement complexity that advantages organizations with legal and engineering resources. Optimization of the competition requires an audit of the enforcement system.

The constitutive purpose must remain visible. In sport, that may include excellence, uncertainty of outcome, bodily expression, community, tradition, and the credible comparison of performances. In games, it may include mastery, discovery, sociality, fiction, or playful disorder. If the metric destroys the reason participants value the activity, the correct response is to redesign the metric or refuse it—not to search harder.

A translation protocol for narrative and play

First identify the level: artifact, participant strategy, platform policy, or rule system. Objectives at one level can be constraints at another. Second identify the state variables the model cannot observe: interpretation, trust, tacit skill, injury, informal norms, or future cultural value. Third identify adaptation: readers learn genres, players learn exploits, athletes change tactics, creators change output. Fourth compare short- and long-horizon effects. Fifth preserve a route for authors, participants, and affected communities to contest the model.

A media or cultural recommender observes outcomes only for items it exposed. Naively comparing clicked and unclicked items confounds relevance with policy. Randomized or structured exploration can estimate counterfactual response, but exploration has ethical and experiential cost. Use conservative domains, user control, and clear exclusions.

Evaluate beyond immediate behavior. Follow satisfaction after use, diversity across a session and over time, ability to find a known item, novelty judged useful, creator concentration, and whether users can reset or escape the profile. Compare with chronological, curated, search-based, and user-configured baselines.

An engagement improvement can reflect compulsion or a difficult exit. Pair behavioral measures with survey, interview, complaint, and well-being evidence. Preserve a nonpersonalized path. A recommendation is a proposal, not a command about what a person should value.

Automatic story metrics often measure overlap, predictability, or classifier judgment. Once optimized, they reward formula, keywording, or surface coherence. Use human reading, source checks, causal graph inspection, and adversarial examples. Preserve passages where reviewers disagree.

A benchmark corpus represents genres, languages, and historical conventions unevenly. Report coverage and avoid universal claims. Generated narrative should be evaluated for attribution, stereotype, privacy, and the effects of deployment, not only text quality.

For writing assistance, measure whether the user can maintain intention, identify unsupported content, revise efficiently, and understand provenance. A fluent continuation that displaces authorship is not automatically assistance.

Games, sport, rules, and training

Games create explicit goals inside designed rule systems; sport adds bodies, institutions, spectators, and culture. They are ideal laboratories for optimization because objectives are visible—and ideal warnings because optimizing the score can still undermine the game.

Strategy and equilibrium

In a finite zero-sum game with payoff matrix A, the row player chooses mixed strategy p and the column player q. The minimax theorem gives

$$\max_p \min_q p^T A q = \min_q \max_p p^T A q. \tag{105}$$

This supports equilibrium strategies in games such as poker subproblems. General games may have multiple equilibria and require beliefs about opponents. Exploitative play can outperform equilibrium against a known weakness while becoming vulnerable to counteradaptation.

Search methods evaluate game trees with minimax, alpha-beta pruning, Monte Carlo tree search, learned value functions, and policy improvement. AlphaGo combined deep networks with tree search and reinforcement learning (Silver et al. 2016). The achievement depended on a clear rule set and simulator—conditions absent from many real institutions.

Game design

Designers tune challenge, pacing, economy, progression, information, social interaction, and accessibility. A balanced character is not one with identical statistics but one that supports viable counterplay and meaningful choice. Live-service telemetry reveals behavior, yet optimizing retention or spending can produce compulsion and distrust.

Procedural content generation searches levels, maps, quests, or items under feasibility and novelty constraints. Human playtesting detects legibility, emotion, and emergent strategies that static metrics miss. Rules should be evaluated for dominant strategies, exploits, accessibility, and community norms.

Sports analytics

Sport optimization covers tactics, roster construction, scheduling, training, nutrition, equipment, and injury prevention. Expected-points or win-probability models translate situations into decision support. Their validity depends on opponent, era, officiating, selection, and strategic reaction.

Training balances stimulus and recovery. The Banister fitness-fatigue model represents performance as the difference between slower positive adaptation and faster negative fatigue responses (Banister et al. 1975):

$$R(t) = R_0 + k_1 \sum_{i=1}^{t-1} u_i \, e^{-(t-i)/\tau_1} - k_2 \sum_{i=1}^{t-1} u_i \, e^{-(t-i)/\tau_2}. \tag{106}$$

$R(t)$ is modeled readiness at time t and R_0 is baseline readiness; u_i is the training impulse at time i; k_1 and k_2 are positive adaptation and fatigue gains; and τ_1 and τ_2 are their decay time constants. The described slower adaptation and faster, initially larger fatigue response requires $\tau_1 > \tau_2$ and $k_2 > k_1$. These parameters are individual and uncertain. Maximizing short-term load can increase injury and burnout. Athlete consent, education, and data privacy constrain monitoring.

Integrity and spectacle

Anti-doping, equipment rules, salary caps, draft systems, and officiating seek a credible contest. Rule changes optimize a bundle: safety, fairness, pace, tradition, broadcast appeal, and participation. Participants adapt, so every rule is a mechanism-design experiment.

When a league or game changes rules, the treatment reaches all participants and they adapt together. Evaluation should use interrupted time, comparable competitions, simulation, and qualitative observation, while acknowledging interference. Separate transient novelty from stable strategy.

Define desired mechanisms before outcomes: reduce dangerous contact, increase viable strategies, shorten dead time, or improve access. Then measure unintended response, enforcement burden, distribution by role or level, and whether participants find the contest more legitimate. A rule can meet a numerical target and undermine play.

Athlete or performer monitoring should inform a conversation, not issue an unquestionable prescription. Estimate response with uncertainty, distinguish population evidence from individual pattern, and preserve clinician or coach judgment within authority. Sudden changes, pain, illness, and psychological state can invalidate the forecast.

Use a safe workload envelope, planned recovery, and review triggers. Data collection should be minimal, consented, and separated from unrelated selection or contract decisions. The person should see and correct the record. Optimization of performance does not authorize ownership of the body.

Difficulty is multidimensional: motor speed, precision, perception, memory, planning, language, ambiguity, time pressure, and tolerance for failure. A single “easy mode” may lower enemy strength and leave the actual barrier untouched. Expose adjustable dimensions and preserve the intended experience where possible.

Accessibility options can be part of the rule system: remapping, timing windows, visual and auditory alternatives, automation, pause, information, and cooperative support. Evaluate with players whose access needs differ. Competitive integrity is not a reason to exclude; it is a design question about categories, disclosure, and meaningful comparison.

Optimization can tune adaptive difficulty, but hidden adaptation may undermine trust or erase accomplishment. Give players control or clear communication. The purpose is sustained meaningful play, not maximized session length.

Field checklist

- Distinguish events, discourse, and reader inference; evaluate generated language beyond likelihood.
- Treat recommendation as intervention; separate player strategy, game design, platform policy, and institutional rules.
- Test tactics and patches under adaptation; protect athlete consent, privacy, health, and appeal rights.

- Audit whether the score serves the activity’s constitutive purpose.

Translation lens. Narrative and competition transfer dynamics, search, equilibrium, and feedback; temporal consequence is preserved, not meaning, legitimacy, or purpose.

Part VI · Practice, Translation, and Frontiers

Chapter 17 · Experimentation, digital twins, human judgment, and failure modes

Evidence-producing optimization

An optimizer is only as credible as its evidence loop. Experiments estimate causal response, simulation explores models, digital twins synchronize selected model states with an operating asset, and human-in-the-loop methods elicit judgment that is difficult to encode in advance.

Design of experiments

One-factor-at-a-time testing cannot reveal interactions efficiently. Factorial and fractional-factorial designs vary inputs together; response-surface methods fit local curvature; blocking controls nuisance variation; randomization protects against unmeasured trends. For response

$$y = \beta_0 + \sum_{i=1}^{p} \beta_i\, x_i + \sum_{i=1}^{p} \beta_{ii}\, x_i^2 + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \beta_{ij}\, x_i x_j + \eta, \tag{107}$$

the pure quadratic terms represent local curvature, the interaction terms show why the best setting of one factor can depend on another, and η is regression noise (National Institute of Standards and Technology 2013). Replication estimates noise. Sequential designs use early results to place later experiments where information or improvement is most valuable.

Online A/B tests estimate treatment effects in live systems. Guardrail metrics protect reliability, safety, and vulnerable groups. Network spillovers, novelty, interference, multiple testing, peeking, and long-term adaptation can invalidate simple comparisons. Stopping rules and analysis plans should be specified before results are inspected.

Simulation optimization

When the objective is an expectation available only through simulation,

$$\min_x J(x) = \mathbb{E}[g(x, \xi)], \tag{108}$$

each evaluation is noisy. Common random numbers can reduce variance when comparing designs. Ranking-and-selection allocates samples among alternatives. Surrogates, stochastic approximation, and Bayesian optimization reduce expensive runs. Validation must compare simulated and observed behavior at the level relevant to the decision (Jones, Schonlau, and Welch 1998; Shahriari et al. 2016).

Digital twins

A digital twin is not simply a detailed simulation. It is a maintained relationship among an asset, data, models, and decisions. The twin’s state estimate should record uncertainty and data freshness. Calibration can make one output accurate while compensating with wrong parameters, so identifiability and multi-condition validation matter.

A twin can support condition monitoring, what-if analysis, predictive control, maintenance, and training. Its governance includes model version, sensor provenance, cybersecurity, operator override, and retirement. High fidelity is useful only where it changes an action.

Human-in-the-loop search

Interactive optimization asks people to compare alternatives, set aspiration levels, adjust constraints, or critique proposals. Pairwise preference data can fit a latent utility model, but preferences may be unstable and constructed during interaction. The interface should show trade-offs and uncertainty without forcing false precision.

Figure 26. Seven-stage evidence loop—question, experiment, simulate, calibrate, decide, operate, and learn—around a central EVIDENCE LOOP.

Figure 26 distinguishes six evidence roles: experiments support causal contrasts, simulations support conditional model claims, twins maintain state relationships, operations reveal consequences, and judgment and revision govern the loop.

Participatory modeling broadens whose knowledge enters. Domain experts reveal tacit constraints; affected communities reveal harms and meanings; operators reveal workarounds. Disagreement should sometimes remain visible rather than averaged into a pseudo-consensus.

Practice note. Close every optimization loop with a measurement plan, a model-update rule, an accountable decision maker, and a condition for stopping or rollback.

Optimization is often presented as a map from a model to a decision. In practice it is a map from an evidence process to a provisional intervention. Experiments, simulations, twins, operators, and post-deployment observations produce different kinds of evidence. Their results cannot simply be pooled. A randomized experiment supports a causal contrast under its assignment and population; a simulation supports a conditional claim about a model; an expert judgment may reveal tacit feasibility; operational telemetry describes a system already shaped by previous decisions.

The field-guide rule is to attach every number to its provenance and every recommendation to a claim boundary. “Alternative A has lower expected cost” is incomplete. Lower under which model, calibrated on which period, with which uncertainties, after which exclusions, and at what decision horizon? Chapter 3’s decision dossier becomes the container for these answers.

Experiments should optimize information, not just response

When experiments are expensive, the next trial can be chosen to improve the product or to reduce uncertainty about the response surface. Those aims may conflict. Exploitation samples a promising region; exploration distinguishes models or exposes interactions. A sequential design should state which information is decision-relevant. Reducing uncertainty everywhere can waste samples on distinctions that would not change the chosen action.

Factorial designs reveal interactions that one-factor-at-a-time work cannot identify efficiently. Blocking, randomization, replication, and predeclared stopping protect inference. These are not ceremonial statistical details. They prevent production drift, operator learning, day-of-week patterns, and selective continuation from being mistaken for treatment effects.

Online experiments add interference and adaptation. A pricing change affects competitors; a feed treatment affects what creators produce; a clinical workflow affects staff behavior across arms. The stable-unit assumption must be argued, not inherited. If spillovers are material, randomize at a cluster or system level, model the network, or treat the experiment as a policy intervention rather than a collection of independent observations.

Guardrails need explicit decision rules. A primary metric can improve while a rare safety outcome deteriorates. Because rare harms produce wide intervals, “not statistically significant” is not evidence of safety. Use prior evidence, severity-weighted thresholds, sequential monitoring, and conservative deployment stages. Define who may stop the experiment and how quickly the system can be restored.

Simulation: conditional truth with an extrapolation hazard

Simulation optimization is powerful because it permits controlled counterfactuals. It is dangerous because an optimizer deliberately searches regions where the model has received least validation. An ordinary forecast may remain near historical use; an optimizer discovers the simulator’s loopholes.

Validation must therefore be decision-centered. Compare not only output distributions but also rankings, sensitivities, constraint margins, failure modes, and the behavior of optimized candidates. A model can reproduce average throughput and misrank scheduling policies. It can match a building’s annual energy and misrepresent peak load. It can match glucose traces under ordinary meals and fail under exercise. These defects matter because they change the decision.

Use several layers of challenge:

- **verification:** does the implementation solve the stated equations or rules;
- **calibration:** do parameters reproduce selected observations;
- **validation:** does the model support the intended class of claims;
- **stress testing:** does it behave plausibly outside calibration conditions;
- **red-teaming:** can the optimizer exploit numerical, physical, or semantic gaps;
- **reconciliation:** when model and world differ, is the difference explained and recorded?

Common random numbers, variance reduction, ranking-and-selection, stochastic approximation, and surrogate models can make simulation search efficient. Efficiency cannot compensate for a wrong boundary. The appropriate baseline is often a simple policy evaluated in the same simulator and in the field.

Digital twins as governed relationships

A digital twin is not an object with an intrinsic fidelity score. It is a maintained relationship among an operating system, sensors, data pipelines, state estimation, models, decisions, and accountable people. The twin can be high fidelity for thermal control and irrelevant for fatigue prognosis. Fidelity must be indexed to a claim.

Every operational twin needs a state-of-knowledge display: data timestamp, missingness, sensor health, model version, parameter uncertainty, prediction interval, and distance from validated conditions. A green status icon that hides stale data is worse than no twin because it manufactures confidence.

Model updates are interventions. Recalibration can alter alarms, maintenance schedules, and controller behavior. Treat model deployment as configuration change with testing, approval, rollback, and versioned evidence. Cybersecurity is part of validity: corrupted sensor data or unauthorized model updates change the represented world. Chapter 9’s assurance chain and Chapter 10’s interface contracts apply without metaphor.

Human judgment is not a residual error term

Operators, patients, engineers, editors, communities, and decision boards know things the model does not. They also have biases, incentives, limited attention, and uneven authority. “Human in the loop” is therefore not a sufficient safeguard. The human needs time, information, competence, an actionable alternative, and protection from retaliation. Overrides must be observable and learnable without becoming a performance target that discourages justified dissent.

Design the division of labor around comparative advantage. Machines can maintain consistency, search large combinatorial spaces, and surface anomalies. People can reinterpret purpose, recognize novel context, negotiate values, and assume responsibility. Some tasks require two independent judgments; others require an appeal body outside the operating chain. The interface should expose uncertainty and causal assumptions, not only a recommendation.

Preference elicitation deserves special caution. A person asked to rank complex alternatives may construct a preference during the process. Pairwise comparisons can reduce cognitive load, but the inferred utility is conditional on framing, information, and fatigue. Preserve the raw comparison, allow "incomparable" and "need more information," and show how sensitive the recommendation is to the fitted preference model.

Failure modes as model diagnostics

The failure modes below are related but distinct.

Proxy gaming occurs when measured performance can be improved without the intended value. **The optimizer's curse** is statistical selection of favorable error; **Goodhart adaptation** occurs when agents change behavior because the measure became a target. They require different countermeasures.

Boundary failure moves burden outside the accounting frame. Lifecycle diagrams and affected-party review expand the boundary; sensitivity analysis cannot fix an omitted population.

Model exploitation occurs because search finds regions where approximation error is favorable. Trust regions, adversarial cases, physical constraints, and staged tests can help.

Brittleness consumes slack and diversity. Robust optimization protects against specified uncertainty; resilience adds detection, graceful degradation, recovery, and adaptation after surprise. The distinction from Chapter 6 should remain explicit.

Optimization theater uses formal precision to prevent scrutiny. Reproducibility, model cards, independent review, and the right to challenge counter it.

Success that erases purpose is institutional. Periodic purpose review, qualitative evidence, and refusal may be the only remedy.

Figure 27 illustrates the adaptive form, not the optimizer's curse. The separation matters: a rotating metric may frustrate gaming but does not remove selection bias, while shrinkage or out-of-sample evaluation may reduce selection bias without changing strategic response.

Stop, rollback, and learn

An optimization project needs stopping rules at three levels. Stop **search** when further computation is unlikely to change the decision or when evidence quality dominates solver error. Stop **deployment** when safety, equity, reliability, or trust triggers cross a threshold. Stop **the project** when the available representation cannot support a legitimate decision.

Rollback is a designed capability. Preserve the previous policy, compatible data, operator knowledge, spare capacity, and authority to restore service. A recommendation that cannot be reversed needs stronger evidence and broader consent. Irreversibility may arise from physical construction, ecological thresholds, institutional lock-in, or loss of skill even when the software itself is reversible.

Post-deployment review should compare predicted and observed outcomes, including distribution and mechanism. Record overrides, near misses, complaints, workaround behavior, and unanticipated externalities. A failed prediction can improve a model; an unexplained organizational workaround may reveal that the modeled constraint was fictitious or that an omitted burden made the recommendation unusable.

The minimum evidence packet

For each optimized intervention, preserve:

- the decision and rejected baselines;
- model, data, code, solver, versions, and random seeds where relevant;
- objective and constraint definitions with owners;
- calibration, validation, sensitivity, and stress-test results;
- evidence gaps and extrapolation flags;
- affected groups, approval, appeal, and override routes;
- monitoring metrics, thresholds, response owners, and review dates;
- rollback assets and the condition for retirement.

This packet is not a guarantee. It is the minimum structure needed for another person to understand what was claimed and decide whether the claim still holds.

Before deployment, ask the team to imagine that the optimization caused a serious failure one year later. Each participant writes a mechanism independently: wrong boundary, proxy gaming, corrupted sensor, strategic response, inaccessible interface, common-cause failure, silent update, lost skill, or unavailable fallback. Group the mechanisms by detection and recovery, then add tests and owners.

Run a rollback drill in the actual environment. Can the prior model, policy, data schema, credentials, configuration, and operating instructions be restored? Is old hardware or manual capacity still available? Does rollback preserve records created under the new system? How long does it take, and who can authorize it outside business hours?

A nominal “off switch” can fail because dependent systems now expect its output, staff no longer know the old process, or data migration is irreversible. Measure recovery as an end-to-end service capability.

List material model assumptions and assign each an evidence strength, consequence if wrong, detectability, and mitigation. The purpose is not to manufacture one risk number. It is to allocate challenge effort to assumptions that are both consequential and weakly observed.

Diversify evidence where common assumptions could fail. Two simulations sharing the same physical law, data, or code are not independent. Use alternate formulations, analytical bounds, experiments, and domain judgment. Track unresolved assumptions through deployment.

When the optimizer selects a region near a weak assumption, raise the risk class and require confirmation. This makes model risk endogenous to search rather than a static appendix.

Failure modes, theater, refusal, and purpose

Optimization failures are often not solver failures. The algorithm finds exactly what the formulation rewards. The defect lies in boundaries, proxies, data, dynamics, incentives, or authority.

Proxy gaming and Goodhart effects

Suppose true value is $V(x)$ but the system optimizes proxy $M(x) = V(x) + \varepsilon(x)$. Selecting the maximum among many candidates preferentially selects positive error:

$$\mathbb{E}[\varepsilon(\hat{x})] > 0, \qquad \hat{x} = \arg\max_{x \in X} M(x), \tag{109}$$

under common conditions. Stronger search can therefore widen the gap between measured and true value. This is the optimizer's curse analyzed by Smith and Winkler (Smith and Winkler 2006). Once people adapt strategically, the measurement process itself changes; that distinct feedback mechanism belongs to the Goodhart family of failures (Goodhart 1975; Muller 2018).

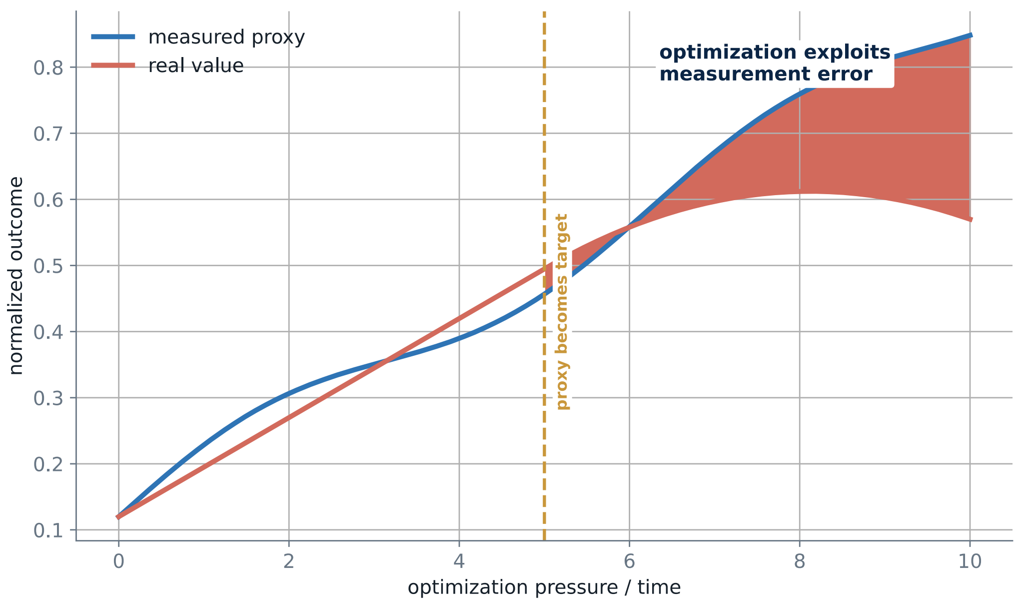


Figure 27. Measured proxy and real value diverge after the proxy becomes a target and optimization exploits measurement error.

Countermeasures include multiple measures, qualitative review, random audits, causal evaluation, rotating indicators, uncertainty penalties, and explicit anti-gaming tests. None substitutes for revisiting the goal.

Boundary errors

A factory minimizes on-site emissions by outsourcing a dirty process. A hospital minimizes length of stay by shifting burden to families. A model minimizes false negatives while flooding staff with alerts. These are boundary failures. Draw the lifecycle, supply chain, feedback loops, affected groups, and externalities before choosing metrics.

Model exploitation

An optimizer searches farther than ordinary use and finds model weaknesses: nonphysical geometry, adversarial inputs, unsafe controller actions, unrealistic financial leverage, or policy loopholes. Validation data sampled near past behavior may not cover the optimized region. Trust regions, feasibility restoration, adversarial testing, and staged deployment limit extrapolation.

Brittleness and hidden coupling

Tight optimization consumes slack. High utilization creates queues; just-in-time inventory creates disruption exposure; aggressive weight reduction reduces damage tolerance; narrow specialization reduces adaptability. Correlated failures defeat redundancy. Robustness requires scenario diversity and attention to common causes.

Technical systems also carry organizational coupling. A dashboard metric can reshape budgets; an automated rule can dissolve expertise; a recommendation can become mandatory through workload. Hidden technical debt in machine-learning systems often resides in data dependencies, feedback, entanglement, and undeclared consumers (Sculley et al. 2015).

Optimization theater

A precise model can legitimate a decision already made, conceal value choices, or overwhelm challenge. Warning signs include undocumented weights, no baseline, no sensitivity analysis, selective scenarios, unexplained exclusions, and no accountable owner. A small transparent model may support better judgment than a large unchallengeable one.

Success that erases purpose

Institutions can optimize a procedure until it ceases to serve its reason: education becomes test production, scholarship becomes citation production, art becomes engagement production, and care becomes throughput production. Periodic purpose reviews should ask whether the activity would still be valued if the metric disappeared.

Practice note. Treat extreme solutions as bug reports about the model until they survive independent domain review, stress tests, and real-world evidence.

At a decision meeting, assign one reviewer to argue from the baseline, one to challenge the boundary, one to inspect uncertainty, one to represent implementation and recovery, and one to trace affected-party consequences. Give them access to model and data assumptions before the presentation.

Require the project team to show a candidate that wins under the objective and fails domain review. If none can be found, either the formulation is extraordinary or challenge has been too weak. Extreme and counterintuitive solutions are valuable tests of what the model permits.

The meeting record should distinguish facts, model-dependent results, preferences, legal or ethical requirements, and unresolved disagreements. A polished frontier chart must not collapse these types.

When a project stops, preserve the reason. Categories include no legitimate authority, insufficient measurement, unacceptable harm, infeasible delivery, no safe fallback, or no improvement over baseline. Record what narrower use remains acceptable and what evidence could reopen the question.

This makes refusal productive. It prevents another team from repeating the same hidden assumptions and can redirect effort to measurement, service capacity, process repair, or a less consequential decision aid. A project portfolio should count well-supported refusal as an outcome, not as absence of output.

Operational dashboards measure what is easy and frequent. Purpose often changes slowly: quality of care, learning, trust, ecological function, artistic vitality, or institutional capability. Create a purpose review with qualitative and long-horizon evidence.

Ask whether the activity would still be valued without the metric; whether workarounds, narrowed capability, or externalized burdens appear; review with affected parties and independent experts.

Purpose monitoring can trigger reframing or retirement even when all service metrics are green. That is not a model failure; it is governance functioning at the correct level.

Field checklist

- Match evidence to claims; predeclare decision-relevant experiments and stopping rules.
- Validate simulations on rankings, sensitivities, margins, and candidates; version twins with uncertainty and rollback.
- Give reviewers time, authority, alternatives, and appeal; distinguish optimizer’s curse, Goodhart adaptation, model exploitation, and boundary error.
- Specify when to stop search or deployment, or refuse the project.

Translation lens. Experimentation and digital twins transfer feedback, state estimation, and sequential design. Preserve claim provenance: model evidence is not world evidence, and human participation is not authority.

Chapter 18 · Worked translations I: infrastructure, health, and adaptive control

How to read the worked translations

The next seven cases exhibit the book's central claim. Each begins in a domain with its own evidence, vocabulary, and authority, then imports structures from other disciplines. The task is not to show that every problem can be written as an objective with constraints. That would be trivial. The task is to show where a transferred structure produces a better decision, which relations it preserves, and which meanings it cannot carry.

Each case follows the same spine — decision and purpose; boundary and affected parties; variables and alternatives; objective architecture; uncertainty and dynamics; method and computation; evidence and validation; governance and failure response; a transfer ledger; a domain-specific deep dive; and a seven-point case checklist — and most cases insert one further domain-specific structural section between the objective architecture and the method. The repeated sequence makes cases comparable. The content inside each heading remains domain-specific.

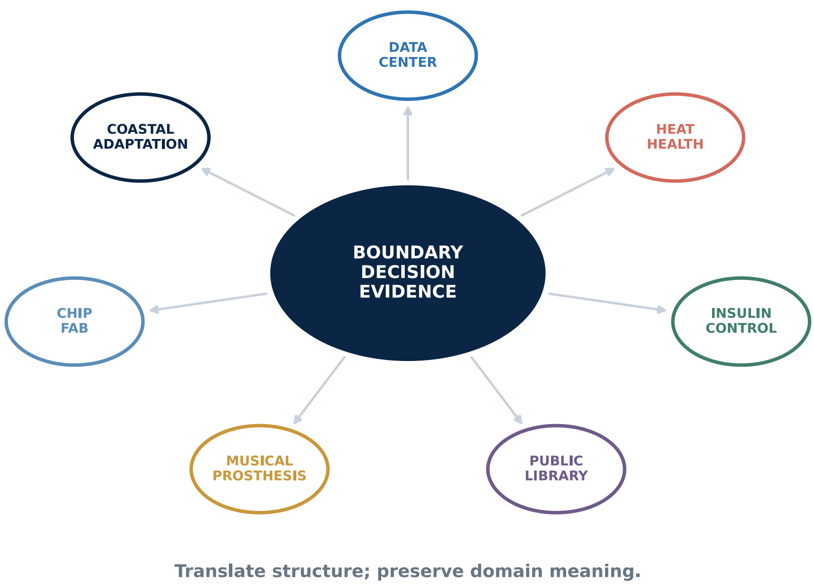


Figure 28. Seven worked translations around shared boundary, decision, and evidence.

Figure 28 is the book's translation map: the seven cases share a grammar of purpose, variables, evidence, and governance, while each transfer ledger records what the mapping fails to preserve.

Case A · A low-carbon, water-aware, resilient data center

Decision and purpose

A cloud operator must decide how to place and time computing work across a regional fleet while renewing hardware and cooling systems. The public description is “reduce carbon.” The operational purpose is wider: deliver useful computation within service commitments while reducing lifecycle greenhouse-gas emissions and water burden, preserving reliability, maintaining security and data-location obligations, and avoiding the transfer of environmental harm to communities with less power.

The decision has several time scales. Every few seconds, schedulers route requests. Every few minutes, controllers adjust cooling and power states. Day-ahead systems defer batch work or purchase electricity. Annual planning chooses capacity, renewable contracts, cooling retrofits, and hardware replacement. Site selection and grid infrastructure create commitments lasting decades. A model that treats these decisions as one static allocation will conceal their different reversibility and evidence requirements.

The baseline matters. Comparing a new schedule with an unconstrained optimum exaggerates benefits. The relevant baselines include current production policy, a simple carbon-threshold policy, a price-only scheduler, and a reliability-first policy with existing capacity. Benefits should be reported against all of them under the same workload traces and grid scenarios.

Boundary and affected parties

The narrow boundary contains servers, network, storage, cooling, and the purchased electricity reported by the operator. The decision-relevant boundary also includes upstream hardware manufacture, construction, refrigerants, backup generation, grid marginal response, water withdrawal and consumption, workload-induced network traffic, hardware disposal, and the user consequences of delay or failure.

Affected parties include service users; tenants with latency, fairness, and data-residency needs; operators and maintenance workers; electricity and water utilities; communities near generating plants and data centers; suppliers; and future populations exposed to emissions. “Carbon-aware” does not imply “just.” A workload shifted away from a high-carbon region can increase water stress elsewhere. A site with low annual average emissions can depend on a marginal generator during the hour when work moves to it. A hardware refresh can lower operating energy while increasing embodied emissions and electronic waste.

Draw the lifecycle before defining the metric. The data flow should distinguish measured site electricity, modeled facility overhead, contractual renewable accounting, estimated marginal emissions, supplier declarations, and scenario-based embodied impacts. These evidence types should not be added without showing their provenance.

Variables and alternatives

At operational scale, let x_{jst} be the fraction or binary assignment of job j to site s and time interval t. Other decisions include server power states z_{st}, cooling set points T_{st}, battery charge and discharge b_{st}, and reserve capacity r_{st}. Planning variables can select cooling technology, renewable contracts, network upgrades, spare capacity, and hardware type and replacement date.

The feasible alternatives are not merely all numerical combinations. Software architecture determines whether work is movable. Data sovereignty limits sites. User-facing requests may not be deferrable. Checkpointing, replication, accelerator availability, data locality, and dependency graphs make migration costly or impossible. Organizational contracts can be harder constraints than hardware capacity.

Representation should therefore begin with workload classes: interactive latency-critical, deadline-constrained batch, interruptible batch, stateful services, regulated data, and safety-relevant services. Each class carries different admissible transformations. Moving an interruptible training job in time is not the same morphism as moving a clinical decision service across jurisdictions.

Objective architecture

A first operational objective might combine electricity cost, marginal emissions, water consumption, service penalties, and wear:

$$\min_{x,z,T,b} \sum_{s,t}[p_{st}E_{st} + \lambda_c c_{st}E_{st} + \lambda_w \omega_{st}W_{st} + \lambda_d D_{st} + \lambda_h H_{st}]. \tag{110}$$

Here p_{st} is electricity price, E_{st} is electricity use, c_{st} an operational emissions factor, W_{st} water consumption, ω_{st} a scarcity or local-impact factor, D_{st} service degradation, and H_{st} hardware or battery wear. Equation (110) is useful for computation and inadequate as governance. Its weights imply substitution: enough latency benefit can purchase water burden; enough cost saving can purchase carbon. The organization may reject those trades.

A safer architecture is layered. Reliability, security, data residency, and emergency-service commitments are hard constraints. Carbon and water receive annual budgets and site-level limits. Tenant fairness receives minimum service guarantees. Cost and residual impact are improved within that feasible region. Planning then compares nondominated portfolios across lifecycle emissions, water, land, cost, adaptability, and recovery time.

Operational and embodied carbon must not be collapsed without time. Manufacturing emissions occur now; operational savings accrue over a device’s remaining life under uncertain grid decarbonization. The replacement decision should compare discounted or physically time-indexed emissions and include reuse. A “more efficient” accelerator may increase total demand through rebound. Report both per-service efficiency and absolute fleet impact.

Constraints, rights, and interfaces

Capacity constraints couple jobs assigned to a site:

$$\sum_j a_{jk}\, x_{jst} \le C_{kst} z_{st} \qquad \text{for each resource } k\text{, site } s\text{, and time } t. \tag{111}$$

Here a_{jk} is the demand of job j for resource k, and C_{kst} is the capacity of resource k at site s and time t. Deadline, precedence, data-locality, and replication constraints define workload feasibility. Thermal dynamics constrain temperatures and ramp rates. Power-system contracts constrain peak draw and battery behavior. Reliability requires reserve margins, failure-domain diversity, and recovery objectives. Security requires authenticated control, least privilege, and separation between optimization and protection systems.

These constraints are owned by different institutions. A scheduler cannot interpret a legal data-residency rule as a soft penalty simply because infeasibility is inconvenient. Nor should a service-level objective be treated as physically immutable when it was chosen without knowledge of environmental burden. Constraint review needs owners, provenance, and a route to renegotiate nonessential commitments.

Interfaces are where local optima fail. A job scheduler emits a load profile to a cooling controller; the cooling controller sees thermal state, not business priority. A battery optimizer interacts with utility tariffs and emergency reserve. A procurement model changes the hardware mix assumed by capacity forecasts. Chapter 7's compositional warning applies: guarantees at each component do not automatically compose when signals, time scales, and failure assumptions differ.

Uncertainty and dynamics

Workload arrivals, job durations, renewable output, marginal emissions, weather, water conditions, equipment failures, and prices are uncertain. Forecast errors are correlated: a heat wave can raise cooling load, increase grid carbon intensity, reduce water availability, and raise service demand simultaneously. Independent scenarios understate tail risk.

Expected-value scheduling is appropriate for frequent, recoverable variation. Risk measures such as CVaR from Chapter 6 can protect tail service delay or cost. Robust sets can protect thermal and capacity feasibility against bounded forecast error. Resilience requires more: detecting model failure, shedding noncritical load, transferring state safely, operating locally during network partition, and recovering without data loss.

Use receding-horizon control for operational decisions. At each interval, update workload, thermal state, grid forecasts, and equipment availability; optimize a horizon; implement only the near-term action; and repeat. But constrain policy churn. Excessive migrations, set-point oscillation, or battery cycling can consume the savings. Stability and wear belong in the model.

Long-term planning should use adaptive pathways. Rather than commit today to one capacity forecast, specify triggers: retrofit cooling if water stress or temperature crosses a threshold; add storage if price volatility and carbon correlation reach a range; defer hardware replacement if grid decarbonization reduces its lifecycle benefit. The option to wait has value when choices are irreversible.

Method and computation

The operational problem combines mixed-integer assignment, queueing, thermal dynamics, and uncertainty. A single monolithic solver may be too slow or brittle. Decompose by time scale and interface.

1. A planning model chooses capacity, technology, and carbon-water budgets under scenarios.
2. A day-ahead mixed-integer or convex approximation schedules movable batch work and energy assets.
3. A fast online policy routes requests and corrects forecast error.
4. Local safety controllers maintain thermal and electrical envelopes regardless of economic instructions.

Queueing results provide sanity checks. Little’s law links average work in system, throughput, and time; high utilization creates rapidly increasing delay (Little 1961). The optimizer should not “discover” a schedule that assumes persistent near-total utilization without queue consequences. Tail latency matters because a request can fan out across many services (Dean and Barroso 2013).

Lagrange multipliers can coordinate decomposed sites: a carbon or network-capacity shadow price communicates scarcity without revealing every local variable. That is a transfer of duality from Chapter 4. It preserves marginal trade-off information under the model. It does not confer moral legitimacy on the price or ensure that communities can be compensated for harm.

Evidence and validation

Data-center demand and efficiency must be measured rather than inferred from headline growth, while water accounting must distinguish withdrawal from consumption and direct cooling from electricity-supply effects (Masanet et al. 2020; Ristic, Madani, and Makuch 2015). Carbon claims likewise depend on the emission factor appropriate to the intervention; average and marginal factors are not interchangeable (Elenes et al. 2022).

The evidence plan begins before deployment. Reconstruct at least a year of workload, weather, electricity, water, failures, and maintenance with known data quality. Back-test forecasts without leakage. Validate energy and thermal models across seasons and load regimes. Compare modeled job completion and tail latency with production. Reconcile facility meters with server telemetry and power-usage-effectiveness estimates.

Carbon validation must state whether factors are average, location-based, market-based, or marginal. The choice changes the decision. Water accounting should distinguish withdrawal from consumption and incorporate local scarcity rather than report liters as globally interchangeable. Embodied impacts should show supplier data, allocation assumptions, uncertainty ranges, and treatment of reuse.

Run the optimizer in shadow mode. It produces actions without controlling the system; operators inspect feasibility, predicted benefit, and surprising behavior. Then use staged deployment by workload class and site, with holdbacks where interference is manageable. Stress tests should include heat waves, grid emergencies, network partition, sensor drift, stale carbon data, correlated hardware failure, and malicious input.

Measure absolute impact as well as modeled savings. Did total energy, emissions, and water change after workload growth and rebound? Did latency burdens fall disproportionately on low-paying tenants? Did operators create manual workarounds? A reduction claimed by the optimizer should be reconciled against bills, meters, service outcomes, and independent lifecycle review.

Governance, monitoring, and failure response

Create a decision board with operations, reliability, security, sustainability, finance, tenant representation, and local environmental expertise. This is not consensus theater. Each role owns distinct constraints and has defined veto or escalation authority. Weight changes, boundary changes, and new workload classes require review.

The operational dashboard should show service, carbon, water, cost, reserve margin, forecast error, and distance from validated conditions. Alert when data is stale or a policy relies on an extrapolated model. Preserve a manual safe policy and the capacity to restore it. Security incidents, major model error, thermal instability, inequitable service, or environmental-budget breach trigger rollback.

Publish an impact report that distinguishes avoided modeled impact, measured operational change, contractual accounting, and lifecycle estimates. Communities should receive information relevant to local water, noise, backup generation, and grid consequences, not only a global carbon total. Complaints and local observations are evidence and must enter review.

Transfer ledger

Queueing theory → workload service. Preserves the relation among arrival, work in system, service capacity, and delay. Forgets the social value of different workloads and the contractual history of service targets.

Optimal control → cooling and batteries. Preserves dynamics, state constraints, forecasts, and receding-horizon correction. Forgets operator practice and political rules governing energy use unless represented.

Dual decomposition → fleet coordination. Preserves marginal scarcity signals and separability. Forgets distributional legitimacy; a shadow price is not compensation or consent.

Lifecycle assessment → technology planning. Preserves material and energy flows across stages under declared allocation rules. Forgets unmeasured local harms, rebound, and contested valuation unless separately governed.

Reliability engineering → climate resilience. Preserves failure modes, redundancy, detection, recovery, and common-cause analysis. Forgets which services society considers essential and who bears curtailed service.

The case succeeds only if these losses remain visible. The optimum is not "the greenest schedule." It is a governed policy whose service, climate, water, security, and equity claims remain separately testable.

Implementation economics and tenant contracts

The schedule can be environmentally superior in the model and commercially impossible under existing contracts. Tenant service objectives, reserved capacity, internal transfer prices, renewable claims, and ownership of efficiency savings determine which actions operating teams will accept. Treat contracts and chargeback as mechanism variables. A tenant charged for delay but not carbon will rationally resist deferral even when the fleet benefits.

Pilot a tariff or service class that exposes a small menu: immediate, deadline-flexible, location-flexible, or carbon-water budgeted service, with truthful disclosure and no coercive degradation of the baseline. Estimate actual flexibility rather than assuming every batch job can move. Protect small tenants from a market in which large customers purchase all low-impact capacity.

Capital planning must reconcile organizational and physical depreciation. Hardware may be financially depreciated while still physically useful, or operationally inefficient while embodied impact makes immediate replacement unattractive. Compare keep, repurpose, resell, and replace pathways. Record rebound assumptions and who receives avoided cost.

Finally, test whether optimization displaces operator attention. A policy that requires constant exception handling can transfer environmental benefit into hidden labor and reliability risk. Staff workload, explainability, and incident recovery belong in the measured implementation cost.

Case checklist

- Define operational, planning, and infrastructure time scales separately.
- Distinguish average, marginal, contractual, and lifecycle carbon claims.
- Weight water by local context and report withdrawal and consumption.
- Preserve reliability, security, residency, and essential-service constraints.
- Validate optimized candidates, not only ordinary operating points.
- Monitor absolute impact, rebound, workload fairness, and operator burden.
- Maintain a safe fallback policy and explicit rollback triggers.

Case B · An equitable heat-health program

Decision and purpose

A metropolitan authority expects more frequent and severe heat. It must choose a portfolio of near-term services and long-term changes: heat alerts, outreach, transport, cooling centers, home cooling and weatherization, tree canopy, shade, cool roofs and pavements, worker protections, clinical surveillance, utility support, and emergency staffing. The purpose is not to minimize the city's average temperature. It is to prevent illness and death, reduce unequal exposure and vulnerability, preserve dignity and access, and adapt the urban fabric without creating new burdens.

The decision is difficult because interventions act on different mechanisms and time scales. An alert can change behavior tomorrow but only if people receive, trust, and can act on it. A cooling center can protect during an event but may be unreachable, stigmatizing, inaccessible, or unsafe. Trees provide shade and other benefits but require years, water, maintenance, and protection from displacement effects. Air conditioning reduces indoor exposure while raising electricity demand and potentially emissions. Housing retrofit can address both heat and energy burden, yet landlord-tenant incentives shape adoption.

This is a portfolio and pathway problem, not a ranking of projects by cost per degree. The authority should decide what to do now, what capability to build, which irreversible commitments to avoid, and which observations will trigger later action.

Boundary and affected parties

The physical boundary includes weather, urban heat islands, buildings, vegetation, energy and water systems, transport, and health services. The social boundary includes housing quality, disability, age, chronic illness, medication, outdoor work, income, language, social isolation, carceral settings, homelessness, immigration status, caregiving, and trust in institutions. Exposure is produced by infrastructure and institutions, not weather alone.

Affected parties include residents, workers, students, patients, caregivers, landlords and tenants, employers, utilities, emergency services, community organizations, transit agencies, and neighboring jurisdictions. The groups used in the model must not be treated as homogeneous. "Older adults" can include independent residents with strong networks and isolated residents without cooling. Neighborhood averages can hide dangerous apartments, street segments, or work sites.

The boundary must include displacement and rebound. A greening project that raises property values can displace the people whose risk justified it. A home-cooling subsidy can lower exposure and raise energy debt if utility support is absent. A center that concentrates services may reduce travel for the agency and increase travel for residents. Distributional effects are part of the intervention, not side notes.

Variables and alternatives

Let x_i denote the scale or selection of intervention i, and let y_{int} allocate service i to neighborhood or group n in period t. Some decisions are continuous, such as outreach hours or subsidy levels; others are binary, such as opening a center or adopting a work-rest rule. Spatial variables locate shade, trees, transport, and facilities. Pathway variables determine when a later action is triggered.

Alternatives should be constructed with communities and operators before optimization. A model can only choose among represented actions. Residents may identify evening transport, trusted-language outreach, pharmacy check-ins, pet accommodation, drinking-water access, or tenant enforcement as decisive options absent from an engineering inventory. Clinicians may identify medications or conditions requiring special protocols. Workers may reveal that a formal rest rule is unenforceable without wage protection.

Include "maintain existing services," "repair reliability first," and "do not build" alternatives. A new cooling center should compete with keeping libraries, clinics, and community facilities open longer. A sensor network should compete with funding outreach and enforcement. The optimizer should not inherit a capital-project bias.

Objective architecture

For group g, scenario ξ, and portfolio x, let $R_g(x,\xi)$ be residual risk of a defined adverse health outcome over a stated period. A pure utilitarian objective would minimize population-weighted expected risk:

$$\min_x \ \textstyle\sum_g N_g \ \mathbb{E}\big[R_g(x,\xi)\big]. \tag{112}$$

That objective can prefer many small improvements for already protected populations over urgent protection for a smaller high-risk group. The field-guide formulation adds group thresholds and a tail-risk term. Let X_H contain accessible portfolios satisfying $C(x) \le B$, $\mathbb{E}\big[R_g(x,\xi)\big] \le \tau_g$ for every priority group, and $\Pr\big(R_g(x,\xi) > \bar{r}_g\big) \le \varepsilon_g$ for every protected group. Then

$$\min_{x\in X_H} \Big(\textstyle\sum_g N_g \ \mathbb{E}\big[R_g(x,\xi)\big] + \lambda\, \mathrm{CVaR}_\alpha\big(L(x,\xi)\big)\Big). \tag{113}$$

Equation (113) combines average benefit, tail protection, group-specific limits, budget, and accessibility. The thresholds are not generated by the solver. They come from public-health evidence, legal duties, capability judgments, and deliberation. Sen's and Nussbaum's capability approaches help explain why access to bodily health, mobility, affiliation, and control cannot be inferred from aggregate welfare alone (Sen 1985; Nussbaum 2011).

Some obligations should remain outside probabilistic trade. An accessible evacuation route, nondiscriminatory service, safe worker practice, and emergency duty cannot be waived because the model assigns low event probability. Similarly, cultural and neighborhood continuity may require procedural protections that a risk score cannot encode.

Constraints, mechanisms, and interaction

The portfolio faces budgets, workforce, land, water, electrical capacity, procurement, seasonal timing, and maintenance constraints. It also faces behavioral and institutional mechanisms. Outreach requires a trusted messenger and current contact path. Cooling requires reliable power and an affordable bill. Shade requires a reachable route at the time of exposure. Worker protection requires enforcement and protection against lost income. Clinical surveillance requires privacy and a plan for response.

Interventions interact. Weatherization can improve cooling efficiency; tree canopy can reduce building load; transit can make centers accessible; utility shutoff protection can make equipment usable; clinical outreach can target people identified through nonintrusive community networks. Other interactions are adverse. Trees can conflict with water constraints or power lines. Reflective surfaces can increase glare or shift radiant load. Added cooling demand can stress the grid during the event.

Represent these mechanisms explicitly where evidence permits. When it does not, preserve them in scenario narratives and feasibility review. A coefficient is not more rigorous than a clearly stated unknown.

Uncertainty and adaptive pathways

Uncertainty includes climate trajectories, event timing, humidity, smoke, power failures, exposure-response relationships, intervention uptake, migration, land-use change, and future budgets. Historical averages are a weak guide under a changing climate (Intergovernmental Panel on Climate Change 2022b). Yet maximizing against an unconstrained worst case can direct all resources to an implausible scenario. Use a plural strategy: stochastic scenarios for frequency, robust constraints for critical service under forecast error, and adaptive pathways for long-lived investments.

An adaptive pathway might fund immediate outreach, facility-hours, worker rules, home audits, and grid-resilience measures; begin canopy and retrofit in priority areas; and reserve land or design options for later centers. Trigger additional cooling capacity when indoor-temperature surveillance, outage frequency, or health outcomes cross predeclared thresholds. Review triggers annually because observation systems and populations change.

The authority should test compound events: heat plus wildfire smoke, heat plus outage, heat plus transit disruption, and repeated hot nights. Resilience is not merely low expected risk; it is the ability to recognize failure, reach isolated people, maintain safe spaces, restore service, and learn after the event.

Method and computation

The planning problem is a mixed-integer, spatial, multiobjective portfolio under uncertainty. Use geospatial preprocessing to estimate hazard and accessibility; epidemiologic models to estimate conditional health response; building and energy models for indoor exposure and grid demand; network models for travel and outreach; and optimization for portfolio comparison. Keep the components separate enough that their assumptions can be challenged.

Start with dominance screening and budget scenarios. Generate a Pareto set across total expected risk, priority-group risk, tail loss, cost, time to benefit, and maintenance burden. Use robust or distributionally robust analysis where parameter uncertainty is material (Ben-Tal, El Ghaoui, and Nemirovski 2009; Shapiro, Dentcheva, and Ruszczyński 2021). Then take a small set of diverse portfolios into deliberation. Do not ask a community board to choose among thousands of solver outputs.

Shadow prices can show where budget, staff, or grid capacity binds. They are questions for investigation, not automatic funding rules. A high shadow value on outreach capacity may justify training and partnerships. A high value on a group-risk threshold may reveal either an expensive commitment or a misspecified intervention set. Extreme values are model diagnostics.

Evidence and validation

The intervention portfolio follows the current WHO heat-health action-plan framework: governance, warning, identification of populations at increased risk, communication, health-system resilience, exposure reduction, surveillance, and monitoring and learning (World Health Organization Regional Office for Europe 2026). Local thresholds and delivery mechanisms still require local epidemiology, service data, and community validation.

Build an evidence chain for each mechanism. Weather and land-surface data support hazard estimates. Housing audits and indoor sensors support exposure estimates, with privacy and sampling protections. Health records support outcome models but may undercount people with poor access to care. Surveys and community organizations support access and trust claims. Facility logs support capacity and use. Utility data support outage and energy-burden analysis.

Validate spatially and temporally. A model calibrated on citywide mortality may not predict neighborhood ambulance demand. A center-access model should be walked and tested by people with mobility, sensory, cognitive, language, and caregiving constraints. An alert should be tested for comprehension and action, not only delivery. Simulated travel time must account for the actual time of day, heat exposure during the trip, and service disruption.

Use pilot deployments where feasible, but do not withhold established protection merely to create a clean experiment. Phased implementation, matched comparison, interrupted time series, and process evidence may be more ethical than individual randomization. Predefine outcomes and analyze distribution, not just average attendance or temperature.

After every event, reconcile forecast, operations, and outcomes. Investigate who did not receive service, which facilities failed, where staff improvised, and which data arrived too late. Near misses and community testimony should update the model even when statistical power is limited.

Governance and accountability

Governance should join public health, emergency management, housing, planning, labor, disability access, utilities, clinicians, community organizations, and residents from priority areas. The authority must publish how priority groups, thresholds, and budgets were chosen. Residents need a route to correct data, challenge facility plans, and report inaccessible or unsafe services.

Create role-specific authority. Public-health officials can trigger emergency actions; access officers can reject inaccessible plans; labor regulators enforce protections; utilities own restoration protocols; community partners can activate outreach and report loss of trust. No weighted vote should permit one role to erase another's statutory duty.

Monitor service reach, indoor exposure, health outcomes, outages, energy burden, worker compliance, center conditions, and displacement indicators. Roll back or redesign interventions that create heat, glare, surveillance, policing, unaffordable bills, or exclusion. Long-term projects need maintenance covenants and anti-displacement measures; planting a tree is not the same as sustaining shade.

Transfer ledger

Climate hazard modeling → public-health planning. Preserves scenario-conditioned heat intensity and frequency. Forgets individual exposure, social protection, and institutional trust.

Reliability engineering → emergency service. Preserves capacity, redundancy, failure domains, and recovery. Forgets why some people cannot or will not use the nominal service.

Portfolio optimization → intervention mix. Preserves budget coupling, interaction, timing, and trade-offs. Forgets unrepresented actions and the legitimacy of priorities.

Control and adaptive pathways → long-horizon adaptation. Preserves observation, triggers, staged commitment, and revision. Forgets that public institutions may fail to act when a trigger occurs unless authority and funding are secured.

Capability ethics → feasibility. Preserves the distinction between formal availability and real ability to remain safe. Resists reduction to preferences, but requires deliberative judgments the optimizer cannot supply.

The mapping changes the optimum because equity is not applied after optimization. It changes feasibility, adds group constraints, changes the value of access interventions, and favors portfolios with trusted delivery and recovery capability.

A distributional audit in space and time

For each candidate portfolio, produce maps and tables of baseline exposure, intervention reach, residual risk, travel or paperwork burden, energy cost, outage exposure, and time to benefit. Show uncertainty and population counts. Do not publish fine-grained health or vulnerability data that can identify individuals; use privacy-preserving aggregation and community review.

Compare at least four lenses: total benefit, worst residual risk, change for priority groups, and people made worse off. A citywide improvement can coexist with one neighborhood receiving heat, glare, policing, construction burden, or displacement. Report these mechanisms separately instead of subtracting them from an aggregate benefit.

Time matters. Emergency services can protect this season; canopy and housing take years. A portfolio that is equitable at maturity can leave a decade-long protection gap. Add interim thresholds and budgets, and monitor whether long-term projects are maintained. If a priority community waits longer for benefit, compensate with immediate measures and decision authority.

The audit should be discussed with people who can verify whether mapped access is real. A route that crosses a dangerous road, a center that excludes pets, or an outreach list that omits undocumented residents can invalidate a nominal allocation.

Case checklist

- Model exposure, vulnerability, and ability to act—not temperature alone.
- Construct interventions with communities and frontline operators.
- Use priority-group and tail-risk constraints alongside total benefit.
- Test compound events, outages, smoke, access, and displacement.
- Validate alerts for action and facilities for real accessibility.
- Publish thresholds, authority, maintenance, and appeal routes.
- Treat after-action testimony and near misses as evidence.

Case C · Closed-loop insulin delivery as adaptive care

Decision and purpose

A closed-loop insulin system repeatedly estimates metabolic state and recommends or delivers insulin. Its purpose is not to minimize average glucose in isolation. It is to help a particular person remain in a clinically appropriate range while sharply limiting dangerous low glucose, avoiding excessive dosing and oscillation, supporting sleep and daily activity, reducing cognitive burden, and preserving the person’s ability to understand, interrupt, and adapt the system with a clinical team.

The decision is safety-critical and asymmetric. Too little insulin can lead to sustained high glucose and acute or long-term harm; too much can produce a rapid emergency. The same measured glucose can require different action depending on trend, recent dose, meal absorption, exercise, illness, stress, alcohol, hormones, sensor quality, and individual physiology. The controller acts under delay: subcutaneous insulin takes time, and continuous glucose sensing measures an interstitial signal with lag and noise.

This case is educational, not a dosing design. Device development and care require current clinical, engineering, cybersecurity, human-factors, and regulatory expertise. The optimization grammar clarifies responsibilities; it does not replace them.

Boundary and affected parties

The system boundary contains the person, sensor, infusion set and pump, insulin, controller, user interface, mobile or embedded software, alarms, charging and consumables, clinician configuration, data services, update process, caregiver access where authorized, and emergency fallback. It also contains daily life: meals are incompletely announced, exercise can be spontaneous, sleep limits attention, and connectivity or supplies can fail.

Affected parties include the user, caregivers selected by the user, clinicians, device and software teams, pharmacists and suppliers, emergency personnel, regulators, and security responders. Their authority is not symmetric. The person living with the system owns goals and burden trade-offs within clinical safety. Clinicians define and review therapeutic parameters. Manufacturers are responsible for device behavior and evidence. Remote service providers should not acquire decision authority merely because they process data.

Do not treat “human override” as a complete safety layer. An override is useful only when the person is awake, informed, physically able, has supplies, understands the alarm, and can act before harm. Automatic fallback must remain safe under realistic human absence.

State, variables, and alternatives

Let G_t be glucose-related state at time t, u_t insulin delivery, d_t known or estimated disturbance such as meal or exercise, and I_t insulin-on-board state. A model may include compartments for glucose appearance and insulin action. The controller estimates hidden state from sensor history, commands, and contextual inputs.

Decision variables include basal modulation, correction micro-boluses where supported, temporary targets, alarm timing, and adaptation parameters. Design variables include sampling, horizon, model structure, constraints, fault detection, interface, and fallback mode. Clinical variables include individualized target range, insulin sensitivity, carbohydrate ratio, active-insulin time, and conditions requiring manual operation.

Alternatives include standard pump therapy, sensor-augmented therapy, predictive low-glucose suspend, hybrid closed loop with meal announcements, and more automated control under specified conditions. Comparison must include the additional interactions, alarms, training, supplies, and failure modes—not only a simulation curve.

Objective architecture

At time t, a model-predictive controller forecasts a horizon H and selects an admissible dose sequence. Define $\phi(G) = w_h\left(G - G_{\text{high}}\right)_+^2 + w_l(G_{\text{low}} - G)_+^2$. Let $\mathcal{U}_t$ contain dose sequences satisfying $0 \le u_{t+k} \le \bar{u}_t$, $I_{t+k} \le \bar{I}_t$, and the state transition $(G_{t+k}, I_{t+k}) = F_{\theta_t}(G_{t+k-1}, I_{t+k-1}, u_{t+k-1}, d_{t+k-1})$. A stylized objective is

$$\min_{u_{t:t+H-1} \in \mathcal{U}_t} \sum_{k=1}^{H} [\phi(G_{t+k}) + r\,(u_{t+k-1} - u_{t+k-2})^2]. \tag{114}$$

The low-glucose weight w_l is much larger than the high-glucose weight over the near horizon because harms are asymmetric. The positive part ensures that only violations beyond the range are penalized in these terms. Dose and insulin-on-board bounds supply safety constraints. Equation (114) illustrates receding-horizon structure (Rawlings, Mayne, and Diehl 2017); a clinical implementation requires richer models, fault handling, verification, and evidence.

A scalar cost is insufficient. Safety requirements should specify maximum delivery, rate of change, response to missing or implausible sensor data, behavior after manual dose, and safe degradation. Human factors add constraints on alarm burden, comprehensibility, and mode visibility. Privacy and security add collection, authentication, update, and access controls. Performance is then evaluated across time in clinically meaningful ranges, events, variability, burden, and subgroup or individual response.

Uncertainty and adaptation

The controller faces parameter uncertainty, state-estimation error, sensor bias and delay, infusion failure, unannounced meals, exercise, illness, and behavioral variation. These are not identically distributed noise. A blocked infusion set changes the plant; a compressed sensor can create structured error; exercise changes physiology; a missed meal announcement changes observability.

Robust design should separate ordinary variability from faults. The controller can adapt within validated parameter bounds while a fault monitor tests sensor consistency, delivery response, and communication. If confidence falls, it transitions to a safe mode, informs the user clearly, and avoids compounding uncertain action. Adaptation must not silently drift outside the evidence envelope.

Personalization is valuable and potentially confounded. If the algorithm changes at the same time as the person's routine, a fitted improvement can be spurious. Use bounded updates, minimum data requirements, conservative priors, change detection, and clinical review. Preserve a versioned history of parameters, triggers, and outcomes so a deterioration can be diagnosed and reversed.

Method and computation

The formal problem combines nonlinear dynamics, partial observation, constraints, and hybrid modes. Model predictive control provides a disciplined architecture because it forecasts, handles constraints, and repeats after new measurement (Rawlings, Mayne, and Diehl 2017). State estimation filters sensor data and propagates uncertainty. Robust or stochastic variants can test parameter and disturbance scenarios. A safety supervisor can enforce invariants independent of the performance optimizer.

Computation must meet deterministic timing and resource limits. Solver failure, numerical ill-conditioning, overflow, time synchronization, battery depletion, and update interruption are system hazards. Verification should cover not only mathematical source but generated code, units, data types, configuration, and hardware integration. Every control output should be traceable to valid inputs, current mode, and a tested software version.

The development process uses simulation, hardware-in-the-loop, controlled studies, and monitored real-world evidence. A high-fidelity simulator can generate rare scenarios, but an optimizer can exploit physiological-model error. Include diverse virtual subjects and challenge cases, then verify behavior with independent models and domain review.

Evidence and validation

Clinical evidence must compare the complete closed-loop system with an appropriate control and report both target-range and safety outcomes. A six-month randomized multicenter trial found improved time in range and illustrates why infusion-set failure, software configuration, and adverse-event monitoring belong in the evidence plan (Brown et al. 2019).

Validation proceeds from component to integrated claim. Test sensor behavior under lag, drift, missing data, compression, and interfering conditions. Test pump delivery, occlusion detection, and command limits. Verify controller code against the formal design. Use hardware-in-the-loop to test timing, communications, battery, and faults. Challenge the complete system with meals, exercise, sleep, illness, sensor replacement, manual dosing, and connectivity loss.

Clinical evaluation should report more than a mean. Show time in relevant ranges, frequency and duration of low and high events, variability, rescue interventions, alarm burden, discontinuation, sleep, quality of life, and differential performance. Decision-curve thinking is useful: the clinical value of a threshold depends on the relative consequence of missing and triggering action, not discrimination alone (Vickers and Elkin 2006).

External validity requires diversity in age, physiology, schedule, language, disability, device experience, access to supplies, and care setting. Average benefit can coexist with an unsafe subgroup or an unusable interface. Missingness and discontinuation are outcomes, not nuisance rows to delete.

Post-deployment monitoring should distinguish device fault, model mismatch, use difficulty, supply interruption, and clinical change. Near misses, overrides, manual-mode episodes, complaints, and cybersecurity signals enter the evidence chain. Updates require regression testing and a reasoned statement of whether existing evidence transfers to the changed system.

Governance and failure response

The person using the device must know current mode, recent delivery, data quality, active alerts, and how to obtain help. Interfaces should avoid silent automation and alarm floods. Training includes ordinary use, sick-day or exercise guidance as clinically directed, sensor or infusion failure, manual fallback, and emergency response.

Clinicians need interpretable summaries and the ability to review parameter history, not an unmanageable stream of raw data. Manufacturers need safety monitoring, secure disclosure, update control, and prompt communication. Regulators require evidence matched to the intended population and claims; adaptive software must remain under lifecycle control (U.S. Food and Drug Administration and Health Canada and Medicines and Healthcare products Regulatory Agency 2021). Data access should be purpose-limited, encrypted, logged, and revocable where possible.

Rollback operates at several levels: revert a model or software update; restore prior parameters; enter a validated safe mode; or discontinue automation with a clinically established alternative. A failure response must not depend on cloud connectivity. Accountability remains with identifiable organizations and professionals, not “the algorithm.”

Transfer ledger

Model predictive control → insulin dosing. Preserves constrained receding-horizon action and correction after observation. Forgets physiological mechanisms omitted from the model and the lived cost of alarms, attention, and trust.

Robust optimization → safety margin. Preserves specified parameter and disturbance uncertainty. Forgets novel faults outside the set; resilience and fallback remain necessary.

Fault-tolerant control → safe degradation. Preserves detection, mode transition, invariants, and recovery. Forgets whether a person can understand and manage the degraded mode.

Clinical decision analysis → threshold choice. Preserves asymmetric consequences and the dependence of action on benefit-harm trade-offs (Pauker and Kassirer 1980). Forgets that preferences and circumstances vary and must be revisited.

Software assurance → therapeutic evidence. Preserves traceability from requirement through code and test. It does not prove clinical benefit; engineering and clinical evidence compose only when their interface assumptions hold.

The core translation is from “control a plant” to “support a person.” The dynamic equations transfer. Ownership of purpose does not. The controller remains a tool inside care, daily life, and a regulated safety system.

Human-factors and alarm workload

Alarm design is a constrained communication problem. The system must convey urgency, cause where known, immediate action, and confidence without overwhelming the person. Alarm frequency affects response; missed or repeated nonactionable alarms change behavior. Evaluate comprehension and action under sleep, exercise, stress, sensory difference, and competing tasks.

Modes should be perceptible. The interface must distinguish automated, limited, fallback, suspended, and manual operation, and explain what delivery continues. A transition should not rely on a transient notification. Confirm critical actions and avoid terminology that implies certainty the system lacks.

Caregiver or clinician sharing should be chosen and revocable by the user within safety and legal requirements. Escalation delays, duplicate alarms, and unclear responsibility can create diffusion rather than protection. Test the complete route from device signal to effective assistance.

Usability evidence should include errors of commission and omission, recovery time, training retention, and burden over weeks—not only success in a supervised session. A system that performs well while users remain highly attentive may fail after automation appropriately reduces attention. Human reliability assumptions must match ordinary life.

The final design record should connect every safety claim to a test and every user-facing mode to training and fallback. It should preserve alternatives rejected for burden as well as those rejected for control performance. When a person's goals or circumstances change, clinicians and the user need enough history to distinguish physiological change from software, parameter, sensor, or behavior change. That traceability is part of continuity of care.

Reviewers should also inspect the counterfactual "automation unavailable for one week." Supplies, instructions, clinical contacts, and skills must support safe continuity. If ordinary care has become dependent on a remote service, proprietary account, or scarce consumable, continuity risk belongs in the system claim. Procurement and support are therefore therapeutic interfaces, not administrative details.

The same review should cover travel, hospitalization, school or work, and emergency handoff, because the device crosses organizations whose records and responsibilities do not automatically compose.

Test the handoff with realistic records, time pressure, and an unfamiliar responder.

Document any information that failed to travel and revise the emergency interface before wider use.

Case checklist

- Define safety asymmetry, individualized purpose, and validated operating conditions.
- Model delays, insulin on board, faults, unannounced disturbances, and fallback.
- Keep safety invariants separate from performance optimization.
- Validate component, code, hardware, integrated behavior, and clinical outcomes.
- Report range, events, variability, burden, discontinuation, and differential effects.
- Version adaptations and preserve a path to prior software and parameters.
- Ensure the user has mode visibility, authority, supplies, and an offline fallback.

Comparative synthesis: what feedback means in three domains

The three cases all use feedback and uncertainty, but their state, actuation, and failure semantics differ.

In the data center, state is largely instrumented equipment and workload; action is scheduling, set point, or capacity; many mistakes are economically recoverable, while outages, security breaches, and environmental limits create harder boundaries. Feedback can run rapidly, and aggregate data is often meaningful.

In heat-health policy, state is distributed across weather, buildings, institutions, and people. Observation is incomplete and shaped by access to care. Action includes public service, infrastructure, law, and trusted communication. Feedback is slow and political: a measured uptake rate reflects both intervention and institution. A missed person is not a residual to be averaged away.

In insulin delivery, state is an individual physiological estimate; action directly changes the body; delay and asymmetry are acute. Feedback is fast but noisy, and safe fallback must remain personal and immediate. Population evidence informs but does not replace individual adaptation.

The shared formal loop—observe, estimate, act, measure—therefore composes with three different constitutions. The data-center constitution protects service, security, resource limits, and communities. The public-health constitution protects group capability, access, and public duty. The care constitution protects bodily safety, autonomy, and clinical responsibility.

The cases also show three uses of robustness. Fleet scheduling uses uncertainty sets and reserve to maintain feasibility. Heat adaptation uses scenarios and pathways because distributions and institutions change. Insulin control uses bounded adaptation plus fault detection because novel faults cannot be treated as ordinary noise. Calling all three “robust optimization” would erase the design differences.

Finally, their evidence loops have different stopping rules. A cloud policy can expand after shadow operation and staged load. A public-health intervention may need action before randomized evidence, with protective baselines and after-event learning. A medical controller requires progressively stronger component, integrated, clinical, and lifecycle evidence. The optimization grammar is stable; the warrant is typed by consequence.

Boundary perturbation exercise

The generative test is to perturb each boundary and observe whether the recommended policy changes for a legitimate reason.

For the data center, first optimize site electricity alone. Then add marginal grid response, water scarcity, embodied hardware, tenant fairness, and community exposure one layer at a time. A schedule that initially moves work toward a low-price site may reverse after water or grid correlation enters. A replacement plan may reverse after embodied impact and reuse enter. Record which decision changes and which layer caused it. This is not sensitivity around a fixed model; it is sensitivity to ontology.

For heat-health, begin with outdoor temperature and citywide health risk. Add indoor exposure, power failure, mobility, language, work, housing tenure, and displacement. The preferred portfolio should move from broad temperature reduction toward trusted delivery, home protection, worker rules, and group thresholds when those mechanisms are material. If it does not, inspect whether the new variables affect feasibility or merely sit in an unused report.

For insulin delivery, begin with a nominal physiological model and mean tracking. Add asymmetric low-glucose harm, delays, insulin on board, sensor faults, exercise, alarm burden, user-selected modes, and offline fallback. The controller should become more conservative under uncertainty, but not uniformly inactive. If every added safeguard is represented as a penalty, the optimizer can trade it away. If every uncertainty produces worst-case dosing, usefulness disappears. Typed constraints and mode logic are the structural response.

This exercise supports boundary sufficiency, not maximal scope. A model that includes everything becomes untestable. Retain an element when it can change action, validate a constraint, explain a failure, or assign authority. Keep other consequences in monitoring and deliberation. Document excluded layers with the condition that would bring them inside.

Comparing evidence without ranking disciplines

The cases use bench measurement, telemetry, epidemiology, simulation, trials, interviews, participatory mapping, human-factors tests, and post-event review. It would be a mistake to rank these on one evidence ladder. Each has a claim type.

Bench and hardware tests support component behavior under controlled conditions. Telemetry supports operational distributions and can be endogenous to policy. Epidemiology supports population associations and causal claims under specified designs. Simulation supports conditional counterfactuals. Clinical study supports benefit and harm for a defined intervention and population. Interviews and participatory work support mechanism, meaning, access, and acceptability claims often absent from recorded variables. After-action review supports reconstruction of a coupled failure.

Composition requires a bridge. A heat model plus an exposure-response curve plus a facility-access model supports a conditional risk estimate only if populations, spatial resolution, behavior, and uncertainty align. A verified controller plus tested pump plus clinical model supports safety only if timing, units, failure modes, and intended use align. Evidence strength is lost at interfaces when bridge assumptions are untested.

The field guide therefore asks for a claim-evidence graph, not a total score. Nodes are claims; edges state inference; sources carry provenance; defeaters name observations that would break the bridge. This representation allows different disciplines to contribute without surrendering their validation standards.

Field checklist

- Distinguish average, marginal, contractual, and lifecycle accounting before comparing infrastructure alternatives.

- Link heat warnings to funded actions, accountable roles, accessible communication, and distributional monitoring.
- Treat closed-loop medical control as a clinical service with alarms, fallback, training, cybersecurity, and adverse-event review.
- Validate feedback at the time scale on which each domain can safely observe and act.
- Preserve a baseline, rollback route, and authority to suspend automation.
- Record what each transfer preserves and what it forgets.

Translation lens for Chapter 18. Infrastructure and health both use feedback, robustness, and staged decisions, but their invariants differ. A data center preserves service and environmental commitments; a heat program preserves capabilities and public duties; an insulin system preserves safety and personal agency. The shared grammar coordinates evidence without making these purposes interchangeable.

Chapter 19 · Worked translations II: knowledge, creativity, fabrication, and climate

Case D · A public library collection and discovery system

Decision and purpose

A public library must allocate a constrained acquisition, licensing, preservation, space, cataloging, programming, and outreach budget. It also wants to improve discovery across shelves, catalogs, and recommendations. The tempting objective is circulation. The actual public purpose is plural: enable informed agency, learning, research, recreation, encounter, cultural memory, local authorship, minority-language use, and access for people underserved by commercial markets.

The institution decides what to acquire, license, retain, relocate, digitize, preserve, display, recommend, and remove; which metadata to create; where to invest staff attention; and how to support physical and digital access. These choices act over different horizons. A display changes this month's discovery. A license may expire. Discarding a local archive can be irreversible. Cataloging creates infrastructure whose value appears across future uses.

The library is not a retailer with public funding. Demand matters because unused collections cannot serve, but present demand is partly caused by past acquisition, language, opening hours, interfaces, education, marketing, transport, disability access, and the recommendation system itself. An optimization that learns only from recorded use will reproduce the exclusions that shaped the record.

Boundary and affected parties

The boundary contains patrons and nonusers, physical branches, mobile and outreach services, staff, collections, publishers and vendors, digital platforms, schools and universities, local archives, community organizations, privacy law, storage and preservation, and the urban or rural transport context. It includes the time needed to browse and the expertise needed to catalog, curate, mediate, and teach.

Affected groups include children, students, researchers, job seekers, migrants, linguistic minorities, disabled patrons, older adults, incarcerated or institutionalized people where served, local creators, people without reliable internet or private space, and future residents who cannot express current demand. Staff are not a generic capacity number: subject knowledge, language, community trust, preservation skill, and accessibility expertise are distinct resources.

Digital borrowing does not erase boundaries. License terms can limit simultaneous use, preservation, privacy, accessibility, and transfer between institutions. A commercial recommendation service can observe reading interests that the library would otherwise protect. Vendor concentration can make short-term convenience a long-term lock-in decision.

Variables and alternatives

Let $x_i \in \{0,1,2,\ldots\}$ be copies or licenses of item i, r_i a retention decision, s_{ib} allocation to branch b, and h_{ikt} exposure of item i through interface or display position k at time t. Additional decisions allocate cataloging, conservation, digitization, programming, and outreach hours.

Item is not always the correct unit. A collection can have value through coverage of a language, subject, historical period, format, reading level, or community record. Series and reference works have complementarities. Archives have provenance and order that should not be decomposed into independent documents. A licensing package can bundle desired and unwanted content. The representation must match the intellectual and material object.

Generate alternatives with librarians and communities. Options include cooperative purchasing, interlibrary lending, rotating deposits, shared preservation, demand-driven acquisition with safeguards, targeted retrospective cataloging, multilingual metadata, sensory-accessible formats, outreach collections, and repair of opening-hour or transport barriers. “Buy or do not buy” is too small a feasible set.

Objective architecture

For collection alternative x, define measures for expected use $U(x)$, subject breadth $B(x)$, linguistic and cultural representation $L(x)$, educational and research support $Q(x)$, preservation value $P(x)$, accessibility $A(x)$, serendipitous exposure $S(x)$, and lifecycle cost $C(x)$. The vector is a deliberation surface, not an assertion that these meanings share a natural scale.

A portfolio model can maximize a declared scalarization while enforcing floors. Let X_L contain licensed portfolios satisfying $L_g(x) \geq \ell_g$ for protected language or community areas, $A_f(x) \geq a_f$ for required accessible formats, $P_j(x) = 1$ for preservation commitments, and $\sum_i c_i\, x_i + C_{\text{staff}}(x) \leq B_{max}$. Then

$$\max_{x \in X_L} \left[U(x) + \lambda_B B(x) + \lambda_Q Q(x) + \lambda_S S(x) - \lambda_C C(x) \right]. \tag{115}$$

Equation (115) places representation, access, preservation, licensing, and staff cost into feasibility. It leaves several choices political and professional: which floors are adequate, how preservation is designated, and how breadth is described. A Pareto set across use, breadth, representation, preservation, cost, and staff burden should accompany any chosen scalarization.

Circulation should be decomposed. A renewal, reference consultation, in-library use, digital access, program use, and archival visit mean different things. Rare use can have high value. Nonuse can reflect invisibility or inaccessibility rather than irrelevance. A book that supports one life-changing inquiry is not well described by a low count; neither is a heavily borrowed but replaceable bestseller automatically a preservation priority.

Discovery as intervention

Search and recommendation determine which collection is encountered. A ranking model may estimate relevance from query, metadata, availability, popularity, and user context. Privacy and intellectual freedom argue against collecting more history than necessary. The system should support anonymous or minimally identified exploration, clear control of personalization, deletion, and a nonpersonalized route.

Popularity creates feedback. Highly ranked items receive exposure, producing more use and still higher rank. Exploration can counter this, but random novelty is not serendipity. Curatorial pathways, thematic juxtaposition, local context, and staff knowledge produce meaningful encounter. The interface can allocate some exposure to underdiscovered material while making the rationale legible.

Fairness is not equal exposure for every item. It may mean language access, representation of community publishing, protection from commercial concentration, or opportunity for a relevant but low-history item to be seen. Evaluate exposure conditional on relevance and collection purpose. Audit whether new or minority-language items are systematically ranked below items with richer metadata.

Uncertainty and dynamics

Demand forecasts are uncertain and endogenous. Curricula change, communities migrate, authors emerge, crises alter information needs, and licenses change. Long-term value is especially uncertain for archives and scholarship. A robust collection retains option value: adaptable space, interoperable metadata, preservation copies, vendor diversity, and cooperative networks.

Model acquisitions over a rolling horizon but protect long-lived commitments from short-term oscillation. If yearly circulation drives automatic weeding, the system can erase a subject before a community program has time to build use. Use minimum review periods, professional assessment, and reversible relocation before irreversible disposal. Scenario analysis should cover budget cuts, branch closure, platform loss, sudden demand, and disaster.

The discovery policy should reserve controlled exploration and measure long-term diversity, not only immediate clicks or loans. Yet experiments must protect privacy and avoid manipulating sensitive inquiry. Treat query logs as potentially sensitive records and restrict secondary use (Dwork and Roth 2014).

Method and computation

The collection problem combines knapsack and set-coverage structure, submodular benefits, facility allocation, and multiobjective choice. Greedy approximation can be useful when coverage exhibits diminishing returns; mixed-integer programming handles bundles, floors, and branch constraints; robust scenarios protect uncertain cost or demand; qualitative review handles significance not captured by data.

The discovery system combines information retrieval, exposure allocation, and human curation. Use offline relevance judgments, privacy-preserving aggregate logs, accessibility tests, and controlled pilots. Maintain independent channels: catalog search, curated lists, browsing maps, staff assistance, and recommendation. Redundancy is epistemic—no single representation should determine discoverability.

Shadow prices can reveal scarce cataloging skill or the cost of a representation floor. They do not tell the library whether the commitment is worthwhile. A high cost can justify capacity investment rather than relaxation. Chapter 4's dual variables remain model-relative.

Evidence and validation

The evidence base includes the public library's institutional mission, not only circulation data. The IFLA–UNESCO manifesto treats the library as infrastructure for education, culture, information, inclusion, and participation, while the Library Bill of Rights makes viewpoint diversity and resistance to censorship explicit constraints (International Federation of Library Associations and Institutions and UNESCO 2022; American Library Association 2019).

Use multiple evidence streams: circulation and hold queues; in-library and reference use; program participation; qualitative inquiry; gap analysis against curricula and community needs; language and format coverage; preservation condition; accessibility testing; and surveys of users and nonusers. Separate absence of demand from absence of access.

Validate the collection model through counterfactual review. Would it have retained materials later judged indispensable? Which communities and subjects lose under alternative weights? Does it recommend packages whose staff or license burden was omitted? Ask subject and community reviewers to inspect both selected and rejected items.

Validate discovery with task-based studies. Can patrons find a known item, explore an unfamiliar subject, distinguish available formats, protect privacy, understand rankings, and recover from spelling or vocabulary differences? Measure exposure and satisfaction across language, accessibility mode, branch, and query type. Automatic relevance metrics are necessary and not sufficient.

Monitor effects after implementation: concentration of loans, diversity of exposure, use of new and local materials, privacy incidents, staff workload, vendor dependence, complaints, and patterns of failed searches. A failed search is evidence about metadata or collection gaps, not merely a zero reward.

Governance and refusal

Collection policy should be public, with defined roles for librarians, preservation specialists, community advisory groups, governing boards, and legal counsel. Staff must be able to override an algorithm with a recorded reason, and repeated overrides should trigger model review rather than staff discipline. Patrons need routes to request, challenge, correct, and appeal.

Refusal protects the institution. The library may refuse engagement optimization, sale of reading data, automatic removal based only on use, opaque vendor ranking, or a recommendation objective that narrows intellectual exposure. It may preserve controversial material while governing display and access according to law and professional ethics. Optimization cannot settle a challenge to intellectual freedom; it can make consequences and consistency visible.

Transfer ledger

Portfolio optimization → collection development. Preserves budget coupling, complementarities, floors, and opportunity cost. Forgets intellectual significance and future community value unless professionals and communities supply them.

Set coverage → breadth. Preserves whether represented items cover declared topics, languages, or formats. Forgets depth, quality, relation among works, and the politics of the taxonomy.

Recommendation → discovery. Preserves relevance and exposure under a chosen representation. Forgets private reasons for inquiry and risks converting past visibility into future canon.

Preservation engineering → cultural memory. Preserves condition, redundancy, media risk, and recovery planning. Forgets why an object matters and who has authority over culturally sensitive material.

Capability theory → library access. Preserves the distinction between nominal availability and real ability to use a resource. Requires context about disability, language, time, transport, safety, and trust.

The system is successful when it enlarges informed agency without turning reading into surveillance or culture into inventory turnover.

Irreversibility and the archival exception

Routine collection items, local ephemera, special collections, and community archives have different reversibility. A common replacement-cost field is misleading when an object is unique, its market disappears, or its provenance depends on an intact collection. Create retention classes and require specialist or community authority before irreversible action.

Digitization is not automatic preservation equivalence. It can improve access while losing material features, scale, marginalia, binding, context, or restrictions defined by donors and communities. Digital files also require format migration, storage integrity, metadata, rights, and funding. Record what a surrogate preserves and what access rules travel with it.

Disaster planning should coordinate environmental control, fire and water response, priority salvage, offsite copies, vendor recovery, and staff training. The optimization objective is recoverable cultural function, not maximum item count saved. Rarely used local records may have priority because loss is permanent.

When algorithms propose weeding or relocation, sample rejected cases and near-threshold items, audit by subject and community, and preserve an appeal period. Reversibility supplies time for evidence and public challenge.

A yearly decision record should show additions, removals, representation and accessibility floors, preservation actions, vendor concentration, staff capacity, privacy incidents, failed searches, and community requests. Publish enough to support accountability without exposing reading histories. The record lets future librarians reconstruct why a low-use item was retained or why a popular package was refused. Collection memory should be as carefully preserved as collection material.

Evaluate the record against the people absent from current data: future residents, nonusers, people whose language metadata is poor, and researchers whose topic is not fashionable. Their interests cannot be estimated precisely. Representation floors, professional stewardship, public requests, and preserved option value provide institutional proxies without pretending to speak completely for them.

Case checklist

- Model collections at the level of works, series, archives, and coverage where appropriate.
- Put representation, accessibility, preservation, privacy, and licenses into feasibility.
- Treat circulation as one evidence stream, not total value.
- Audit feedback between ranking, exposure, use, and future acquisition.
- Validate with nonusers, failed searches, qualitative inquiry, and professional review.
- Preserve reversible review before discard and multiple routes to discovery.
- Publish policy, override, request, challenge, and refusal mechanisms.

Case E · A musical prosthesis co-designed as an instrument

Decision and purpose

A musician with a limb difference, injury, progressive condition, or changing mobility wants a device that supports performance. The design team must choose geometry, attachment, materials, joints, sensing, control mapping, feedback, appearance, maintenance, and fabrication. If the task is framed as replacing an average anatomical function, the design can be mechanically competent and musically alien. The purpose is to support this musician’s technique, repertoire, bodily comfort, expressive range, stage identity, and ability to adapt the device over time.

“Prosthesis” and “instrument interface” name overlapping but nonidentical objects. One framing emphasizes clinical function and bodily safety; another emphasizes an expressive system learned through practice. The project may change type during development. A device intended to imitate wrist motion may become a new controller with gestures unavailable to conventional technique. The musician decides whether that difference is compensation, extension, or an unwanted departure.

The baseline is not a standardized body. Compare alternatives with the musician's present technique, existing device if any, adapted instrument, environmental change, assistance, and a range of newly designed interfaces. Sometimes modifying the instrument or performance setup yields more benefit than adding complexity to the body-worn device.

Boundary and participants

The boundary contains the musician's body and sensation, device, instrument, repertoire, posture, stage movement, clothing, room, amplification, rehearsal and travel, fabrication, charging or consumables, repair, clinician or therapist where involved, ensemble collaborators, and audience context. It includes setup time and the consequences of failure during performance.

Participants include the musician as decision owner, prosthetist or orthotist where applicable, clinicians and therapists, mechanical and electronics engineers, instrument makers, software and control designers, materials and fabrication specialists, teachers or ensemble partners selected by the musician, and accessibility or performance staff. Each contributes evidence; none can substitute for the musician's account of embodiment and expression.

Privacy includes motion and physiological data. A stage device can also be a visible identity object. Appearance is not cosmetic residue. The musician may want continuity, concealment, deliberate visibility, cultural reference, or a form that belongs to the instrument. The design process must not optimize normalization against the user's wishes.

Variables and alternatives

Mechanical variables include length, joint axes, stiffness, damping, mass distribution, socket or attachment geometry, contact pressure, range, release mechanism, and material. Sensing variables include modality, placement, sampling, filtering, calibration, and redundancy. Control variables map sensed gesture q and context c to musical or mechanical command y.

Alternatives span passive mechanical adapters, body-powered mechanisms, myoelectric or inertial sensing, switches, pressure and touch surfaces, gaze or foot control, hybrid mappings, instrument modification, and software transformation. A modular interface can support different terminal devices or mappings for practice, studio, and stage.

The feasible space must include donning, removal, cleaning, repair, transport, and operation when power or a sensor fails. A technically expressive system that requires unavailable specialist maintenance is not feasible. Nor is a beautiful prototype whose attachment causes pain after a rehearsal rather than a laboratory trial.

Objective architecture

Let x describe hardware and π a control mapping. Evaluation can include task accuracy A, expressive coverage E, comfort K, learning burden L, fatigue or load F, reliability R_{rel}, setup time S, maintainability M, mass m, and cost C. These measures have different status.

Safety, tissue protection, structural integrity, electrical safety, secure release, and clinically required limits are constraints. The musician may define essential musical gestures or repertoire passages as coverage constraints. Within feasibility, a Pareto search can expose alternatives across expression, learning, fatigue, mass, and cost:

$$\min_{x,\pi} \left(-E(x,\pi),\ L(x,\pi),\ F(x,\pi),\ m(x),\ C(x,\pi)\right) \quad \text{subject to} \quad g_{\text{safety}}(x) \le 0,\ A_r(x,\pi) \ge \underline{a}_r\ \forall r \in \mathcal{R}. \tag{116}$$

The repertoire set $\mathcal{R}$ is selected with the musician and should include demanding, ordinary, and personally important tasks. Equation (116) does not define expressive quality. E may combine reachable dynamics, timing, articulation, continuous control, gesture distinction, and qualitative evaluation. It remains an evolving model of possibility.

Optimization should retain diverse mechanisms. If all criteria are scalarized early, the system may choose a compromise that is mediocre everywhere. A musician may prefer a light passive device with limited generality for one repertoire and a powered interface for another. A family of tools can dominate a universal device.

Control mapping and learnability

The mapping $\pi: q, c \mapsto y$ should be evaluated for controllability, observability, consistency, latency, resolution, and expressive affordance. A one-to-one anatomical imitation is not automatically learnable; familiar sensor changes can produce unfamiliar musical results. Conversely, a nonlinear mapping can become embodied through practice if it is predictable and gives usable feedback.

Calibration should be quick, transparent, and recoverable. The system should show when a signal is outside its trained range or a sensor has shifted. Adaptive mapping can compensate for fatigue or placement, but it can also move the instrument under the performer's hands. Bound adaptation, expose it, and allow the musician to freeze or restore a mapping.

Chapter 10's control theory transfers through feedback and constraints. Chapter 15's music theory adds structure: timing deviation, timbre, phrasing, and tension can be meaningful rather than error. A controller that minimizes deviation from a score may suppress precisely the gesture the device should enable.

Uncertainty and robustness

Body state, skin condition, sweat, clothing, electrode or sensor placement, fatigue, venue temperature, wireless interference, battery, and instrument setup vary. The system needs robustness across expected variation and a graceful response to faults. Design experiments should distinguish device variability from learning and from day-to-day bodily change.

Worst-case design can become heavy and restrictive. Use risk-based layers: hard limits for injury and unsafe release; robust margins for ordinary placement or load variation; detection and fallback for rare faults; and modular replacement for damage. A passive safe state may preserve limited playing rather than end a performance abruptly.

Long-term uncertainty includes changing repertoire, condition, technique, and technology. Adjustable geometry, open interfaces, replaceable sensors, documented calibration, and local fabrication preserve option value. A tightly integrated proprietary device can optimize initial performance and create dependency.

Co-design method

Begin with contextual observation and interviews, then map meaningful activities rather than generic motions. Ask the musician to identify passages, gestures, situations, discomforts, and ambitions. Create low-fidelity mechanical and interface prototypes before expensive integration. Use deliberately contrasting alternatives to reveal preferences: light versus capable, anatomical versus novel, continuous versus discrete control, visible versus discreet form.

Run cycles of fitting, short tasks, rehearsal, reflection, and redesign. Studio critique from Chapter 15 becomes a method of preference learning and reframing. The musician can say not only which alternative is better but that the comparison asks the wrong question. Record this as a model change.

Use engineering optimization inside the cycle for stress, geometry, topology, component selection, or controller tuning. Keep interfaces parameterized so the result can return to physical prototype quickly. A finite-element optimum that changes after real contact pressure or gesture should be updated without treating the user’s body as noise.

Evidence and validation

Accessible digital-instrument research emphasizes adaptability, user participation, iterative prototyping, and interdisciplinary development; musical-interface research adds low and stable latency, learnability, and long-term capacity for virtuosity (Frid 2019; Wessel and Wright 2001). These are design claims to test with the performer, not universal weights to import.

Evidence combines engineering tests, clinical or ergonomic assessment where required, task performance, learning curves, rehearsal and stage trials, maintenance trials, and first-person report. Bench tests cover load, fatigue, impact, release, ingress, battery, electromagnetic conditions, latency, and fault response. Fit tests cover pressure, skin, range, discomfort, and duration.

Musical evaluation uses a representative task battery and open performance. Quantitative measures can include timing, error, dynamic range, repeatability, setup time, and endurance. Qualitative evidence covers agency, embodiment, sound, expressive sufficiency, attention, confidence, appearance, and interaction with collaborators. Record trade-offs rather than converting every response to a mean score.

Validation must extend beyond a demonstration. Test an entire rehearsal, travel and setup, rapid recovery from miscalibration, changing venue, and repair by the intended support network. Novelty can temporarily improve attention or mask burden. Longitudinal review distinguishes learning from fatigue and identifies changing needs.

The strongest validation is not resemblance to a normative movement. It is reliable support for the musician's declared activities, safety, and authorship across realistic contexts.

Governance, ownership, and repair

The musician controls goals, acceptable appearance, data sharing, and adoption. Clinicians own clinical judgments; engineers own safety and technical claims; fabricators own material and process control. Responsibilities and warranties should be explicit at interfaces. If research is involved, participation and withdrawal must not jeopardize access to a working device.

Document parts, software, calibration, maintenance, known limits, data flow, and emergency procedures in accessible language. Define who can update firmware or mappings and how a prior version is restored. Prefer interoperable formats and locally repairable components where feasible. Secure remote features without making essential operation cloud-dependent.

Success may create a design others want. Reuse requires consent and revalidation: bodies, techniques, and purposes differ. Do not present one musician's solution as a universal optimization. The transferable object is the co-design method and interface architecture, not the fitted geometry or weights.

Transfer ledger

Biomechanical optimization → device geometry. Preserves loads, motion, contact, mass, and material behavior under modeled conditions. Forgets sensation, identity, expressive intention, and learning.

Control theory → gesture mapping. Preserves state, feedback, latency, stability, constraints, and fault response. Forgets that deviation can be expressive and that the performer may intentionally redefine the controlled quantity.

Music geometry → repertoire coverage. Preserves selected pitch, timing, or voice-leading relations (Tymoczko 2006). Forgets timbre, historical style, bodily technique, and personal meaning.

Human-centered design → co-authorship. Preserves iterative evaluation in use and the user's authority to reframe (Norman 2013; Schön 1983). It does not eliminate power differences or guarantee access to fabrication and care.

Modular systems engineering → repair and evolution. Preserves interfaces, replaceability, and option value. Forgets the tacit craft through which a comfortable and responsive device is made.

The morphism from prosthesis to instrument is intentionally incomplete. It preserves controllable action and expands possibility; it refuses to define a person's body as a deficit to be minimized.

Failure-mode and effects review

Walk through loss of power, sensor drift, broken attachment, software crash, wireless interference, accidental release, stuck actuator, moisture, impact, skin change, and unavailable replacement parts. For each mode, record effect on body, instrument, performance, data, and recovery; detection; safe state; and owner.

The safest response is context-dependent. Immediate release can prevent injury and drop or damage an instrument. Holding position can preserve equipment and apply unsafe load. The musician and relevant clinicians and engineers should define priorities for rehearsal, stage, travel, and ordinary use.

Practice fault recovery deliberately. The performer should be able to recognize the mode, switch mapping or device, continue in a limited way where safe, or stop without confusion. Ensemble collaborators can know a communication cue without receiving private health data.

Track failures and repairs longitudinally. A design that requires frequent expert adjustment may be unsuitable even if each failure is minor. Maintenance evidence should update geometry, materials, spares, documentation, and local support. Reliability is part of expressive freedom because fear of failure changes performance.

At handoff, create a performer-owned dossier: fitted geometry, safe limits, components, source and build files where permitted, mapping versions, calibration, spares, maintenance, fault actions, data permissions, and contacts. Record the musician's own evaluation and unresolved trade-offs. The dossier supports repair by a future team without converting intimate data into an unrestricted technical asset.

If the project produces a general platform, separate common modules from personal fitting and mapping. Validate the platform's interface and failure behavior, then repeat co-design for every new musician. A reusable connector can preserve mechanical compatibility; it cannot certify comfort, repertoire, identity, or control preference for another body.

Authorship and credit should follow the musician's contribution to concept, testing, mapping, and performance, not disappear behind a generic "user feedback" label.

Any public demonstration should use language and imagery approved by the musician.

Case checklist

- Frame the purpose with the musician, not against a normative body.
- Compare body-worn, instrument, software, environmental, and hybrid alternatives.
- Keep tissue, structural, electrical, and release safety as constraints.
- Evaluate expressive range, embodiment, learning, maintenance, and appearance.
- Bound and expose adaptive control; preserve calibration and version rollback.
- Test full rehearsals, stage faults, travel, setup, and local repair.
- Transfer the process and interfaces, not one person's fitted optimum.

Case F · A yield-aware, energy-aware semiconductor fabrication system

Decision and purpose

A semiconductor manufacturer must plan product mix, wafer starts, routes, batches, equipment qualification, preventive maintenance, dispatching, sampling, rework, and energy use across a fabrication facility. The naive objective is throughput or cost per wafer. The real purpose is to deliver conforming devices on time with high and stable yield, protect workers and environment, maintain traceability, preserve learning, and use scarce equipment, materials, energy, and water responsibly.

The system is re-entrant: a wafer can revisit equipment classes many times at different process layers. Routes contain hundreds or thousands of operations, qualification rules, queue-time limits, batching constraints, setup families, and inspections. Processing times are variable; equipment fails; recipes drift; engineering experiments share tools with production. A local dispatch rule can improve immediate utilization and increase cycle time, contamination risk, or downstream congestion.

Decisions range from seconds to years. Dispatching chooses the next lot. Maintenance and qualification act over days. Starts and product mix act over weeks. Tool purchases and fab design act over years. The model should coordinate these horizons without pretending that a day-ahead schedule determines a decade of technology.

Boundary and affected parties

The operational boundary includes wafer fabrication, metrology, material handling, utilities, maintenance, process engineering, quality, planning, and information systems. A lifecycle boundary includes ultrapure water, chemicals and gases, electricity, abatement, upstream materials, masks, packaging and test, equipment manufacture, waste, logistics, and the downstream energy or capability of the chips.

Affected parties include operators and technicians, process and equipment engineers, customers, suppliers, nearby communities, utility and water systems, emergency services, regulators, and future users of the technology. Safety and environmental compliance are not production penalties. Exposure limits, incompatible materials, exhaust and abatement, emergency states, and waste handling define feasibility.

Data boundaries matter. A measurement can reveal proprietary process information; supplier and customer data may be contractually restricted. Yet hiding interfaces can prevent root-cause analysis. Governance should enable traceability with controlled access rather than create disconnected local optimizers.

Variables and alternatives

Let x_{lomt} indicate whether lot l begins operation o on tool m at time t. Planning variables s_{pt} set wafer starts for product p; q_{mt} select qualification or maintenance; e_{mt} set operating or idle-energy modes; and a_{kt} allocate engineering sampling or metrology capacity.

Alternatives include dispatch rules, batch formation, setup sequencing, qualification calendars, preventive or condition-based maintenance, sampling strategies, rework policies, buffer limits, cross-training, spare strategy, and capacity expansion. Process changes and equipment recipes are governed experiments, not ordinary schedule variables.

The representation should preserve lot identity, route stage, recipe and reticle compatibility, contamination class, priority, due date, queue-time window, hold state, and genealogy. Aggregating all work in an equipment family can erase layer-specific constraints. Conversely, a lot-level exact model may be computationally useless for long-horizon planning. Use multiple resolutions with explicit reconciliation.

Objective architecture

Operational objectives include cycle time, on-time delivery, work in process, yield risk, setup and qualification loss, energy, water, maintenance, and schedule stability. A stylized rolling-horizon objective is

$$\min_x \left[\sum_l w_l\, T_l(x) + \lambda_W \sum_t W_t\,(x) + \lambda_E \sum_t E_t\,(x) + \lambda_Q \sum_l Q_l\,(x) + \lambda_\Delta \parallel x - x^{\mathrm{prev}} \parallel_1\right] \quad (117)$$

subject to route precedence, tool and operator capacity, qualification, batching, contamination, queue-time, maintenance, safety, and due-date commitments. T_l is tardiness, W_t work in process or congestion, E_t energy, Q_l a validated yield-risk indicator, and the last term discourages schedule churn. Equation (117) is a coordination model, not a claim that yield can be purchased with enough delivery benefit.

Critical yield and safety rules belong in constraints or escalation gates. Environmental budgets can be hard annual and event-level limits. Within feasibility, Pareto comparison should show cost, cycle time, delivery risk, energy-water burden, and learning capacity. Engineering experiments consume short-term capacity and create information that can improve future yield; their value is sequential.

Do not optimize reported yield without defining denominator and disposition. Scrap, rework, test limits, product mix, sampling, and deferred failure can change the metric. Track physical material flows and final customer quality alongside accounting yield.

Coupled physics, queues, and learning

Fabrication joins three models. The **process model** links recipes, equipment state, environment, and material response. The **flow model** links starts, routes, capacity, queues, and completion. The **learning model** links measurement and experiments to changes in process belief. Optimizing one while freezing the others produces systematic failure.

High utilization creates queue amplification. A bottleneck tool with variable availability can set fab-wide cycle time. Adding work upstream can make every tool look busy while delaying output. WIP caps, workload control, and bottleneck-aware dispatch can outperform utilization maximization. Little’s law from Chapter 8 supplies an aggregate invariant, while discrete-event simulation represents route and failure detail.

Yield models are vulnerable to selection and feedback. Lots sent to additional metrology are not random; engineers sample suspected problems. A model trained on measured lots can confuse sampling policy with process risk. If the predictor then chooses sampling, it changes the data. Preserve random or designed audit samples and model the selection mechanism.

Uncertainty and robustness

Uncertainty includes arrivals, process time, tool failure and repair, yield drift, material variation, measurement error, demand, supplier interruption, and utility conditions. Some uncertainty is aleatory; some signals incomplete knowledge or a new failure mode. A robust schedule should retain recovery room without protecting equally against every imagined case.

Use scenario and risk analysis for correlated bottleneck failures, utility curtailment, contamination events, and demand changes. Maintain slack, qualified alternative tools, spare parts, cross-trained staff, and safe inventory for critical materials. These are resilience assets whose value is invisible in steady-state throughput.

Maintenance creates an exploration-exploitation problem. Deferring work increases capacity today and failure risk tomorrow. Fixed intervals can waste useful life; purely predictive maintenance can miss novel failures or become dependent on a fragile sensor. Combine condition evidence with engineering limits, mandatory tasks, and opportunistic maintenance during known idle periods.

Method and computation

Long-horizon planning can use linear or mixed-integer models with aggregated capacity and scenario demand. Detailed scheduling uses constraint programming, decomposition, rolling-horizon mixed-integer methods, dispatch rules, or simulation-based search. Real-time execution needs fast policies and exception handling rather than fragile exact schedules.

Digital twins combine equipment, process, and flow models, but each claim needs validation. A twin that predicts cycle time may not predict defect mechanisms. Surrogates can accelerate recipe or layout search; uncertainty and physical constraints should limit extrapolation. Automatic differentiation and inverse design can connect chip architecture, physical design, and manufacturability, yet the abstraction boundary must remain visible (Mirhoseini et al. 2021).

Energy-aware scheduling can move noncritical utilities or batch steps when grid conditions improve, subject to material queue time, thermal stability, and production risk. Hardware efficiency should be evaluated end to end: a faster accelerator may reduce time per computation and raise total demand. Energy per operation, facility energy, embodied impact, and absolute use answer different questions (Horowitz 2014; Hennessy and Patterson 2019).

Evidence and validation

Semiconductor process control couples statistical monitoring, experimental design, yield modeling, equipment diagnosis, and run-to-run adaptation (May and Spanos 2006; Del Castillo and Montgomery 1995). Equipment availability and utilization should use a stable state vocabulary such as SEMI E10 so that suppliers, operators, and capacity models count time consistently (SEMI 2022).

Trace every model input to manufacturing execution, equipment, metrology, maintenance, utility, or laboratory records with time, units, version, and quality flags. Align clocks and lot genealogy. Missing or manually corrected records should remain visible. Recipe and software version are part of experimental identity.

Validate flow models on queue distributions, cycle-time tails, bottleneck shifts, tool utilization, holds, and recovery from downtime—not only monthly average output. Validate yield models across tools, products, layers, time, and sampling policies. Use holdout periods and prospective shadow evaluation. Investigate calibration where the decision changes, such as sampling or maintenance thresholds.

Before a scheduling policy controls production, replay historical disruptions, run discrete-event simulations, operate in advisory mode, and pilot in a bounded area with safe override. Compare against simple bottleneck and due-date rules. Monitor schedule churn and operator workarounds; they reveal constraints the model omitted.

Process optimization requires designed experiments, engineering review, material characterization, reliability testing, and change control. A simulated optimum cannot bypass qualification. Extreme recipes or geometries are model-exploitation candidates until independent physical evidence supports them.

Governance and failure response

Separate production authority, process-change authority, safety/environmental authority, and model ownership. A scheduler can sequence approved work; it cannot approve a recipe or relax an exposure control. Operators must be able to place a lot or tool on hold. Repeated manual holds should trigger root-cause and model review.

Monitor delivery, cycle time distribution, WIP, yield and escapes, sampling bias, maintenance deferral, energy, water, chemicals, alarms, schedule churn, and distance from validated conditions. Roll back a dispatch or model update if it increases unsafe work, hidden queues, quality escapes, instability, or environmental breach. Preserve the previous rule set and compatible data.

Learning must be institutional. Record why a model changed, what evidence supported it, which products and tools are covered, and what observations would falsify it. Avoid making a single proprietary model an unchallengeable control point.

Transfer ledger

Job-shop scheduling → re-entrant fab flow. Preserves resource capacity, precedence, setup, batching, and due dates. Forgets process-layer semantics and contamination unless represented.

Queueing → cycle time. Preserves aggregate relations among WIP, throughput, variability, and delay. Forgets lot genealogy and rare holds.

Statistical process control → yield learning. Preserves baseline variation and change detection. Forgets causal mechanism and can be distorted by adaptive sampling.

Reliability optimization → maintenance. Preserves failure probability, downtime, spares, and lifecycle cost. Forgets tacit technician evidence and novel failure mechanisms.

Hardware-software co-design → production planning. Preserves coupled performance across architecture, mapping, and physical implementation. Forgets supply-chain, labor, and environmental consequences unless the boundary is expanded.

The fabricated chip is a frozen optimization trace, but the fab is a living evidence system. Its objective is not constant maximum utilization; it is reliable production with learning and safe capacity to recover.

Rare defects and controlled change

Average yield can improve while a rare reliability mechanism worsens. Sampling plans should combine risk-based metrology with designed audit samples that estimate escapes. Track defect families, spatial patterns, tool and chamber history, material lots, and downstream test. A classifier that merges rare defects into “other” may erase the signal that matters most.

Process and dispatch changes should have a change hypothesis, covered products and tools, preconditions, validation, monitoring window, and rollback. Avoid changing multiple coupled controls without a design capable of separating effects. Emergency fixes need retrospective review.

Statistical limits should account for adaptive recipes and sampling. A stable chart can reflect a controller hiding variation; an alarm can reflect a product-mix change. Link process control to physical mechanism and genealogy. When causality is uncertain, quarantine and investigate rather than optimize disposition labels.

Customer and field evidence closes the loop. Final test is not perfect observation of lifetime behavior. Returns, reliability tests, and downstream integration can reveal failure modes absent from wafer metrics. Governance should prevent delivery pressure from redefining a quality escape as acceptable without proper authority.

The change dossier should link requirement, experiment, model, recipe, equipment and software versions, affected genealogy, acceptance limits, review, release, and monitoring. Preserve negative and inconclusive results so a later optimization does not repeat unsafe regions. If a schedule, maintenance policy, or yield model changes the sampling process, mark that change in the data lineage before retraining or trend comparison.

Close the economic loop as well. Expedite, scrap, rework, energy, water, maintenance, and customer escape costs can be recorded by different systems and owners. Reconcile them before declaring improvement. Otherwise an optimizer can lower the planning cost by moving burden into engineering holds, utilities, field support, or later reliability loss.

Supplier changes require the same lineage: material identity, qualification, affected products, substitution logic, and observation window. A nominally equivalent input can interact with a process margin the scheduling model never sees.

Audit emergency substitutions after normal supply returns and retire undocumented workarounds.

Feed the audit into supplier qualification, process limits, inventory policy, and future disruption scenarios.

Assign a named owner.

Case checklist

- Separate planning, dispatch, process change, and infrastructure horizons.
- Preserve route stage, qualification, contamination, queue-time, and genealogy.
- Model process, flow, and learning loops together at their interfaces.
- Keep safety, environmental, and critical-quality rules outside ordinary trade-offs.
- Audit adaptive sampling and optimization-induced data feedback.
- Validate tails, disruptions, optimized policies, and operator workarounds.
- Maintain holds, safe rules, prior versions, and recovery capacity.

Case G · A coastal adaptation pathway under deep uncertainty

Decision and purpose

A coastal region must decide how to reduce flood, erosion, salinity, and storm risk while sustaining homes, livelihoods, ecosystems, infrastructure, public access, and cultural relationships to place. Available actions include protection, accommodation, restoration, warning and recovery capability, changes to development, and voluntary or regulated relocation. No single “sea-level number” determines the choice. The region must act before uncertainty resolves and preserve the ability to change course.

The purpose is not to maximize protected property value. That objective privileges wealth already capitalized in land, can direct protection toward the most expensive assets, and can erase renters, informal use, heritage, ecosystems, and future mobility. The decision must protect life and critical services, distribute burdens justly, avoid preventable ecological and fiscal lock-in, and maintain legitimate choices for communities.

Different actions have different reversibility. Updating building codes and preserving setback space can keep options open. A major barrier or levee creates physical, ecological, financial, and political commitment. Retreat changes property, identity, tax base, and social networks. Waiting also acts: it permits new development, rising exposure, and loss of land needed for future adaptation.

Boundary and affected parties

The physical system includes coastline, rivers, groundwater, sediment, wetlands, dunes, drainage, waves and surge, rainfall, subsidence, buildings, roads, ports, water and wastewater, power, communications, and evacuation routes. The social system includes ownership and tenancy, insurance and finance, local government revenue, employment, schools, health, heritage, indigenous or customary rights, public access, and receiving communities.

Boundaries follow water and institutions, not municipal lines. An upstream defense can redirect risk; a seawall can change erosion alongshore; pumping can affect groundwater and subsidence. Protecting one tax base can shift regional infrastructure cost. Relocation creates obligations in receiving areas. A benefit-cost study confined to the project footprint is structurally incomplete.

Affected parties include residents and renters, indigenous and historically rooted communities, businesses and workers, fishers and farmers, port users, utilities, emergency services, insurers and lenders, tourists, ecosystems represented through law and stewardship, neighboring jurisdictions, and future residents. Participation must occur before alternatives and objectives are fixed.

Variables and pathways

Let a_t be the adaptation action selected at decision period t, z_t the built and ecological system state, and ξ_t observed climate and socioeconomic conditions. Actions can build or raise defenses, restore wetlands, elevate structures, upgrade drainage, protect critical facilities, change land use, buy property, support relocation, or invest in warning and recovery.

The central object is a policy π mapping observations to actions, not one finished design. A pathway might restore wetlands and restrict new exposure now, elevate a critical road when nuisance flooding passes a threshold, begin voluntary acquisition under specified conditions, and construct a surge barrier only if sea level and asset configuration enter a range where it remains effective and acceptable.

Trigger design is itself a decision. A physical threshold may be observed too late because planning and construction take years. Use lead indicators and decision lead time. Triggers can include event frequency, drainage failure, insurance availability, maintenance cost, salinity, ecosystem condition, population exposure, or loss of evacuation reliability. Define measurement, review interval, responsible authority, funding source, and action lead time for each.

Objective architecture

For policy π and scenario trajectory ξ, let $L(\pi, \xi)$ include mortality and health risk, displacement, service interruption, property and infrastructure loss, ecosystem damage, and recovery burden. Let $C(\pi, \xi)$ be public and private lifecycle cost, and let $I_g(\pi, \xi)$ be residual capability or risk for group g.

A robust adaptive formulation can minimize expected and worst-regret performance while enforcing safety and distributional floors. With $J(\pi, \xi) = L(\pi, \xi) + C(\pi, \xi)$, define $J^*(\xi) = \min_{\pi' \in \Pi_{safe}} J(\pi', \xi)$. Let Π_{safe} contain policies satisfying the critical-service risk bound, every protected-group floor, the legal and ecological action set $a_t \in A(z_t, \xi_{0:t})$, and the transition $z_{t+1} = F(z_t, a_t, \xi_t)$. Then

$$\min_{\pi \in \Pi_{\text{safe}}} \left(\mathbb{E}_\xi [L(\pi, \xi) + C(\pi, \xi)] + \lambda \max_{\xi \in \Xi} [J(\pi, \xi) - J^*(\xi)] \right). \tag{118}$$

The regret term compares a pathway with the best policy that would have been chosen with hindsight in each scenario. Equation (118) favors policies that remain acceptable across futures rather than one policy optimized for a single probability distribution. The feasible set A contains legal, ecological, engineering, financial, and consent requirements.

Not every value belongs in L. Human life, cultural continuity, public access, and ecosystem integrity should not be monetized without showing the ethical and legal assumptions. Use a dashboard and Pareto comparison. Where rights or irreversible thresholds apply, use constraints and vetoes. Planetary and ecological boundaries are not ordinary preferences (Rockström et al. 2009).

Deep uncertainty and option value

Uncertainty spans sea-level pathways, ice-sheet response, storm behavior, rainfall, subsidence, erosion, ecosystem response, population, development, technology, insurance, migration, and political capacity (Intergovernmental Panel on Climate Change 2022b). Assigning one precise joint distribution can manufacture confidence. Scenario discovery asks which combinations cause a policy to fail; robust decision making asks which policies perform acceptably across many plausible futures.

Option value comes from actions that preserve future alternatives: setback land, modular defenses, adaptable buildings, transferable finance, open evacuation routes, ecosystem migration corridors, and institutions capable of staged relocation. These assets can look inefficient in a most-likely scenario. Their value appears when the future or public preference changes.

Irreversibility demands asymmetric evidence. A reversible pilot can proceed with uncertainty and monitoring. A barrier that transforms an estuary or a relocation program that dissolves a community requires deeper evidence, consent, compensation, and alternatives. “Delay until certainty” is not neutral when it allows exposure and lock-in to grow.

Coupled engineering and social mechanisms

Hydrodynamic and geomorphic models estimate inundation, waves, erosion, and sediment response. Structural and geotechnical models assess defenses and foundations. Network models assess evacuation and critical-service access. Ecological models assess habitat and protective function. Economic models estimate direct and indirect losses. Social research examines mobility, trust, heritage, tenure, and the ability to use programs.

These components do not compose automatically. A flood-depth map becomes a damage estimate through inventories and vulnerability curves; it becomes a household decision through finance, information, tenure, health, and attachment. Each mapping forgets information. Maintain intermediate products and uncertainty instead of passing only an expected-dollar layer into optimization.

Nature-based measures illustrate coupling. Wetland or dune restoration can reduce waves, create habitat, and provide recreation, but performance depends on sediment, space, maintenance, and event conditions. A model should not treat “one hectare restored” as a fixed risk reduction. Nor should uncertainty be used to dismiss ecological co-benefits while engineered protection is modeled with false precision.

Method and computation

Use exploratory modeling over a wide scenario ensemble rather than one forecast. Simulate candidate policies through time, including triggers and construction lead. Compute expected loss, tail loss, regret, critical-service failure, distributional outcomes, ecosystem indicators, fiscal exposure, and stranded investment. Identify scenarios where each policy violates requirements.

Dynamic programming can solve stylized staged choices; mixed-integer models handle project timing and network investment; robust optimization handles bounded uncertainty; real-options analysis clarifies the value of waiting and flexibility; multiobjective visualization supports deliberation. The final process is not an automatic selection. A small number of distinct pathways should be brought to communities and decision bodies with assumptions and failure regions visible.

Stress tests should include compound flooding, defense failure, evacuation during power loss, repeated events before recovery, insurance withdrawal, funding delay, population change, and neighboring adaptation. A pathway that is robust physically and impossible institutionally is not robust.

Evidence and validation

Dynamic adaptive policy pathways formalize staged actions, monitoring signposts, and transfer points across many plausible futures (Haasnoot et al. 2013; Kwakkel, Haasnoot, and Walker 2015). Coastal portfolios must also confront the distributional and institutional reality of protection, accommodation, and retreat; managed retreat is a risk response with governance and justice requirements, not merely a location variable (Hino, Field, and Mach 2017; Intergovernmental Panel on Climate Change 2022a).

Validate hazard models against tide gauges, storm records, high-water marks, topography, subsidence observations, and event hindcasts, while acknowledging nonstationarity. Validate drainage, building, infrastructure, and ecosystem components at the scale of the decision. Compare model ensembles and explain structural differences instead of averaging them away.

Validate social and distributional claims with property and tenancy data, accessibility analysis, interviews, participatory mapping, historical records, and after-action evidence. Official property data can omit informal residence, cultural use, or people without secure tenure. Community knowledge can identify routes, gathering places, and failure mechanisms missing from maps.

Pilot warnings, shelter routes, retrofit delivery, and voluntary programs. Monitor uptake and reasons for nonparticipation. A low uptake rate may reveal distrust, paperwork, inaccessible finance, landlord control, or an unacceptable offer—not irrationality.

After events, reconcile forecast extent, infrastructure performance, communication, evacuation, service loss, recovery time, displacement, and ecological effects. Update the pathway only through a documented review that distinguishes new evidence from changed values or budgets.

Governance, justice, and failure response

Create a regional body with municipal, infrastructure, emergency, housing, environmental, indigenous or customary, community, labor, and scientific representation. Define decision rights, not merely consultation. Some actions require local consent or statutory process. Publish scenarios, maps, model limits, property and distributional effects, and the basis for thresholds.

Finance must follow the pathway. A trigger without authorized funds is a decorative control law. Create mechanisms for maintenance, emergency action, compensation, relocation support, and receiving-community investment. Protect renters and people without clear title. Avoid buyout designs that reward only property value or produce patchwork abandonment.

Monitoring includes hazard, defense condition, ecosystem state, critical services, exposure, insurance and housing cost, displacement, participation, public access, and trust. Trigger review if impacts shift to another area, maintenance fails, ecological thresholds are crossed, or protected groups fall below the capability floor. Preserve emergency alternatives if the preferred long-term project is delayed.

Transfer ledger

Robust control → adaptation pathway. Preserves state observation, constraints, feedback, and policy revision. Forgets that authority, finance, and consent determine whether an action follows a trigger.

Real options → flexible infrastructure. Preserves the value of waiting, staging, and reversibility. Forgets nonmarket attachments and distribution of who can wait safely.

Reliability engineering → regional resilience. Preserves failure domains, redundancy, critical services, recovery, and common cause. Forgets histories of neglect and unequal capacity to recover.

Ecosystem modeling → nature-based protection. Preserves material flows and conditional protective function. Forgets cultural relationship, intrinsic value, and governance of land and water.

Welfare analysis → loss comparison. Preserves declared consequences under a valuation rule. Forgets rights, procedural justice, and meanings that should remain nonfungible.

The pathway is not optimized once. It is a commitment to observe, deliberate, act, and revise without allowing uncertainty to become either false precision or an excuse for inaction.

Relocation as a complete service system

Relocation cannot be represented as parcels acquired. It is a pathway involving information, appraisal, legal help, finance, housing search, employment, school, health and social care, cultural continuity, moving, site restoration, and integration in receiving communities. Offers must be timely enough that people can choose before repeated loss removes choice.

Eligibility rules should protect renters, informal occupants, shared and inherited ownership, and people with accessibility needs. Compensation based only on market value can be insufficient to obtain equivalent safe housing and can ignore business, network, and cultural loss. Monitor who accepts, who cannot, and who remains in a fragmented service area.

Voluntary programs can become coercive if services, insurance, or maintenance are withdrawn before a viable alternative exists. Publish service-transition rules and preserve emergency response. Community-scale or group relocation options may retain networks better than isolated buyouts, but require land and receiving-area participation.

After acquisition, govern demolition, contamination, land stewardship, public access, and ecological restoration. A checkerboard of empty parcels can increase hazard and maintenance burden. The optimization should include the future landscape and fiscal system, not end at transaction count.

Every pathway review should publish a decision register: observations since the last review, trigger status, model and scenario changes, projects completed and delayed, maintenance condition, group and ecological outcomes, funds available, dissent, and the next reversible and irreversible acts. This prevents a long-horizon pathway from becoming a diagram that survives while its assumptions, authority, and implementation quietly disappear.

Case checklist

- Model a policy with triggers, lead times, and funding—not one terminal project.
- Expand the boundary along water, ecosystems, infrastructure, and displacement.
- Keep safety, rights, cultural authority, and ecological thresholds visible.
- Test expected loss, tail loss, regret, critical services, and group capabilities.

- Preserve option value through setbacks, modularity, corridors, and institutions.
- Validate physical and social mappings separately and retain intermediate uncertainty.
- Monitor distribution, maintenance, ecosystem condition, trust, and receiving areas.

Comparative synthesis: portfolios, identity, and irreversibility

The four cases share a portfolio structure. A library allocates money, space, attention, and exposure. A musician selects a family of mechanical and control possibilities. A fab allocates capacity, maintenance, experiments, and environmental resources. A coast selects staged interventions and preserved options. In each, portfolio mathematics reveals coupling and opportunity cost. In none does it supply purpose.

Their objects differ in identity. A book can be replaceable as a commodity and unique as part of an archive. A fitted device is inseparable from one musician's body and practice. A wafer lot has genealogy through repeated operations. A coastal place contains ecological and social relations not captured by parcels. The model must know which differences are equivalent and which destroy the object.

Irreversibility also takes four forms. A library can irreversibly discard cultural evidence. A device can create bodily harm or dependency but is often physically modifiable. A process change can propagate latent defects into many products. Coastal construction or relocation can transform ecosystems and communities for generations. Evidence and participation should become stronger as reversibility decreases.

Each case contains an endogenous measurement loop. Library exposure creates circulation. Device adaptation changes technique and sensor data. Fab sampling follows suspected risk. Coastal investment changes development and insurance. Historical data is therefore a trace of prior policy, not a neutral demand distribution. Exploration, audit samples, and qualitative inquiry prevent the loop from confirming itself.

The transfer ledgers show why category theory belongs in a field guide. The useful morphisms preserve coverage, feedback, flow, interface, and option relations. Their kernels contain cultural meaning, embodiment, tacit craft, historical injustice, and legitimacy. These are not vague leftovers. They determine whether the composite system serves its purpose. A translation is mature when the receiving discipline can identify and govern what the mathematical map cannot carry.

Four interface patterns

Across the cases, four interface patterns recur.

The **coverage interface** appears in the library and the coast. A collection covers subjects, languages, formats, and memories; an adaptation portfolio covers hazards, places, groups, and time. Set coverage can show gaps, but the definition of an element and adequate coverage is local. One copy does not provide access if format or discovery fails; one project does not provide protection if maintenance or service transition fails.

The **embodiment interface** appears in the prosthesis and, less visibly, the library. A control mapping becomes usable through a musician's learning and sensation. A discovery system becomes useful through a patron's language, attention, privacy, and ability to reach material. An abstract input-output model preserves command or relevance and forgets bodily and social conditions. Co-design restores some of that context.

The **genealogy interface** appears in the fab and archives. A wafer carries route, tool, recipe, material, and measurement history; an archival object carries provenance, order, custody, and relation. Aggregation improves planning and can destroy the identity needed for quality or meaning. Preserve genealogy at the evidence layer even when the optimizer uses a coarser state.

The **pathway interface** appears in coastal adaptation and device evolution. Present choices change future feasible sets. A barrier, discarded archive, proprietary component, or frozen interface can close options. Staged commitment, modularity, reversibility, and trigger rules preserve option value. These design qualities are not merely lower expected cost; they are capacity to learn and change.

These patterns are candidates for reuse in a domain not covered here. The analyst should not import their metrics. Import the question: what constitutes coverage, embodiment, genealogy, and pathway in the new setting, and who can validate the answer?

Negative transfer tests

A cross-disciplinary analogy earns trust by surviving attempts to break it.

Library as inventory. If stock-turn logic recommends removing rare local history, the transfer from retail forgot preservation and future users. Repair by typed retention, professional review, and archival genealogy.

Prosthesis as robot manipulator. If trajectory accuracy recommends a device the musician experiences as painful or alien, the transfer from control forgot embodiment and authorship. Repair through safety constraints, co-design, and expressive evaluation.

Fab as generic job shop. If utilization and due dates send an incompatible layer through a tool or defer metrology that protects yield, the transfer from scheduling forgot process semantics and learning. Repair through typed routes, qualification, and controlled sampling.

Coast as asset portfolio. If expected property loss directs protection toward wealth and treats relocation as a transaction, the transfer from finance forgot rights, community, ecosystems, and receiving places. Repair through capability floors, consent, service systems, and ecological constraints.

The test generalizes. Construct a candidate optimum under the imported model. Ask a domain expert and an affected participant to find a case that is numerically good and substantively wrong. Identify the missing relation, then decide whether it belongs in representation, feasibility, evidence, governance, or refusal. Repeat until remaining failures are understood and monitored.

This is not an endless demand for completeness. It is a disciplined search for the kernel of the translation. The best transfer ledger includes at least one observed counterexample and the design change it caused.

Field checklist

- Protect rights, mission, cultural memory, and expressive agency before optimizing use or throughput.
- Co-design accessible instruments for long-term musical control, repair, authorship, and changing capability.
- Separate semiconductor yield learning from uncontrolled process change and use stable equipment-state definitions.
- Build coastal pathways around signposts, option value, finance, consent, and complete relocation services.
- Test irreversible choices against distribution, identity, maintenance, and future feasible alternatives.
- Preserve the differences among knowledge, creativity, fabrication, and adaptation when translating methods.

Translation lens for Chapter 19. A library, musical prosthesis, fabrication system, and coast all benefit from portfolios, feedback, interfaces, and option value. The mapping succeeds only when discovery is not reduced to demand, embodiment is not reduced to motion, yield is not reduced to utilization, and place is not reduced to property value.

Chapter 20 · Future frontiers, stopping rules, refusal, and synthesis

Computational and institutional frontiers

Optimization is expanding from solving fixed mathematical programs to co-designing models, experiments, machines, institutions, and explanations. Several frontiers are especially consequential.

Automated modeling and scientific discovery

Machine learning can propose hypotheses, surrogate expensive simulators, infer differential equations, design experiments, and search molecules or materials. Physics-informed neural networks include differential-equation residuals in a learning objective (Raissi, Perdikaris, and Karniadakis 2019). The challenge is not only predictive accuracy but identifiability, extrapolation, uncertainty, and discovery of mechanisms.

Automated machine learning and neural architecture search optimize pipelines that themselves optimize models (Hutter, Kotthoff, and Vanschoren 2019; Elsken, Metzen, and Hutter 2019). These nested loops amplify data leakage, benchmark overfitting, and compute inequality. Reproducible evaluation and resource accounting must scale with automation.

Differentiable systems and inverse design

Automatic differentiation makes gradients available through simulators, renderers, physics solvers, and programs. Inverse design can optimize shapes, materials, optics, controls, and manufacturing jointly. Discrete choices and discontinuities remain difficult; surrogate gradients can be useful without being faithful. A gradient is evidence about a computational graph, not necessarily the world.

Composable and distributed optimization

Large systems demand decomposition across organizations and privacy boundaries. Distributed optimization, federated learning, market mechanisms, and category-theoretic interfaces seek local computation with global consistency. The frontier is compositional assurance: can guarantees about components be assembled into a guarantee about the whole?

Open games model strategic components with explicit inputs, outputs, observations, and continuations (Ghani et al. 2018). Decorated cospans model systems assembled along interfaces (Fong 2015). These approaches suggest software-like libraries of optimization components, while context dependence limits naive reuse.

Quantum and hybrid optimization

Quantum algorithms investigate new representations of search, sampling, linear algebra, and energy minimization. The quantum approximate optimization algorithm alternates problem and mixing operators in a parameterized circuit (Farhi, Goldstone, and Gutmann 2014). Near-term devices face noise, limited depth, and optimization landscapes with barren plateaus (Preskill 2018; McClean et al. 2018; Cerezo et al. 2021). Claims of advantage require end-to-end comparisons including encoding, error mitigation, classical optimization, and hardware cost.

AI agents and institutional optimization

Agents that plan, call tools, write code, negotiate, or operate infrastructure create long-horizon optimization with partial observability. Capability must be paired with permissions, sandboxes, audit trails, reversibility, and human authority. Evaluations should include deception, specification gaming, correlated failure, and recovery—not just task completion.

At institutional scale, algorithmic systems may allocate work, credit, care, or attention. Democratic oversight and legal contestability must evolve as quickly as technical capability. The future question is not simply whether machines can optimize more, but whether societies can govern what they optimize.

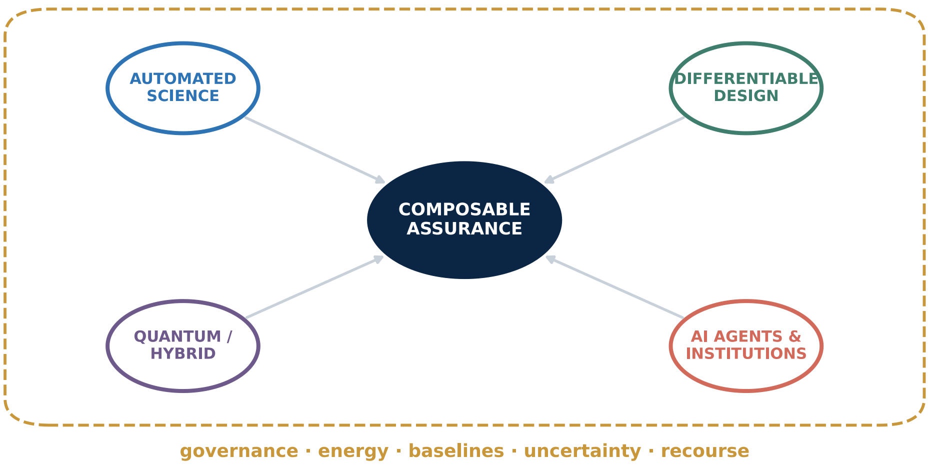


Figure 29. Automated science, differentiable design, quantum or hybrid methods, and AI agents and institutions feed composable assurance.

Practice note. Demand an end-to-end baseline. A frontier method is an improvement only after data, interfaces, energy, validation, human work, and residual risk are counted.

Future optimization systems will search larger spaces, differentiate through more of the world, coordinate more agents, and update more quickly. These capabilities can improve science, design, infrastructure, and care. They also increase the distance between an apparently local objective and its distributed effects. The central frontier is therefore not a more powerful solver by itself. It is an institution able to specify, observe, challenge, and stop powerful search.

Automated modeling makes this visible. A system can choose features, architectures, experiments, tools, and even evaluation criteria. Every additional loop creates another opportunity for leakage and self-confirmation. If a model proposes experiments that generate its next training data, it shapes the evidence by which it is judged. If an agent writes the tests used to evaluate its code, correlated blind spots can pass through. Independent baselines and holdouts must remain outside the optimized loop.

Differentiable programming extends gradient information through simulators, renderers, physical models, and software. The gradient is exact for the computational graph and conditional on its approximations. It should be accompanied by a provenance path: which variables were held fixed, which discontinuities were smoothed, which constraints were relaxed, and which world measurements calibrate the result. Chapter 4's local sensitivity becomes a systems claim only after these interfaces are validated.

Quantum and hybrid approaches likewise require end-to-end comparison. Encoding, state preparation, error, measurement, classical outer-loop search, and hardware energy belong in the boundary. A speedup for an inner kernel is not automatically a faster, cheaper, or more reliable decision process (Preskill 2018; Cerezo et al. 2021). The same rule applies to specialized accelerators, distributed computing, and learned surrogates.

Compositional assurance

Category theory offers a disciplined aspiration: describe systems by objects, morphisms, composition, and invariants so that local reasoning can be reused. Decorated cospans model components joined through interfaces (Fong 2015); open games expose strategic components with context from the future (Ghani et al. 2018). These constructions help specify what a component consumes, produces, and promises.

Optimization resists naive composition. Local feasible sets can become globally inconsistent. Two stable controllers can interact unstably. Two fair allocations can compose into an unfair pipeline. A privacy guarantee can be consumed through repeated use. A locally optimal policy can change another component's data distribution. The assurance task is therefore to write a composition rule with side conditions.

For each interface, record:

- semantic type and units of inputs and outputs;
- timing, ordering, and freshness assumptions;
- uncertainty and error contract;
- resource and privacy budget consumed;
- safety and security invariants preserved;
- authority to reject, override, or degrade;
- observations available for detecting contract failure.

A system-level guarantee is valid only if these assumptions hold simultaneously. This is category theory used as engineering grammar rather than ornament: composition is allowed where morphisms type-check and required invariants are preserved. Where the diagram does not commute, the mismatch is evidence, not an inconvenience to hide.

Optimization constitutions

As systems become more autonomous, projects need a constitution before they need a larger objective. The constitution defines legitimate purpose, prohibited actions, decision rights, evidence standards, monitoring, appeal, and conditions for revision. It distinguishes what the optimizer may search from what the institution has authority to choose.

A useful constitution contains seven clauses.

1. **Purpose.** Name the human, social, scientific, or ecological capability the system serves and the baseline it seeks to improve.
2. **Jurisdiction.** Define the people, assets, places, time horizons, and externalities inside the decision boundary.
3. **Rights and limits.** State legal, safety, ethical, environmental, and procedural requirements that cannot be traded by a weight.
4. **Evidence.** Define which claims require simulation, experiment, professional review, participation, or monitored deployment.
5. **Authority.** Assign who can approve objectives, change constraints, deploy, override, investigate, and stop.
6. **Contestability.** Give affected parties understandable notice, correction, explanation, and appeal appropriate to the stakes.
7. **Sunset and repair.** Specify review dates, rollback assets, incident response, retirement, and restoration of capabilities the system may displace.

These clauses turn governance from a general aspiration into interfaces. A regulator, operator, model owner, community body, or user can hold different authority. “Human oversight” is replaced by named acts and response times.

Stopping search

More optimization is not always valuable. Stop computational search when the expected value of another iteration is less than its resource and delay cost, or when solver uncertainty is already small relative to model and preference uncertainty. A simple decision rule is

$$\text{continue search only if} \qquad \mathbb{E}[\mathit{\Delta} V_{n+1} \mid \mathcal{I}_n] > C_{\text{compute}} + C_{\text{delay}} + C_{\text{risk}}, \tag{119}$$

where $\mathcal{I}_n$ is current information and $\mathit{\Delta} V_{n+1}$ is decision-relevant improvement, not merely a lower numerical objective. Equation (119) is qualitative unless these terms are estimated, but it forces the right comparison.

Practical stopping indicators include stable decisions across plausible models, a certified bound adequate for the choice, repeated alternatives within measurement error, exhausted evaluation budget, or a finding that new computation cannot resolve the contested value. Report the best candidate, bound or gap, sensitivity, and reason for stopping.

Stopping deployment

Deployment is an experiment with responsibilities. Stop or roll back when a safety invariant fails; when impact leaves the validated region; when data or model health is compromised; when a protected group crosses a threshold; when operators cannot understand or safely manage the recommendation; when strategic behavior changes the mechanism; or when monitoring is insufficient to know.

Triggers should combine quantitative thresholds and qualitative authority. A rare serious incident may justify action before a rate estimate is precise. A pattern of justified overrides or community complaints can reveal a structural problem before the primary metric changes. Restoration time is part of design; an emergency stop that leaves no functioning service is not a safe stop.

Updates require proportional revalidation. A display change may alter human action even if the model is unchanged. A new data source may change privacy and selection. A solver or hardware change can affect numerics. A new population or jurisdiction can invalidate evidence. Version the complete decision system, not only model weights.

Stopping the project

Refusal is appropriate when the institution lacks a legitimate objective, the necessary decision is outside its authority, harms cannot be made acceptable, evidence cannot support the claim, affected people cannot meaningfully contest the system, or the project would destroy a capability essential for safe fallback. Some decisions should remain deliberative, professional, personal, or political even when analysis informs them.

Refusal is not inaction. It may redirect work toward measurement, service design, capacity, rights protection, interface repair, or a less centralized tool. A hospital may refuse automatic discharge optimization and build better care-transition evidence. A school may refuse a single teacher score and improve observation and support. A cultural institution may refuse personalized surveillance and invest in contextual discovery.

Satisficing is often the responsible alternative (Simon 1955, 1956). Find a feasible action that meets safety, rights, service, and evidence thresholds, then preserve slack and learning capacity. The last decimal of objective value is rarely worth institutional brittleness.

A final synthesis

The field guide can be compressed to four distinctions.

A problem is not its program. The world, purpose, evidence, authority, and intervention surround the mathematical representation. A perfectly solved wrong program remains wrong.

Values have types. When the reduction conditions in Chapter 1 fail, targets, constraints, penalties, vetoes, rights, and process requirements are not interchangeable weights. Their type determines which trade is legal or meaningful.

Translation is a morphism, not an isomorphism. Queueing can illuminate a clinic; feedback can illuminate rehearsal; portfolio theory can illuminate a library. Each mapping preserves relations and forgets meanings. Responsible transfer names both.

Optimization is a governed learning loop. Frame, represent, search, test, decide, observe, revise, and sometimes stop. Evidence and authority must travel with the result.

The seven worked translations demonstrate generative coverage. None uses exactly the same model, because no transdisciplinary method should erase the disciplines it connects. They share a grammar: boundary, alternatives, typed values, uncertainty, evidence, governance, and a transfer ledger. That grammar can be applied to a field not named in this book. Start by asking which structures recur, what the new domain treats as valid evidence, who owns the objective, and what must survive the translation.

Figure 29 should be read as a map of coupled responsibilities. Automated science needs experimental independence; differentiable design needs model provenance; compositional systems need interface contracts; quantum methods need end-to-end baselines; institutional optimization needs legitimacy and repair.

Assess maturity across formulation, evidence, computation, integration, governance, and repair. A system may be advanced in one and primitive in another.

At **exploratory** maturity, the model generates hypotheses and alternatives; outputs are not operational authority. At **advisory** maturity, validated outputs support a named human process with baselines and documented override. At **bounded automation**, the system acts inside verified constraints with monitoring and fallback. At **adaptive operation**, updates occur under lifecycle evidence and change control. At **institutional maturity**, purpose, distribution, appeal, incident learning, and retirement are governed.

Do not promote a system because one metric improves. Require the evidence and organizational capability of the next level. High-stakes projects can deliberately remain advisory. Automation is a design choice, not an inevitable measure of progress.

Maturity can fall when staff, data, suppliers, law, or context changes. Review the operating institution as well as the model.

Archaeological fieldwork is a useful stress test because search can destroy its own evidence. A team must choose survey area, excavation units, depth and sequence, sampling, conservation, documentation, laboratory analysis, and allocation of limited time and expertise. The objective is not to maximize artifacts recovered. It is to produce defensible knowledge while preserving context, respecting law and community authority, protecting people and material, and leaving future researchers meaningful options.

The boundary includes landscape, site, stratigraphy, artifacts and ecofacts, remote sensing, archives, laboratory capacity, conservation, land ownership, descendant and indigenous communities, permitting, funding, weather, security, curation, and publication. A find is valuable partly through relation: layer, association, orientation, material, absence, and excavation history. Removing an object without context can increase count and destroy information.

Decisions are sequential and partially observed. Noninvasive survey and test pits update beliefs about site structure. Excavation reveals and irreversibly changes the state. A policy can choose the next action based on observations, but the action determines what future evidence remains. This resembles Bayesian experimental design and adaptive control, with an unusual state constraint: some information-bearing structures cannot be restored after observation.

A stylized objective could value expected information gain, research coverage, preservation, community priorities, and cost. It should impose constraints for permits, culturally sensitive material, human remains, safety, conservation capacity, recording standards, and agreed authority. The information-gain term depends on a model of hypotheses and observations; it will favor distinctions expressible in that model. Interpretive value and community meaning do not become complete because a posterior entropy decreases.

Use a portfolio of methods: archival research, pedestrian and geophysical survey, environmental sampling, limited excavation, conservation, and analysis. Optimize the sequence so results from less destructive methods inform later action. Reserve unexcavated areas and samples for future techniques. Value of information and option value from Chapter 6 become material stewardship.

Evidence validation includes calibration and ground truth for survey instruments, recording consistency, chain of custody, stratigraphic review, independent interpretation, laboratory controls, and reconciliation with historical, linguistic, and community knowledge. Negative evidence must be interpreted against sampling and preservation. Publication should include methods, uncertainty, data and access rules, and competing readings.

Governance is not an after-the-fact ethics panel. Communities may have legal, cultural, or sovereign authority over sites, access, interpretation, display, repatriation, and data. A technically promising excavation can be refused. Field safety and conservation teams can stop work. The decision dossier should record why material was removed, what remains, and who can authorize future use.

The transfer ledger is revealing. **Experimental design** preserves sequential information and sampling efficiency; it forgets that observation can be destructive and culturally governed. **Optimal stopping** preserves the comparison between further information and cost; it forgets duties to leave material undisturbed. **Information theory** preserves distinctions among modeled hypotheses; it forgets meanings outside the hypothesis set. **Portfolio optimization** preserves budget and method complementarity; it forgets that one excavation can transform authority and place.

The example passes the field-guide test because the grammar produces a nonobvious result: the optimum may be to search less, document better, preserve an option, or refuse excavation. Optimization is not synonymous with extraction.

A language community may allocate resources among teaching, family transmission, speaker support, documentation, publishing, media, terminology, events, digital tools, archives, and institutional recognition. A platform might propose maximizing enrolled learners, daily active users, test scores, or corpus size. Those measures can support operations and cannot define revitalization.

The purpose belongs to the community and may include intergenerational use, cultural continuity, sovereignty, expressive range, public presence, speaker well-being, and domains in which the language can live. The boundary includes fluent and heritage speakers, learners, families, teachers, elders, artists, schools, government, technology platforms, archives, dialects, orthographies, land and ceremony, and historical suppression. Data collection and publication can expose knowledge the community considers restricted.

Decisions are dynamic. Training teachers increases later capacity. Creating media can motivate use. Terminology work enables new domains. A dictionary preserves forms and can freeze one standard. Speech technology can improve access and appropriate voices without consent. A recommendation system can increase exposure and reward short, standardized content. Every intervention changes the language ecology that produces future data.

A portfolio model can track reach, proficiency, speaker support, domain coverage, accessibility, cost, and resilience. Hard constraints should encode community data governance, consent, restricted knowledge, fair compensation, dialect inclusion, and the authority to approve uses. Minimum investment in fluent speakers or family transmission may protect the source of learning rather than maximize learner count.

Uncertainty concerns participation, institutional continuity, migration, technology, funding, and the effect of interventions. Adaptive pathways can build immediate speaker and teacher capacity, pilot programs, monitor real use, and expand only with community approval. Preserve multiple delivery modes so dependence on one platform does not create lock-in.

Evidence should combine participation and learning data with observed use, speaker and family accounts, domain presence, material quality, teacher burden, and cultural assessment. A standardized test can measure selected competence and miss confidence, identity, interaction, or use with elders. Corpus size can grow through duplicated or synthetic material without increasing living language. Qualitative evidence is not a concession; it addresses the construct.

Governance assigns the community decision rights over objectives, data, models, publication, access, and commercial use. Technologists and funders provide tools and resources, not ownership. Speakers should be compensated and able to withdraw particular recordings where agreements permit. Systems need provenance down to recording, speaker permission, transformation, and downstream model.

The transfer ledger again limits the analogy. **Education optimization** preserves resource, sequence, and learning-support structure; it forgets that language transmission is relationship and identity. **Information retrieval** preserves discovery of documented material; it forgets cultural protocol and the value of oral context. **Machine learning** preserves statistical patterns and can support tools; it forgets authority and can standardize away variation. **Ecological resilience** preserves diversity, redundancy, and dependence among institutions; biological fitness must not be imported as a ranking of people or languages.

The recommended policy can therefore prioritize speaker time, teacher formation, community-controlled archives, family contexts, creative production, and open but governed technical infrastructure over the metric with the largest user count. The grammar is generative because it identifies where optimization can coordinate resources and where community authority must set the type of the problem.

Retrospective stress tests: running the ledger backward

The seven worked translations are prospective. These two cases instead run the seven questions backward from an observed failure or accumulated production evidence, recording which questions would and would not have warned. Because their sources already helped shape the grammar, they are consistency checks rather than out-of-sample validation.

Retrospective case R1 · Rankings and reactive institutions

Espeland and Sauder documented how published law-school rankings became constitutive rather than merely descriptive: schools redistributed attention, managed reported categories, and adapted admissions and organizational practice to a commensurated public measure. Their evidence concerns professional schools and ranking institutions, not a generic public-sector program; the lesson is about reactivity when an external measure becomes an internal target (Espeland and Sauder 2007).

Run the seven questions backward. Q3, what is better, flags the conversion of plural educational purposes into an ordinal score. Q4 flags endogenous adaptation and self-fulfilling feedback. Q6 flags the need to validate institutional consequences, not only ranking reproducibility. Q7 flags authority, burden, and the absence of recourse against a privately produced ordering. Q1 and Q5 warn only if the boundary includes schools' strategic responses rather than treating rank as passive information. Q2 warns only if admissibility encodes mission, rights, and protected exclusions rather than merely data eligibility.

Backward transfer ledger. Commensuration preserves comparability and ordering across heterogeneous schools. It forgets mission, educational quality not represented by inputs, local context, and the mechanisms by which a published position changes behavior. The repair is not a more elaborate single score: retain multiple evidence channels, audit gaming and redistribution, disclose uncertainty and category construction, and assign authority to challenge or retire the measure.

The counterexample changes the design rule. A ranking should be evaluated as an intervention in an institution, with predeclared reactivity scenarios and a stopping condition if adaptation degrades the underlying purpose. The field guide would not have predicted every response; it would have required the response channel to exist in the model and governance plan.

Retrospective case R2 · Hidden technical debt in machine-learning systems

Sculley and colleagues synthesized production experience across machine-learning systems rather than reporting one isolated post-mortem. They identified boundary erosion through data dependencies, entanglement, feedback loops, undeclared consumers, configuration debt, and changes that improve a model component while increasing system maintenance risk (Sculley et al. 2015).

Run the questions backward. Q1 flags that the decision object is the production system, not model parameters alone. Q4 flags feedback and changing data dependencies. Q6 flags system-level validation, monitoring, and recovery. Q7 flags ownership of dependencies and changes. Q3 warns if maintenance, failure, and operator burden are part of what counts as better; without them it can bless a locally improved model. Q2 and Q5 alone are insufficient: a feasible training run and a strong search procedure do not expose hidden consumers or coupled failure modes.

Backward transfer ledger. Component optimization preserves local loss, resource, and visible interface assumptions; it forgets entanglement, lineage, downstream consumers, feedback-created data, and safe-change cost. Repair with typed data and service contracts, provenance, dependency inventories, shadow deployment, rollback, and before-and-after system tests. Later Google production practice operationalized many of these repairs across several dozen teams through data, model, pipeline, canary, monitoring, and rollback tests (Breck et al. 2017).

The retrospective result is discriminating rather than triumphant. Several field-guide questions flag the failure only when the boundary is drawn broadly enough and maintenance is admitted as value. The schema is therefore a prompt for disciplined inquiry, not a guarantee that its user will choose the right ontology.

What travels back from humanities and arts

Transfer in optimization is often narrated in one direction: mathematics supplies methods and the humanities supply examples or caveats. The reverse direction is less often formalized and methodologically useful. Humanities and arts contribute procedures for discovering when the object, frame, and criterion are changing under interpretation and use.

Critique can function as adversarial preference elicitation. A counter-reading asks which assumptions make a proposed optimum coherent, whose purposes are naturalized, and which alternative breaks the ranking. Retrospective case R1 is a concrete instance: reading ranking as an intervention exposes feedback omitted by a passive measurement model. Counter-reading generates discriminating cases that reveal missing variables, constraints, and authorities. Reflective practice turns surprise into frame revision rather than a nuisance residual (Schön 1983; Eco 1990).

Framing-loop analysis asks whether search helps produce the objective it later claims to satisfy. A ranking creates attention; a recommendation system creates exposure; a design prototype teaches users what to want; a public narrative changes which futures appear possible. In such systems the map from preference to outcome loops back into preference formation. The objective must be monitored as a state, not treated as fixed input.

Interpretive plurality supplies a rigorous alternative to forced consensus. When several readings are warranted by evidence and purpose, the output should be a set of interpretations with provenance, tensions, and scope—not a single score selected by hidden tie-breaking. This resembles a set-valued argmin, but the analogy must preserve reasons and authority as well as membership.

Narrative and historical analysis add path dependence. Sequence, voice, genre, institutional memory, and the counterfactual from which change is judged affect meaning. Two systems with the same current state can carry different obligations because they arrived through dispossession, repair, consent, or promise. A Markov representation is inadequate when history is constitutive rather than merely predictive (Genette 1980; Moretti 2005).

Reverse transfer ledger. Critical interpretation preserves contestability and exposes suppressed premises; it can forget operational scale unless paired with explicit decisions. Framing-loop analysis preserves endogeneity of purpose; it can become unfalsifiable unless linked to observations. Plural reading preserves warranted difference; it can obscure action unless the institution states how disagreement affects choice. Narrative history preserves path and voice; it can overfit one account unless counter-readings and archives remain available.

The practical implication is a richer optimization workflow. Before fixing an objective, commission a counter-reading. During pilots, observe whether the intervention changes the frame. At evaluation, preserve justified disagreement and the evidence supporting it. At governance review, ask which histories and voices authorize revision. These are not decorative ethics steps; they are methods for finding model error that ordinary sensitivity analysis cannot see.

Stopping, refusal, and final synthesis

Optimization is appropriate when decisions are consequential, alternatives exist, evidence can discriminate among them, and the model can be governed. It is not automatically appropriate because a quantity can be measured.

The optimization charter

Before building a model, write a one-page charter.

1. **Purpose:** What change is sought, and why?
2. **Decision:** Who can choose what, and on what cadence?
3. **Boundary:** Which lifecycle, feedback, and affected groups are included?

4. **Values:** Which outcomes are objectives, constraints, safeguards, or deliberative questions?
5. **Evidence:** What data and causal assumptions support the model?
6. **Uncertainty:** Which unknowns, shifts, and adversaries matter?
7. **Authority:** Who approves, operates, contests, audits, and retires it?
8. **Exit:** What triggers rollback, redesign, or abandonment?

If these questions cannot be answered, more algorithmic sophistication usually deepens confusion.

Match method to consequence

Use the lightest method that supports the decision. A checklist may outperform an optimizer when choices are few and values are qualitative. A spreadsheet may be enough for transparent trade studies. Convex or linear optimization is preferable when its structure fits and certificates matter. Simulation, mixed-integer, stochastic, bilevel, or learning methods earn their complexity only when the decision needs it.

A mature solution reports more than x^*. The recommendation is a bundle of action and warrant: selected decision, objective value, feasibility evidence, optimality or approximation gap, sensitivity, assumptions, validation evidence, and accountable owner.

Stopping rules

Stop numerical search when additional improvement is smaller than decision-relevant uncertainty or implementation cost. Stop data collection when the expected value of information is below its burden. Stop deployment when safeguards fail or benefits do not materialize. Stop metric refinement when disagreement is about values rather than measurement.

Approximate solutions are often rational. A robust, explainable, timely solution can dominate a fragile exact one. Simon's satisficing is not laziness; it is optimization under the cost of optimizing (Simon 1956).

Reasons to refuse optimization

Refuse or reframe when:

- the proposed objective violates rights or dignity;
- affected people lack meaningful voice or recourse;
- the intervention's causal pathway is unknown and consequence is high;
- measurement would create unacceptable surveillance;
- the model would legitimate a decision that cannot be responsibly delegated;
- irreversibility and uncertainty make preservation of options paramount;
- the activity's value depends on freedom from ranking or instrumental control.

Care, friendship, play, mourning, citizenship, scholarship, and art contain dimensions that can be supported by optimization but damaged by total optimization.

A final bird's-eye view

Across calculus, chips, clinics, cities, markets, poems, and music, the recurring pattern is a loop: frame, represent, search, test, decide, observe, and revise.

The arrows matter more than the box labeled "solve." Category theory offers a disciplined analogy: useful composition preserves structure across interfaces. Optimization practice succeeds when mathematical structure, domain meaning, evidence, and accountable authority survive every translation.

The deepest optimization skill is therefore not finding an extremum. It is knowing what should remain invariant while the world, the model, and our values change.

For a new domain, begin with one page and resist software.

Frame. Name purpose, decision level, baseline, boundary, affected parties, authority, and refusal conditions. **Represent.** List variables, alternatives, consequence mechanisms, typed values, uncertainty, time, and adaptation. **Structure.** Identify convexity, discreteness, dynamics, conservation, sparsity, symmetry, interfaces, and available information. **Search.** Choose a method whose inductive bias matches that structure and whose cost matches the decision. **Test.** Build the claim-evidence matrix, baselines, stress cases, and domain checker. **Decide.** Show alternatives, uncertainty, rights, and dissent to the authorized process. **Operate.** Monitor invariants, outcomes, model health, distribution, and purpose. **Repair.** Keep rollback, appeal, update, and retirement real.

At every step, write a transfer ledger. Which structure came from another discipline? What does it preserve? What does it forget? Which local evidence and authority repair the loss? This short habit prevents a borrowed equation from becoming an imported worldview.

A project should hold a formal stopping conversation before launch and at review. Ask whether additional search can change the choice; whether uncertainty is numerical, empirical, or normative; whether the next evidence is worth its cost and delay; whether deployment is reversible; and whether people responsible for fallback can actually execute it.

Record three independent decisions: stop computing, authorize or refuse action, and schedule reopening. The mathematically best candidate can be refused. An approximate candidate can be authorized with monitoring. A successful deployment can be retired when purpose or context changes.

Optimization becomes mature when stopping is not interpreted as failure of ambition. The aim is not the endless reduction of a score. It is a defensible improvement that remains answerable to the world.

Final field checklist

- State the purpose and baseline before naming the objective.
- Type every value as target, constraint, penalty, veto, right, or process requirement.
- Keep model uncertainty separate from contested preference and absent authority.

- Validate optimized candidates at the boundary where they will act.
- Name what each disciplinary translation preserves and forgets.
- Assign approval, override, incident, appeal, rollback, and retirement authority.
- Stop search when model or value uncertainty dominates numerical improvement.
- Stop deployment when invariants, evidence, or controllability fail.
- Refuse systems whose purpose, authority, evidence, or repair cannot be made legitimate.

Translation lens. Optimization becomes transdisciplinary not when every discipline shares one objective, but when different disciplines can compose their partial models without surrendering their standards of evidence, harm, agency, and meaning.

Acknowledgements

The author thanks the independent referee whose detailed reading sharpened the book's structure, mathematics, references, navigation, and illustrations. Responsibility for the synthesis and any remaining errors belongs to the author.

About the idea's author

Dmitry Vostokov is a software diagnostics researcher, engineer, and author whose work connects mathematical structure, computing systems, evidence, and practical problem solving. His broader writing develops pattern-oriented and transdisciplinary languages for diagnosing and improving complex technical systems.

Appendices

Appendix A · Mathematical toolkit

This appendix collects the notation and results most often needed to read optimization work. It is a map, not a substitute for a course in analysis, probability, numerical linear algebra, or algorithms. For the full teaching treatment, see Appendix F, Units 8–22; this appendix is the summary card.

Sets, vectors, and matrices

- $x \in X$ means that x belongs to feasible or ambient set X.
- $\mathbb{R}^n$ is n-dimensional real space.
- $\| x \|_p = (\sum_{i=1}^{n} |x_i|^p)^{1/p}$ for $p \geq 1$; $\| x \|_\infty = \max_{1 \leq i \leq n} |x_i|$.
- $x^T y$ is an inner product in Euclidean coordinates.
- $A \succcurlyeq 0$ means symmetric matrix A is positive semidefinite: $x^T A x \geq 0$ for all x.
- $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ are extreme eigenvalues for a symmetric matrix.

Norm choice expresses geometry. An ℓ_1 ball has corners that encourage sparse solutions; an ℓ_2 ball is rotationally symmetric; an ℓ_∞ ball bounds each coordinate.

Differential quantities

For $f: \mathbb{R}^n \to \mathbb{R}$, the gradient is

$$\nabla f(x) = [\partial f / \partial x_1 \quad \cdots \quad \partial f / \partial x_n]^T. \tag{120}$$

The Hessian is $\nabla^2 f(x)$ with entries $\partial^2 f / (\partial x_i \, \partial x_j)$. For vector function $r: \mathbb{R}^n \to \mathbb{R}^m$, the Jacobian J_r collects first derivatives. The directional derivative in direction d is

$$Df(x)[d] = \lim_{t \downarrow 0} \frac{f(x+td) - f(x)}{t}. \tag{121}$$

Automatic differentiation evaluates derivatives of a program by repeatedly applying the chain rule. Forward mode is efficient for few inputs; reverse mode is efficient for few scalar outputs, subject to storage and control-flow costs.

Probability and statistics

Expectation, variance, and covariance are

$$\mathbb{E}[X], \qquad \mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}X)^2], \qquad \mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}X)(Y - \mathbb{E}Y)]. \tag{122}$$

Conditional probability obeys

$$P(A \mid B) = \frac{P(B|A)P(A)}{P(B)}. \tag{123}$$

A confidence interval is a procedure with repeated-sampling coverage under assumptions; it is not generally a posterior probability that the parameter lies in the realized interval. A

Bayesian credible interval is a posterior probability statement conditional on model and prior. Prediction intervals address future observations and are usually wider.

Optimization statements

The expression

$$x^* \in \arg\min_{x \in X} f(x) \tag{124}$$

returns minimizers; $\min_{x \in X} f(x)$ returns the minimum value. An infimum may exist without being attained. A local minimum has no better feasible point in a neighborhood. A global minimum has no better feasible point anywhere in X.

For smooth unconstrained optimization, stationarity $\nabla f(x^*) = 0$ is necessary at an interior local optimum. It is sufficient for global optimality only with additional structure such as convexity. For constrained problems, use feasible directions, normal cones, or KKT conditions.

Convergence vocabulary

If errors $e_k \to 0$:

- linear convergence means roughly $e_{k+1} \leq c e_k$ for $0 < c < 1$;
- superlinear means $e_{k+1}/e_k \to 0$;
- quadratic means $e_{k+1} \leq c e_k^2$ near the solution;
- sublinear rates include $O(1/k)$ and $O\left(1/\sqrt{k}\right)$.

Iteration count is only one cost. Derivative evaluation, factorization, memory, communication, precision, and parallel efficiency determine wall-clock behavior.

Approximation and error

Distinguish four errors:

1. **Representation error:** the model omits or simplifies reality.
2. **Estimation error:** parameters are inferred from finite or biased data.
3. **Discretization error:** continuous objects are approximated on a mesh or basis.
4. **Optimization error:** the algorithm stops short of the model optimum.

Driving optimization error below the other three may waste computation and create false confidence.

Appendix B · Method cards

Each card states when a method is a plausible first choice and what to verify. Hybrid methods are common.

Linear programming

Use when: objectives and constraints are linear and decisions continuous.

Strengths: mature solvers, dual values, certificates, scalability with sparsity.

Watch: wrong linearization, units, ill-conditioning, degeneracy, hidden integrality.

Report: primal/dual feasibility, objective, gap, tolerances, sensitivity.

Convex optimization

Use when: the objective and feasible set are convex or can be represented by disciplined transformations.

Strengths: global optimality, duality, composable modeling.

Watch: domain errors, scaling, solver tolerances, converting hard constraints into penalties.

Report: formulation class, constraint qualification, residuals, dual certificate.

Mixed-integer optimization

Use when: choices are discrete and linear or convex relaxations are informative.

Strengths: exactness certificates, rich logical modeling, strong commercial and open solvers.

Watch: weak big-M constants, symmetry, huge time-indexed formulations, false precision.

Report: incumbent, best bound, gap, runtime, node count, formulation choices.

Constraint programming and SAT/SMT

Use when: logical, temporal, combinatorial, or global constraints dominate.

Strengths: propagation, expressive discrete structure, proof of infeasibility.

Watch: weak propagation, encoding size, missing optimization bounds.

Report: solver theory, encoding, solution/proof status, limits.

Dynamic programming

Use when: optimal substructure and a manageable state summarize the past.

Strengths: exact recursion, policy outputs, interpretable value functions.

Watch: curse of dimensionality, discretization, incorrect Markov state.

Report: state/action definitions, boundary conditions, approximation error.

Gradient and Newton-type methods

Use when: variables are continuous and derivatives are available or estimable.

Strengths: high-dimensional scaling; rapid local convergence for second-order methods.

Watch: saddle points, conditioning, noisy gradients, line-search failure, nonphysical regions.

Report: initialization, scaling, derivative checks, stopping criterion, trajectory variability.

Proximal and splitting methods

Use when: objectives decompose into smooth and structured nonsmooth terms.

Strengths: sparsity, constraints via projections, distributed decomposition.

Watch: step-size rules, slow accuracy, primal-dual residual interpretation.

Report: operator/prox definitions, residuals, convergence assumptions.

Derivative-free and Bayesian optimization

Use when: evaluations are expensive, noisy, and low- to moderate-dimensional.

Strengths: works with black boxes; Bayesian methods quantify surrogate uncertainty.

Watch: dimensionality, categorical variables, nonstationarity, unsafe evaluations, acquisition overhead.

Report: evaluation budget, initial design, noise model, constraints, best-observed trace.

Evolutionary and population methods

Use when: the landscape is discontinuous, multimodal, structured, or creative diversity matters.

Strengths: parallel exploration, flexible representation, Pareto sets.

Watch: large evaluation budgets, representation bias, weak baselines, premature convergence.

Report: operators, population size, seeds, budget, diversity and convergence plots.

Robust optimization

Use when: uncertainty sets are defensible and constraint failure matters.

Strengths: explicit worst-case protection; tractable counterparts for some sets.

Watch: overconservatism, arbitrary uncertainty sets, correlated errors.

Report: uncertainty construction, protection level, price of robustness.

Stochastic and chance-constrained optimization

Use when: probability distributions or scenarios are credible.

Strengths: expected-value, tail, recourse, service-level formulations.

Watch: rare events, scenario bias, dependence, sample size, time inconsistency.

Report: distribution/scenarios, seed, sampling error, out-of-sample stress tests.

Multiobjective optimization

Use when: trade-offs should remain visible.

Strengths: Pareto frontier, preference elicitation, avoids premature scalarization.

Watch: too many objectives, incomparable scales, dominated displays, hidden veto constraints.

Report: objective definitions, normalization, dominance rule, selected point, rationale.

Simulation optimization

Use when: a validated simulator is the evaluable model.

Strengths: complex queues, networks, agent behavior, uncertainty.

Watch: simulation bias, correlated noise, long transients, optimizer exploiting simulator artifacts.

Report: calibration, validation, warm-up, replications, variance reduction, uncertainty.

Reinforcement learning and optimal control

Use when: actions affect future states and repeated feedback is available.

Strengths: policies rather than one-time decisions; explicit dynamics or learned value.

Watch: unsafe exploration, reward misspecification, partial observation, distribution shift.

Report: state/action/reward, horizon, simulator fidelity, off-policy evaluation, safety limits.

Human-in-the-loop and participatory methods

Use when: preferences, tacit knowledge, or legitimacy cannot be fixed in advance.

Strengths: reveals constraints and meanings, supports negotiation, builds ownership.

Watch: participation theater, fatigue, power imbalance, unstable preference models.

Report: who participated, what authority they had, disagreements, revisions, recourse.

Critical interpretation and counter-reading

Use when: a metric, category, model, or proposed optimum may suppress meanings, alternatives, or power relations.

Strengths: reveals hidden premises; generates adversarial cases; distinguishes disagreement about facts, frames, authority.

Watch: critique without a decision link, unbounded suspicion, tokenized participation, treating one counter-reading as final.

Report: focal text or artifact, situated readers, primary and counter-readings, evidence, suppressed alternatives, resulting model change.

Framing-loop analysis

Use when: search, measurement, ranking, recommendation, or design changes the preferences, demand, categories, or evidence used to evaluate it.

Strengths: detects endogenous objectives, performative metrics, feedback-created data, shifts in the feasible imagination.

Watch: causal stories without observation, confusing ordinary adaptation with corruption, monitoring only the original metric.

Report: initial frame, intervention, feedback channel, observed frame change, affected groups, stopping trigger, reframing decision.

Plural interpretation and comparative reading

Use when: evidence supports several non-equivalent readings and premature aggregation would erase purpose, voice, or context.

Strengths: preserves warranted disagreement, provenance, scope conditions, consequences of alternative interpretations.

Watch: false balance, interpretations detached from evidence, hidden selection among readings, indecision without a governance rule.

Report: interpretation set, supporting evidence, conflicts and shared ground, who may select for which decision, what remains unresolved.

Appendix C · Domain translation matrix

The matrix is a bird's-eye index. It shows recurring objects without asserting that the disciplines are interchangeable.

Table 2. Domain translation matrix: decisions, objectives, invariants, and evidence.

Domain	Typical decisions	Common objectives	Hard constraints and invariants	Evidence and validation
Mathematics	point, function, measure, proof path	value, bound, convergence	topology, algebra, feasibility	theorem, counterexample, numerical residual
Algorithms	representation, operation sequence	time, memory, approximation	correctness, resource model	proof, benchmark, adversarial cases
Software	architecture, code, deployment	quality vector, lead time	semantics, security, compatibility	tests, profiles, incidents, users
Systems/cloud	placement, scheduling, replication	latency, throughput, cost, carbon	capacity, service-level objectives, consistency	traces, load tests, failure drills
AI/data	model, data, threshold, policy	loss, calibration, utility	privacy, safety, compute	holdout, causal test, monitoring
Hardware	architecture, placement, routing	PPA, yield, reliability	physics, timing, fabrication rules	sign-off, simulation, silicon
Mechanical/civil	shape, size, material, sequence	mass, stiffness, lifecycle value	equilibrium, codes, buildability	analysis, test, inspection, field data
Process/materials	flows, conditions, composition	yield, purity, energy, hazard	balances, thermodynamics, safety	experiment, pilot, plant history
Energy/environment	capacity, dispatch, protection	cost, emissions, resilience	carbon, reliability, ecological limits	scenarios, lifecycle data, monitoring
Transport/robotics	route, trajectory, fleet, control	time, energy, access, safety	dynamics, collision, service	simulation, trials, operations
Biology/ecology	intervention, reserve, synthetic design	persistence, function, restoration	evolution, conservation, consent	field/lab data, replication
Clinical medicine	test, treat, monitor, refer	benefit, harm, preference fit	consent, contraindication, standard of care	trials, calibration, shared decision
Public health	program and resource portfolio	population health, equity, resilience	rights, capacity, access	surveillance, causal evaluation, deliberation
Economics/finance	allocation, mechanism, portfolio	welfare, profit, risk	budgets, incentives, regulation	market data, stress tests, causal evidence
Policy/law	intervention, rule, eligibility	public outcomes, legitimacy	rights, due process, authority	impact evaluation, audit, challenge
Human factors	interface, function allocation, workload	performance, safety, inclusion	capability, fatigue, accessibility	observation, usability test, incident
Architecture/design	form, material, program	function, place, delight, carbon	site, code, craft, access	prototype, simulation, critique, use

Domain	Typical decisions	Common objectives	Hard constraints and invariants	Evidence and validation
Art/music/literature	rule, form, gesture, revision	expressive intention, surprise	medium, provenance, internal coherence	critique, performance, interpretation
Games/sport	strategy, rules, training	win, balance, spectacle, health	rules, integrity, bodies	match data, playtest, athlete report

Table 2 should be read across rows before it is read down columns: the shared grammar is useful only when each domain's admissibility rules and evidence remain intact.

Structural correspondences

Some patterns transfer reliably:

- **Conservation:** flow balance, mass balance, probability normalization, accounting identities.
- **Duality:** quantities and prices, flows and potentials, primal plans and certificates.
- **Feedback:** controllers, markets, learning systems, policy response, critique and revision.
- **Interfaces:** software APIs, physical ports, organizational handoffs, categorical boundaries.
- **Regularization:** mathematical penalties, engineering margins, institutional safeguards.
- **Pareto frontiers:** visible trade-offs when no single ordering is legitimate.

Other correspondences are dangerous. Biological fitness is not moral worth; market price is not social value; engagement is not well-being; statistical normality is not health; compression is not interpretation; equilibrium is not justice.

Appendix D · Reference validation and audit trail

Scope and selection standard

The generated bibliography contains **193 cited records** and excludes uncited source records. Selection prioritizes foundational monographs, peer-reviewed research, official standards, legislation, and guidance from accountable public bodies. The seven worked translations include domain evidence for the specific claims on which their formulations depend.

A DOI, ISBN, arXiv identifier, standard number, or official catalog page establishes bibliographic identity; it does not certify the argument made from the source. Citation coverage and record structure were checked throughout. Link liveness and the current applicability of every standard, law, guideline, and software practice remain time-dependent.

Reproducible checks

1. **Structure.** Citation keys are unique; every cited key resolves; each record has a title, year, and accountable author or institution; container fields are checked by type.
2. **Coverage.** Every in-text citation appears in the generated bibliography, and every generated record is cited.
3. **Identifiers.** DOI syntax, arXiv identifiers, ISBN length, standard numbers, and official-source URLs are checked against primary metadata for corrected and newly added records.
4. **Rendering.** ISBNs that would otherwise be hidden by the citation style are repeated in an explicit note. A later-edition identifier is not substituted for an original edition merely to fill a field.
5. **Link hygiene.** Known dead, TLS-failing, stale, or edition-mismatched links are omitted or replaced only when an authoritative record identifies the same work or a clearly labeled authorized reprint.

Correction ledger

- The NIST SSDF and AI RMF records preserve **National Institute of Standards and Technology** as literal institutional authorship; no semicolon is introduced by name parsing.
- Schrijver and Spivak carry visible ISBN notes alongside official catalog pages. Other otherwise-unlinked books expose their recorded ISBNs in notes; pre-ISBN originals are labeled rather than assigned identifiers from later editions.
- Bendsøe and Sigmund is aligned to the second corrected printing dated 2004, with the Springer DOI and hardcover ISBN stated together.
- Campbell’s 1976 working paper links to the authorized 2011 reprint and says so; the DOI is not misrepresented as an identifier for an unaltered 1976 digital issue.
- Farhi, Goldstone, and Gutmann is explicitly identified as arXiv preprint **arXiv:1411.4028**.

- Publisher ampersands and particles in personal and institutional names are protected from author-list parsing. The ledger describes only records that remain in the generated bibliography.
- Version 26 added four cited records—Sen (1985), Kantorovich (1960), Kaplan et al. (2020), and Hoffmann et al. (2022). Version 27 replaced an uncited 2017 conference record with Breck et al. (2017), removed the city from the Sen (1985) imprint, and left 193 cited records.

Limits and date

The audit snapshot is **22 August 2026**. Structural and coverage checks apply to the complete generated bibliography. Primary metadata was re-read for every new worked-translation source and every record named in the correction ledger. Link checking is targeted rather than a promise of permanent availability. Readers must consult current official sources before operational use in medicine, engineering, law, standards compliance, safety, climate adaptation, or software assurance.

The cross-disciplinary synthesis remains interpretation by the author. Citations identify evidence and intellectual lineage; they do not make the cited authors responsible for the synthesis or resolve disagreements among fields.

Appendix E · Extended figure descriptions

These descriptions supplement the concise captions and the alternative text embedded with each vector figure. They identify layout, reading order, and the relation the figure is intended to clarify.

Figure 1

Caption. Bird's-eye view: purpose and values at center; model, evidence, search, and seven domains around them; governance, uncertainty, and feedback spanning the map.

Description. A concentric map puts PURPOSE & VALUES centrally; MODEL, EVIDENCE, and SEARCH around them; seven domain boxes outside; and GOVERNANCE · UNCERTAINTY · FEEDBACK above.

Figure 2

Caption. Seven-stage optimization lifecycle—frame, represent, search, test, decide, observe, revise—around value in context.

Description. Seven nodes circle VALUE IN CONTEXT clockwise: FRAME, REPRESENT, SEARCH, TEST, DECIDE, OBSERVE, and REVISE; REVISE returns to FRAME.

Figure 3

Caption. Structural fingerprint of an optimization problem across five axes: decision, geometry, information, dynamics, and objectives.

Description. A five-branch tree starts at OPTIMIZATION PROBLEM: DECISION (continuous, discrete, function/policy), GEOMETRY (convex, nonconvex, combinatorial), INFORMATION (deterministic, stochastic, adversarial), DYNAMICS (static, sequential, equilibrium), and OBJECTIVES (single, multiple, lexicographic). A footer lists constraints, derivatives, scale, symmetry, evidence, and consequence.

Figure 4

Caption. A timeline from ancient reasoning about the good, through calculus, variational principles, industrial operations research, convexity and complexity, to data-driven, compositional, and governed optimization.

Description. A horizontal timeline alternates milestone labels above and below a central line, moving from ancient deliberation to calculus, operations research, complexity, learning, and governed composition.

Figure 5

Caption. Narrow-valley, multiple-basin, and saddle contour landscapes with distinct search directions.

Description. Three contour panels show a narrow valley with gold paths, multiple basins with gold/red trajectories, and a saddle with red/green directional arrows.

Figure 6

Caption. Feasible region, objective contours, active constraints, normal vectors, and the KKT balance at an optimum.

Description. A polygonal feasible region is crossed by objective contours. The optimum lies where two constraints are active; one objective gradient and two constraint normals are drawn with their vector-balance equation.

Figure 7

Caption. A 2 × 2 comparison of expected-value, risk-averse, robust, and adaptive decisions under uncertainty.

Description. A 2 × 2 grid contrasts optimize the mean, control the tail, protect the admitted set, and observe and revise.

Figure 8

Caption. A Pareto front showing dominated designs, nondominated alternatives, a knee region, aspiration point, and the effect of changing scalarization weights.

Description. A trade-off curve separates dominated points from nondominated alternatives. A knee, aspiration point, and two tangent scalarizations show different selection logics.

Figure 9

Caption. Closed loop from goal through controller or policy, system or world, and outcome, with disturbance and sensor or estimator feedback.

Description. GOAL → CONTROLLER / POLICY → SYSTEM / WORLD → OUTCOME. Disturbance enters the system; outcome returns through SENSOR / ESTIMATOR. The footer names model, delay, saturation, noise, and override.

Figure 10

Caption. Transport coupling between three source and three target masses; arrow thickness shows moved mass.

Description. Three source masses connect to three target masses. Varying arrow thickness shows transported mass; coupling π minimizes total movement cost.

Figure 11

Caption. Sequential composition of SYSTEM A and SYSTEM B through an interface, with f, g, $g \circ f$, a local objective, and a constraint.

Description. SYSTEM A (X→Y) passes through INTERFACE Y to SYSTEM B (Y→Z) by f then g; a curved arrow shows g∘f. Lower boxes name objective and constraint.

Figure 12

Caption. Algorithm families positioned by available structural information and by evaluation scale and cost.

Description. A two-axis chart positions seven method families by structural information (horizontal) and evaluation scale and cost (vertical): Bayesian, evolutionary, derivative-free, stochastic-gradient, gradient, Newton/interior-point, and specialized combinatorial methods.

Figure 13

Caption. Six-layer stack from purpose and decision to hardware and energy.

Description. Six narrowing layers run from PURPOSE & DECISION through mathematical structure, algorithm family, representation, and runtime to HARDWARE & ENERGY.

Figure 14

Caption. Service flow from workload to outcomes, with observation and autoscaling feedback.

Description. Workload flows through QUEUE & ADMISSION and SCHEDULER & SERVICE to outcomes; OBSERVE and AUTOSCALE return feedback. Footer metrics include latency, throughput, reliability, cost, and carbon.

Figure 15

Caption. A radar chart comparing runtime, reliability, security, maintainability, delivery, and usability.

Description. A filled blue radar profile spans runtime, reliability, security, maintainability, delivery, and usability.

Figure 16

Caption. Coupled engineering disciplines exchanging shared design variables, states, sensitivities, and requirements.

Description. Several engineering discipline boxes exchange bidirectional arrows around shared design variables and requirements, emphasizing coupled analysis.

Figure 17

Caption. Hardware trade space around performance, power, area, and architecture or layout, with security, verification, yield, software, and temperature.

Description. PERFORMANCE, POWER, and AREA surround ARCHITECTURE & LAYOUT; security, verification, yield, software, and temperature extend the hardware trade space.

Figure 18

Caption. Contextual-fitness landscape with accessible variation and environmental change.

Description. Contextual fitness is plotted over a phenotype/genotype neighborhood; a gold accessible-variation path crosses a green landscape, and a red marker shows environmental change.

Figure 19

Caption. Two clinical thresholds divide posterior probability into OBSERVE, TEST / ELICIT, and TREAT regions.

Description. Test and treatment thresholds divide a 0-to-1 probability axis into OBSERVE, TEST / ELICIT, and TREAT; a central arrow shows increasing posterior probability.

Figure 20

Caption. Health-system allocation from needs, budget, and workforce to prevention, primary care, acute care, and readiness, governed by equity, access, rights, and geography.

Description. POPULATION NEEDS and BUDGET & WORKFORCE feed ALLOCATION PORTFOLIO, which branches to PREVENTION, PRIMARY CARE, ACUTE CARE, and READINESS under equity, access, rights, and geography.

Figure 21

Caption. A governance loop linking public goals, causal evidence, policy design, implementation, measurement, contestation, and revision.

Description. A circular governance process moves from public goals and causal evidence through policy design, implementation, measurement, contestation, and revision.

Figure 22

Caption. A responsible-optimization frame with rights as boundaries, stakeholder capabilities, distributional outcomes, and planetary constraints.

Description. A bounded responsible-optimization frame places rights at the border, planetary constraints outside ordinary trade-offs, and stakeholder capabilities and distributional outcomes inside.

Figure 23

Caption. A creative design space in which constraints shape a field of possibilities and critique redirects the search.

Description. A cloud of creative alternatives is shaped by constraint boundaries. A critique arrow redirects the search toward a revised region rather than a fixed optimum.

Figure 24

Caption. An eight-axis musical design radar spanning pitch, rhythm, timbre, gesture, space, form, expectation, and culture.

Description. A gold radar profile spans eight labeled axes: pitch, rhythm, timbre, gesture, space, form, expectation, and culture.

Figure 25

Caption. A narrative search diagram connecting world events, plot selection, discourse order, voice, reader inference, and revision.

Description. A directed narrative pipeline moves from world events to plot selection, discourse order, voice, reader inference, and revision, with feedback from interpretation to later choices.

Figure 26

Caption. Seven-stage evidence loop—question, experiment, simulate, calibrate, decide, operate, and learn—around a central EVIDENCE LOOP.

Description. Seven nodes circle EVIDENCE LOOP clockwise: QUESTION, EXPERIMENT, SIMULATE, CALIBRATE, DECIDE, OPERATE, and LEARN; LEARN returns to QUESTION.

Figure 27

Caption. Measured proxy and real value diverge after the proxy becomes a target and optimization exploits measurement error.

Description. Blue measured proxy and coral real value are plotted over optimization pressure/time. After the vertical proxy becomes target marker, shading shows their divergence and exploited measurement error.

Figure 28

Caption. Seven worked translations around shared boundary, decision, and evidence.

Description. BOUNDARY, DECISION, EVIDENCE sends arrows to DATA CENTER, HEAT HEALTH, INSULIN CONTROL, PUBLIC LIBRARY, MUSICAL PROSTHESIS, CHIP FAB, and COASTAL ADAPTATION.

Figure 29

Caption. Automated science, differentiable design, quantum or hybrid methods, and AI agents and institutions feed composable assurance.

Description. AUTOMATED SCIENCE, DIFFERENTIABLE DESIGN, QUANTUM / HYBRID, and AI AGENTS & INSTITUTIONS point inward to COMPOSABLE ASSURANCE, inside a frame whose footer lists governance, energy, baselines, uncertainty, and recourse.

Appendix F · Mathematics for Optimization

A school-mathematics-to-advanced-mathematics companion for this volume

Appendix Version 13 · Independent mathematics companion included in Optimization Across Disciplines, Version 33

This appendix begins below the level usually assumed by an optimization textbook. It revises arithmetic, fractions, algebra, graphs, functions, sets, logic, and proof before building linear algebra, calculus, probability, statistics, optimization, dynamics, transport, information theory, topology, functional ideas, complexity, and category theory. The purpose is not to replace a full course in each subject. It is to make every mathematical statement in the volume readable, traceable to prerequisites, and usable with appropriate caution.

The teaching order is a dependency graph rather than a prestige ladder. Advanced objects are compositions of simpler structures: a Bellman equation reuses functions, probability, sums, and optimization; a KKT system reuses algebra, derivatives, geometry, and logic; a categorical account reuses functions and composition while making interfaces explicit. Readers may enter at any unit, but should follow prerequisite links when notation becomes opaque.

Coverage contract. The atlas accounts for all 124 numbered equations in the volume; two illustrative displays introduced in Chapters 1 and 7 are deliberately unnumbered and outside it. Each entry gives location, prerequisites, a plain-language reading, recurring symbols, and context. The teaching units derive and illustrate representative ideas. Exercises and solutions are deliberately omitted, as requested.

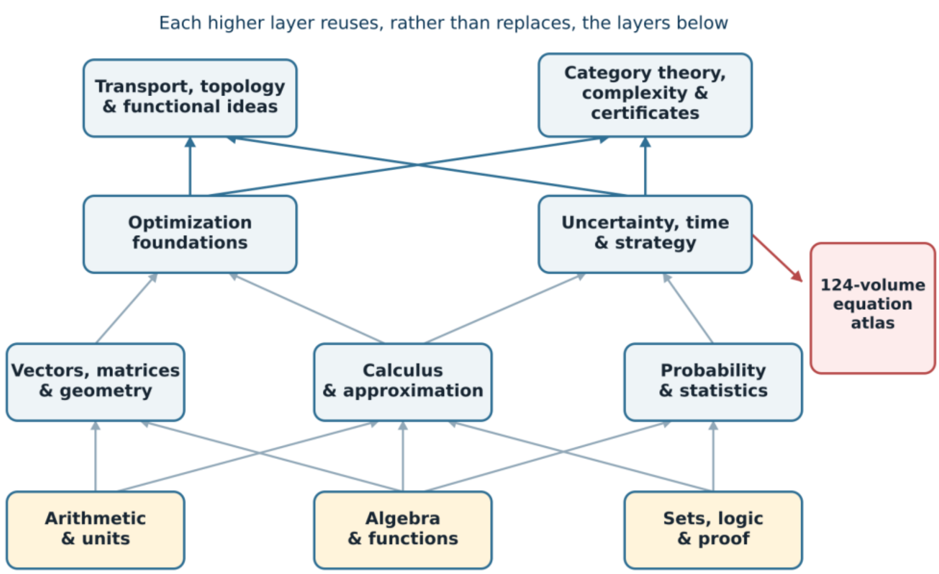


Figure F.1. Dependency map from arithmetic and logic through calculus, probability, optimization, dynamics, transport, category theory, and the 124-equation atlas.

The right-hand atlas is not a terminal layer separate from the rest. It is an index of morphisms back into the teaching graph: every volume equation points to the earlier structures it composes.

How to use this appendix

Read Units 1–7 if school mathematics feels rusty; use Units 8–22 to rebuild the standard undergraduate foundations; use Units 23–35 for the core mathematics of optimization; and use Units 36–41 as bridges to the advanced language employed in the volume. Unit 42 demonstrates cross-disciplinary translation. The atlas then supports equation-by-equation lookup.

A displayed equation should be read in four passes: identify the objects and their types; state what is varied and what is fixed; read the relation or optimization operator in words; and check assumptions, units, and claim scope. This prevents notation from functioning as an authority symbol detached from meaning.

Appendix F figure map and descriptions

1. Figure F.1: dependency map. Three foundation blocks - arithmetic and units; algebra and functions; sets, logic, and proof - feed vectors and geometry, calculus and approximation, and probability and statistics. Higher arrows connect these to optimization, uncertainty, transport, topology, category theory, complexity, certificates, and the equation atlas.
2. Figure F.2: numbers, ratios, and units. A number line orders values from -2 to 4. A ratio of 3:5 becomes the fraction 3/5 and 60 percent, while 72 km/h becomes 20 m/s after multiplication by 1000/3600; both paths preserve the represented quantity.
3. Figure F.3: functions, derivatives, and integrals. Three aligned panels show the same U-shaped function: its input-output graph, a tangent line marking local slope at a point, and a shaded area marking accumulated value over an interval. The panels distinguish value, rate of change, and accumulation.
4. Figure F.4: vectors and matrices. Vectors u and v share an origin and combine to u + v. A matrix A then maps a rectangle to a rotated, scaled, or sheared parallelogram, presenting the matrix as a composable linear transformation rather than merely a rectangular array of numbers.
5. Figure F.5: gradients and curvature. Nested elliptical contours surround a minimum. At a sample point, the red gradient crosses contours toward steepest increase, while the green negative-gradient direction points through lower contours toward lower values; the contour geometry depicts local curvature.
6. Figure F.6: Bayesian updating. Prior probability branches into likelihood and evidence, which combine at the posterior. The posterior informs a decision threshold; a decision can produce new data or revision that feeds back into the next update, separating inference, decision, and learning.

7. Figure F.7: convexity and KKT geometry. Curved objective contours meet a polygonal convex feasible set at a boundary optimum. The objective gradient and active-constraint normal point in opposing directions there, visualizing the first-order KKT balance created by an active constraint. A marked unconstrained minimum outside the set shows why the descent direction is blocked at the boundary.
8. Figure F.8: numerical iteration and evidence. The main path runs from scaling and initialization through model evaluation, step computation, acceptance or rejection, and stopping or iteration. Residuals, optimality gap, units, and sensitivity inform the step, while a feedback arc returns a revised iterate, trust region, or fidelity.
9. Figure F.9: Pareto reasoning. Gray points are dominated and gold points form the nondominated front for two objectives that are both minimized. A circled knee identifies a region where further improvement in one objective requires a comparatively large sacrifice in the other, without declaring a unique best choice.
10. Figure F.10: risk and CVaR. A loss distribution marks expected loss near its center and a value-at-risk threshold in the right tail. The tail beyond that quantile is shaded; conditional value at risk averages losses within this shaded region, distinguishing frequency from tail severity.
11. Figure F.11: Bellman and control feedback. State enters a policy, which selects an action applied to the system or world. The resulting reward returns to policy evaluation and the next state loops into the following decision, so future value becomes part of the present comparison.
12. Figure F.12: Markov chains and queues. The upper row shows neighboring states connected by forward and backward transition arrows. The lower row follows arrivals into a waiting line, through service, and to departure.
13. Figure F.13: optimal transport. Three source masses connect to three target masses through labeled transport links. Splitting a source across destinations illustrates a coupling rather than one-to-one matching, and the optimization box states the governing objective: minimize transported mass multiplied by distance.
14. Figure F.14: categorical composition. The upper chain maps input object X through model f, decision g, and consequence h to Y, with a single arc showing the composite morphism. The lower pair shows parallel components joined by a shared interface or contract; composition is valid only when meanings and constraints align.
15. Figure F.15: complexity and certificates. The upper sequence connects problem size to algorithmic cost, approximation or residual, and a claim with explicit scope. Below, feasible witnesses, bound or dual witnesses, and validation evidence are linked as complementary forms of assurance rather than interchangeable proof.
16. Figure F.16: method families. Four structural problem classes - continuous smooth, convex structured, discrete or logical, and black-box or global - map to representative algorithm families. All paths converge on certificates and validation, emphasizing that problem structure determines admissible claims as well as solver choice.

Part F.1 · School mathematics rebuilt

Unit 1 · Numbers, arithmetic, and units

Prerequisites. none beyond counting.

Why it matters in this volume. Every optimization model eventually compares quantities. Cost, mass, latency, probability, utility, carbon, dose, and musical distance are not interchangeable merely because a computer stores them as numbers.

A number has a magnitude and often a unit. Addition combines like quantities; multiplication creates a new scale; division forms a rate or ratio. Negative numbers encode direction or deficit, while zero may be a genuine quantity, a boundary, or an unavailable measurement. Estimation should precede calculation: an answer with the wrong order of magnitude is suspect even when the arithmetic is internally consistent.

Dimensional analysis treats units algebraically. A valid equation has compatible dimensions on both sides. Scaling replaces a dimensional variable by a dimensionless one, often improving numerical conditioning. Significant figures communicate measurement precision, not decorative formatting. Scientific notation separates order of magnitude from leading digits.

$$72\,\mathrm{km/h} = 72\,\frac{1000\,\mathrm{m}}{3600\,\mathrm{s}} = 20\,\mathrm{m/s} \tag{F.1}$$

Unit factors equal one, so the represented speed is unchanged.

$$x_{\text{scaled}} = \frac{x - x_0}{s} \tag{F.2}$$

Subtract a reference value and divide by a meaningful scale.

Worked reading. If a server processes 2,400 requests in 60 seconds, its observed throughput is 40 requests per second. That rate does not by itself state latency, burst tolerance, or maximum sustainable load. The mathematical value and the operational claim must remain distinct.

Common traps.

- Adding meters to seconds
- Treating a rounded display value as exact
- Ignoring whether zero means none, unknown, or below detection

Where it reappears. Chapters 3, 4, 8, 10, 11, 12, and the worked translations repeatedly use units, rates, scaling, and residuals.

Sources. [F1; F2]

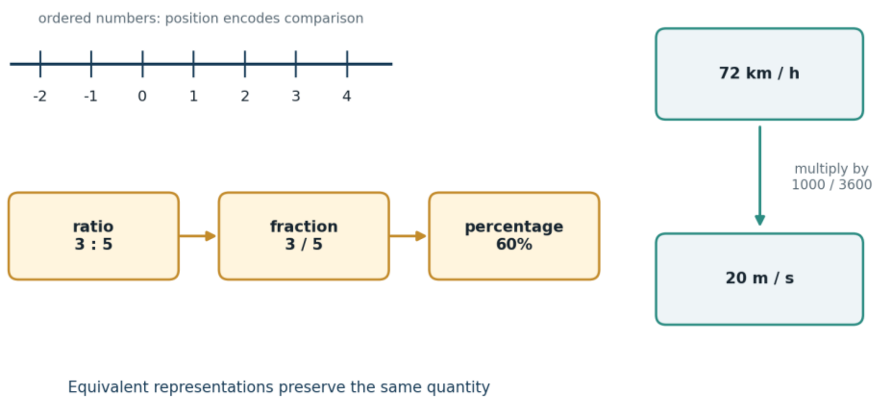


Figure F.2. Equivalent representations of numbers, ratios, percentages, and unit conversion.

Unit 2 · Fractions, ratios, proportions, and percentages

Prerequisites. Unit 1.

Why it matters in this volume. Mixtures, allocations, probabilities, utilization, error rates, fairness metrics, and portfolio weights are all forms of part-to-whole reasoning.

A fraction is a number, a ratio is a comparison, and a percentage is a ratio per hundred. The same symbol can play different roles: 3/5 may denote a quotient, a probability, a slope, or a share. A proportion asserts equality between two ratios; cross-multiplication is valid because both denominators are nonzero and the same equality-preserving multiplication is applied to both sides.

Weighted averages use nonnegative shares that sum to one. Relative change divides a change by a baseline, so a percentage change is undefined when the baseline is zero and unstable when it is very small. Percentage points and percent change are different: moving from 20% to 25% is a rise of five percentage points and a relative increase of 25%.

$$\sum_{i=1}^{n} w_i = 1, \qquad \bar{x}_w = \sum_{i=1}^{n} w_i\, x_i, \qquad w_i \geq 0 \tag{F.3}$$

Nonnegative normalized weights form a convex combination and therefore a weighted average.

$$\text{relative change} = \frac{x_{\text{new}} - x_{\text{old}}}{x_{\text{old}}} \tag{F.4}$$

Change is measured relative to a nonzero baseline.

Worked reading. A hospital allocating 40%, 35%, and 25% of a flexible budget has weights summing to one. If the outcomes being averaged use different denominators or populations, however, the weighted average can be meaningless even though the arithmetic is correct.

Common traps.

- Confusing percentage points with percent change
- Averaging ratios with incompatible denominators
- Normalizing weights that should instead represent absolute capacity

Where it reappears. Allocation, mixture, portfolio, probability, utilization, and multiobjective examples throughout Chapters 6, 8, 11–13, and 18–19.

Sources. [F1; F2]

Unit 3 · Powers, roots, exponentials, and logarithms

Prerequisites. Units 1–2 and basic algebra.

Why it matters in this volume. Growth, decay, discounting, learning curves, entropy, likelihood, numerical conditioning, and algorithmic complexity all use powers or logarithms.

A power repeats multiplication. A root reverses a power on an appropriate domain. An exponential has the variable in the exponent and models proportional growth or decay. A logarithm asks which exponent produces a number. Logarithms turn multiplication into addition, which is why log-likelihoods and decibels are computationally convenient.

The natural exponential and logarithm use base e. For positive a and b, $\log(ab)=\log a+\log b$. Exponential growth is not merely 'fast'; its defining property is a constant relative rate. For $a>0$, the equation $x^2=a$ has the two real solutions $\pm\sqrt{a}$; for $a=0$ it has one real solution, and for $a<0$ it has none. The symbol $\sqrt{a}$ conventionally denotes the nonnegative principal root when $a\geq 0$.

$$a^m a^n = a^{m+n}, \qquad \log(ab) = \log a + \log b \tag{F.5}$$

Exponent and logarithm rules express inverse algebraic structures.

$$x(t) = x_0 e^{rt} \tag{F.6}$$

A constant relative rate r produces exponential change.

Worked reading. If a failure probability is the product of many small conditional factors, computing the logarithm of the product as a sum can avoid underflow. The transformation preserves ordering because the logarithm is strictly increasing on positive numbers.

Common traps.

- Using logarithms of nonpositive real numbers
- Assuming every increase is exponential
- Dropping units from an exponent, which must be dimensionless

Where it reappears. Complexity, probability, information, finance, reliability, and learning models in Chapters 4, 8, 9, 10, 12, 13, and 20.

Sources. [F2; F3]

Unit 4 · Algebra: expressions, equations, and inequalities

Prerequisites. Units 1–3.

Why it matters in this volume. Optimization models are algebraic descriptions of what may change, what must remain true, and how consequences are computed.

An expression names a quantity; an equation states that two expressions are equal; an inequality states an order. A variable may be an unknown to solve for, a decision under control, a measured input, or a random quantity. The role matters. Solving an equation applies reversible operations that preserve the solution set. Solving an inequality requires extra care: multiplying by a negative number reverses the order.

A system collects several simultaneous requirements. Substitution eliminates variables; elimination combines equations; factorization exposes roots or structure. An identity is true for all admissible values, whereas an equation to solve is true only for particular values. Constraints may be strict, non-strict, equality, logical, or set-membership statements.

$$ax + b = c \quad \Rightarrow \quad x = \frac{c-b}{a}, \qquad a \neq 0 \tag{F.7}$$

Undo addition and multiplication while recording the nonzero condition.

$$-2x \leq 6 \quad \Rightarrow \quad x \geq -3 \tag{F.8}$$

Division by a negative reverses the inequality.

Worked reading. For a budget $p_1x_1+p_2x_2 \leq B$, the coefficients are unit prices, the variables are quantities, and the right-hand side is available money. The inequality encodes admissibility, not a preference to spend the entire budget.

Common traps.

- Cancelling a quantity that might be zero
- Changing an equation on only one side
- Treating every variable as a decision variable

Where it reappears. The master form and every mathematical chapter, especially Chapters 1, 4–7, 10–13, and 18–19.

Sources. [F2]

Unit 5 · Coordinates, graphs, and functions

Prerequisites. Units 1–4.

Why it matters in this volume. Objectives, constraints, response surfaces, calibration curves, trajectories, and Pareto fronts are functions viewed through graphs.

A function assigns exactly one output to each admissible input. Its domain states which inputs are allowed; its codomain states the kind of output; its image is what is actually produced. A graph makes the assignment geometric. Slope measures change in output per unit change in input. Intercepts, turning points, discontinuities, and asymptotes are structural clues, not merely drawing features.

Composition applies one function after another. An inverse reverses a one-to-one function on its image. Piecewise functions describe rules that change by region. Level sets collect inputs with the same output and become contour lines or surfaces in higher dimensions.

$$(g \circ f)(x) = g\big(f(x)\big) \tag{F.9}$$

Composition passes the output of f into g.

$$m = \frac{f(x_2)-f(x_1)}{x_2-x_1} \tag{F.10}$$

A secant slope is average change over an interval.

Worked reading. A latency penalty may be flat below a service objective and rise sharply above it. Modeling it as one global straight line erases the threshold; a piecewise or smooth-threshold function represents the intended semantics more faithfully.

Common traps.

- Forgetting to state the domain
- Assuming a plotted curve proves behavior outside the plotted range
- Confusing f^{-1} with $1/f$

Where it reappears. Response surfaces, calibration, thresholds, objectives, and transformations throughout the volume.

Sources. [F2; F3]

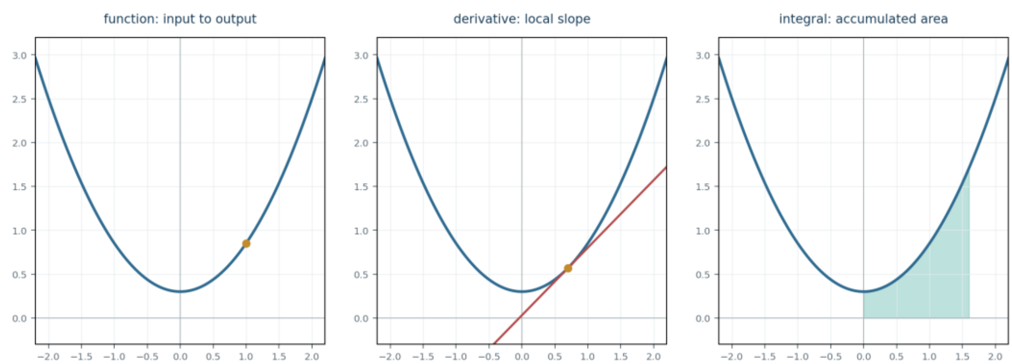


Figure F.3. A function, its local derivative, and its accumulated integral.

Unit 6 · Sequences, series, and summation notation

Prerequisites. Units 1–5.

Why it matters in this volume. Iterations, time steps, discounted returns, accumulated costs, sample averages, and finite resource totals are indexed collections.

A sequence is an ordered list generated by a rule. Sigma notation compresses a finite or infinite sum and makes the index range explicit. Reindexing changes labels, not values, when done consistently. A series is the sum of sequence terms; infinite series require a convergence question before algebraic manipulation is justified.

Geometric sequences multiply by a constant ratio. Arithmetic sequences add a constant difference. Recurrences define later terms from earlier ones and are the simplest form of dynamics. Discount factors reduce the weight of distant terms but do not automatically guarantee convergence if rewards grow too quickly.

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2} \tag{F.11}$$

A closed form replaces a linear number of additions.

$$\sum_{t=0}^{\infty} \gamma^t = \frac{1}{1-\gamma}, \qquad |\gamma| < 1 \tag{F.12}$$

The infinite geometric series converges only under the stated condition.

Worked reading. A ten-step cost $\sum_t c_t$ is finite regardless of discounting. An infinite-horizon cost needs assumptions about γ and c_t. Writing the summation symbol does not discharge those assumptions.

Common traps.

- Omitting index limits
- Manipulating divergent series as if finite
- Confusing an iteration counter with physical time

Where it reappears. Algorithms, dynamic programming, control, reinforcement learning, finance, and lifecycle accounting in Chapters 4, 7–13, and 18–19.

Sources. [F3]

Unit 7 · Sets, logic, quantifiers, and proof

Prerequisites. Units 1–6.

Why it matters in this volume. Feasible sets, events, state spaces, categories, guarantees, and certificates are all built from sets and logical statements.

A set is determined by membership, not order. Union means ‘in either’; intersection means ‘in both’; complement means ‘not in the set’ relative to a stated universe. Logic distinguishes implication from equivalence and universal claims from existential claims. Negating ‘for every’ produces ‘there exists a counterexample’; negating ‘there exists’ produces ‘for every, not.’

A proof is a reproducible chain from assumptions to conclusion. Direct proof, contrapositive, contradiction, induction, and construction answer different logical shapes. A counterexample refutes a universal claim but does not establish a competing universal claim. Necessary and sufficient conditions point in opposite implication directions.

$$A \subseteq B \quad \Leftrightarrow \quad \forall x\,(x \in A \Rightarrow x \in B) \tag{F.13}$$

Subset inclusion is a quantified implication.

$$\neg\big(\forall x\, P(x)\big) \quad \Leftrightarrow \quad \exists x\, \neg P(x) \tag{F.14}$$

One counterexample negates a universal statement.

Worked reading. To claim that every local minimum of a convex function is global, one must state the convexity and domain assumptions. A single nonconvex counterexample shows why the qualifier matters.

Common traps.

- Reversing an implication
- Treating absence of evidence as proof of impossibility
- Hiding assumptions inside notation

Where it reappears. Feasibility, optimality conditions, verification, complexity, rights constraints, and category theory throughout Chapters 1–7 and 20.

Sources. [F4]

Part F.2 · Linear structures and geometry

Unit 8 · Vectors and linear combinations

Prerequisites. Part F.1.

Why it matters in this volume. The principal decision object in continuous optimization is often a vector: a jointly chosen collection whose components interact.

A vector can represent displacement, features, allocations, states, gradients, or coefficients. Addition combines compatible vectors; scalar multiplication changes magnitude and possibly direction. A linear combination forms a weighted sum. Span describes everything reachable by linear combinations; independence means no listed direction is redundant.

Coordinates describe a vector relative to a basis. Changing basis changes coordinates but not the underlying vector. This distinction is essential when variables are rescaled or represented in different units. Linear structure lets local behavior be summarized by matrices and covectors.

$$v = \sum_{i=1}^{n} \alpha_i \, b_i \tag{F.15}$$

Coordinates α_i express v in basis vectors b_i.

$$\alpha u + \beta v \in V \tag{F.16}$$

A vector space is closed under linear combinations.

Worked reading. A portfolio weight vector and a physical-force vector obey the same algebra but different semantics. The analogy preserves linear combination; it does not identify risk with force.

Common traps.

- Adding vectors with incompatible meanings or units
- Confusing a vector with its coordinate column
- Assuming a long list of features is linearly independent

Where it reappears. Chapters 4–7, machine learning, signal processing, control, structural design, finance, and the worked translations.

Sources. [F5]

Unit 9 · Matrices, linear maps, and systems of equations

Prerequisites. Unit 8.

Why it matters in this volume. Matrices encode coupled linear relationships, transformations, constraints, dynamics, covariance, and curvature approximations.

A matrix is best understood as a linear map. Matrix-vector multiplication forms weighted combinations of columns. A linear system Ax=b asks for an input whose image equals b. Rank measures the dimension of the effective image. A singular matrix loses information in at least one direction and cannot have an ordinary inverse.

Matrix multiplication is composition: AB applies B first and A second. Order therefore matters. Transpose exchanges input and output roles under the Euclidean inner product. Eigenvectors preserve direction under a transformation; singular values measure stretching without requiring a square or symmetric matrix.

$$Ax = b \tag{F.17}$$

The columns of A combine with coefficients x to produce b.

$$A(Bx) = (AB)x \tag{F.18}$$

Matrix multiplication represents composition of linear maps.

$$Av = \lambda v \text{ for some } v \neq 0 \tag{F.19}$$

An eigenvector is a nonzero vector that keeps its direction and is scaled by λ.

Worked reading. If constraints are nearly linearly dependent, the system may be solvable yet sensitive. An inverse printed by software is not evidence of a well-conditioned model; residuals and condition estimates must be examined.

Common traps.

- Multiplying matrices in the wrong order
- Assuming every square matrix is invertible
- Using explicit inversion when solving a system is safer and cheaper

Where it reappears. Linear, quadratic, conic and semidefinite optimization; Kalman filtering; control; machine learning; and hardware/system models.

Sources. [F5]

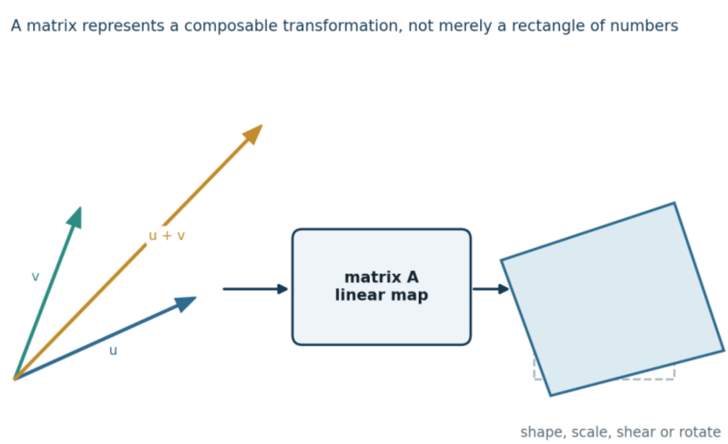


Figure F.4. Vectors, addition, and a matrix viewed as a composable linear transformation.

Unit 10 · Inner products, norms, distance, and projection

Prerequisites. Units 8–9.

Why it matters in this volume. Optimization needs a notion of size, similarity, direction, and nearest feasible point. Those notions are chosen mathematical structure, not automatic facts.

An inner product measures alignment and induces a norm. A norm measures magnitude and obeys positivity, homogeneity, and the triangle inequality. A metric measures distance. Different norms encode different geometries: ℓ_1 favors axis-aligned sparsity, ℓ_2 is rotationally symmetric, and $\ell\infty$ measures the largest component.

Projection selects a nearest point in a set under a chosen norm. For a nonempty closed convex set under Euclidean distance, the projection exists and is unique. Orthogonality converts approximation into geometry: least squares makes the residual orthogonal to the model's column space.

$$\langle x, y\rangle = x^{\top}y, \qquad \| x \|_2 = \sqrt{x^{\top}x} \tag{F.20}$$

The Euclidean norm is induced by the standard inner product.

$$\Pi_C(v) = \arg\min_{x\in C} \| x - v \|_2 \tag{F.21}$$

Projection chooses the nearest feasible point.

Worked reading. Standardizing one feature but not another changes Euclidean distance and therefore nearest neighbors, gradients, and regularization. Geometry must be justified by the meaning and scale of the variables.

Common traps.

- Calling every dissimilarity a metric
- Forgetting which norm defines 'steepest'
- Assuming projection onto a nonconvex set is unique

Where it reappears. Gradient methods, regularization, signal and image recovery, clustering, transport, trust regions, and category-aware interfaces.

Sources. [F5; F10]

Unit 11 · Graphs, networks, and combinatorics

Prerequisites. Part F.1 and Unit 8.

Why it matters in this volume. Schedules, routes, dependencies, causal structures, circuits, social interactions, narratives, and category diagrams can be represented as nodes and relations.

A graph has vertices and edges. Direction represents asymmetric relation; weights represent cost, capacity, probability, or strength. Paths compose edges. Trees are connected graphs without cycles. A directed acyclic graph supports a partial order and appears in scheduling and causal reasoning.

Combinatorics counts structured possibilities. The multiplication principle counts independent stages; combinations ignore order; permutations retain it. Exponential growth in combinations explains why discrete optimization cannot normally enumerate every candidate.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{F.22}$$

The binomial coefficient counts k-element subsets.

$$\text{cost}(P) = \sum_{e \in P} c_e \tag{F.23}$$

A path cost is the sum of its edge costs when costs are additive.

Worked reading. Choosing 20 items from 100 already yields more than 5×10^{20} subsets. A compact problem statement can therefore describe a search space far beyond exhaustive inspection.

Common traps.

- Confusing a graph drawing with the abstract graph
- Treating correlation networks as causal graphs
- Assuming edge costs remain additive when congestion couples paths

Where it reappears. Chapters 2, 5, 7–11, 13, 16, and the library, semiconductor, routing, and narrative cases.

Sources. [F6; F7]

Part F.3 · Change, approximation, and continuous models

Unit 12 · Limits and continuity

Prerequisites. Functions, algebra, and sequences.

Why it matters in this volume. Derivatives, integrals, convergence, and numerical error all depend on controlled limiting behavior.

A limit describes what a function approaches as its input approaches a point; it need not equal the function's value there. Continuity means small input changes produce small output changes locally. In optimization, continuity helps establish existence of extrema on compact feasible sets, but does not ensure uniqueness or easy computation.

Convergence specifies a mode: variables, function values, distributions, or residuals may converge differently. Numerical sequences can appear stable while converging to the wrong result because of discretization or modeling error.

$$\lim_{x \to a} f(x) = L \tag{F.24}$$

Values of f(x) approach L as x approaches a.

$$x_k \to x^* \quad \Leftrightarrow \quad \forall \varepsilon > 0\ \exists K\ \forall k \geq K\colon \parallel x_k - x^* \parallel < \varepsilon \tag{F.25}$$

Convergence eventually enters every requested tolerance.

Worked reading. A step-function tariff is discontinuous at its threshold. A smooth approximation may help a solver but changes the economics near the threshold. The approximation error must be treated as part of the model.

Common traps.

- Substituting before checking that a limit is well-defined
- Equating a flat numerical trace with mathematical convergence
- Assuming continuity implies differentiability

Where it reappears. Chapters 4, 7, 9–11, 17, and any discussion of convergence, discretization, or trust regions.

Sources. [F3]

Unit 13 · Derivatives and one-dimensional optimization

Prerequisites. Unit 12.

Why it matters in this volume. A derivative translates local change into a number and supplies the first tool for finding stationary candidates.

The derivative is the limit of a difference quotient. Its sign indicates local increase or decrease; its magnitude measures sensitivity in output units per input unit. At an interior differentiable local optimum, the derivative must be zero, but zero derivative also occurs at maxima, saddles in higher dimensions, and flat inflection points.

The chain rule differentiates compositions and therefore mirrors the structure of computational graphs. Product and quotient rules describe how sensitivities combine. Nondifferentiable points require one-sided derivatives or generalized derivatives.

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{F.26}$$

The derivative is a limiting local slope.

$$\frac{d}{dx} g\big(f(x)\big) = g'\big(f(x)\big) f'(x) \tag{F.27}$$

The chain rule propagates local sensitivity through composition.

Worked reading. For $f(x)=(x-3)^2+2$, $f'(x)=2(x-3)$, so $x=3$ is stationary. Since $f''(3)=2>0$, it is a strict local minimum; the quadratic form also shows it is global.

Common traps.

- Declaring every stationary point a minimum
- Ignoring derivative units
- Using finite differences with a step so small that subtraction error dominates

Where it reappears. Chapter 4, gradient verification, response surfaces, engineering sensitivities, learning, and control.

Sources. [F3; F9]

Unit 14 · Gradients, Jacobians, Hessians, and covectors

Prerequisites. Units 8–13.

Why it matters in this volume. Multivariable optimization needs derivatives that distinguish scalar outputs, vector outputs, and curvature.

The differential of a scalar function is a covector: it consumes a direction and returns the first-order change. An inner product identifies that covector with a gradient vector. A Jacobian collects derivatives of a vector-valued map. A Hessian collects second derivatives of a scalar function and approximates curvature.

The gradient depends on geometry. Under the Euclidean inner product, the negative gradient is steepest descent per unit ℓ_2 length. Changing variables or preconditioning changes the represented direction. Positive-definite Hessian at a stationary point is sufficient for a strict local minimum; semidefinite curvature at one point is inconclusive.

$$f(x+d) = f(x) + \nabla f(x)^{\top} d + o(\| d \|) \tag{F.28}$$

The gradient represents the first-order differential in Euclidean coordinates.

$$H_f(x) = \nabla^2 f(x) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right] \tag{F.29}$$

The Hessian records local second-order interaction.

Worked reading. For a two-variable objective, contour lines show equal values. The gradient is normal to a smooth contour. A direction tangent to the contour has zero first-order change even though second-order change may remain.

Common traps.

- Treating the gradient as coordinate-free
- Confusing a Jacobian with a Hessian
- Inferring global convexity from one positive-semidefinite Hessian

Where it reappears. Chapter 4, KKT geometry, automatic differentiation, machine learning, structures, control, and physical inverse design.

Sources. [F3; F5; F9]

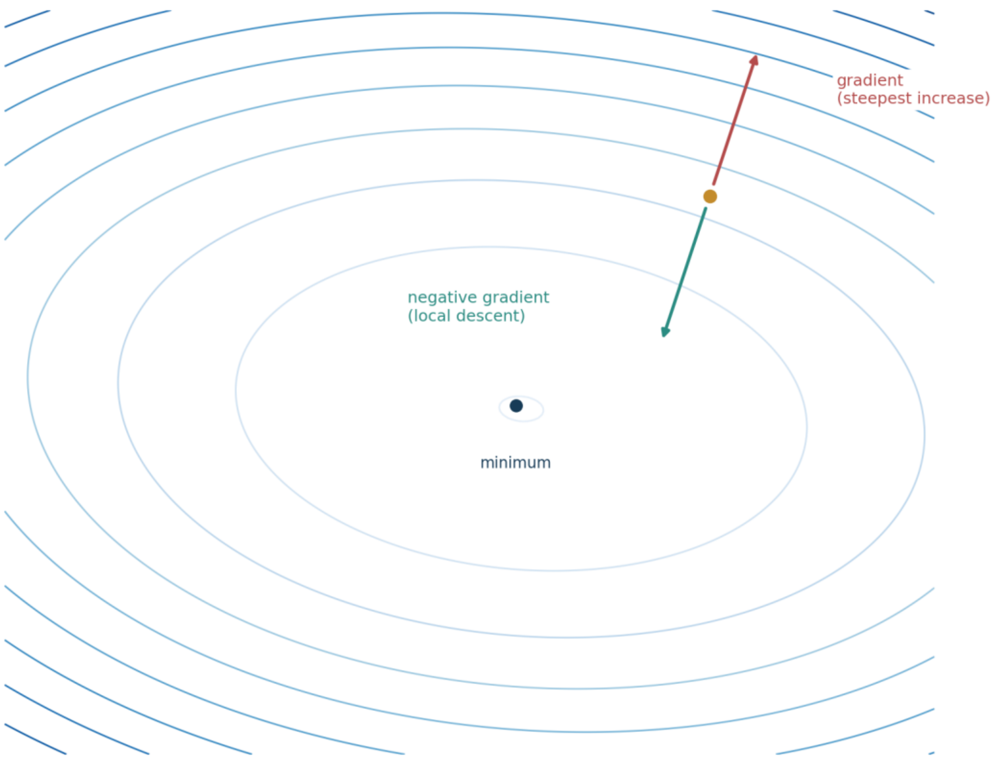


Figure F.5. Objective contours with a gradient, a descent direction, and the minimum.

Unit 15 · Taylor approximation, curvature, and conditioning

Prerequisites. Unit 14.

Why it matters in this volume. Solvers predict improvement from local models, and sensitivity analysis predicts how solutions move when data change.

Taylor expansion approximates a smooth function by polynomial information at a reference point. First order gives a tangent model; second order adds curvature. The remainder states what has been omitted. A trust region restricts steps to a neighborhood where the approximation has evidence.

Conditioning belongs to the problem: it measures response to perturbations. Stability belongs to the algorithm: it measures whether the computation amplifies error beyond the problem's inherent sensitivity. Scaling can improve numerical behavior without changing the underlying decision, provided transformations are tracked correctly.

$$f(x+d) \approx f(x) + \nabla f(x)^{\mathsf{T}} d + \frac{1}{2} d^{\mathsf{T}} H_f(x) d \tag{F.30}$$

A local quadratic model combines slope and curvature.

$$\kappa(A) = \| A \| \| A^{-1} \| \tag{F.31}$$

For invertible A and an induced matrix norm, this condition number bounds the worst-case relative amplification of perturbations.

Worked reading. A narrow curved valley can have a modest objective value error but large variable uncertainty along its flat direction. Reporting only the objective hides practical instability.

Common traps.

- Treating an approximation sign as equality
- Calling an ill-conditioned answer inaccurate without separating data sensitivity from solver error
- Changing scale without transforming tolerances and units

Where it reappears. Trust regions, scaling, residuals, digital twins, sensitivity, inverse design, and deployment envelopes throughout Chapters 4, 10, 11, 17–19.

Sources. [F5; F9]

Unit 16 · Integration and differential equations

Prerequisites. Units 6 and 12–14.

Why it matters in this volume. Accumulated cost, energy, exposure, probability, work, inventory, and state evolution are expressed through integrals and differential equations.

A definite integral accumulates infinitesimal contributions and can be understood as a limit of sums. The fundamental theorem connects accumulated area with differentiation. An ordinary differential equation specifies a rate of change; an initial condition selects a trajectory among possible solutions.

Numerical integration and time stepping introduce discretization error. A smaller step usually reduces truncation error but increases computation and may expose roundoff or stiffness. Conservation laws and units provide essential checks.

$$\int_a^b f(t)\, dt = \lim_{n \to \infty} \sum_{i=1}^{n} f(t_i^*) \Delta t \tag{F.32}$$

Integration is the limit of increasingly fine accumulated sums.

$$\dot{x}(t) = F(x(t), u(t), t), \qquad x(0) = x_0 \tag{F.33}$$

A controlled differential equation defines state evolution from an initial condition.

Worked reading. If power P(t) is measured in watts, energy over an interval is ∫P(t)dt in joules. Summing power readings without multiplying by the sample interval produces the wrong unit and usually the wrong answer.

Common traps.

- Forgetting the constant of integration in indefinite integration
- Assuming every differential equation has a unique global solution
- Ignoring numerical step-size and stiffness effects

Where it reappears. Energy, dose and exposure, control, climate, process systems, robotics, and dynamic worked translations.

Sources. [F3]

Unit 17 · Functionals and the calculus of variations

Prerequisites. Units 14–16.

Why it matters in this volume. When the decision is an entire curve, field, shape, signal, or policy, the objective is a function of a function—a functional.

The calculus of variations perturbs a candidate function by a small admissible function and studies first-order change. Under regularity and endpoint conditions, stationarity yields the Euler–Lagrange equation. Boundary conditions and admissible function spaces are part of the problem, not afterthoughts.

Discretize-then-optimize turns a functional problem into a finite vector problem. Optimize-then-discretize derives continuous optimality conditions first. The two routes need not agree at finite resolution; mesh refinement and adjoint checks compare them.

$$J[y] = \int_a^b L\left(t, y(t), y'(t)\right) dt \tag{F.34}$$

A functional assigns a number to an entire function y.

$$\frac{\partial L}{\partial y} - \frac{d}{dt}\left(\frac{\partial L}{\partial y\prime}\right) = 0 \tag{F.35}$$

The Euler–Lagrange equation is a stationarity condition for admissible path variations.

Worked reading. The shortest path in a curved geometry is not found by optimizing each point independently. The path must be varied as a whole while respecting endpoints and geometry.

Common traps.

- Varying a path outside the admissible class
- Omitting boundary conditions
- Assuming discretization preserves every continuous constraint

Where it reappears. Chapter 4, optimal control, geodesics, shape and topology optimization, signal processing, and continuous-time policy design.

Sources. [F14]

Part F.4 · Probability, statistics, and evidence

Unit 18 · Probability, events, and random variables

Prerequisites. Sets, functions, fractions, and sums.

Why it matters in this volume. Uncertainty in observations, parameters, outcomes, environments, and other agents is represented probabilistically in many—but not all—optimization problems.

A probability model is a probability space $(\Omega, \mathcal{F}, P)$: Ω is the sample space, $\mathcal{F}$ is a σ-algebra of events, and P maps each event to [0, 1], with $P(\Omega)=1$ and countable additivity for every countable pairwise-disjoint family of events. The complement rule $P(A^c)=1-P(A)$ follows from these axioms. A real-valued random variable is a measurable function $X:(\Omega,\mathcal{F})\to(\mathbb{R},\mathcal{B}(\mathbb{R}))$. Its distribution describes probability over values, not a single uncertain value hidden in a box.

Discrete distributions use probability masses; continuous distributions use densities whose integrals give probabilities. A density value can exceed one because it is not itself an event probability. Independence is a structural factorization claim, stronger than zero correlation.

$$P(\Omega) = 1, \qquad P(A_1 \cup A_2 \cup \cdots) = \sum_{i=1}^{\infty} P(A_i), \qquad P(A^c) = 1 - P(A) \tag{F.36}$$

Normalization and countable additivity imply the complement rule for every event A.

$$P(X \in A) = \int_A p_X(x)\, dx \tag{F.37}$$

A continuous probability is area under the density over an event.

Worked reading. A 1% component failure probability does not imply a 1% system failure probability. Dependence, common causes, redundancy, and system logic determine how component events compose.

Common traps.

- Treating a density as a probability
- Assuming uncorrelated means independent
- Assigning probabilities without stating the reference class or time horizon

Where it reappears. Chapters 3, 6, 9–13, 17–19: uncertainty, learning, medicine, reliability, finance, climate, and evidence.

Sources. [F8]

Unit 19 · Expectation, variance, covariance, and correlation

Prerequisites. Unit 18 and algebra.

Why it matters in this volume. Objectives under uncertainty often compare expected consequences, dispersion, covariance, or tail behavior.

Expectation is a probability-weighted average. Variance measures expected squared deviation from the mean. Covariance measures joint linear variation; correlation standardizes covariance to a dimensionless number between −1 and 1 when variances are positive.

Expectation is linear even without independence. Variance of a sum includes covariance terms. Mean and variance do not determine every distribution, and two distributions with the same moments can have different tails and decision implications.

$$\mathbb{E}[X] = \sum_x x\ P(X = x) \quad \text{or} \quad \int x p_X(x)\, dx \tag{F.38}$$

Expectation averages values using their probability law.

$$\mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}X)^2] \tag{F.39}$$

Variance is mean squared deviation from the mean.

$$\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\mathrm{Cov}(X, Y) \tag{F.40}$$

Dependence contributes a covariance term.

Worked reading. Diversification works when components do not move perfectly together. Simply adding individual variances ignores covariance and can understate or overstate system risk.

Common traps.

- Interpreting expectation as the outcome that will occur
- Using correlation as evidence of causation
- Comparing variances of quantities with different units without normalization

Where it reappears. Risk, portfolios, learning, reliability, public health, and stochastic optimization in Chapters 6, 9, 12, 13, 18, and 19.

Sources. [F8; F12]

Unit 20 · Conditional probability, Bayes' rule, and likelihood

Prerequisites. Units 18–19.

Why it matters in this volume. Diagnosis, filtering, calibration, learning, and adaptive decisions update beliefs after evidence arrives.

Conditional probability restricts attention to cases where the condition holds. Bayes' rule reverses the direction of conditioning by combining a prior probability with a likelihood and normalizing by the evidence. A likelihood is the observed-data probability viewed as a function of a parameter; it is not a probability distribution over the parameter until combined with a prior and normalized.

Base rates matter. A highly sensitive test can still have a low positive predictive value when the condition is rare. Sequential updating is coherent when the data model and conditional-independence assumptions are correct.

$$P(H \mid D) = \frac{P(D|H)P(H)}{P(D)} \tag{F.41}$$

Posterior equals likelihood times prior divided by evidence.

$$P(D) = \sum_h P\,(D \mid H = h)P(H = h) \tag{F.42}$$

The evidence averages the likelihood over hypotheses.

Worked reading. With prevalence 1%, sensitivity 90%, and false-positive rate 5%, a positive test does not imply 90% probability of disease. Bayes' rule combines all three quantities and yields about 15.4% under the simplified assumptions.

Common traps.

- Ignoring the base rate
- Swapping P(H|D) and P(D|H)
- Calling a likelihood a posterior probability

Where it reappears. Clinical thresholds, Bayesian optimization, state estimation, machine learning, security, and adaptive control.

Sources. [F8]

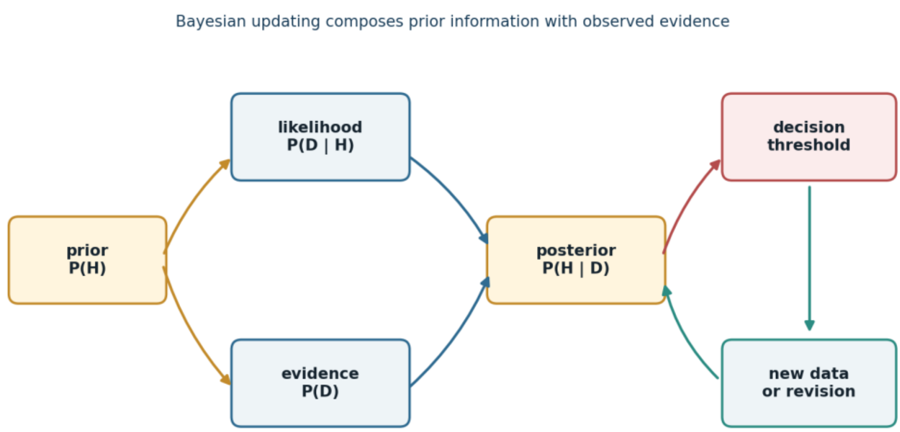


Figure F.6. Bayesian updating from prior and likelihood to posterior and revised decisions.

Unit 21 · Sampling, estimation, uncertainty intervals, and causality

Prerequisites. Units 18–20.

Why it matters in this volume. Optimization consumes estimates. If evidence is biased, confounded, shifted, or selectively measured, a precisely solved model can optimize the wrong world.

A population is the target collection; a sample is observed data. An estimator is a rule mapping data to an estimate. Bias, variance, and consistency describe repeated-sampling behavior. Confidence intervals concern a procedure's coverage under assumptions; they are not generally posterior probability statements about a fixed parameter.

Causal inference asks what would change under intervention. Randomization makes assignment independent of potential outcomes by design; in a finite sample, covariates are balanced only in expectation. Observational identification requires explicit assumptions about confounding, selection, measurement, interference, and positivity. Prediction alone does not answer an intervention question.

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{F.43}$$

The sample mean estimates a population mean under a sampling model.

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{Var}(\hat{\theta}) + \mathrm{Bias}(\hat{\theta})^2 \tag{F.44}$$

Mean squared error decomposes into variance and squared bias.

Worked reading. A hospital risk model may predict readmission accurately because treatment is encoded in the data. Using its association as a causal treatment effect can recommend the opposite intervention. The estimand must be specified before the estimator.

Common traps.

- Treating a convenience sample as representative
- Interpreting statistical significance as practical importance
- Optimizing a predictor as if it were a causal response surface

Where it reappears. Measurement and evidence in Chapter 3; ML in Chapter 9; medicine and policy in Chapters 12–14; experiments and digital twins in Chapter 17.

Sources. [F4; F8; F11]

Unit 22 · Stochastic processes and Markov models

Prerequisites. Sequences, probability, and matrices.

Why it matters in this volume. Queues, reliability, control, finance, disease progression, ecology, and reinforcement learning evolve through uncertain time.

A stochastic process is an indexed family of random variables. A Markov process makes the next-state distribution depend on the present state rather than the full recorded history, conditional on that state. The assumption is about the chosen state representation; omitted memory can make a process appear non-Markovian.

A transition matrix contains conditional probabilities and maps a state distribution forward. Stationary distributions are unchanged by the transition map, but stationarity need not imply rapid mixing or independence over time.

$$P(X_{t+1} = j \mid X_t = i, X_{t-1}, \dots) = P(X_{t+1} = j \mid X_t = i) \tag{F.45}$$

The chosen present state screens off earlier history.

$$\mu_{t+1}^{\top} = \mu_t^{\top} P \tag{F.46}$$

A row distribution advances through the transition matrix.

Worked reading. If machine degradation depends on cumulative thermal exposure, ‘current temperature’ alone is not a Markov state. Adding a sufficient damage variable may restore the property.

Common traps.

- Assuming every time series is Markov
- Confusing a stationary distribution with a static process
- Ignoring uncertainty dependence across time

Where it reappears. Chapters 6–13, particularly queues, filtering, reliability, control, reinforcement learning, health pathways, and finance.

Sources. [F13]

Part F.5 · Optimization foundations and methods

Unit 23 · Formulating an optimization problem

Prerequisites. Parts F.1–F.4.

Why it matters in this volume. The mathematical grammar of the volume begins by separating choices, admissibility, evaluation, uncertainty, and authority.

A decision variable names what the procedure may choose. The feasible set names admissible choices. An objective or preference relation ranks consequences. Parameters and data are not decision variables unless the system is authorized to change them. Argmin denotes the set of minimizers; min denotes the attained optimal value when it exists.

A formulation is a model, not the world. Equivalent algebraic forms may differ numerically. Multiple objectives may be preserved rather than prematurely aggregated. Rights, safety, and logical consistency may belong as hard constraints or decision rights rather than soft penalties.

$$x^* \in \arg\min_{x \in \mathcal{X}} f(x) \tag{F.47}$$

x* is one minimizing choice in the feasible set.

$$p^* = \min_{x \in \mathcal{X}} f(x) \tag{F.48}$$

p* is the best attained objective value.

Worked reading. For staffing, x may be integer shift assignments, X the legal and coverage constraints, and f a cost-plus-service criterion. Historical staffing is a baseline, not automatically a constraint or objective.

Common traps.

- Optimizing variables the decision maker cannot control
- Conflating argmin with minimum value
- Hiding stakeholder trade-offs in an unexplained scalar score

Where it reappears. Bird's-eye view, Chapters 1–3, all mathematical chapters, and every worked translation.

Sources. [F9; F15]

Unit 24 · Convexity and supporting geometry

Prerequisites. Functions, vectors, inequalities, and derivatives.

Why it matters in this volume. Convexity is the principal local-to-global structure used in the volume.

A convex set contains every line segment between its points. A convex function lies below its chords. Minimizing a convex function over a convex set has no non-global local minima. Strict convexity can imply uniqueness; strong convexity provides quantitative curvature.

Convexity is preserved by nonnegative sums, affine composition, pointwise maxima, and other composition rules. Recognizing structure is often more valuable than choosing a fashionable algorithm. Convexity of an objective alone is insufficient if the feasible set is nonconvex.

$$\theta x + (1-\theta)y \in C, \qquad 0 \le \theta \le 1 \tag{F.49}$$

Every segment between feasible points stays feasible.

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta)f(y) \tag{F.50}$$

The function lies below the chord joining two graph points.

Worked reading. The squared Euclidean distance to a fixed point is convex. A binary restriction on the variables makes the feasible set discrete and therefore changes the global structure even if the formula is unchanged.

Common traps.

- Calling a problem convex because the objective looks bowl-shaped
- Assuming strict convexity guarantees existence
- Inferring convexity from a small plot

Where it reappears. Chapter 4, relaxations in Chapter 5, risk and multiobjective formulations in Chapter 6, and numerous domain models.

Sources. [F9; F15]

Unit 25 · Lagrange multipliers and KKT conditions

Prerequisites. Units 14, 23, and 24.

Why it matters in this volume. Constraints change stationarity from ‘zero gradient’ to a balance between objective and active-constraint normals.

The Lagrangian combines the objective with weighted constraint functions. Multipliers measure the local objective value of relaxing appropriately scaled constraints. KKT conditions combine primal feasibility, dual feasibility, stationarity, and complementary slackness.

Constraint qualifications are required for necessity claims. Under convexity, a feasible primal-dual point satisfying KKT conditions certifies global optimality. Outside convexity, KKT generally provides local stationarity evidence, not a global certificate.

$$\mathcal{L}(x,\lambda,\nu) = f(x) + \lambda^{\mathsf{T}} g(x) + \nu^{\mathsf{T}} h(x) \tag{F.51}$$

The Lagrangian attaches multipliers to inequality and equality constraints.

$$\nabla_x \mathcal{L}(x^*,\lambda^*,\nu^*) = 0, \quad \lambda_i^* g_i(x^*) = 0 \tag{F.52}$$

Stationarity and complementary slackness form part of KKT.

Worked reading. An inactive capacity constraint has slack and therefore zero multiplier under complementary slackness. A positive multiplier indicates local value of additional capacity only within the model and scaling used.

Common traps.

- Reading multipliers as universal prices
- Ignoring constraint qualifications
- Treating KKT satisfaction as a global certificate for a nonconvex model

Where it reappears. Chapter 4, constrained engineering, allocation, policy, control, and worked translations.

Sources. [F9; F15; F16]

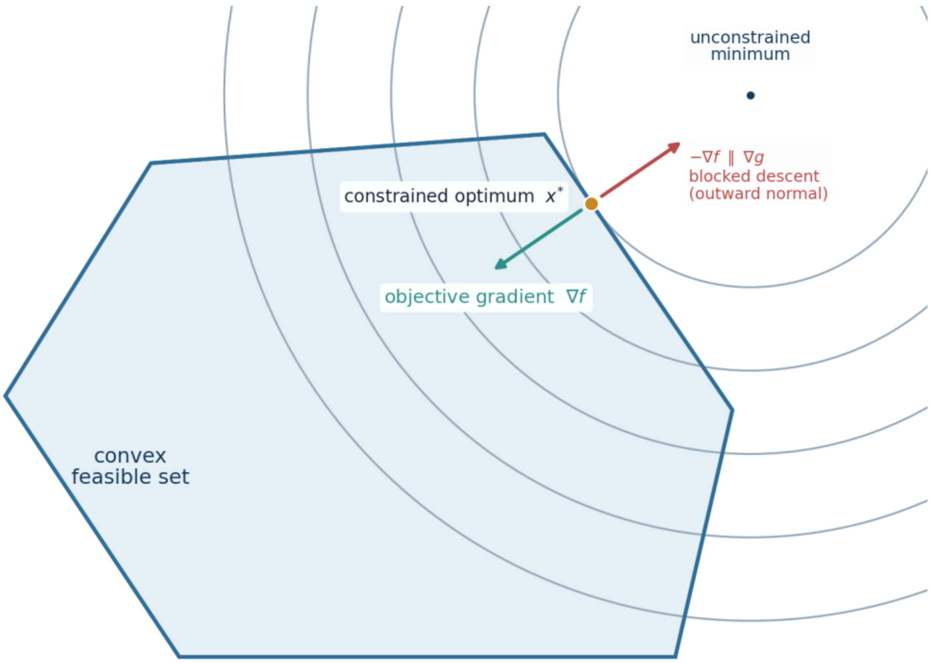


Figure F.7. Convex feasible geometry and the KKT balance at an active boundary.

Unit 26 · Duality, shadow values, and sensitivity

Prerequisites. Unit 25.

Why it matters in this volume. Duality supplies bounds, decompositions, interpretations, and certificates.

The dual function is the infimum of the Lagrangian over primal variables and therefore gives a lower bound for a minimization problem. Maximizing that bound forms the dual problem. Weak duality always orders dual and primal values. Strong duality requires hypotheses such as convexity plus an appropriate regularity condition.

Sensitivity connects small parameter changes to optimal value changes. Dual multipliers often provide first-order sensitivities, but only locally and under regularity. Near degeneracy, active-set changes can make sensitivity nonsmooth or set-valued.

$$q(\lambda, \nu) = \inf_{x} \mathcal{L}(x, \lambda, \nu), \qquad \lambda \geq 0 \tag{F.53}$$

The dual function is a lower bound indexed by feasible multipliers.

$$d^* \leq p^* \tag{F.54}$$

Weak duality places the best dual bound below the primal optimum.

Worked reading. If a resource multiplier is 12 objective units per resource unit, one additional unit is locally worth about 12—provided the active set, model, and units remain stable. It is not a forecast for arbitrarily large expansion.

Common traps.

- Assuming zero duality gap without checking assumptions
- Extrapolating a local shadow value
- Interpreting a multiplier without its units and constraint scaling

Where it reappears. Chapter 4; linear and conic optimization; markets; capacity planning; resource allocation; and certificates.

Sources. [F9; F15; F16]

Unit 27 · Linear, quadratic, conic, and semidefinite forms

Prerequisites. Algebra, matrices, convexity, and duality.

Why it matters in this volume. Many domain models compile into standard structures with mature algorithms and interpretable certificates.

Linear programming uses affine objectives and constraints. Quadratic programming adds a quadratic objective; positive-semidefinite curvature preserves convexity. Conic optimization constrains affine expressions to a cone, unifying nonnegative, second-order, and semidefinite forms. Semidefinite programming optimizes over symmetric positive-semidefinite matrices.

Standard form is a translation target, not a claim that the world is linear. Piecewise-linear approximation, lifting, and relaxation may expose structure while adding variables or approximation error.

$$\min_{x} c^{\mathsf{T}} x \quad \text{s.t.} \quad Ax = b,\ x \geq 0 \tag{F.55}$$

A linear program has an affine objective and affine feasibility.

$$\min_{x} \frac{1}{2} x^{\mathsf{T}} Q x + c^{\mathsf{T}} x \quad \text{s.t.} \quad Ax \leq b \tag{F.56}$$

A quadratic program is convex when Q is positive semidefinite.

$$X \succcurlyeq 0 \tag{F.57}$$

A symmetric matrix X is positive semidefinite.

Worked reading. A covariance matrix must be positive semidefinite because every linear combination has nonnegative variance. That same cone appears in control, relaxations, and experimental design.

Common traps.

- Assuming every quadratic is convex
- Using a relaxation without measuring recovery error
- Treating standard-form coefficients as unitless

Where it reappears. Chapter 5 and applications in control, covariance, hardware, structures, energy, manufacturing, finance, and learning.

Sources. [F9; F15]

Unit 28 · Discrete, integer, and combinatorial optimization

Prerequisites. Unit 11 and formulation.

Why it matters in this volume. Assignments, schedules, routes, layouts, logic, selections, and program transformations involve indivisible choices.

Integer variables encode counts or categories; binary variables encode yes/no choices. Relaxing integrality creates a continuous bound but may produce an unusable fractional decision. Branch-and-bound partitions the search space while using bounds to prune regions. Cutting planes remove fractional solutions without removing integer-feasible ones.

Constraint programming emphasizes propagation over domains; SAT/SMT emphasizes logical satisfiability; dynamic programming exploits repeated substructure. Representation can change practical difficulty without changing the underlying decision.

$$x_i \in \{0,1\} \tag{F.58}$$

A binary variable selects or excludes item i.

$$\sum_i w_i x_i \le B \tag{F.59}$$

Selected items must respect a capacity or budget.

Worked reading. A fractional solution selecting 0.4 of a bridge project may be a useful lower bound but is not an implementable plan. Rounding can violate budget, safety, or coupling constraints, so recovery must be validated.

Common traps.

- Rounding each variable independently
- Calling a heuristic solution optimal without a bound
- Using an integer variable when a continuous proportion is physically realizable

Where it reappears. Chapter 5; cloud scheduling; compiler and circuit design; manufacturing; transport; public allocation; libraries; narrative and game structures.

Sources. [F6; F7]

Unit 29 · Gradient, Newton, proximal, and trust-region algorithms

Prerequisites. Units 13–17 and 23–27.

Why it matters in this volume. Algorithms turn mathematical structure and information access into candidate-generating procedures.

Gradient descent follows first-order local information. Newton's method solves a local quadratic model using curvature. Quasi-Newton methods estimate curvature from gradient changes. Proximal methods preserve nonsmooth structure such as norms and indicator functions. Trust-region methods restrict steps to where a model is credible.

A stopping rule should report residuals, gap, iteration or evaluation limits, and the kind of claim supported. Automatic differentiation differentiates implemented code, not intended mathematics; directional checks compare them.

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \tag{F.60}$$

Gradient descent moves opposite the local Euclidean gradient.

$$x_{k+1} = x_k - H(x_k)^{-1} \nabla f(x_k) \tag{F.61}$$

Newton's step solves the local quadratic stationarity equation.

$$\mathrm{prox}_{\alpha g}(v) = \arg\min_x \left(g(x) + \frac{\|x-v\|_2^2}{2\alpha} \right) \tag{F.62}$$

A proximal map balances structure g with distance from v.

Worked reading. A method that needs fewer iterations may still use more wall time or energy if each iteration solves a large physical simulation. Evaluation cost, memory traffic, parallelism, and synchronization belong to the algorithmic cost model.

Common traps.

- Comparing algorithms only by iteration count
- Accepting a solver status without residuals
- Differentiating through discontinuities without specifying the derivative notion

Where it reappears. Chapter 4, algorithm stacks in Chapters 7–10, digital twins, inverse design, and worked translations.

Sources. [F9; F10]

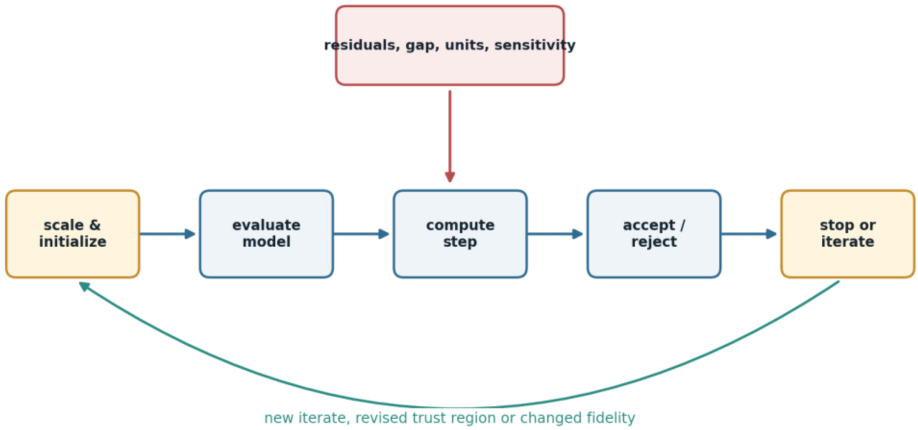


Figure F.8. A numerical optimization loop with scaling, model evaluation, steps, acceptance, and explicit evidence.

Unit 30 · Global, derivative-free, and surrogate optimization

Prerequisites. Functions, probability, and formulation.

Why it matters in this volume. Some objectives are rugged, discontinuous, noisy, expensive, proprietary, or available only through experiments and simulation.

Derivative-free methods compare function values, local interpolation models, direct searches, or populations. Global optimization combines upper bounds from feasible candidates with lower bounds from relaxations, partitions, or interval reasoning. Bayesian optimization uses a probabilistic surrogate and an acquisition rule to choose informative evaluations.

A surrogate changes the information problem. It should represent predictive uncertainty and be validated in the region where decisions are proposed. Global claims require explicit bounds or exhaustive structure; multiple starts alone are evidence of exploration, not proof.

$$x_{n+1} \in \arg\max_{x \in \mathcal{X}} a_n(x) \tag{F.63}$$

An acquisition function chooses the next evaluation using the current surrogate.

$$L_k \le f^* \le U_k \tag{F.64}$$

Global methods narrow lower and upper bounds on the optimum.

Worked reading. When a laboratory experiment takes a day, selecting the next experiment can matter more than optimizing a cheap algebraic subproblem. The objective includes information value and experimental risk.

Common traps.

- Calling the best sampled point globally optimal
- Using a surrogate outside its evidence region
- Ignoring observation noise when comparing candidates

Where it reappears. Chapter 5, biomedical discovery, materials, manufacturing, creative search, experiments, and digital twins.

Sources. [F9; F17]

Unit 31 · Multiple objectives and Pareto reasoning

Prerequisites. Functions, sets, and formulation.

Why it matters in this volume. Real decisions often preserve several values that cannot be reduced without a normative choice.

A point dominates another when it is no worse in every objective and better in at least one, with directions stated consistently. Nondominated points form a Pareto set; their objective images form a Pareto front. Scalarization converts a vector objective into one comparison but introduces weights, reference points, or constraints.

Weighted sums can miss nonconvex parts of a front. Normalization changes the meaning of weights. A knee is a region where small improvement in one objective requires large sacrifice in another, but knee identification itself depends on scale and geometry.

$$\min_{x \in \mathcal{X}} \big(f_1(x), \dots, f_m(x)\big) \tag{F.65}$$

A vector objective preserves several criteria rather than imposing a total order.

$$\min_{x \in \mathcal{X}} \sum_{i=1}^{m} w_i \, f_i(x), \qquad w_i \geq 0 \tag{F.66}$$

A weighted sum is one explicit scalarization.

Worked reading. Power, performance, area, temperature, yield, and verification effort do not share natural units. A transparent front plus stakeholder deliberation is often more honest than a hidden composite score.

Common traps.

- Calling a nondominated point uniquely optimal
- Comparing weights before normalizing objectives
- Treating preference elicitation as purely technical

Where it reappears. Chapter 6 and nearly every domain chapter, especially hardware, health allocation, policy, sustainability, art, music, and literature.

Sources. [F18]

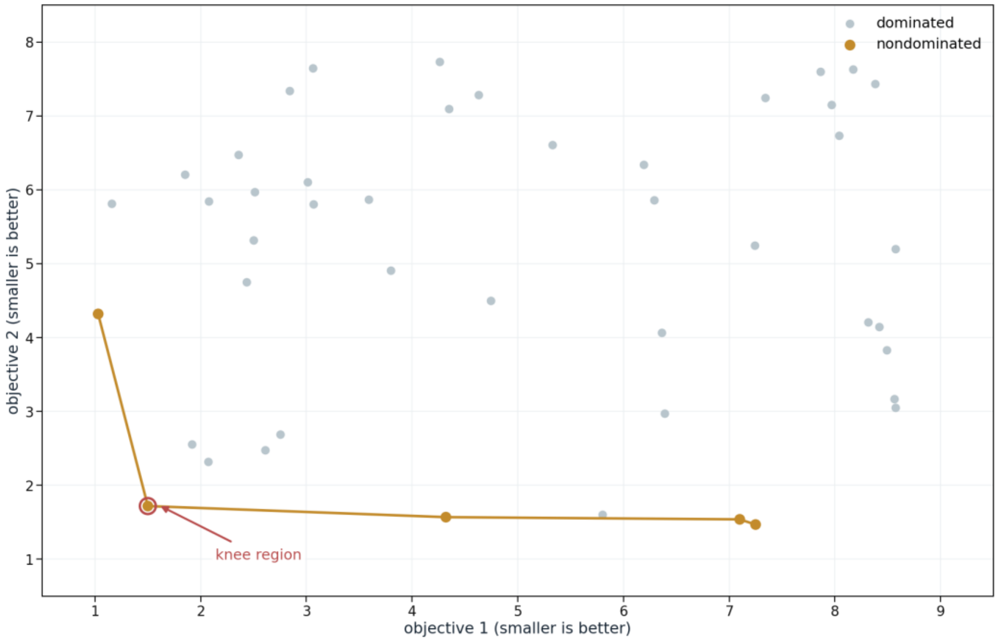


Figure F.9. Dominated and nondominated alternatives on a two-objective Pareto front.

Part F.6 · Uncertainty, time, and strategic interaction

Unit 32 · Expected value, risk, robustness, and distributional ambiguity

Prerequisites. Probability, optimization, and convexity.

Why it matters in this volume. Decision makers may care about average performance, tails, worst cases, adaptability, or ambiguity about the probability law.

Stochastic optimization minimizes an expectation or another distributional functional. Risk measures such as CVaR emphasize poor outcomes. Robust optimization protects against every parameter in an uncertainty set. Distributionally robust optimization protects against every probability law in an ambiguity set. These are different attitudes and should not be conflated.

Worst-case protection can be conservative when the set is too broad and unsafe when it omits plausible joint extremes. Scenario methods approximate distributions or uncertainty sets and require sampling and confidence statements.

$$\min_x \mathbb{E}[f(x,\xi)] \tag{F.67}$$

Expected-value optimization averages over uncertain ξ.

$$\mathrm{CVaR}_\alpha(L) = \min_\eta \left(\eta + \frac{1}{1-\alpha}\mathbb{E}[(L-\eta)_+]\right) \tag{F.68}$$

CVaR averages losses beyond an optimized threshold.

$$\min_x \max_{\xi \in \mathcal{U}} f(x,\xi) \tag{F.69}$$

Robust optimization protects against every modeled uncertainty in U.

Worked reading. Two designs can have equal expected cost while one has a small probability of catastrophic loss. Tail-sensitive or hard-safety constraints expose the difference that the mean suppresses.

Common traps.

- Calling a robust solution risk-free
- Choosing an uncertainty set without calibration
- Treating CVaR as the same as maximum loss

Where it reappears. Chapter 6; safety, security, medicine, finance, energy, climate, manufacturing, and every worked translation.

Sources. [F17; F19; F20]

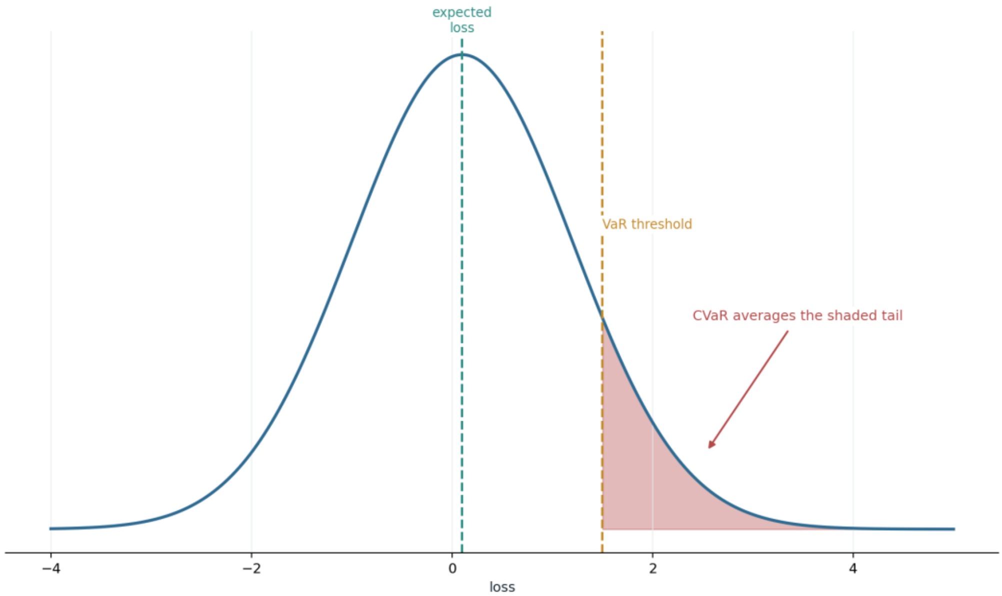


Figure F.10. Expected loss, a value-at-risk threshold, and the tail averaged by conditional value at risk.

Unit 33 · Dynamic programming, optimal control, MPC, and reinforcement learning

Prerequisites. Sequences, probability, differential equations, and optimization.

Why it matters in this volume. When actions change future states, present choices must include downstream consequences and information.

Dynamic programming uses the principle of optimality: after the first action, the remaining policy must be optimal for the resulting state under the same model. Bellman equations express value recursively. Optimal control applies this reasoning to dynamical systems. Model predictive control repeatedly solves a finite-horizon problem, applies the first action, observes, and replans.

Reinforcement learning estimates value functions, models, or policies from interaction. Exploration changes the data-generating process and may carry real risk. State choice, reward design, discounting, off-policy evaluation, and distribution shift determine what is learned.

$$V_t(s) = \min_a (c_t(s,a) + \mathbb{E}[V_{t+1}(S') \mid s,a]) \tag{F.70}$$

Bellman recursion combines immediate cost with optimal continuation value.

$$J(\pi) = \mathbb{E}_\pi[\textstyle\sum_{t=0}^{\infty} \gamma^t r_t], 0 \leq \gamma < 1, |r_t| \leq R < \infty. \tag{F.71}$$

A policy is evaluated by expected discounted return.

Worked reading. A controller with a perfect short-horizon model can still create long-horizon damage if the terminal cost and constraints omit slow state variables. Receding-horizon feedback does not repair a misspecified objective automatically.

Common traps.

- Treating reward as the same as purpose
- Assuming the Markov state is sufficient
- Deploying exploration without safety constraints and oversight

Where it reappears. Chapter 7, cloud scheduling, robotics, control, adaptive care, climate pathways, games, and sport training.

Sources. [F21; F22; F23; F24]

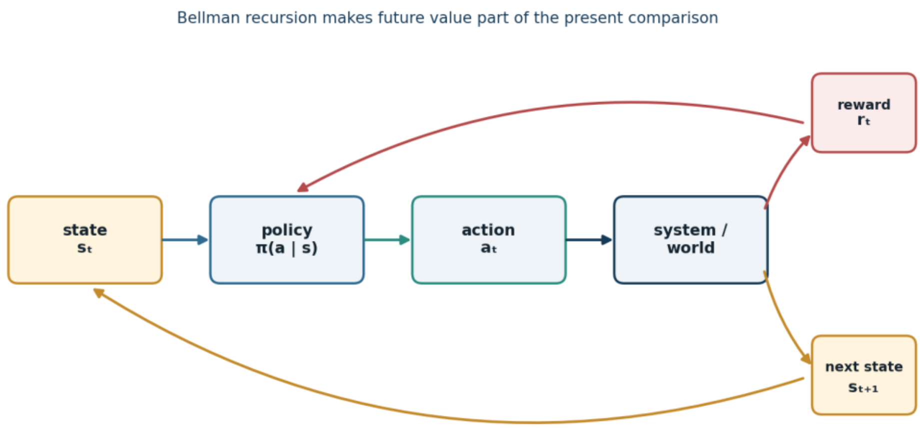


Figure F.11. State, policy, action, system, reward, and next-state feedback in Bellman recursion.

Unit 34 · Games, Nash equilibrium, mechanisms, and bilevel problems

Prerequisites. Probability, optimization, and logic.

Why it matters in this volume. Other agents adapt. Policies, markets, security controls, recommendations, and competitions can change the behavior that generates outcomes.

A game specifies players, strategies, information, and payoffs. A Nash equilibrium is a strategy profile where no player benefits from unilateral deviation. It predicts mutual best response under the model, not fairness, efficiency, uniqueness, or social desirability.

Mechanism design chooses rules so strategic responses produce desired properties. Bilevel optimization represents a leader choosing while anticipating a follower's optimization. The follower's problem may be set-valued, discontinuous, or misspecified.

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \quad \forall s_i \tag{F.72}$$

No player improves by deviating alone from a Nash equilibrium.

$$\min_x F\big(x, y^*(x)\big) \quad \text{where} \quad y^*(x) \in \arg\min_y f(x, y) \tag{F.73}$$

A leader anticipates a follower's optimized response.

Worked reading. A congestion-pricing policy changes route choices; evaluating it with fixed historical traffic ignores the response it is designed to induce. Equilibrium modeling helps, but behavioral and institutional evidence remains necessary.

Common traps.

- Equating equilibrium with optimum
- Assuming players know and optimize the modeled payoff
- Ignoring multiple follower optima in a bilevel model

Where it reappears. Chapter 6; security; markets and finance; policy and law; media; games and sport; Goodhart effects.

Sources. [F25]

Unit 35 · Queueing, reliability, and resource systems

Prerequisites. Probability, stochastic processes, and rates.

Why it matters in this volume. Latency, congestion, availability, maintenance, hospital flow, transport, and service capacity depend on arrival, service, and failure processes.

A queue couples random arrivals with finite service. Utilization near one can produce rapidly growing delay even when average capacity exceeds average demand. Little's law relates long-run average number in system, throughput, and average time under broad stability and conservation conditions.

Reliability composes component failure behavior through system structure. Series systems fail if any required component fails; redundant systems require dependence and common-cause analysis. Availability includes repair and restoration, not only failure probability.

$$L = \lambda W \tag{F.74}$$

Average number equals throughput times average time under Little's-law conditions.

$$A = \frac{\mathrm{MTBF}}{\mathrm{MTBF}+\mathrm{MTTR}} \tag{F.75}$$

A simple steady-state availability ratio combines mean up and repair times.

Worked reading. At 95% utilization, a small demand forecast error can create a large delay error. Capacity planning should compare distributions and tail service objectives, not only average work per average capacity.

Common traps.

- Using Little's law without a stable flow boundary
- Adding independent failure probabilities when failures share causes
- Equating utilization with user-visible service quality

Where it reappears. Chapters 8, 10–13, 18, and 19: cloud systems, hardware, manufacturing, transport, health systems, and public services.

Sources. [F26; F27]

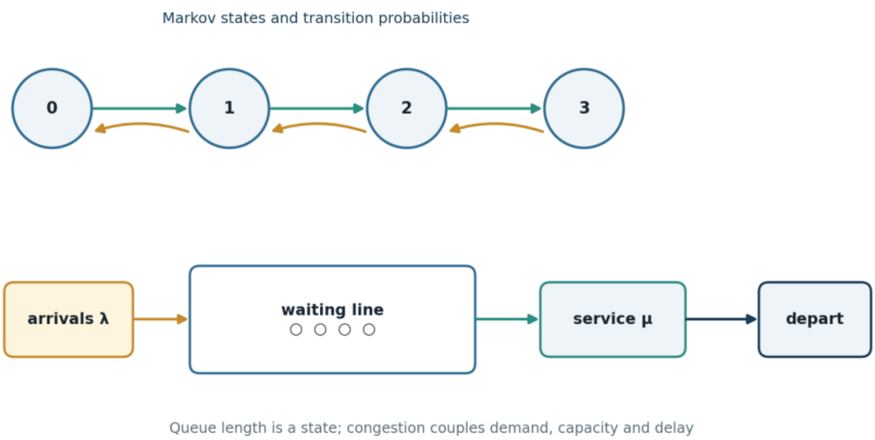


Figure F.12. Markov transitions and a queue with arrivals, waiting, service, and departure.

Part F.7 · Advanced bridges used in the volume

Unit 36 · Metric spaces and topology

Prerequisites. Sets, functions, and distance.

Why it matters in this volume. Optimization in curves, distributions, shapes, manifolds, and infinite-dimensional spaces needs language beyond ordinary coordinates.

A metric space supplies a distance and therefore notions of convergence, openness, and continuity. Topology retains neighborhood and continuity structure even when no particular distance is privileged. Compactness supports existence results by preventing minimizing sequences from escaping without a convergent subsequence.

Connectedness distinguishes one-piece feasible regions from separated components. Homeomorphism preserves topological structure, not lengths or angles. Topology optimization in engineering uses the word in a related but application-specific sense: material connectivity may change during design.

$$d(x,z) \leq d(x,y) + d(y,z) \tag{F.76}$$

A metric obeys the triangle inequality.

$$x_n \to x \quad \Leftrightarrow \quad d(x_n, x) \to 0 \tag{F.77}$$

Metric convergence reduces to distance tending to zero.

Worked reading. A feasible set can be bounded in a coordinate plot yet fail to include its limiting boundary. Then an infimum may exist without an attainable minimum. Closedness and compactness clarify the distinction.

Common traps.

- Assuming every topology comes from the pictured Euclidean distance
- Equating boundedness with compactness in arbitrary spaces
- Using 'topology' without distinguishing mathematical from engineering usage

Where it reappears. Chapters 4, 7, 11, 15, and 20: manifolds, transport, topology optimization, creative spaces, and frontiers.

Sources. [F28]

Unit 37 · Function spaces, dual spaces, and infinite-dimensional optimization

Prerequisites. Vectors, norms, calculus, and topology.

Why it matters in this volume. Policies, signals, trajectories, fields, probability densities, and shapes are functions, so their optimization lives in a space of functions.

A normed function space treats functions as vectors. Completeness ensures Cauchy sequences converge within the space. A Hilbert space has an inner product and supports orthogonal projection. A dual space consists of continuous linear functionals; gradients are naturally dual objects before geometry identifies them with vectors.

Linear operators map functions to functions or numbers. Adjoints reverse an operator relative to inner products and make sensitivities computable without differentiating once per parameter. Weak convergence can retain boundedness and existence where strong convergence is unavailable.

$$\langle f, g\rangle_{L^2} = \int f(t)g(t)\,dt \tag{F.78}$$

The L² inner product generalizes Euclidean dot products to functions.

$$\langle Ax, y\rangle = \langle x, A^* y\rangle \tag{F.79}$$

The adjoint transfers an operator across an inner product.

Worked reading. In PDE-constrained design, one adjoint solve can produce the gradient with respect to many parameters. The result remains a derivative of the discretized or continuous model chosen, so adjoint verification is still required.

Common traps.

- Treating all function spaces as interchangeable
- Calling a covector a vector without stating the pairing
- Ignoring boundary conditions and regularity

Where it reappears. Chapter 4, signals and control, inverse design, PDE models, optimal transport, and future differentiable systems.

Sources. [F14]

Unit 38 · Optimal transport and geometry of distributions

Prerequisites. Probability, optimization, metrics, and linear programming.

Why it matters in this volume. Optimal transport compares distributions by the cost of moving mass rather than comparing values point by point.

A coupling is a joint distribution with prescribed marginals. The transport problem chooses a coupling minimizing expected movement cost. Wasserstein distance arises from particular costs and metric assumptions. Entropic regularization makes computation smoother and often faster while spreading mass and changing the exact problem.

Transport supports domain adaptation, matching, imaging, generative models, allocation, and geometric comparison. The units of the cost matter: moving probability mass through physical distance differs from moving semantic mass through an embedding.

$$\min_{\gamma\in\Pi(\mu,\nu)} \int c(x,y)\,d\gamma(x,y) \tag{F.80}$$

Choose a coupling with marginals μ and ν that minimizes movement cost.

$$W_p(\mu,\nu) = \left(\inf_{\gamma\in\Pi(\mu,\nu)} \int d(x,y)^p\,d\gamma\right)^{1/p} \tag{F.81}$$

Wasserstein distance is optimal p-power transport cost converted back to distance units.

Worked reading. Two histograms with nearby shifted peaks can be far under pointwise difference but close under transport. That usefulness depends on the ground distance representing meaningful movement.

Common traps.

- Using an arbitrary embedding distance as ground truth
- Forgetting marginal constraints
- Reporting an entropically regularized value as the exact unregularized optimum

Where it reappears. Chapter 7 and applications to data science, imaging, matching, distribution shift, and cross-domain translation.

Sources. [F29; F30]

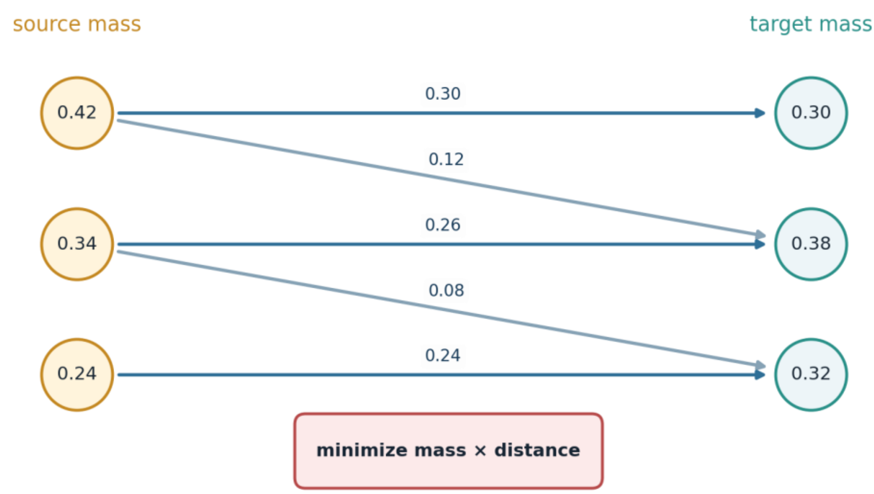


Figure F.13. A transport coupling moving source mass to target mass while minimizing mass-weighted distance.

Unit 39 · Information, entropy, and divergence

Prerequisites. Probability and logarithms.

Why it matters in this volume. Compression, uncertainty, learning, privacy, exploration, communication, and model comparison use information-theoretic quantities.

Shannon entropy is expected self-information. It is largest for a uniform distribution over a finite set and zero for a certain outcome. Cross-entropy scores predictions against observed distributions. Kullback–Leibler divergence measures directed information discrepancy and is not a metric: it is asymmetric and can be infinite.

Mutual information measures dependence by comparing a joint distribution with the product of marginals. Information measures depend on representation, discretization, and reference distribution; they do not automatically measure meaning or social value.

$$H(X) = -\sum_x p(x) \log p(x) \tag{F.82}$$

Entropy averages the information content of outcomes.

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)} \tag{F.83}$$

KL divergence penalizes coding P with a model Q.

Worked reading. A highly entropic message stream can be random noise or rich signal. Entropy quantifies distributional unpredictability, not semantic usefulness, without an additional model.

Common traps.

- Calling KL divergence a distance
- Comparing entropy values computed with different log bases without converting units
- Equating statistical information with meaning

Where it reappears. Chapters 7–9, signals and communication, privacy, active learning, Bayesian optimization, and creative/narrative domains.

Sources. [F31; F32]

Unit 40 · Category theory and compositional optimization

Prerequisites. Sets, functions, logic, and linear maps.

Why it matters in this volume. Category theory provides language for assembling optimization components while keeping interfaces and transformations explicit.

A category has objects, morphisms between objects, identity morphisms, and associative composition. Functions form a category of sets; linear maps form a category of vector spaces; processes can form categories when interfaces compose. A functor maps objects and morphisms while preserving identity and composition. A natural transformation compares functors coherently across every object.

A monoidal product models parallel combination; when the monoidal structure is cartesian, it is the categorical product. Composition models sequential connection. An optimization component can expose inputs, outputs, constraints, costs, states, and certificates. Composition is sound only when interfaces and semantics align; categorical notation does not make an invalid translation valid.

$$h \circ (g \circ f) = (h \circ g) \circ f \tag{F.84}$$

Associativity makes multi-stage composition unambiguous.

$$F(g \circ f) = F(g) \circ F(f), \qquad F(1_X) = 1_{F(X)} \tag{F.85}$$

A functor preserves composition and identities.

$$\eta_Y \circ F(f) = G(f) \circ \eta_X \tag{F.86}$$

Naturality states coherent comparison across a morphism f.

Worked reading. A model-to-solver pipeline is not automatically a functor: changing representation must preserve the relevant feasible decisions and objective meaning. If a compilation drops a constraint, composition has failed semantically even when types match.

Common traps.

- Treating analogy as isomorphism
- Assuming composability from matching data types alone
- Using categorical vocabulary without specifying objects and morphisms

Where it reappears. Chapter 7, compositional systems, software/hardware co-design, workflows, evidence chains, and future automated science.

Sources. [F33; F34; F35]

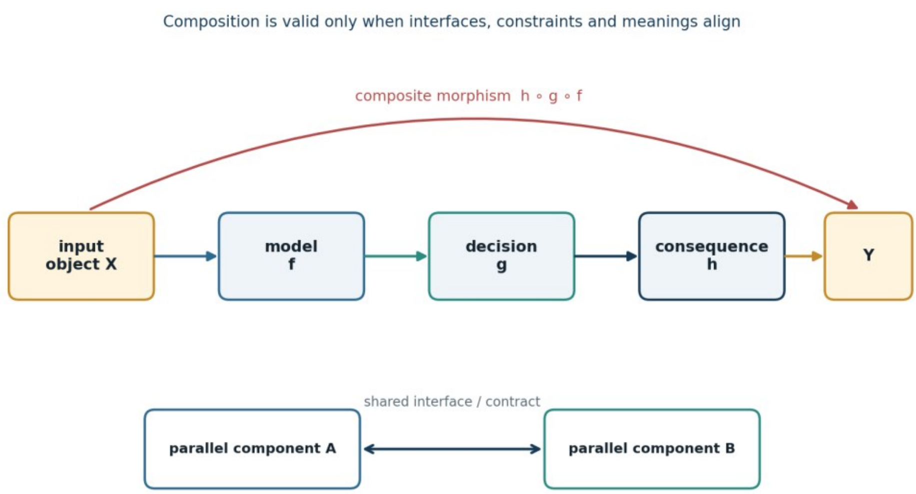


Figure F.14. Sequential and parallel composition of optimization components with explicit interfaces.

Unit 41 · Complexity, approximation, and certificates

Prerequisites. Logic, combinatorics, algorithms, and optimization.

Why it matters in this volume. A mathematical answer is operational only when it can be computed, checked, and delivered within resource and decision limits.

Time complexity counts operations as input size grows; space complexity counts memory. Big-O is an asymptotic upper bound, not a stopwatch prediction. Complexity classes describe families of decision problems and reductions. NP-hardness concerns worst-case scalable exact solution, not impossibility of solving every useful instance.

Approximation algorithms provide explicit quality ratios or additive bounds. Randomized algorithms provide probability statements. Optimization certificates include feasible witnesses, dual bounds, residuals, gaps, proof logs, or verified enclosures. Validation evidence addresses whether the model and data support deployment—the certificate and validation answer different questions.

$$T(n) \in O(n^2) \tag{F.87}$$

Runtime grows no faster than a constant multiple of n^2 beyond some threshold.

$$f(\hat{x}) \leq \alpha f(x^*), \qquad \alpha \geq 1 \tag{F.88}$$

For nonnegative minimization, an approximation ratio $\alpha \geq 1$ bounds the achieved objective relative to the optimum.

$$\text{gap} = U - L \tag{F.89}$$

Upper and lower bounds quantify remaining objective uncertainty.

Worked reading. A solver can produce a tiny primal-dual gap for a model whose data are obsolete. Conversely, a valuable policy may have strong field evidence without a formal optimality certificate. Trustworthy practice records both layers.

Common traps.

- Reading asymptotic notation as an exact runtime
- Calling NP-hardness a reason not to model structure
- Confusing optimality evidence with real-world validity

Where it reappears. Chapters 2, 4, 5, 7–10, 17, and 20; approximation, verification, stopping rules, and computational frontiers.

Sources. [F6; F7]

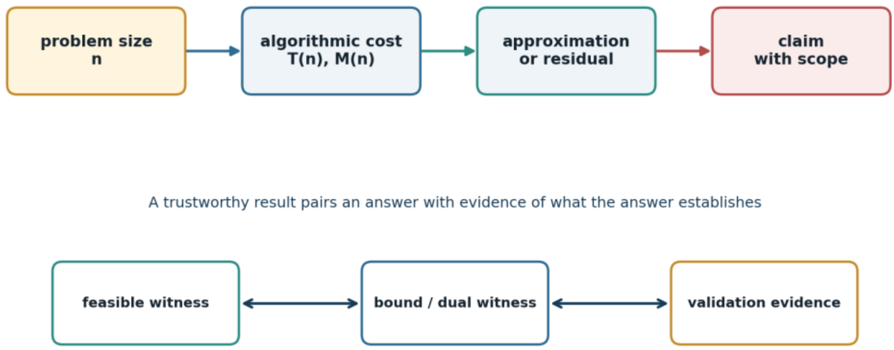


Figure F.15. Problem size, computational cost, approximation, scoped claims, and complementary certificate types.

Part F.8 · Cross-disciplinary mathematical bridges

Unit 42 · Reading mathematics across domains

Prerequisites. Units 1–41.

Why it matters in this volume. The same formal pattern may travel across disciplines, but its variables, evidence, units, and ethical meaning do not travel automatically.

In cloud capacity, Little's law connects throughput, time, and concurrent work, while a tail constraint protects service quality. In machine learning, a regularized loss balances fit with an inductive bias whose operational meaning must be defended. In control, state equations and Bellman recursion connect present action to future consequence. In medicine, Bayes' rule updates risk while causal assumptions determine treatment effects.

In finance, covariance and tail risk distinguish diversification from average return. In structural and semiconductor design, matrices, PDEs, constraints, and integer choices couple physics with manufacturability. In music, distances and constraints can describe voice leading without reducing artistic value to a single score. In literature, graphs and orderings can illuminate narrative alternatives without asserting that interpretation is a numeric optimum.

$$\text{formal similarity} \neq \text{semantic identity} \tag{F.90}$$

A shared equation is a translation aid, not proof that domains are the same.

Worked reading. For every translation, identify the object, morphism, preserved structure, lost structure, evidence, and decision authority. This is the categorical discipline of asking whether an analogy is merely a mapping or a stronger structure-preserving correspondence.

Common traps.

- Copying a method without its assumptions
- Suppressing units and stakeholders during translation
- Confusing a useful model with a complete account of the domain

Where it reappears. All domain chapters and the seven worked translations in Chapters 18 and 19.

Sources. [F6; F7; F9; F10; F11; F12; F13; F14; F15; F16; F17; F18; F19; F20; F21; F22; F23; F24; F25; F26; F27; F28; F29; F30; F31; F32; F33; F34; F35]

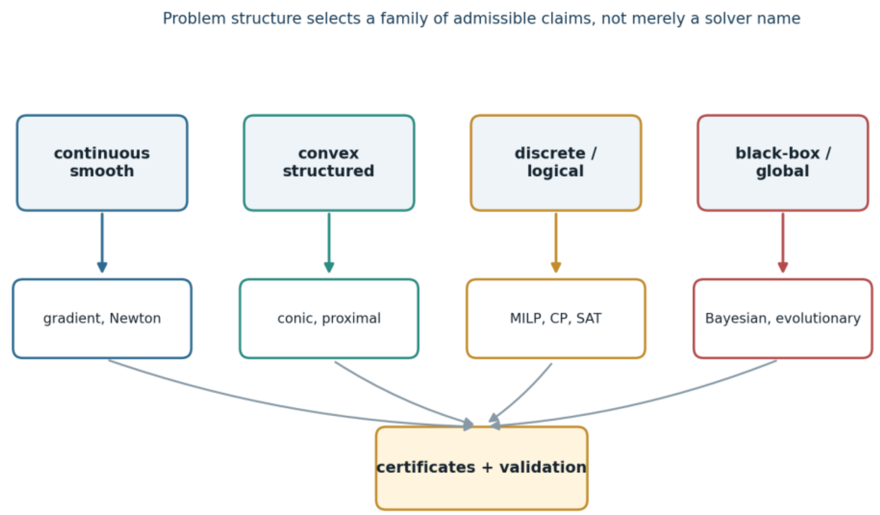


Figure F.16. Problem structures mapped to algorithm families, certificates, and validation.

Part F.9 · Equation atlas for the volume

The atlas follows the volume's unchanged equation sequence. 'Volume equation' means the numbered equation already present in the main text. Atlas displays reproduce the mathematical statement for study; the label at right uses **V.1–V.124** so it cannot be confused with a new appendix equation. Local context governs symbol meaning; the notation concordance records recurring conventions rather than overriding local definitions.

Volume equation 1 · Preface: Optimization as a shared language

Location. Preface: Optimization as a shared language.

Prerequisites. Units 23–31.

Read it as. Choose the subscripted variable or policy from its admissible set to attain the displayed best value.

Recurring symbols. mathcal X: a feasible or decision space; x: a decision, state, or generic variable; local text determines the role; f: an objective or generic function

$$x^* \in \arg\min_{x\in\mathcal{X}} f(x). \tag{V.1}$$

Context in the volume. This is the seed from which much of the book grows. Yet the seed does not tell us who chose f, how $\mathcal{X}$ was constructed, whether measurements are reliable, whether other people respond strategically, or whether the problem changes while it is being solved. Those questions connect optimization to statistics, control, economics, ethics, politics, design, and interpretation.

Structural tags. optimization.

Volume equation 2 · Bird's-eye view

Location. Bird's-eye view.

Prerequisites. Units 22 and 33.

Read it as. The next iterate or state is computed from the current quantities according to the displayed update rule.

Recurring symbols. t: time or a continuous index; P: probability, transition matrix, or distribution

$$W_t \overset{M}{\to} P_t \overset{A}{\to} S_t \overset{I}{\to} W_{t+1} \overset{E}{\to} \widehat{W}_{t+1}, \tag{V.2}$$

Context in the volume. where W_t is the world at time t, M is modeling, P_t is the formalized problem, A is an algorithm or deliberative procedure, S_t is a selected solution, I is implementation, and E is observation or evaluation. The next problem is not necessarily the old problem with new data: implementation may change incentives, institutions, technologies, populations, meanings, and available actions.

Structural tags. dynamics/control.

Volume equation 3 · The master form

Location. Bird's-eye view — The master form.

Prerequisites. Units 23–31, Unit 32.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. pi: a policy, probability, or mathematical constant; rho: a risk/aggregation functional, utilization, or correlation; theta: a parameter, mixture weight, or preference parameter; xi: an uncertain quantity or scenario; x: a decision, state, or generic variable; local text determines the role; F: a vector-valued map, aggregate objective, or transition model

Context in the volume. A broad mathematical template is

$$\min_{x,\pi} \rho(F(x,\pi,\xi);\, \theta) \tag{V.3}$$

Structural tags. optimization, uncertainty/risk.

Volume equation 4 · The master form

Location. Bird's-eye view — The master form.

Prerequisites. Units 1–7, Unit 32.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. pi: a policy, probability, or mathematical constant; xi: an uncertain quantity or scenario; g: an inequality constraint or generic function; i: an item, component, sample, or constraint index; x: a decision, state, or generic variable; local text determines the role; m: a count, dimension, or slope when defined locally

Context in the volume. … subject to

$$g_i(x,\pi,\xi) \le 0, \qquad i = 1, \dots, m, \tag{V.4}$$

Structural tags. school algebra, uncertainty/risk.

Volume equation 5 · The master form

Location. Bird's-eye view — The master form.

Prerequisites. Units 1–7, Unit 32.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. pi: a policy, probability, or mathematical constant; xi: an uncertain quantity or scenario; h: an equality constraint or generic function; j: a second index, commonly for equality constraints or states; x: a decision, state, or generic variable; local text determines the role; p: probability, dimension, exponent, density, or primal value according to context

Context in the volume. … subject to

$$h_j(x,\pi,\xi) = 0, \qquad j = 1, \dots, p, \tag{V.5}$$

Structural tags. school algebra, uncertainty/risk.

Volume equation 6 · The master form

Location. Bird's-eye view — The master form.

Prerequisites. Units 1–7.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. pi: a policy, probability, or mathematical constant; mathcal X: a feasible or decision space; x: a decision, state, or generic variable; local text determines the role; R: a reward, requirement, relation, or real-number set; k: an iteration, time step, or generic index

$$x \in \mathcal{X}, \qquad \pi \in \Pi, \qquad R_k(x,\pi) \geq r_k. \tag{V.6}$$

Context in the volume. Here x is a design or allocation, π is a policy, ξ represents uncertainty, F is a vector of consequences, ρ aggregates or orders those consequences using parameters θ, g_i and h_j encode inequality and equality constraints, and $R_k \geq r_k$ represents rights, resilience, reliability, or other non-negotiable requirements. Different disciplines instantiate the same form differently; some reject the aggregation ρ and preserve a set of incomparable alternatives.

Structural tags. mathematical notation.

Volume equation 7 · Purpose, alternatives, and decision systems

Location. Chapter 1 · Purpose, alternatives, and the grammar of optimization — Purpose, alternatives, and decision systems.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted variable or policy from its admissible set to attain the displayed best value.

Recurring symbols. mathcal X: a feasible or decision space; x: a decision, state, or generic variable; local text determines the role; f: an objective or generic function; y: an output, response, path, or second variable

$$\arg\min_{x \in \mathcal{X}} f(x) = \{x \in \mathcal{X} : f(x) \leq f(y) \text{ for all } y \in \mathcal{X}\}. \tag{V.7}$$

Context in the volume. The notation distinguishes the *optimal value* $f^* = \inf_{x\in\mathcal{X}} f(x)$ from the *optimizer* x^*. The distinction matters: a best value may be approached but never attained; many different candidates may attain it; or the infimum may be $-\infty$. Existence, uniqueness, and computability are separate questions.

Structural tags. school algebra, optimization.

Volume equation 8 · Purpose, alternatives, and decision systems

Location. Chapter 1 · Purpose, alternatives, and the grammar of optimization — Purpose, alternatives, and decision systems.

Prerequisites. Units 18–22, Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. mathbb E: expectation under a stated probability law; t: time or a continuous index; L: a Lagrangian integrand, loss, queue length, or bound; x: a decision, state, or generic variable; local text determines the role

$$\min_{a,t} \mathbb{E}\left[L\left(x_{a,t}\right)\right] + c(a,t), \tag{V.8}$$

Context in the volume. than to minimizing L alone. This is why approximation algorithms, anytime methods, heuristics, and satisficing are principled tools rather than merely inferior substitutes. Simon's bounded rationality treats decision procedures as adapted to limited information and computation, not as defective copies of omniscient maximization.

Structural tags. probability/statistics, optimization.

Volume equation 9 · Objectives, constraints, values, and authority

Location. Chapter 1 · Purpose, alternatives, and the grammar of optimization — Objectives, constraints, values, and authority.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. F: a vector-valued map, aggregate objective, or transition model; x: a decision, state, or generic variable; local text determines the role; f: an objective or generic function; k: an iteration, time step, or generic index

Context in the volume. Suppose a decision x produces a consequence vector

$$F(x) = \left(f_1(x), f_2(x), \dots, f_k(x)\right). \tag{V.9}$$

Structural tags. school algebra, optimization.

Volume equation 10 · Objectives, constraints, values, and authority

Location. Chapter 1 · Purpose, alternatives, and the grammar of optimization — Objectives, constraints, values, and authority.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Add the indexed contributions over the stated range, keeping each weight, unit, and constraint interpretation explicit.

Recurring symbols. sum: summation over the displayed index range; J: an objective functional, Jacobian, or performance criterion; x: a decision, state, or generic variable; local text determines the role; i: an item, component, sample, or constraint index; k: an iteration, time step, or generic index; f: an objective or generic function

$$J(x) = \sum_{i=1}^{k} w_i \, f_i(x), \qquad w_i \geq 0, \tag{V.10}$$

Context in the volume. looks neutral but makes substantive commitments. Units must be normalized; weights encode trade-offs; nonlinearities and thresholds may be lost; and some Pareto-efficient points cannot be recovered by weighted sums when the attainable objective set is nonconvex. Alternatives include lexicographic priority, goal programming, reference-point methods, outranking, constrained optimization, and deliberative selection from a Pareto set.

Structural tags. school algebra, optimization.

Volume equation 11 · Objectives, constraints, values, and authority

Location. Chapter 1 · Purpose, alternatives, and the grammar of optimization — Objectives, constraints, values, and authority.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; f: an objective or generic function; g: an inequality constraint or generic function; i: an item, component, sample, or constraint index; h: an equality constraint or generic function; j: a second index, commonly for equality constraints or states

Context in the volume. For a differentiable program

$$\min_{x} f(x) \quad \text{subject to} \quad g_i(x) \leq 0,\ h_j(x) = 0, \tag{V.11}$$

Structural tags. school algebra, optimization.

Volume equation 12 · Objectives, constraints, values, and authority

Location. Chapter 1 · Purpose, alternatives, and the grammar of optimization — Objectives, constraints, values, and authority.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Add the indexed contributions over the stated range, keeping each weight, unit, and constraint interpretation explicit.

Recurring symbols. lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; nu: an equality multiplier or probability measure; sum: summation over the displayed index range; x: a decision, state, or generic variable; local text determines the role; f: an objective or generic function; i: an item, component, sample, or constraint index; m: a count, dimension, or slope when defined locally; g: an inequality constraint or generic function; j: a second index, commonly for equality constraints or states; p: probability, dimension, exponent, density, or primal value according to context; h: an equality constraint or generic function

$$\mathcal{L}(x, \lambda, \nu) = f(x) + \sum_{i=1}^{m} \lambda_i \, g_i(x) + \sum_{j=1}^{p} \nu_j \, h_j(x). \tag{V.12}$$

Context in the volume. The multipliers λ_i and ν_j quantify local sensitivity to constraints under regularity conditions. Economists interpret them as shadow prices; engineers interpret them as marginal value or burden; governance should resist the temptation to interpret every multiplier as a legitimate market price.

Structural tags. school algebra, optimization.

Volume equation 13 · Search cost, baselines, and levels of intervention

Location. Chapter 1 · Purpose, alternatives, and the grammar of optimization — Search cost, baselines, and levels of intervention.

Prerequisites. Units 23–31, Unit 32.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. xi: an uncertain quantity or scenario; t: time or a continuous index; x: a decision, state, or generic variable; local text determines the role; L: a Lagrangian integrand, loss, queue length, or bound; C: a set, cost, covariance, or constant

$$\min_{a,t,x} \mathbb{E}[L(x; \xi) \mid a, t] + C_{\text{search}}(a, t) + C_{\text{delay}}(t) + C_{\text{change}}(x, x_0), \tag{V.13}$$

Context in the volume. where a is a method, t is the effort or stopping time, x_0 is the current practice, and ξ represents uncertain consequences. This formulation extends the simpler search-cost model in Equation (8) by adding delay and change costs. It explains why the most sophisticated solver is not automatically the best intervention. A fast transparent approximation can dominate an exact but delayed answer; an existing safe practice can dominate a fragile improvement whose transition cost is ignored. This is bounded rationality made explicit rather than treated as an embarrassment.

Structural tags. optimization, uncertainty/risk.

Volume equation 14 · Representation, complexity, and equivalent decisions

Location. Chapter 2 · Problem structure, taxonomy, and historical foundations — Representation, complexity, and equivalent decisions.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. sum: summation over the displayed index range; x: a decision, state, or generic variable; local text determines the role; y: an output, response, path, or second variable; F: a vector-valued map, aggregate objective, or transition model; p: probability, dimension, exponent, density, or primal value according to context; i: an item, component, sample, or constraint index; n: sample size, dimension, horizon, or problem-size parameter; j: a second index, commonly for equality constraints or states; m: a count, dimension, or slope when defined locally; f: an objective or generic function

$$\min_{(x,y)\in\mathcal{F}_p}\left(\sum_{i=1}^{n}\sum_{j=1}^{m} d_i\, c_{ij}x_{ij}+\sum_{j=1}^{m} f_j\, y_j\right). \tag{V.14}$$

Context in the volume. The algebra exposes fixed costs, assignment costs, and coupling. Yet domain meaning still determines whether distance c_{ij} represents travel time, accessibility, cultural acceptability, network latency, or ecological disruption. The same formulation can support emergency stations, content replicas, warehouses, or clinics, but the validation evidence is different in each case. Equation (14) is therefore a reusable syntax with typed semantics.

Structural tags. school algebra, optimization.

Volume equation 15 · Measurement and the evidence chain

Location. Chapter 3 · Measurement, evidence, governance, and the optimization lifecycle — Measurement and the evidence chain.

Prerequisites. Units 1–7.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. y: an output, response, path, or second variable; m: a count, dimension, or slope when defined locally; z: an auxiliary or third variable

$$y = m(z,c)+\varepsilon, \tag{V.15}$$

Context in the volume. where m is a measurement process that depends on context c, and ε represents error. Optimizing y changes incentives around m and may change the relationship between y and z. This is the mechanism beneath Goodhart-like failures: when a proxy becomes a target, selection pressure concentrates precisely on the gap between proxy and purpose.

Structural tags. school algebra.

Volume equation 16 · Calculus, local geometry, and numerical truth

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Calculus, local geometry, and numerical truth.

Prerequisites. Units 1–7, Units 12–17, Units 23–31, Units 22 and 33.

Read it as. The next iterate or state is computed from the current quantities according to the displayed update rule.

Recurring symbols. nabla: gradient or derivative operator; x: a decision, state, or generic variable; local text determines the role; k: an iteration, time step, or generic index; f: an objective or generic function

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \tag{V.16}$$

Context in the volume. where the step size α_k may be fixed, scheduled, or chosen by line search. The equation is simple; its behavior depends on geometry. Poor scaling produces narrow valleys and oscillation. Saddle points can slow or redirect progress. Noise turns the path into a stochastic process. Constraints may require projection or a different geometry.

Structural tags. school algebra, calculus, optimization, dynamics/control.

Volume equation 17 · Calculus, local geometry, and numerical truth

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Calculus, local geometry, and numerical truth.

Prerequisites. Units 1–7, Units 12–17, Units 23–31, Units 22 and 33.

Read it as. The next iterate or state is computed from the current quantities according to the displayed update rule.

Recurring symbols. nabla: gradient or derivative operator; x: a decision, state, or generic variable; local text determines the role; k: an iteration, time step, or generic index; H: a hypothesis, Hessian, entropy, or Hamiltonian; f: an objective or generic function

$$x_{k+1} = x_k - H(x_k)^{-1} \nabla f(x_k). \tag{V.17}$$

Context in the volume. Near a well-behaved solution it can converge rapidly, but forming and solving with the Hessian can be expensive. Quasi-Newton methods such as BFGS update an approximation using gradient differences. Conjugate-gradient and Krylov methods exploit large sparse structure. Trust-region methods restrict each local model to a region where it is credible.

Structural tags. school algebra, calculus, optimization, dynamics/control.

Volume equation 18 · Calculus, local geometry, and numerical truth

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Calculus, local geometry, and numerical truth.

Prerequisites. Units 1–7, Units 12–17, Units 23–31.

Read it as. Accumulate the integrand over the displayed interval, space, path, or coupling.

Recurring symbols. int: integration over the displayed domain; J: an objective functional, Jacobian, or performance criterion; y: an output, response, path, or second variable; L: a Lagrangian integrand, loss, queue length, or bound; t: time or a continuous index

Context in the volume. Matrix variables appear in control, covariance estimation, semidefinite programming, and quantum information. Function variables appear when the decision is a curve, field, or policy. The calculus of variations considers a functional such as

$$J[y] = \int_a^b L\left(t, y(t), y'(t)\right) dt. \tag{V.18}$$

Structural tags. school algebra, calculus, optimization.

Volume equation 19 · Calculus, local geometry, and numerical truth

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Calculus, local geometry, and numerical truth.

Prerequisites. Units 1–7, Units 12–17, Units 23–31.

Read it as. The derivatives quantify local sensitivity; equality or balance supplies a stationarity or evolution condition.

Recurring symbols. partial: partial derivative symbol; y: an output, response, path, or second variable

$$\frac{\partial L}{\partial y} - \frac{d}{dt}\left(\frac{\partial L}{\partial y\prime}\right) = 0. \tag{V.19}$$

Context in the volume. This passage from optimizing points to optimizing paths underlies geodesics, mechanics, shape design, image processing, and optimal control.

Structural tags. school algebra, calculus, optimization.

Volume equation 20 · Calculus, local geometry, and numerical truth

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Calculus, local geometry, and numerical truth.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; F: a vector-valued map, aggregate objective, or transition model; f: an objective or generic function; g: an inequality constraint or generic function

Context in the volume. Many practical objectives are composite: a smooth loss plus a norm, indicator, hinge, maximum, or piecewise tariff. For

$$\min_x F(x) = f(x) + g(x), \tag{V.20}$$

Structural tags. school algebra, optimization.

Volume equation 21 · Calculus, local geometry, and numerical truth

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Calculus, local geometry, and numerical truth.

Prerequisites. Units 1–7, Units 8–10, Units 23–31.

Read it as. Choose the subscripted variable or policy from its admissible set to attain the displayed best value.

Recurring symbols. alpha: a step size, confidence/tail level, weight, or scalar; g: an inequality constraint or generic function; x: a decision, state, or generic variable; local text determines the role

$$\mathrm{prox}_{\alpha g}(v) = \arg\min_{x}\left(g(x) + \frac{1}{2\alpha} \| x - v \|_2^2\right). \tag{V.21}$$

Context in the volume. The proximal step preserves structure that smoothing can blur: sparsity, constraints, or a sharp penalty. Equation (21) appears in regularized estimation, imaging, signal recovery, and resource allocation. The translation is sound when the regularizer encodes a defensible prior or cost. It is not sound when sparsity is preferred merely because the algorithm supports it.

Structural tags. school algebra, linear algebra, optimization.

Volume equation 22 · Convexity, duality, and certificates

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Convexity, duality, and certificates.

Prerequisites. Units 23–31.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. theta: a parameter, mixture weight, or preference parameter; f: an objective or generic function; x: a decision, state, or generic variable; local text determines the role; y: an output, response, path, or second variable

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta)f(y), \qquad 0 \le \theta \le 1. \tag{V.22}$$

Context in the volume. The inequality says that the graph lies below its chords. Convex problems have a remarkable local-to-global property: every local minimum is global. When f is strictly convex, the minimizer is unique if it exists. Strong convexity and smoothness give quantitative convergence rates and condition numbers.

Structural tags. optimization.

Volume equation 23 · Convexity, duality, and certificates

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Convexity, duality, and certificates.

Prerequisites. Units 1–7, Units 23–31.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; nu: an equality multiplier or probability measure; q: a dual function, probability, or auxiliary quantity; x: a decision, state, or generic variable; local text determines the role

$$q(\lambda,\nu)=\inf_{x}\mathcal{L}(x,\lambda,\nu),\qquad \lambda\geq 0. \tag{V.23}$$

Context in the volume. Every dual-feasible pair produces a lower bound on a minimization problem. The dual problem maximizes this bound:

Structural tags. school algebra, optimization.

Volume equation 24 · Convexity, duality, and certificates

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Convexity, duality, and certificates.

Prerequisites. Units 23–31.

Read it as. Choose the subscripted decision so the displayed quantity is as large as the constraints allow.

Recurring symbols. lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; nu: an equality multiplier or probability measure; q: a dual function, probability, or auxiliary quantity

$$\max_{\lambda\geq 0,\nu} q(\lambda,\nu). \tag{V.24}$$

Context in the volume. Weak duality always gives $d^{*}\leq p^{*}$. For a convex problem satisfying Slater's condition, strong duality gives $d^{*}=p^{*}$. More generally, strong duality requires hypotheses that must be stated rather than inferred from convexity alone. A matching primal solution and dual bound form an optimality certificate.

Structural tags. optimization.

Volume equation 25 · Convexity, duality, and certificates

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Convexity, duality, and certificates.

Prerequisites. Units 1–7, Units 12–17, Units 23–31.

Read it as. The derivatives quantify local sensitivity; equality or balance supplies a stationarity or evolution condition.

Recurring symbols. nabla: gradient or derivative operator; sum: summation over the displayed index range; f: an objective or generic function; x: a decision, state, or generic variable; local text determines the role; i: an item, component, sample, or constraint index; m: a count, dimension, or slope when defined locally; g: an inequality constraint or generic function; j: a second index, commonly for equality constraints or states; p: probability, dimension, exponent, density, or primal value according to context; h: an equality constraint or generic function

Context in the volume. For differentiable convex problems, the KKT conditions are

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \nu_j^* \nabla h_j(x^*) = 0, \tag{V.25}$$

Structural tags. school algebra, calculus, optimization.

Volume equation 26 · Convexity, duality, and certificates

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Convexity, duality, and certificates.

Prerequisites. Units 1–7, Units 23–31.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. g: an inequality constraint or generic function; i: an item, component, sample, or constraint index; x: a decision, state, or generic variable; local text determines the role; h: an equality constraint or generic function; j: a second index, commonly for equality constraints or states

Context in the volume. For differentiable convex problems, the KKT conditions are

$$g_i(x^*) \le 0, \qquad h_j(x^*) = 0, \tag{V.26}$$

Structural tags. school algebra, optimization.

Volume equation 27 · Convexity, duality, and certificates

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Convexity, duality, and certificates.

Prerequisites. Units 1–7, Units 23–31.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. i: an item, component, sample, or constraint index; g: an inequality constraint or generic function; x: a decision, state, or generic variable; local text determines the role

$$\lambda_i^* \ge 0, \qquad \lambda_i^* g_i(x^*) = 0. \tag{V.27}$$

Context in the volume. The last condition is complementary slackness: either an inequality is inactive and its multiplier is zero, or it is active and may carry a nonzero shadow value. Even for convex problems, necessity of the KKT conditions requires an appropriate constraint qualification, such as Slater's condition in the standard differentiable inequality-constrained setting. Under convexity, KKT conditions are sufficient for global optimality when a feasible primal-dual point satisfies them; outside convexity they are generally only local necessary conditions under regularity and do not certify a global optimum.

Structural tags. school algebra, optimization.

Volume equation 28 · Convexity, duality, and certificates

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Convexity, duality, and certificates.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. f: an objective or generic function; y: an output, response, path, or second variable; x: a decision, state, or generic variable; local text determines the role

$$f^*(y) = \sup_x \big(\langle y, x\rangle - f(x)\big) \tag{V.28}$$

Context in the volume. translates between primal and dual representations. Nonsmooth regularizers often have simple proximal maps:

Structural tags. school algebra, optimization.

Volume equation 29 · Convexity, duality, and certificates

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Convexity, duality, and certificates.

Prerequisites. Units 1–7, Units 8–10, Units 23–31.

Read it as. Choose the subscripted variable or policy from its admissible set to attain the displayed best value.

Recurring symbols. gamma: a discount factor, coupling, or scalar; f: an objective or generic function; x: a decision, state, or generic variable; local text determines the role

$$\mathrm{prox}_{\gamma f}(v) = \arg\min_x \left(f(x) + \tfrac{1}{2\gamma} \parallel x - v \parallel^2\right). \tag{V.29}$$

Context in the volume. This restates the proximal map of Equation (21), using f for the generic nonsmooth term and γ for the proximal parameter. Splitting methods exploit this to decompose objectives into parts whose individual proximal operations are easy.

Structural tags. school algebra, linear algebra, optimization.

Volume equation 30 · Convexity, duality, and certificates

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Convexity, duality, and certificates.

Prerequisites. Units 23–31.

Read it as. Choose the subscripted decision so the displayed quantity is as large as the constraints allow.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role

$$\max_{x_1,x_2\geq 0} 3x_1 + 5x_2 \quad \text{subject to} \quad x_1 + 2x_2 \leq 8,\ 3x_1 + 2x_2 \leq 12. \tag{V.30}$$

Context in the volume. Its dual assigns nonnegative shadow values y_1, y_2 to the two resources:

Structural tags. optimization.

Volume equation 31 · Convexity, duality, and certificates

Location. Chapter 4 · Continuous optimization: calculus, convexity, duality, and certificates — Convexity, duality, and certificates.

Prerequisites. Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. y: an output, response, path, or second variable

$$\min_{y_1,y_2\geq 0} 8y_1 + 12y_2 \quad \text{subject to} \quad y_1 + 3y_2 \geq 3,\ 2y_1 + 2y_2 \geq 5. \tag{V.31}$$

Context in the volume. Weak duality says every dual-feasible resource valuation upper-bounds every primal-feasible production value. When primal and dual values coincide, the pair certifies global optimality. The multipliers also give local sensitivity: if a resource limit increases slightly and the active set remains stable, the corresponding dual value estimates marginal benefit. That interpretation is conditional. Large changes can alter the basis; discrete production invalidates marginal divisibility; market, environmental, or labor consequences may not be present in the objective.

Structural tags. optimization.

Volume equation 32 · Linear and conic forms as compilation targets

Location. Chapter 5 · Structured search: linear, conic, discrete, global, and derivative-free methods — Linear and conic forms as compilation targets.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role

$$\min_{x} c^T x \quad \text{subject to} \quad Ax = b,\ x \geq 0. \tag{V.32}$$

Context in the volume. Its feasible set is a polyhedron. Extreme points encode basic feasible solutions; the simplex method travels among them, while interior-point methods move through the interior. Linear programming became a general language for production, transport, diet, blending, scheduling relaxations, network flow, and resource allocation.

Structural tags. school algebra, optimization.

Volume equation 33 · Linear and conic forms as compilation targets

Location. Chapter 5 · Structured search: linear, conic, discrete, global, and derivative-free methods — Linear and conic forms as compilation targets.

Prerequisites. Units 23–31.

Read it as. Choose the subscripted decision so the displayed quantity is as large as the constraints allow.

Recurring symbols. y: an output, response, path, or second variable; A: a matrix, event, or set

$$\max_{y} b^T y \quad \text{subject to} \quad A^T y \le c. \tag{V.33}$$

Context in the volume. Primal and dual variables often express two views of one system: quantities versus marginal values, flows versus potentials, or plans versus certificates.

Structural tags. optimization.

Volume equation 34 · Linear and conic forms as compilation targets

Location. Chapter 5 · Structured search: linear, conic, discrete, global, and derivative-free methods — Linear and conic forms as compilation targets.

Prerequisites. Units 23–31.

Read it as. Interpret the displayed quadratic expression as an objective whose value is compared or minimized; the surrounding constraints determine the admissible decisions.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role

$$\frac{1}{2} x^T Q x + c^T x \tag{V.34}$$

Context in the volume. with linear constraints. If $Q \succcurlyeq 0$, the problem is convex. Quadratic programs appear in least squares, portfolio optimization, support-vector machines, model predictive control, and sequential quadratic programming. Equality-constrained QPs reduce to structured linear systems; active-set and interior-point methods address inequalities.

Structural tags. optimization.

Volume equation 35 · Linear and conic forms as compilation targets

Location. Chapter 5 · Structured search: linear, conic, discrete, global, and derivative-free methods — Linear and conic forms as compilation targets.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role

$$\min_{x} c^T x \quad \text{subject to} \quad Ax = b,\ x \in K, \tag{V.35}$$

Context in the volume. where K is a convex cone. Choosing the nonnegative orthant yields linear programming. Second-order cones express norm constraints such as $\| Ax + b \|_2 \leq c^T x + d$. The positive-semidefinite cone yields semidefinite programming (SDP), where a symmetric matrix variable X must satisfy $X \succcurlyeq 0$.

Structural tags. school algebra, optimization.

Volume equation 36 · Discrete search as proof construction

Location. Chapter 5 · Structured search: linear, conic, discrete, global, and derivative-free methods — Discrete search as proof construction.

Prerequisites. Units 23–31, Units 11 and 28.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; i: an item, component, sample, or constraint index

$$\min_x c^T x \quad \text{subject to} \quad Ax \leq b, \quad x_i \in \mathbb{Z} \text{ for } i \in I. \tag{V.36}$$

Context in the volume. Integrality transforms geometry. The continuous relaxation may have an attractive fractional solution that has no physical meaning. Exact methods combine several ideas.

Structural tags. optimization, discrete/combinatorial.

Volume equation 37 · Discrete search as proof construction

Location. Chapter 5 · Structured search: linear, conic, discrete, global, and derivative-free methods — Discrete search as proof construction.

Prerequisites. Units 1–7.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. f: an objective or generic function; A: a matrix, event, or set; B: a matrix, set, or budget

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B) \tag{V.37}$$

Context in the volume. whenever $A \subseteq B$ and $e \notin B$. Submodularity supports strong greedy approximations for coverage, sensor placement, influence, and summarization. It is a discrete analogue of convexity in some respects, though the analogy must be used carefully.

Structural tags. mathematical notation.

Volume equation 38 · Global, black-box, and surrogate search

Location. Chapter 5 · Structured search: linear, conic, discrete, global, and derivative-free methods — Global, black-box, and surrogate search.

Prerequisites. Units 1–7.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. f: an objective or generic function; y: an output, response, path, or second variable; x: a decision, state, or generic variable; local text determines the role; g: an inequality constraint or generic function

$$f(y) \geq f(x) + g^T(y - x) \quad \text{for all } y. \tag{V.38}$$

Context in the volume. Subgradient methods are robust but can be slow. Proximal-gradient methods are effective for composite objectives $f(x) + r(x)$ with smooth f and simple nonsmooth r:

Structural tags. mathematical notation.

Volume equation 39 · Global, black-box, and surrogate search

Location. Chapter 5 · Structured search: linear, conic, discrete, global, and derivative-free methods — Global, black-box, and surrogate search.

Prerequisites. Units 1–7, Units 12–17, Units 22 and 33.

Read it as. The next iterate or state is computed from the current quantities according to the displayed update rule.

Recurring symbols. alpha: a step size, confidence/tail level, weight, or scalar; nabla: gradient or derivative operator; x: a decision, state, or generic variable; local text determines the role; k: an iteration, time step, or generic index; f: an objective or generic function

$$x_{k+1} = \mathrm{prox}_{\alpha r}\big(x_k - \alpha \nabla f(x_k)\big). \tag{V.39}$$

Context in the volume. This pattern underlies sparsity-inducing regularization, constrained projections, and operator splitting.

Structural tags. school algebra, calculus, dynamics/control.

Volume equation 40 · Uncertainty, risk, and four attitudes to action

Location. Chapter 6 · Uncertainty, multiple objectives, games, and hierarchy — Uncertainty, risk, and four attitudes to action.

Prerequisites. Units 18–22, Units 23–31, Unit 32.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. xi: an uncertain quantity or scenario; mathbb E: expectation under a stated probability law; x: a decision, state, or generic variable; local text determines the role; f: an objective or generic function

$$\min_{x} \ \mathbb{E}_{\xi}[f(x, \xi)] \tag{V.40}$$

Context in the volume. is appropriate only when a probability model is credible and average performance captures the decision maker's concern. Rare harms, ambiguity, nonstationarity, and irreversibility require other criteria.

Structural tags. probability/statistics, optimization, uncertainty/risk.

Volume equation 41 · Uncertainty, risk, and four attitudes to action

Location. Chapter 6 · Uncertainty, multiple objectives, games, and hierarchy — Uncertainty, risk, and four attitudes to action.

Prerequisites. Units 18–22, Units 23–31, Unit 32.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. xi: an uncertain quantity or scenario; mathbb E: expectation under a stated probability law; x: a decision, state, or generic variable; local text determines the role; Q: a quadratic matrix, action-value function, or distribution

Context in the volume. Two-stage stochastic programming separates a here-and-now decision x from recourse $y(\xi)$ after uncertainty is observed:

$$\min_x c^T x + \mathbb{E}_\xi[Q(x,\xi)], \tag{V.41}$$

Structural tags. probability/statistics, optimization, uncertainty/risk.

Volume equation 42 · Uncertainty, risk, and four attitudes to action

Location. Chapter 6 · Uncertainty, multiple objectives, games, and hierarchy — Uncertainty, risk, and four attitudes to action.

Prerequisites. Units 1–7, Units 23–31, Unit 32.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. xi: an uncertain quantity or scenario; Q: a quadratic matrix, action-value function, or distribution; x: a decision, state, or generic variable; local text determines the role; y: an output, response, path, or second variable; q: a dual function, probability, or auxiliary quantity; h: an equality constraint or generic function

$$Q(x,\xi) = \min_y\{q(\xi)^T y : W(\xi)y = h(\xi) - T(\xi)x,\ y \geq 0\}. \tag{V.42}$$

Context in the volume. Scenario trees generalize to multiple stages but grow rapidly. Decomposition, sampling, and approximate dynamic programming manage this complexity.

Structural tags. school algebra, optimization, uncertainty/risk.

Volume equation 43 · Uncertainty, risk, and four attitudes to action

Location. Chapter 6 · Uncertainty, multiple objectives, games, and hierarchy — Uncertainty, risk, and four attitudes to action.

Prerequisites. Unit 32.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. xi: an uncertain quantity or scenario; g: an inequality constraint or generic function; x: a decision, state, or generic variable; local text determines the role

Context in the volume. A chance constraint

$$\Pr(g(x,\xi) \le 0) \ge 1-\varepsilon \tag{V.43}$$

Structural tags. uncertainty/risk.

Volume equation 44 · Uncertainty, risk, and four attitudes to action

Location. Chapter 6 · Uncertainty, multiple objectives, games, and hierarchy — Uncertainty, risk, and four attitudes to action.

Prerequisites. Units 1–7, Units 18–22, Units 23–31, Unit 32.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. alpha: a step size, confidence/tail level, weight, or scalar; eta: a threshold or auxiliary optimization variable; mathbb E: expectation under a stated probability law; L: a Lagrangian integrand, loss, queue length, or bound

$$\mathrm{CVaR}_\alpha(L) = \min_\eta \left[\eta + \tfrac{1}{1-\alpha}\mathbb{E}[(L-\eta)_+]\right]. \tag{V.44}$$

Context in the volume. Here $(z)_+ = \max(z,0)$ is the positive part. This representation makes CVaR optimization tractable in many sampled problems.

Structural tags. school algebra, probability/statistics, optimization, uncertainty/risk.

Volume equation 45 · Uncertainty, risk, and four attitudes to action

Location. Chapter 6 · Uncertainty, multiple objectives, games, and hierarchy — Uncertainty, risk, and four attitudes to action.

Prerequisites. Units 23–31, Unit 32.

Read it as. Compare a decision against an inner worst case or competing choice, respecting the order of the two optimizations.

Recurring symbols. xi: an uncertain quantity or scenario; mathcal U: an uncertainty set; x: a decision, state, or generic variable; local text determines the role; f: an objective or generic function

$$\min_{x} \max_{\xi \in \mathcal{U}} f(x, \xi). \tag{V.45}$$

Context in the volume. The set encodes epistemic judgment. If too small, protection is illusory; if too large, solutions are unusably conservative. Budgeted uncertainty provides a tunable price of robustness. Distributionally robust optimization instead considers a set $\mathcal{P}$ of plausible probability distributions:

Structural tags. optimization, uncertainty/risk.

Volume equation 46 · Uncertainty, risk, and four attitudes to action

Location. Chapter 6 · Uncertainty, multiple objectives, games, and hierarchy — Uncertainty, risk, and four attitudes to action.

Prerequisites. Units 18–22, Units 23–31, Unit 32.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. xi: an uncertain quantity or scenario; mathbb E: expectation under a stated probability law; x: a decision, state, or generic variable; local text determines the role; P: probability, transition matrix, or distribution; f: an objective or generic function

$$\min_{x} \sup_{P \in \mathcal{P}} \mathbb{E}_{P}[f(x, \xi)]. \tag{V.46}$$

Context in the volume. Ambiguity sets may be defined by moments, divergences, or optimal-transport distances.

Structural tags. probability/statistics, optimization, uncertainty/risk.

Volume equation 47 · Multiple objectives and uncertain preferences

Location. Chapter 6 · Uncertainty, multiple objectives, games, and hierarchy — Multiple objectives and uncertain preferences.

Prerequisites. Unit 32.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. i: an item, component, sample, or constraint index; x: a decision, state, or generic variable; local text determines the role

$$u_i(x_i^*, x_{-i}^*) \geq u_i(x_i, x_{-i}^*) \quad \text{for every player } i \text{ and action } x_i. \tag{V.47}$$

Context in the volume. No player can profit by unilateral deviation. For every finite game, Nash's existence theorem guarantees at least one equilibrium in mixed strategies; a pure-strategy equilibrium need not exist. Equilibrium is not the same as collective optimality, fairness, or stability under learning. Mechanism design reverses the problem: choose the rules so that strategic behavior produces desirable outcomes.

Structural tags. uncertainty/risk.

Volume equation 48 · Multiple objectives and uncertain preferences

Location. Chapter 6 · Uncertainty, multiple objectives, games, and hierarchy — Multiple objectives and uncertain preferences.

Prerequisites. Units 23–31, Unit 32.

Read it as. Choose the subscripted variable or policy from its admissible set to attain the displayed best value.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; y: an output, response, path, or second variable; F: a vector-valued map, aggregate objective, or transition model; z: an auxiliary or third variable; f: an objective or generic function; g: an inequality constraint or generic function

$$\min_{x,y} F(x,y) \quad \text{subject to} \quad y \in \arg\min_{z}\{f(x,z): g(x,z) \le 0\}. \tag{V.48}$$

Context in the volume. They model leader-follower games, pricing, hyperparameter tuning, adversarial learning, infrastructure planning, and policy response. Even linear bilevel problems can be hard because the lower-level solution map is discontinuous or set-valued.

Structural tags. optimization, uncertainty/risk.

Volume equation 49 · Decisions through time

Location. Chapter 7 · Dynamics, transport, category theory, algorithms, and verification — Decisions through time.

Prerequisites. Units 1–7, Units 22 and 33, Units 36–40.

Read it as. The next iterate or state is computed from the current quantities according to the displayed update rule.

Recurring symbols. t: time or a continuous index

$$s_{t+1} = T(s_t, a_t, w_t) \tag{V.49}$$

Context in the volume. describe a system with state s_t, action a_t, and disturbance w_t. A policy π maps observations to actions. The objective may discount future costs:

Structural tags. school algebra, dynamics/control, advanced geometry/composition.

Volume equation 50 · Decisions through time

Location. Chapter 7 · Dynamics, transport, category theory, algorithms, and verification — Decisions through time.

Prerequisites. Units 1–7, Units 18–22, Units 36–40.

Read it as. Average the displayed consequence under the stated probability law.

Recurring symbols. gamma: a discount factor, coupling, or scalar; pi: a policy, probability, or mathematical constant; mathbb E: expectation under a stated probability law; sum:

summation over the displayed index range; J: an objective functional, Jacobian, or performance criterion; t: time or a continuous index

Context in the volume. … describe a system with state s_t, action a_t, and disturbance w_t. A policy π maps observations to actions. The objective may discount future costs:

$$J^{\pi}(s_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \, c(s_t, a_t)], 0 \le \gamma < 1, |c(s,a)| \le C < \infty. \quad \text{(V.50)}$$

Structural tags. school algebra, probability/statistics, advanced geometry/composition.

Volume equation 51 · Decisions through time

Location. Chapter 7 · Dynamics, transport, category theory, algorithms, and verification — Decisions through time.

Prerequisites. Units 1–7, Units 18–22, Units 23–31, Units 36–40.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. gamma: a discount factor, coupling, or scalar; mathbb E: expectation under a stated probability law; V: a vector space or value function

$$V^*(s) = \min_a [c(s,a) + \gamma \, \mathbb{E}[V^*(s') \mid s, a]]. \quad \text{(V.51)}$$

Context in the volume. This recursive decomposition is the foundation of dynamic programming, Markov decision processes, and much reinforcement learning. Its power comes with the curse of dimensionality: the state space can grow exponentially.

Structural tags. school algebra, probability/statistics, optimization, advanced geometry/composition.

Volume equation 52 · Decisions through time

Location. Chapter 7 · Dynamics, transport, category theory, algorithms, and verification — Decisions through time.

Prerequisites. Units 1–7, Units 12–17, Units 36–40.

Read it as. Accumulate the integrand over the displayed interval, space, path, or coupling.

Recurring symbols. int: integration over the displayed domain; J: an objective functional, Jacobian, or performance criterion; x: a decision, state, or generic variable; local text determines the role; Φ: a terminal cost; L: a Lagrangian integrand, loss, queue length, or bound; t: time or a continuous index

Context in the volume. For continuous dynamics $\dot{x} = f(x, u, t)$ with terminal cost Φ and running cost L, define the objective

$$J[u] = \Phi\big(x(T)\big) + \int_0^T L\,(x(t), u(t), t)\,dt, \quad \text{(V.52)}$$

Structural tags. school algebra, calculus, advanced geometry/composition.

Volume equation 53 · Decisions through time

Location. Chapter 7 · Dynamics, transport, category theory, algorithms, and verification — Decisions through time.

Prerequisites. Units 1–7, Units 36–40.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; H: a hypothesis, Hessian, entropy, or Hamiltonian; x: a decision, state, or generic variable; local text determines the role; t: time or a continuous index; L: a Lagrangian integrand, loss, queue length, or bound

$$H(x,u,\lambda,t)=L(x,u,t)+\lambda^T f(x,u,t). \tag{V.53}$$

Context in the volume. Necessary conditions couple state, costate, and pointwise control selection. The Hamilton-Jacobi-Bellman equation provides a dynamic-programming view. Linear-quadratic control yields analytic structure; nonlinear constrained control typically requires numerical methods.

Structural tags. school algebra, advanced geometry/composition.

Volume equation 54 · Decisions through time

Location. Chapter 7 · Dynamics, transport, category theory, algorithms, and verification — Decisions through time.

Prerequisites. Units 1–7, Units 23–31, Units 22 and 33, Unit 32, Units 36–40.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. V: a vector space or value function; t: time or a continuous index; A: a matrix, event, or set

$$V_t(s)=\min_{a\in\mathcal{A}(s)}\mathbb{E}\big[c_t(s,a,\xi_t)+V_{t+1}\big(T(s,a,\xi_t)\big)\big]. \tag{V.54}$$

Context in the volume. Equation (54) is simultaneously an optimization equation and an interface: the future is summarized by V_{t+1}. Dynamic programming, model predictive control, and reinforcement learning differ in how they represent or learn that interface. Exact tables are possible for small states; approximate value functions, policy gradients, tree search, and fitted models trade bias, variance, and computation at larger scales.

Structural tags. school algebra, optimization, dynamics/control, uncertainty/risk, advanced geometry/composition.

Volume equation 55 · Transport and geometry as semantics

Location. Chapter 7 · Dynamics, transport, category theory, algorithms, and verification — Transport and geometry as semantics.

Prerequisites. Units 12–17, Units 23–31, Units 36–40.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. gamma: a discount factor, coupling, or scalar; mu: a probability measure, service rate, mean, or distribution; nu: an equality multiplier or probability measure; int: integration over the displayed domain; x: a decision, state, or generic variable; local text determines the role; y: an output, response, path, or second variable

$$\min_{\gamma \in \Pi(\mu,\nu)} \int c(x,y)\, d\gamma(x,y). \tag{V.55}$$

Context in the volume. For $c(x,y) = d(x,y)^p$, the optimum defines the p-Wasserstein distance after taking the pth root. Unlike pointwise divergences, transport respects the ground geometry: nearby mass is cheaper to move than distant mass.

Structural tags. calculus, optimization, advanced geometry/composition.

Volume equation 56 · Transport and geometry as semantics

Location. Chapter 7 · Dynamics, transport, category theory, algorithms, and verification — Transport and geometry as semantics.

Prerequisites. Units 8–10, Units 36–40.

Read it as. Interpret the displayed quantity as an entropic regularization penalty: ε scales the divergence between the coupling γ and the product measure μ⨂ν.

Recurring symbols. gamma: a discount factor, coupling, or scalar; mu: a probability measure, service rate, mean, or distribution; nu: an equality multiplier or probability measure

$$\varepsilon\, \mathrm{KL}(\gamma \;\|\; \mu \otimes \nu) \tag{V.56}$$

Context in the volume. and enables fast Sinkhorn iterations. The regularization smooths and accelerates but also changes the solution; ε is a modeling as well as numerical parameter.

Structural tags. linear algebra, advanced geometry/composition.

Volume equation 57 · Category theory and composition

Location. Chapter 7 · Dynamics, transport, category theory, algorithms, and verification — Category theory and composition.

Prerequisites. Units 36–40.

Read it as. Read the arrow as a typed declaration: P maps an input in X to an output in Y.

Recurring symbols. P: probability, transition matrix, or distribution; X: a random variable, matrix, state space, or set

$$P: X \to Y. \tag{V.57}$$

Context in the volume. Sequential composition connects the output of one component to the input of another. Parallel composition uses a monoidal product:

Structural tags. advanced geometry/composition.

Volume equation 58 · Category theory and composition

Location. Chapter 7 · Dynamics, transport, category theory, algorithms, and verification — Category theory and composition.

Prerequisites. Units 36–40.

Read it as. Read the arrow as a typed declaration for parallel composition: $P_1 \otimes P_2$ maps $X_1 \otimes X_2$ to $Y_1 \otimes Y_2$.

Recurring symbols. P: probability, transition matrix, or distribution; X: a random variable, matrix, state space, or set

$$(P_1 \otimes P_2): (X_1 \otimes X_2) \to (Y_1 \otimes Y_2). \tag{V.58}$$

Context in the volume. The total cost might add, take a maximum, or combine in another monoidal way. The choice of monoid is part of the semantics. Decorated cospans and related constructions provide formal languages for open networks whose internal decorations may be circuits, dynamical systems, resource requirements, or optimization data.

Structural tags. advanced geometry/composition.

Volume equation 59 · Category theory and composition

Location. Chapter 7 · Dynamics, transport, category theory, algorithms, and verification — Category theory and composition.

Prerequisites. Units 23–31, Units 36–40.

Read it as. Choose the subscripted variable or policy from its admissible set to attain the displayed best value.

Recurring symbols. mathcal X: a feasible or decision space; P: probability, transition matrix, or distribution

$$\mathrm{Argmin}(P) \subseteq \mathcal{X}_P, \tag{V.59}$$

Context in the volume. possibly equipped with order, topology, probability, or approximation structure. Functorial behavior may be recovered only under additional assumptions or by enriching the codomain.

Structural tags. optimization, advanced geometry/composition.

Volume equation 60 · Algorithms, representations, and resource models

Location. Chapter 8 · Algorithms, computing systems, software, and cloud platforms — Algorithms, representations, and resource models.

Prerequisites. Units 1–7.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. C: a set, cost, covariance, or constant; A: a matrix, event, or set; n: sample size, dimension, horizon, or problem-size parameter

$$C(A,n) = w_t T(A,n) + w_m M(A,n) + w_i I(A,n) + w_e E(A,n), \tag{V.60}$$

Context in the volume. where T is time, M memory, I input/output, and E energy. The weights depend on the deployment environment. External-memory algorithms optimize transfers between storage levels; cache-oblivious algorithms seek good locality without hard-coding cache parameters; parallel algorithms trade total work against span, synchronization, and communication.

Structural tags. school algebra.

Volume equation 61 · Algorithms, representations, and resource models

Location. Chapter 8 · Algorithms, computing systems, software, and cloud platforms — Algorithms, representations, and resource models.

Prerequisites. Units 1–7.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. rho: a risk/aggregation functional, utilization, or correlation; f: an objective or generic function; x: a decision, state, or generic variable; local text determines the role

$$f(\hat{x}) \le \rho f(x^*) \tag{V.61}$$

Context in the volume. for ratio $\rho \ge 1$. Streaming algorithms control memory and passes. Fixed-parameter algorithms isolate a small structural parameter. Learned indexes and learned heuristics exploit observed distributions, but need fallback guarantees when the distribution shifts.

Structural tags. mathematical notation.

Volume equation 62 · Algorithms, representations, and resource models

Location. Chapter 8 · Algorithms, computing systems, software, and cloud platforms — Algorithms, representations, and resource models.

Prerequisites. Units 1–7.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. R: a reward, requirement, relation, or real-number set; x: a decision, state, or generic variable; local text determines the role; B: a matrix, set, or budget; C: a set, cost, covariance, or constant; A: a matrix, event, or set

$$R(x) = \left(T_{50}(x), T_{99}(x), M(x), B(x), E(x), C(x), A(x)\right), \tag{V.62}$$

Context in the volume. where the components may represent median and tail latency, memory, bandwidth, energy, monetary cost, and availability. Equation (62) invites Pareto analysis and explicit service constraints. It also prevents a local microbenchmark from silently standing in for user experience.

Structural tags. school algebra.

Volume equation 63 · Systems, queues, and cloud capacity

Location. Chapter 8 · Algorithms, computing systems, software, and cloud platforms — Systems, queues, and cloud capacity.

Prerequisites. Units 1–7.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; L: a Lagrangian integrand, loss, queue length, or bound

$$L = \lambda W. \tag{V.63}$$

Context in the volume. It is a conservation law, not a scheduling policy. As utilization approaches capacity, latency often rises nonlinearly. Batching improves throughput but adds waiting. Replication reduces read latency and increases availability while creating consistency and coordination costs. Backpressure protects downstream components but can propagate upstream.

Structural tags. school algebra.

Volume equation 64 · Systems, queues, and cloud capacity

Location. Chapter 8 · Algorithms, computing systems, software, and cloud platforms — Systems, queues, and cloud capacity.

Prerequisites. Units 1–7.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. p: probability, dimension, exponent, density, or primal value according to context

$$S(p) = \frac{1}{s+(1-s)/p}. \tag{V.64}$$

Context in the volume. Gustafson’s law instead asks how problem size can scale with available processors. Both are idealizations. Real scaling is limited by memory bandwidth, communication, load imbalance, synchronization, and energy. Roofline-style reasoning compares arithmetic intensity with compute and bandwidth ceilings.

Structural tags. school algebra.

Volume equation 65 · Systems, queues, and cloud capacity

Location. Chapter 8 · Algorithms, computing systems, software, and cloud platforms — Systems, queues, and cloud capacity.

Prerequisites. Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; mu: a probability measure, service rate, mean, or distribution; x: a decision, state, or generic variable; local text determines the role; C: a set, cost, covariance, or constant

$$\min_{x}\ C_{\text{money}}(x) + \lambda C_{\text{latency}}(x) + \mu C_{\text{carbon}}(x) \tag{V.65}$$

Context in the volume. subject to capacity, affinity, security, and service-level constraints. Autoscaling is a feedback controller; if delayed or overly aggressive, it oscillates. Reliability engineering uses redundancy, graceful degradation, fault containment, and recovery objectives rather than assuming failure can be optimized away.

Structural tags. optimization.

Volume equation 66 · Systems, queues, and cloud capacity

Location. Chapter 8 · Algorithms, computing systems, software, and cloud platforms — Systems, queues, and cloud capacity.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. sum: summation over the displayed index range; R: a reward, requirement, relation, or real-number set; z: an auxiliary or third variable

$$\min_{r\in\mathcal{R}} \left(\textstyle\sum_{s,z} c_{sz}\, r_{sz} + \lambda_E E(r)\right). \tag{V.66}$$

Context in the volume. The feasible-set definition paired with Equation (66) makes three structures visible: integer capacity, chance or risk constraints for latency, and survivability under failure sets. The model should include startup delay, state replication, data locality, deployment surge, and recovery workload. A capacity plan that is feasible only in steady state can fail during the change that implements it.

Structural tags. school algebra, optimization.

Volume equation 67 · Learning as nested, decision-focused optimization

Location. Chapter 9 · Machine learning, artificial intelligence, security, privacy, and assurance — Learning as nested, decision-focused optimization.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted variable or policy from its admissible set to attain the displayed best value.

Recurring symbols. lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; theta: a parameter, mixture weight, or preference parameter; sum: summation over the displayed index range; n: sample size, dimension, horizon, or problem-size parameter; i: an item, component, sample, or constraint index; f: an objective or generic function; x: a decision, state, or generic variable; local text determines the role; y: an output, response, path, or second variable; R: a reward, requirement, relation, or real-number set

$$\hat{\theta} \in \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell\left(f_{\theta}(x_i), y_i\right) + \lambda R(\theta). \tag{V.67}$$

Context in the volume. The loss ℓ encodes an error model; the regularizer R encodes preference or capacity control. Training error is not deployment error. Generalization depends on hypothesis class, data dependence, sampling, optimization, and shift. Cross-validation estimates out-of-sample performance only when splits respect time, groups, leakage paths, and intended use.

Structural tags. school algebra, optimization.

Volume equation 68 · Learning as nested, decision-focused optimization

Location. Chapter 9 · Machine learning, artificial intelligence, security, privacy, and assurance — Learning as nested, decision-focused optimization.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted variable or policy from its admissible set to attain the displayed best value.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; L: a Lagrangian integrand, loss, queue length, or bound; X: a random variable, matrix, state space, or set

$$a^*(x) = \arg\min_{a} \mathbb{E}[L(a, Y) \mid X = x]. \tag{V.68}$$

Context in the volume. Thresholds therefore depend on consequences and capacity, not merely model discrimination.

Structural tags. school algebra, optimization.

Volume equation 69 · Learning as nested, decision-focused optimization

Location. Chapter 9 · Machine learning, artificial intelligence, security, privacy, and assurance — Learning as nested, decision-focused optimization.

Prerequisites. Units 1–7.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. All symbols are defined by the adjacent volume context; consult the notation concordance for recurring conventions.

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \Pr[M(D') \in S] + \delta \tag{V.69}$$

Context in the volume. for all measurable S. Privacy budget, accuracy, and participation remain coupled.

Structural tags. mathematical notation.

Volume equation 70 · Learning as nested, decision-focused optimization

Location. Chapter 9 · Machine learning, artificial intelligence, security, privacy, and assurance — Learning as nested, decision-focused optimization.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted variable or policy from its admissible set to attain the displayed best value.

Recurring symbols. theta: a parameter, mixture weight, or preference parameter; sum: summation over the displayed index range; x: a decision, state, or generic variable; local text determines the role; A: a matrix, event, or set; y: an output, response, path, or second variable; P: probability, transition matrix, or distribution

$$a^*(x) \in \arg\max_{a \in \mathcal{A}} \sum_{y \in \mathcal{Y}} P_\theta\,(y \mid x)\, U(a, y; x). \tag{V.70}$$

Context in the volume. Equation (70) separates probabilistic estimation from utility and permits abstention, referral, or information gathering. Calibration matters because predicted probabilities enter the consequence calculation. Class balance and a confusion matrix are insufficient when harms differ, capacity is constrained, or the action changes future labels.

Structural tags. school algebra, optimization.

Volume equation 71 · Security, privacy, and adversarial optimization

Location. Chapter 9 · Machine learning, artificial intelligence, security, privacy, and assurance — Security, privacy, and adversarial optimization.

Prerequisites. Units 18–22, Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; X: a random variable, matrix, state space, or set; P: probability, transition matrix, or distribution; H: a hypothesis, Hessian, entropy, or Hamiltonian; C: a set, cost, covariance, or constant

$$\min_{x \in X} \mathbb{E}_{a \sim P(a|x)}[H(x, a)] + C(x), \tag{V.71}$$

Context in the volume. where investment x changes both harm H and the distribution of attacks. Static risk scores miss strategic adaptation. Game theory, robust optimization, and red-team exercises supply complementary views.

Structural tags. probability/statistics, optimization.

Volume equation 72 · Security, privacy, and adversarial optimization

Location. Chapter 9 · Machine learning, artificial intelligence, security, privacy, and assurance — Security, privacy, and adversarial optimization.

Prerequisites. Units 18–22, Units 23–31, Unit 32.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. alpha: a step size, confidence/tail level, weight, or scalar; lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; xi: an uncertain quantity or scenario; x: a decision, state, or generic variable; local text determines the role; L: a Lagrangian integrand, loss, queue length, or bound

$$\min_{x} \ \mathbb{E}[L(x,\xi)] + \lambda \mathrm{CVaR}_{\alpha}\big(L(x,\xi)\big). \tag{V.72}$$

Context in the volume. Backups must be isolated and restored in exercises; incident response must be rehearsed; recovery objectives must reflect dependency chains. Diversity can reduce common-mode failure but increases maintenance burden.

Structural tags. probability/statistics, optimization, uncertainty/risk.

Volume equation 73 · Security, privacy, and adversarial optimization

Location. Chapter 9 · Machine learning, artificial intelligence, security, privacy, and assurance — Security, privacy, and adversarial optimization.

Prerequisites. Units 23–31.

Read it as. Compare a decision against an inner worst case or competing choice, respecting the order of the two optimizations.

Recurring symbols. theta: a parameter, mixture weight, or preference parameter; x: a decision, state, or generic variable; local text determines the role; y: an output, response, path, or second variable; f: an objective or generic function

$$\min_{\theta} \ \mathbb{E}_{(x,y)}\left[\max_{\delta\in\Delta(x)} \ell(f_{\theta}(x+\delta),y)\right]. \tag{V.73}$$

Context in the volume. The set $\Delta(x)$ is a threat model, not a natural fact. A small norm ball may be mathematically convenient yet fail to represent semantic manipulation, data poisoning, model extraction, prompt injection, or supply-chain compromise. Robustness inside Δ should not be advertised as general security.

Structural tags. optimization.

Volume equation 74 · Systems engineering, co-design, and exchanged envelopes

Location. Chapter 10 · Systems engineering, structures, signals, control, hardware, and co-design — Systems engineering, co-design, and exchanged envelopes.

Prerequisites. Units 23–31, Units 22 and 33.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; y: an output, response, path, or second variable; m: a count, dimension, or slope when defined locally; f: an objective or generic function

Context in the volume. Multidisciplinary design optimization can be written schematically as

$$\min_{x,y_1,\dots,y_m} f(x, y_1, \dots, y_m) \tag{V.74}$$

Structural tags. optimization, dynamics/control.

Volume equation 75 · Systems engineering, co-design, and exchanged envelopes

Location. Chapter 10 · Systems engineering, structures, signals, control, hardware, and co-design — Systems engineering, co-design, and exchanged envelopes.

Prerequisites. Units 1–7, Units 22 and 33.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. i: an item, component, sample, or constraint index; x: a decision, state, or generic variable; local text determines the role; y: an output, response, path, or second variable; m: a count, dimension, or slope when defined locally

$$r_i(x, y_1, \dots, y_m) = 0, \qquad i = 1, \dots, m, \tag{V.75}$$

Context in the volume. and system constraints. The shared design x might contain geometry and material choices; y_i might represent aerodynamic, structural, thermal, control, or economic states. Architectures differ in whether they solve discipline consistency inside each evaluation, distribute targets, or optimize all states simultaneously.

Structural tags. school algebra, dynamics/control.

Volume equation 76 · Structures and physical feasibility

Location. Chapter 10 · Systems engineering, structures, signals, control, hardware, and co-design — Structures and physical feasibility.

Prerequisites. Units 23–31, Units 22 and 33.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; m: a count, dimension, or slope when defined locally

Context in the volume. A structural sizing problem may minimize mass using a vector of fixed element mass coefficients m

$$\min_{x} m^T x \tag{V.76}$$

Structural tags. optimization, dynamics/control.

Volume equation 77 · Structures and physical feasibility

Location. Chapter 10 · Systems engineering, structures, signals, control, hardware, and co-design — Structures and physical feasibility.

Prerequisites. Units 1–7, Units 23–31, Units 22 and 33.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. rho: a risk/aggregation functional, utilization, or correlation; sum: summation over the displayed index range; f: an objective or generic function; V: a vector space or value function

$$\min_{\rho} f^T u(\rho) \quad \text{subject to} \quad K(\rho)u = f, \quad \sum_e v_e \rho_e \leq V. \tag{V.77}$$

Context in the volume. Penalization encourages near-binary densities; filters and length-scale constraints suppress mesh-dependent checkerboards. The mathematically optimal geometry may still need support removal, minimum radii, standardized stock, inspection access, and tolerances.

Structural tags. school algebra, optimization, dynamics/control.

Volume equation 78 · Signals, estimation, sensing, and control

Location. Chapter 10 · Systems engineering, structures, signals, control, hardware, and co-design — Signals, estimation, sensing, and control.

Prerequisites. Units 8–10, Units 22 and 33.

Read it as. Interpret the displayed squared norm as a least-squares residual or objective measuring mismatch between Ax and y.

Recurring symbols. y: an output, response, path, or second variable

$$\| Ax - y \|_2^2. \tag{V.78}$$

Context in the volume. Weighted least squares reflects unequal noise; regularization stabilizes ill-posed inversion; sparsity penalties support compressed representations. Fourier and wavelet transforms choose bases in which signals or operators become simple. Wiener and Kalman filters optimize mean-square estimation under stated stochastic models. Model mismatch, non-Gaussian tails, and sensor bias require robust or nonlinear alternatives.

Structural tags. linear algebra, dynamics/control.

Volume equation 79 · Signals, estimation, sensing, and control

Location. Chapter 10 · Systems engineering, structures, signals, control, hardware, and co-design — Signals, estimation, sensing, and control.

Prerequisites. Units 1–7, Units 18–22, Units 22 and 33.

Read it as. Add the indexed contributions over the stated range, keeping each weight, unit, and constraint interpretation explicit.

Recurring symbols. sum: summation over the displayed index range; H: a hypothesis, Hessian, entropy, or Hamiltonian; X: a random variable, matrix, state space, or set; x: a decision, state, or generic variable; local text determines the role; p: probability, dimension, exponent, density, or primal value according to context

Context in the volume. For a discrete source X, entropy is

$$H(X) = -\sum_x p\,(x)\log_2 p(x). \qquad \text{(V.79)}$$

Structural tags. school algebra, probability/statistics, dynamics/control.

Volume equation 80 · Signals, estimation, sensing, and control

Location. Chapter 10 · Systems engineering, structures, signals, control, hardware, and co-design — Signals, estimation, sensing, and control.

Prerequisites. Units 1–7, Units 18–22, Units 23–31, Units 22 and 33.

Read it as. Choose the subscripted decision so the displayed quantity is as large as the constraints allow.

Recurring symbols. C: a set, cost, covariance, or constant; p: probability, dimension, exponent, density, or primal value according to context; x: a decision, state, or generic variable; local text determines the role; X: a random variable, matrix, state space, or set

$$C = \max_{p(x)} I(X;Y), \qquad \text{(V.80)}$$

Context in the volume. under the channel model and constraints. Coding approaches these limits with delay, complexity, and energy costs. Network design adds routing, congestion, spectrum allocation, interference, fairness, and resilience.

Structural tags. school algebra, probability/statistics, optimization, dynamics/control.

Volume equation 81 · Signals, estimation, sensing, and control

Location. Chapter 10 · Systems engineering, structures, signals, control, hardware, and co-design — Signals, estimation, sensing, and control.

Prerequisites. Units 1–7, Units 22 and 33.

Read it as. The next iterate or state is computed from the current quantities according to the displayed update rule.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; t: time or a continuous index; y: an output, response, path, or second variable

Context in the volume. For a linear system

$$x_{t+1} = Ax_t + Bu_t, \qquad y_t = Cx_t, \tag{V.81}$$

Structural tags. school algebra, dynamics/control.

Volume equation 82 · Signals, estimation, sensing, and control

Location. Chapter 10 · Systems engineering, structures, signals, control, hardware, and co-design — Signals, estimation, sensing, and control.

Prerequisites. Units 1–7, Units 22 and 33.

Read it as. Add the indexed contributions over the stated range, keeping each weight, unit, and constraint interpretation explicit.

Recurring symbols. sum: summation over the displayed index range; J: an objective functional, Jacobian, or performance criterion; t: time or a continuous index; x: a decision, state, or generic variable; local text determines the role

$$J = \sum_{t=0}^{\infty} (x_t^T Q x_t + u_t^T R u_t). \tag{V.82}$$

Context in the volume. Optimality relies on stabilizability, detectability, and the model. Robust control bounds uncertainty; adaptive control updates parameters; model predictive control repeatedly solves a finite-horizon constrained problem using new state estimates. Delay and saturation can destabilize a controller that looks optimal in an instantaneous model.

Structural tags. school algebra, dynamics/control.

Volume equation 83 · Hardware beyond PPA

Location. Chapter 10 · Systems engineering, structures, signals, control, hardware, and co-design — Hardware beyond PPA.

Prerequisites. Units 1–7, Units 23–31, Units 22 and 33.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. sum: summation over the displayed index range; x: a decision, state, or generic variable; local text determines the role; i: an item, component, sample, or constraint index; y: an output, response, path, or second variable

$$\min_{(x_i, y_i)} \sum_{e \in E} w_e \ \mathrm{HPWL}(e), \tag{V.83}$$

Context in the volume. subject to density and block constraints. True performance depends on routed parasitics, congestion, timing, voltage drop, and thermal coupling.

Structural tags. school algebra, optimization, dynamics/control.

Volume equation 84 · Processes, manufacturing, and safe intensification

Location. Chapter 11 · Processes, materials, manufacturing, energy, climate, robotics, and transport — Processes, manufacturing, and safe intensification.

Prerequisites. Units 1–7, Units 36–40.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. F: a vector-valued map, aggregate objective, or transition model; z: an auxiliary or third variable; x: a decision, state, or generic variable; local text determines the role

$$F(z, x) = 0, \tag{V.84}$$

Context in the volume. where z contains temperatures, pressures, flows, and compositions, and x contains design or control choices. The optimizer minimizes cost or environmental impact subject to these equations and inequalities. Heat integration, reactor-separator-recycle structure, utilities, and equipment sizing create nonconvex mixed-integer nonlinear programs.

Structural tags. school algebra, advanced geometry/composition.

Volume equation 85 · Energy, climate, conservation, and circularity

Location. Chapter 11 · Processes, materials, manufacturing, energy, climate, robotics, and transport — Energy, climate, conservation, and circularity.

Prerequisites. Units 1–7, Units 23–31, Units 36–40.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. sum: summation over the displayed index range; x: a decision, state, or generic variable; local text determines the role; t: time or a continuous index

$$\min_{x} \sum_{t=0}^{T} \frac{C_t(x_t)}{(1+r)^t} \tag{V.85}$$

Context in the volume. subject to demand balance, reliability, build rates, network limits, emissions, and technology constraints. Hourly or subhourly operations must be represented well enough that a capacity plan remains feasible. Representative periods reduce computation but can erase rare stress events and long storage cycles.

Structural tags. school algebra, optimization, advanced geometry/composition.

Volume equation 86 · Energy, climate, conservation, and circularity

Location. Chapter 11 · Processes, materials, manufacturing, energy, climate, robotics, and transport — Energy, climate, conservation, and circularity.

Prerequisites. Units 1–7, Units 36–40.

Read it as. Add the indexed contributions over the stated range, keeping each weight, unit, and constraint interpretation explicit.

Recurring symbols. sum: summation over the displayed index range; j: a second index, commonly for equality constraints or states; i: an item, component, sample, or constraint index; f: an objective or generic function; V: a vector space or value function

$$\sum_{j:(j,i)\in E} f_{ji} + s_i = \sum_{j:(i,j)\in E} f_{ij} + d_i, \qquad i \in V, \tag{V.86}$$

Context in the volume. where s_i and d_i represent supply and demand. Equation (86) can describe molecules, electricity under an approximation, vehicles, data, or money, but each domain adds different physics and loss. The equation is a structural correspondence, not an assertion that the entities are interchangeable.

Structural tags. school algebra, advanced geometry/composition.

Volume equation 87 · Robotics and transportation in open environments

Location. Chapter 11 · Processes, materials, manufacturing, energy, climate, robotics, and transport — Robotics and transportation in open environments.

Prerequisites. Units 1–7, Units 12–17, Units 36–40.

Read it as. Accumulate the integrand over the displayed interval, space, path, or coupling.

Recurring symbols. int: integration over the displayed domain; J: an objective functional, Jacobian, or performance criterion; x: a decision, state, or generic variable; local text determines the role; Φ: a terminal cost; L: a Lagrangian integrand, loss, queue length, or bound; t: time or a continuous index

$$J = \Phi\big(x(T)\big) + \int_0^T L\,(x(t), u(t), t)\,dt \tag{V.87}$$

Context in the volume. subject to dynamics $\dot{x} = f(x, u, t)$, boundary conditions, and path constraints. Direct transcription discretizes the trajectory into a nonlinear program; indirect methods use necessary conditions from optimal control. Fuel, time, heating, noise, comfort, risk, and control effort may conflict.

Structural tags. school algebra, calculus, advanced geometry/composition.

Volume equation 88 · Evolution and contextual biological optimality

Location. Chapter 12 · Biology, clinical medicine, public health, and biomedical discovery — Evolution and contextual biological optimality.

Prerequisites. Units 1–7.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. i: an item, component, sample, or constraint index; p: probability, dimension, exponent, density, or primal value according to context

$$w_i = w_i(p), \tag{V.88}$$

Context in the volume. and the landscape moves as the population moves. Replicator dynamics model frequency change:

Structural tags. school algebra.

Volume equation 89 · Evolution and contextual biological optimality

Location. Chapter 12 · Biology, clinical medicine, public health, and biomedical discovery — Evolution and contextual biological optimality.

Prerequisites. Units 1–7, Units 22 and 33.

Read it as. Add the indexed contributions over the stated range, keeping each weight, unit, and constraint interpretation explicit.

Recurring symbols. sum: summation over the displayed index range; p: probability, dimension, exponent, density, or primal value according to context; i: an item, component, sample, or constraint index; j: a second index, commonly for equality constraints or states; m: a count, dimension, or slope when defined locally

$$\dot{p}_i = p_i\big(w_i(p) - \bar{w}(p)\big), \qquad \bar{w}(p) = \textstyle\sum_{j=1}^{m} p_j\, w_j(p). \tag{V.89}$$

Context in the volume. Evolutionarily stable strategies describe resistance to invasion under a game, not moral or social desirability. Traits can be by-products, exaptations, or consequences of constraints; adaptationist stories need alternatives and evidence.

Structural tags. school algebra, dynamics/control.

Volume equation 90 · Evolution and contextual biological optimality

Location. Chapter 12 · Biology, clinical medicine, public health, and biomedical discovery — Evolution and contextual biological optimality.

Prerequisites. Units 1–7.

Read it as. The left and right expressions are ordered; the inequality usually encodes feasibility, a bound, or dominance.

Recurring symbols. All symbols are defined by the adjacent volume context; consult the notation concordance for recurring conventions.

$$Sv = 0, \qquad \ell \le v \le u, \tag{V.90}$$

Context in the volume. then optimizes a proxy such as biomass production $c^T v$. The model is powerful because conservation constrains possibilities, but its objective and steady-state assumptions must be experimentally checked. Multiobjective and regulatory extensions help explain trade-offs among growth, yield, stress, and by-products.

Structural tags. school algebra.

Volume equation 91 · Clinical decisions and treatment policies

Location. Chapter 12 · Biology, clinical medicine, public health, and biomedical discovery — Clinical decisions and treatment policies.

Prerequisites. Units 1–7, Units 18–22.

Read it as. Add the indexed contributions over the stated range, keeping each weight, unit, and constraint interpretation explicit.

Recurring symbols. sum: summation over the displayed index range; x: a decision, state, or generic variable; local text determines the role; P: probability, transition matrix, or distribution

$$EU(a \mid x) = \sum_{d\in\mathcal{D}} P\,(D = d \mid x)\, U(a, d; x). \tag{V.91}$$

Context in the volume. The preferred action maximizes expected utility, subject to consent and clinical constraints. If treatment benefit increases with disease probability while treatment harm does not, a threshold probability can separate “do not treat” from “treat”. A threshold is not universal: it changes with disease severity, treatment burden, comorbidity, alternatives, and the person’s preferences.

Structural tags. school algebra, probability/statistics.

Volume equation 92 · Clinical decisions and treatment policies

Location. Chapter 12 · Biology, clinical medicine, public health, and biomedical discovery — Clinical decisions and treatment policies.

Prerequisites. Units 1–7, Units 18–22.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. P: probability, transition matrix, or distribution

$$\frac{P(D|T)}{1-P(D|T)} = \frac{P(T|D)}{P(T|\neg D)}\frac{P(D)}{1-P(D)}. \tag{V.92}$$

Context in the volume. Likelihood ratios characterize a test, while pretest odds place it in context. Testing has value only if results can change an action or meaningfully inform the person.

Structural tags. school algebra, probability/statistics.

Volume equation 93 · Clinical decisions and treatment policies

Location. Chapter 12 · Biology, clinical medicine, public health, and biomedical discovery — Clinical decisions and treatment policies.

Prerequisites. Units 1–7, Units 12–17.

Read it as. Accumulate the integrand over the displayed interval, space, path, or coupling.

Recurring symbols. int: integration over the displayed domain; Q: a quadratic matrix, action-value function, or distribution; t: time or a continuous index

$$Q = \int_0^T u\,(t)\,dt, \tag{V.93}$$

Context in the volume. but utilities vary across people and contexts. Disability, age, and chronic illness raise ethical questions that a scalar cannot settle. Clinical constraints, rights, and individual preference should remain visible.

Structural tags. school algebra, calculus.

Volume equation 94 · Public health and allocation

Location. Chapter 12 · Biology, clinical medicine, public health, and biomedical discovery — Public health and allocation.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted decision so the displayed quantity is as large as the constraints allow.

Recurring symbols. sum: summation over the displayed index range; x: a decision, state, or generic variable; local text determines the role; i: an item, component, sample, or constraint index; n: sample size, dimension, horizon, or problem-size parameter; B: a matrix, set, or budget

$$\max_x \sum_{i=1}^{n} B_i\,(x_i) \tag{V.94}$$

Context in the volume. subject to budget, staff, beds, geography, eligibility, and minimum-service constraints. The objective immediately raises questions: Are benefits measured as deaths averted, quality-adjusted life years, unmet need, financial protection, or capability? Are gains weighted by severity, disadvantage, or reciprocity? Cost-effectiveness supplies evidence, not a complete allocation rule.

Structural tags. school algebra, optimization.

Volume equation 95 · Biomedical discovery and translation

Location. Chapter 12 · Biology, clinical medicine, public health, and biomedical discovery — Biomedical discovery and translation.

Prerequisites. Units 1–7.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; B: a matrix, set, or budget; H: a hypothesis, Hessian, entropy, or Hamiltonian

$$U(d) = B(d) - \lambda H(d), \tag{V.95}$$

Context in the volume. where both benefit B and harm H are uncertain and heterogeneous. Phase transitions in development are real-option decisions: continue, redesign, partner, or stop after new evidence. Portfolio optimization must resist incentives to hide negative information.

Structural tags. school algebra.

Volume equation 96 · Markets, finance, and endogenous organizations

Location. Chapter 13 · Economics, operations research, public policy, law, and education — Markets, finance, and endogenous organizations.

Prerequisites. Units 1–7, Units 23–31, Units 22 and 33.

Read it as. Choose the subscripted decision so the displayed quantity is as large as the constraints allow.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; p: probability, dimension, exponent, density, or primal value according to context; m: a count, dimension, or slope when defined locally

$$\max_{x \geq 0} u(x) \quad \text{subject to} \quad p^T x \leq m. \tag{V.96}$$

Context in the volume. A firm chooses inputs and outputs to maximize profit. Prices coordinate these decisions in competitive models. General equilibrium asks whether individually optimizing choices and market clearing can coexist. Welfare theorems require strong assumptions and do not select a distribution of endowments. Arrow’s result shows that no social-welfare aggregation rule can satisfy a set of plausible conditions simultaneously when there are at least three alternatives.

Structural tags. school algebra, optimization, dynamics/control.

Volume equation 97 · Markets, finance, and endogenous organizations

Location. Chapter 13 · Economics, operations research, public policy, law, and education — Markets, finance, and endogenous organizations.

Prerequisites. Units 1–7, Units 23–31, Units 22 and 33.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; mu: a probability measure, service rate, mean, or distribution

$$\min_{w} \; 1/2\, w^T \Sigma w - \lambda \mu^T w, \qquad \mathbf{1}^T w = 1. \tag{V.97}$$

Context in the volume. It established diversification as an optimization problem, but estimated returns are unstable and covariance changes during stress. Robust allocation, scenario analysis, liquidity limits, transaction costs, leverage, taxes, and tail risk matter. Conditional value at risk averages losses beyond a quantile and supports convex formulations.

Structural tags. school algebra, optimization, dynamics/control.

Volume equation 98 · Operations research as institutional engineering

Location. Chapter 13 · Economics, operations research, public policy, law, and education — Operations research as institutional engineering.

Prerequisites. Units 1–7, Units 23–31, Units 22 and 33.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. q: a dual function, probability, or auxiliary quantity

Context in the volume. A simple newsvendor chooses quantity q before random demand D:

$$\min_q c_o\,\mathbb{E}[(q-D)_+] + c_u\,\mathbb{E}[(D-q)_+]. \tag{V.98}$$

Structural tags. school algebra, optimization, dynamics/control.

Volume equation 99 · Operations research as institutional engineering

Location. Chapter 13 · Economics, operations research, public policy, law, and education — Operations research as institutional engineering.

Prerequisites. Units 1–7, Units 22 and 33.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. F: a vector-valued map, aggregate objective, or transition model; q: a dual function, probability, or auxiliary quantity

$$F_D(q^*) = \frac{c_u}{c_u+c_o}. \tag{V.99}$$

Context in the volume. This compact result shows how costs translate into a service level—and how a wrong cost model creates the wrong stock policy.

Structural tags. school algebra, dynamics/control.

Volume equation 100 · Policy, law, delivery, and accountability

Location. Chapter 13 · Economics, operations research, public policy, law, and education — Policy, law, delivery, and accountability.

Prerequisites. Units 1–7, Units 22 and 33.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. All symbols are defined by the adjacent volume context; consult the notation concordance for recurring conventions.

$$\mathrm{ATE} = \mathbb{E}[Y(1) - Y(0)]. \tag{V.100}$$

Context in the volume. Only one potential outcome is observed for each unit. Randomization, natural experiments, adjustment, and structural models identify effects under different assumptions. A model that accurately predicts unemployment does not necessarily reveal which intervention reduces it.

Structural tags. school algebra, dynamics/control.

Volume equation 101 · Ethics, fairness, and moral uncertainty

Location. Chapter 14 · Psychology, ethics, sustainability, philosophy, history, and language — Ethics, fairness, and moral uncertainty.

Prerequisites. Units 23–31, Unit 32.

Read it as. Compare a decision against an inner worst case or competing choice, respecting the order of the two optimizations.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; i: an item, component, sample, or constraint index

$$\max_{x} \min_{i} u_i(x), \tag{V.101}$$

Context in the volume. captures one protective intuition but not the full theory. Ethical frameworks should guide questions and institutions rather than be caricatured as formulas.

Structural tags. optimization, uncertainty/risk.

Volume equation 102 · Humanities, interpretation, and historical path dependence

Location. Chapter 14 · Psychology, ethics, sustainability, philosophy, history, and language — Humanities, interpretation, and historical path dependence.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted decision so the displayed quantity is as large as the constraints allow.

Recurring symbols. theta: a parameter, mixture weight, or preference parameter; sum: summation over the displayed index range; t: time or a continuous index; p: probability, dimension, exponent, density, or primal value according to context

$$\max_{\theta} \sum_{t=1}^{T} \log p_\theta(w_t \mid w_{<t}), \tag{V.102}$$

Context in the volume. but probable continuation is not the same as truth, relevance, or literary value. Translation trades semantic content, tone, rhythm, cultural reference, and audience. No single alignment score captures all of them.

Structural tags. school algebra, optimization.

Volume equation 103 · Music: geometry, composition, performance, and listening

Location. Chapter 15 · Architecture, design, visual art, music, and performance — Music: geometry, composition, performance, and listening.

Prerequisites. Units 1–7.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. f: an objective or generic function; n: sample size, dimension, horizon, or problem-size parameter

$$f_n = f_0\, 2^{n/12}. \tag{V.103}$$

Context in the volume. This makes transposition uniform while approximating just intervals. Other temperaments optimize different compromises among interval purity, key color, modulation, and instrument construction. Tuning is therefore an historical multiobjective design, not a solved natural constant.

Structural tags. school algebra.

Volume equation 104 · Music: geometry, composition, performance, and listening

Location. Chapter 15 · Architecture, design, visual art, music, and performance — Music: geometry, composition, performance, and listening.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. pi: a policy, probability, or mathematical constant; sum: summation over the displayed index range; C: a set, cost, covariance, or constant; p: probability, dimension, exponent, density, or primal value according to context; q: a dual function, probability, or auxiliary quantity; k: an iteration, time step, or generic index; i: an item, component, sample, or constraint index; m: a count, dimension, or slope when defined locally

$$C(p,q) = \min_{\pi,k} \textstyle\sum_{i=1}^{m} w_i \left|p_i - q_{\pi(i)} - 12k_i\right|, \tag{V.104}$$

Context in the volume. where m is the number of voices, $w_i \geq 0$ weights the relative cost of moving voice i, π is a permutation matching voices, and each integer k_i represents an octave shift. The cost captures motion, not expressive function by itself.

Structural tags. school algebra, optimization.

Volume equation 105 · Games, sport, rules, and training

Location. Chapter 16 · Literature, narrative, media, games, sport, and competition — Games, sport, rules, and training.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Compare a decision against an inner worst case or competing choice, respecting the order of the two optimizations.

Recurring symbols. p: probability, dimension, exponent, density, or primal value according to context; q: a dual function, probability, or auxiliary quantity

$$\max_p \min_q p^T A q = \min_q \max_p p^T A q. \quad \text{(V.105)}$$

Context in the volume. This supports equilibrium strategies in games such as poker subproblems. General games may have multiple equilibria and require beliefs about opponents. Exploitative play can outperform equilibrium against a known weakness while becoming vulnerable to counteradaptation.

Structural tags. school algebra, optimization.

Volume equation 106 · Games, sport, rules, and training

Location. Chapter 16 · Literature, narrative, media, games, sport, and competition — Games, sport, rules, and training.

Prerequisites. Units 1–7.

Read it as. Add the indexed contributions over the stated range, keeping each weight, unit, and constraint interpretation explicit.

Recurring symbols. sum: summation over the displayed index range; R: a reward, requirement, relation, or real-number set; t: time or a continuous index; k: an iteration, time step, or generic index; i: an item, component, sample, or constraint index

$$R(t) = R_0 + k_1 \sum_{i=1}^{t-1} u_i \, e^{-(t-i)/\tau_1} - k_2 \sum_{i=1}^{t-1} u_i \, e^{-(t-i)/\tau_2}. \quad \text{(V.106)}$$

Context in the volume. $R(t)$ is modeled readiness at time t and R_0 is baseline readiness; u_i is the training impulse at time i; k_1 and k_2 are positive adaptation and fatigue gains; and τ_1 and τ_2 are their decay time constants. The described slower adaptation and faster, initially larger fatigue response requires $\tau_1 > \tau_2$ and $k_2 > k_1$. These parameters are individual and uncertain. Maximizing short-term load can increase injury and burnout. Athlete consent, education, and data privacy constrain monitoring.

Structural tags. school algebra.

Volume equation 107 · Evidence-producing optimization

Location. Chapter 17 · Experimentation, digital twins, human judgment, and failure modes — Evidence-producing optimization.

Prerequisites. Units 1–7.

Read it as. Add the indexed contributions over the stated range, keeping each weight, unit, and constraint interpretation explicit.

Recurring symbols. eta: a threshold or auxiliary optimization variable; sum: summation over the displayed index range; y: an output, response, path, or second variable; i: an item, component, sample, or constraint index; p: probability, dimension, exponent, density, or primal value according to context; x: a decision, state, or generic variable; local text determines the role; j: a second index, commonly for equality constraints or states

$$y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \sum_{i=1}^{p} \beta_{ii} x_i^2 + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \beta_{ij} x_i x_j + \eta, \tag{V.107}$$

Context in the volume. the pure quadratic terms represent local curvature, the interaction terms show why the best setting of one factor can depend on another, and η is regression noise. Replication estimates noise. Sequential designs use early results to place later experiments where information or improvement is most valuable.

Structural tags. school algebra.

Volume equation 108 · Evidence-producing optimization

Location. Chapter 17 · Experimentation, digital twins, human judgment, and failure modes — Evidence-producing optimization.

Prerequisites. Units 1–7, Units 23–31, Unit 32.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. xi: an uncertain quantity or scenario; x: a decision, state, or generic variable; local text determines the role; J: an objective functional, Jacobian, or performance criterion; g: an inequality constraint or generic function

$$\min_{x} J(x) = \mathbb{E}[g(x, \xi)], \tag{V.108}$$

Context in the volume. each evaluation is noisy. Common random numbers can reduce variance when comparing designs. Ranking-and-selection allocates samples among alternatives. Surrogates, stochastic approximation, and Bayesian optimization reduce expensive runs. Validation must compare simulated and observed behavior at the level relevant to the decision.

Structural tags. school algebra, optimization, uncertainty/risk.

Volume equation 109 · Failure modes, theater, refusal, and purpose

Location. Chapter 17 · Experimentation, digital twins, human judgment, and failure modes — Failure modes, theater, refusal, and purpose.

Prerequisites. Units 1–7, Units 23–31.

Read it as. Choose the subscripted variable or policy from its admissible set to attain the displayed best value.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; X: a random variable, matrix, state space, or set

$$\mathbb{E}[\varepsilon(\hat{x})] > 0, \qquad \hat{x} = \arg\max_{x \in X} M(x), \tag{V.109}$$

Context in the volume. under common conditions. Stronger search can therefore widen the gap between measured and true value. This is the optimizer's curse analyzed by Smith and Winkler. Once people adapt strategically, the measurement process itself changes; that distinct feedback mechanism belongs to the Goodhart family of failures.

Structural tags. school algebra, optimization.

Volume equation 110 · Case A · A low-carbon, water-aware, resilient data center

Location. Chapter 18 · Worked translations I: infrastructure, health, and adaptive control — Case A · A low-carbon, water-aware, resilient data center.

Prerequisites. Units 1–7, Units 23–31, Units 22 and 33.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. sum: summation over the displayed index range; x: a decision, state, or generic variable; local text determines the role; z: an auxiliary or third variable; t: time or a continuous index; p: probability, dimension, exponent, density, or primal value according to context; h: an equality constraint or generic function; H: a hypothesis, Hessian, entropy, or Hamiltonian

$$\min_{x,z,T,b} \sum_{s,t}[p_{st}E_{st} + \lambda_c c_{st}E_{st} + \lambda_w \omega_{st} W_{st} + \lambda_d D_{st} + \lambda_h H_{st}]. \tag{V.110}$$

Context in the volume. Here E_{st} is electricity use, c_{st} an operational emissions factor, W_{st} water consumption, ω_{st} a scarcity or local-impact factor, D_{st} service degradation, and H_{st} hardware or battery wear. Equation (110) is useful for computation and inadequate as governance. Its weights imply substitution: enough latency benefit can purchase water burden; enough cost saving can purchase carbon. The organization may reject those trades.

Structural tags. school algebra, optimization, dynamics/control.

Volume equation 111 · Case A · A low-carbon, water-aware, resilient data center

Location. Chapter 18 · Worked translations I: infrastructure, health, and adaptive control — Case A · A low-carbon, water-aware, resilient data center.

Prerequisites. Units 1–7, Units 22 and 33.

Read it as. Add the indexed contributions over the stated range, keeping each weight, unit, and constraint interpretation explicit.

Recurring symbols. sum: summation over the displayed index range; j: a second index, commonly for equality constraints or states; x: a decision, state, or generic variable; local text determines the role; C: a set, cost, covariance, or constant; z: an auxiliary or third variable; k: an iteration, time step, or generic index; t: time or a continuous index

$$\sum_j a_{jk}\, x_{jst} \le C_{kst} z_{st} \qquad \text{for each resource } k\text{, site } s\text{, and time } t. \tag{V.111}$$

Context in the volume. Deadline, precedence, data-locality, and replication constraints define workload feasibility. Thermal dynamics constrain temperatures and ramp rates. Power-system contracts constrain peak draw and battery behavior. Reliability requires reserve margins, failure-domain diversity, and recovery objectives. Security requires authenticated control, least privilege, and separation between optimization and protection systems.

Structural tags. school algebra, dynamics/control.

Volume equation 112 · Case B · An equitable heat-health program

Location. Chapter 18 · Worked translations I: infrastructure, health, and adaptive control — Case B · An equitable heat-health program.

Prerequisites. Units 1–7, Units 23–31, Units 22 and 33, Unit 32.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. xi: an uncertain quantity or scenario; sum: summation over the displayed index range; x: a decision, state, or generic variable; local text determines the role; g: an inequality constraint or generic function; R: a reward, requirement, relation, or real-number set

Context in the volume. For group g, scenario ξ, and portfolio x, let $R_g(x,\xi)$ be residual risk of a defined adverse health outcome over a stated period. A pure utilitarian objective would minimize population-weighted expected risk:

$$\min_x \sum_g N_g \; \mathbb{E}\big[R_g(x,\xi)\big]. \tag{V.112}$$

Structural tags. school algebra, optimization, dynamics/control, uncertainty/risk.

Volume equation 113 · Case B · An equitable heat-health program

Location. Chapter 18 · Worked translations I: infrastructure, health, and adaptive control — Case B · An equitable heat-health program.

Prerequisites. Units 1–7, Units 18–22, Units 23–31, Units 22 and 33, Unit 32.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. alpha: a step size, confidence/tail level, weight, or scalar; lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; xi: an uncertain quantity or scenario; sum: summation over the displayed index range; x: a decision, state, or generic variable; local text determines the role; X: a random variable, matrix, state space, or set; H: a hypothesis, Hessian, entropy, or Hamiltonian; g: an inequality constraint or generic function; R: a reward, requirement, relation, or real-number set; L: a Lagrangian integrand, loss, queue length, or bound

$$\min_{x \in X_H} \left(\sum_g N_g \ \mathbb{E}\left[R_g(x,\xi)\right] + \lambda \, \mathrm{CVaR}_\alpha\left(L(x,\xi)\right) \right). \tag{V.113}$$

Context in the volume. Equation (113) combines average benefit, tail protection, group-specific limits, budget, and accessibility. The thresholds are not generated by the solver. They come from public-health evidence, legal duties, capability judgments, and deliberation. Sen's and Nussbaum's capability approaches help explain why access to bodily health, mobility, affiliation, and control cannot be inferred from aggregate welfare alone.

Structural tags. school algebra, probability/statistics, optimization, dynamics/control, uncertainty/risk.

Volume equation 114 · Case C · Closed-loop insulin delivery as adaptive care

Location. Chapter 18 · Worked translations I: infrastructure, health, and adaptive control — Case C · Closed-loop insulin delivery as adaptive care.

Prerequisites. Units 1–7, Units 23–31, Units 22 and 33.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. sum: summation over the displayed index range; t: time or a continuous index; H: a hypothesis, Hessian, entropy, or Hamiltonian; k: an iteration, time step, or generic index

$$\min_{u_{t:t+H-1} \in \mathcal{U}_t} \sum_{k=1}^{H} \left[\phi(G_{t+k}) + r\,(u_{t+k-1} - u_{t+k-2})^2\right]. \tag{V.114}$$

Context in the volume. The low-glucose weight w_l is much larger than the high-glucose weight over the near horizon because harms are asymmetric. The positive part ensures that only violations beyond the range are penalized in these terms. Dose and insulin-on-board bounds supply safety constraints. Equation (114) illustrates receding-horizon structure; a clinical implementation requires richer models, fault handling, verification, and evidence.

Structural tags. school algebra, optimization, dynamics/control.

Volume equation 115 · Case D · A public library collection and discovery system

Location. Chapter 19 · Worked translations II: knowledge, creativity, fabrication, and climate — Case D · A public library collection and discovery system.

Prerequisites. Units 23–31.

Read it as. Choose the subscripted decision so the displayed quantity is as large as the constraints allow.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; X: a random variable, matrix, state space, or set; L: a Lagrangian integrand, loss, queue length, or bound; B: a matrix, set, or budget; Q: a quadratic matrix, action-value function, or distribution; C: a set, cost, covariance, or constant

$$\max_{x\in X_L}\left[U(x)+\lambda_B B(x)+\lambda_Q Q(x)+\lambda_S S(x)-\lambda_C C(x)\right]. \tag{V.115}$$

Context in the volume. Equation (115) places representation, access, preservation, licensing, and staff cost into feasibility. It leaves several choices political and professional: which floors are adequate, how preservation is designated, and how breadth is described. A Pareto set across use, breadth, representation, preservation, cost, and staff burden should accompany any chosen scalarization.

Structural tags. optimization.

Volume equation 116 · Case E · A musical prosthesis co-designed as an instrument

Location. Chapter 19 · Worked translations II: knowledge, creativity, fabrication, and climate — Case E · A musical prosthesis co-designed as an instrument.

Prerequisites. Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. pi: a policy, probability, or mathematical constant; x: a decision, state, or generic variable; local text determines the role; L: a Lagrangian integrand, loss, queue length, or bound; F: a vector-valued map, aggregate objective, or transition model; m: a count, dimension, or slope when defined locally; C: a set, cost, covariance, or constant; g: an inequality constraint or generic function; A: a matrix, event, or set; R: a reward, requirement, relation, or real-number set

$$\min_{x,\pi}\left(-E(x,\pi),\ L(x,\pi),\ F(x,\pi),\ m(x),\ C(x,\pi)\right)\quad \text{subject to}\quad g_{\text{safety}}(x)\le 0,\ A_r(x,\pi)\ge \underline{a}_r\ \forall r\in\mathcal{R}. \tag{V.116}$$

Context in the volume. The repertoire set $\mathcal{R}$ is selected with the musician and should include demanding, ordinary, and personally important tasks. Equation (116) does not define expressive quality. E may combine reachable dynamics, timing, articulation, continuous control, gesture distinction, and qualitative evaluation. It remains an evolving model of possibility.

Structural tags. optimization.

Volume equation 117 · Case F · A yield-aware, energy-aware semiconductor fabrication system

Location. Chapter 19 · Worked translations II: knowledge, creativity, fabrication, and climate — Case F · A yield-aware, energy-aware semiconductor fabrication system.

Prerequisites. Units 1–7, Units 8–10, Units 23–31.

Read it as. Choose the subscripted decision so the displayed objective is as small as the constraints allow.

Recurring symbols. sum: summation over the displayed index range; x: a decision, state, or generic variable; local text determines the role; t: time or a continuous index; Q: a quadratic matrix, action-value function, or distribution

$$\min_{x} \left[\sum_{l} w_l T_l(x) + \lambda_W \sum_t W_t(x) + \lambda_E \sum_t E_t(x) + \lambda_Q \sum_l Q_l(x) + \lambda_\Delta \parallel x - x^{\text{prev}} \parallel_1\right] \tag{V.117}$$

Context in the volume. subject to route precedence, tool and operator capacity, qualification, batching, contamination, queue-time, maintenance, safety, and due-date commitments. T_l is tardiness, W_t work in process or congestion, E_t energy, Q_l a validated yield-risk indicator, and the last term discourages schedule churn. Equation (117) is a coordination model, not a claim that yield can be purchased with enough delivery benefit.

Structural tags. school algebra, linear algebra, optimization.

Volume equation 118 · Case G · A coastal adaptation pathway under deep uncertainty

Location. Chapter 19 · Worked translations II: knowledge, creativity, fabrication, and climate — Case G · A coastal adaptation pathway under deep uncertainty.

Prerequisites. Units 23–31, Unit 32.

Read it as. Compare a decision against an inner worst case or competing choice, respecting the order of the two optimizations.

Recurring symbols. lambda: a Lagrange multiplier, arrival rate, eigenvalue, or scalar; pi: a policy, probability, or mathematical constant; xi: an uncertain quantity or scenario; L: a Lagrangian integrand, loss, queue length, or bound; C: a set, cost, covariance, or constant; J: an objective functional, Jacobian, or performance criterion

$$\min_{\pi \in \Pi_{\text{safe}}} \left(\mathbb{E}_\xi[L(\pi,\xi) + C(\pi,\xi)] + \lambda \max_{\xi \in \Xi}[J(\pi,\xi) - J^*(\xi)]\right). \tag{V.118}$$

Context in the volume. The regret term compares a pathway with the best policy that would have been chosen with hindsight in each scenario. Equation (118) favors policies that remain acceptable across futures rather than one policy optimized for a single probability distribution. The feasible set A contains legal, ecological, engineering, financial, and consent requirements.

Structural tags. optimization, uncertainty/risk.

Volume equation 119 · Computational and institutional frontiers

Location. Chapter 20 · Future frontiers, stopping rules, refusal, and synthesis — Computational and institutional frontiers.

Prerequisites. Unit 32.

Read it as. Continue search only when the expected decision-relevant gain from another iteration exceeds the combined compute, delay, and risk costs.

Recurring symbols. V: a vector space or value function; n: sample size, dimension, horizon, or problem-size parameter; C: a set, cost, covariance, or constant

$$\text{continue search only if} \qquad \mathbb{E}[\Delta V_{n+1} \mid \mathcal{I}_n] > C_{\text{compute}} + C_{\text{delay}} + C_{\text{risk}}, \tag{V.119}$$

Context in the volume. where $\mathcal{I}_n$ is current information and ΔV_{n+1} is decision-relevant improvement, not merely a lower numerical objective. Equation (119) is qualitative unless these terms are estimated, but it forces the right comparison.

Structural tags. uncertainty/risk.

Volume equation 120 · Differential quantities

Location. Appendix A · Mathematical toolkit — Differential quantities.

Prerequisites. Units 1–7, Units 8–10, Units 12–17.

Read it as. The derivatives quantify local sensitivity; equality or balance supplies a stationarity or evolution condition.

Recurring symbols. nabla: gradient or derivative operator; partial: partial derivative symbol; f: an objective or generic function; x: a decision, state, or generic variable; local text determines the role; n: sample size, dimension, horizon, or problem-size parameter

Context in the volume. For $f: \mathbb{R}^n \to \mathbb{R}$, the gradient is

$$\nabla f(x) = [\partial f / \partial x_1 \quad \cdots \quad \partial f / \partial x_n]^T. \tag{V.120}$$

Structural tags. school algebra, linear algebra, calculus.

Volume equation 121 · Differential quantities

Location. Appendix A · Mathematical toolkit — Differential quantities.

Prerequisites. Units 1–7.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. f: an objective or generic function; x: a decision, state, or generic variable; local text determines the role; t: time or a continuous index

$$Df(x)[d] = \lim_{t \downarrow 0} \frac{f(x+td) - f(x)}{t}. \tag{V.121}$$

Context in the volume. Automatic differentiation evaluates derivatives of a program by repeatedly applying the chain rule. Forward mode is efficient for few inputs; reverse mode is efficient for few scalar outputs, subject to storage and control-flow costs.

Structural tags. school algebra.

Volume equation 122 · Probability and statistics

Location. Appendix A · Mathematical toolkit — Probability and statistics.

Prerequisites. Units 1–7, Units 18–22.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. X: a random variable, matrix, state space, or set

Context in the volume. Expectation, variance, and covariance are

$$\mathbb{E}[X], \qquad \mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}X)^2], \qquad \mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}X)(Y - \mathbb{E}Y)]. \quad \text{(V.122)}$$

Structural tags. school algebra, probability/statistics.

Volume equation 123 · Probability and statistics

Location. Appendix A · Mathematical toolkit — Probability and statistics.

Prerequisites. Units 1–7, Units 18–22.

Read it as. Read the equality from left to right as a definition, identity, update, or balance, using the surrounding section to determine its claim.

Recurring symbols. P: probability, transition matrix, or distribution; A: a matrix, event, or set; B: a matrix, set, or budget

$$P(A \mid B) = \frac{P(B|A)P(A)}{P(B)}. \quad \text{(V.123)}$$

Context in the volume. A confidence interval is a procedure with repeated-sampling coverage under assumptions; it is not generally a posterior probability that the parameter lies in the realized interval. A Bayesian credible interval is a posterior probability statement conditional on model and prior. Prediction intervals address future observations and are usually wider.

Structural tags. school algebra, probability/statistics.

Volume equation 124 · Optimization statements

Location. Appendix A · Mathematical toolkit — Optimization statements.

Prerequisites. Units 23–31, Units 22 and 33.

Read it as. Choose the subscripted variable or policy from its admissible set to attain the displayed best value.

Recurring symbols. x: a decision, state, or generic variable; local text determines the role; X: a random variable, matrix, state space, or set; f: an objective or generic function

$$x^* \in \arg\min_{x \in X} f(x) \quad \text{(V.124)}$$

Context in the volume. returns minimizers; $\min_{x \in X} f(x)$ returns the minimum value. An infimum may exist without being attained. A local minimum has no better feasible point in a neighborhood. A global minimum has no better feasible point anywhere in X.

Structural tags. optimization, dynamics/control.

Part F.10 · Notation concordance

Notation is typed: a symbol's role is determined by its local definition and mathematical object. The entries below record recurring conventions in the volume and appendix. When a local definition differs, the local definition has priority.

x. a decision, state, or generic variable; local text determines the role.

y. an output, response, path, or second variable.

z. an auxiliary or third variable.

t. time or a continuous index.

i. an item, component, sample, or constraint index.

j. a second index, commonly for equality constraints or states.

k. an iteration, time step, or generic index.

m. a count, dimension, or slope when defined locally.

n. sample size, dimension, horizon, or problem-size parameter.

p. probability, dimension, exponent, density, or primal value according to context.

q. a dual function, probability, or auxiliary quantity.

f. an objective or generic function.

g. an inequality constraint or generic function.

h. an equality constraint or generic function.

F. a vector-valued map, aggregate objective, transition model, or functor.

G. a functor or generic map.

J. an objective functional, Jacobian, or performance criterion.

L. a Lagrangian function or integrand, loss, queue length, or bound.

M. a finite penalty coefficient or maintainability measure, according to local context.

A. a matrix, event, set, scalar availability, or linear operator.

B. a matrix, set, or budget.

C. a set, cost, covariance, or constant.

H. a hypothesis, Hessian, entropy, or Hamiltonian.

P. probability, transition matrix, distribution, or an open component.

Q. a quadratic matrix, action-value function, or distribution.

R. a reward, requirement, relation, or real-number set.

U. an uncertainty set or an upper bound, according to local context.

V. a vector space or value function.

X. a feasible or decision space, random variable, matrix, state space, or set.

α. a step size, confidence/tail level, weight, or scalar.

β. a coefficient, weight, or scalar.

γ. a discount factor, coupling, or scalar.

ε. measurement or estimation error, a tolerance, an accuracy or privacy parameter, or a regularization weight, according to local context.

η. a threshold, auxiliary optimization variable, or natural transformation.

θ. a parameter, mixture weight, or preference parameter.

λ. a Lagrange multiplier, arrival rate, eigenvalue, or scalar.

μ. a probability measure, service rate, mean, or distribution.

ν. an equality multiplier or probability measure.

ξ. an uncertain quantity or scenario.

π. a policy, probability, or mathematical constant.

Π. a policy set or a set of couplings.

ρ. a risk/aggregation functional, utilization, or correlation.

σ. standard deviation, singular value, noise scale, or strategy.

Φ. a terminal cost.

Ω. a sample space or domain.

E. expectation under a stated probability law.

∇. gradient or derivative operator.

∂. a partial-derivative symbol or subdifferential operator, according to context.

∑. summation over the displayed index range.

∫. integration over the displayed domain.

Part F.11 · Mathematics glossary

Adjoint. An operator that transfers a linear map across an inner product; central to efficient sensitivity calculations.

Affine. A linear expression plus a constant offset.

Ambiguity set. A set of probability distributions considered plausible in distributionally robust optimization.

Argmin. The set of admissible arguments at which an objective attains its minimum.

Bayes' rule. The identity that updates a prior using a likelihood and normalizing evidence.

Bellman equation. A recursion relating present value to immediate consequence and optimal continuation value.

Binary variable. A decision variable restricted to zero or one.

Certificate. Checkable evidence supporting feasibility, a bound, optimality, approximation, or another scoped claim.

Compact set. A set with the finite-cover property; in Euclidean space, equivalently closed and bounded.

Complementary slackness. The KKT relation stating that an inequality's slack and multiplier cannot both be nonzero.

Condition number. A measure of worst-case sensitivity of an output or solution to small input perturbations.

Conditional probability. Probability evaluated within the event specified after the conditioning bar.

Convex function. A function whose value at a mixture is no greater than the same mixture of endpoint values.

Convex set. A set containing the line segment between every pair of its points.

Correlation. Standardized covariance; a measure of linear association, not causation.

Covariance. Expected product of centered random variables, measuring joint linear variation.

CVaR. Conditional value at risk; an average of losses in a specified upper tail.

Derivative. A limiting local rate of change.

Differential. The linear map giving first-order change; for scalar functions it is naturally a covector.

Dual problem. An optimization problem providing bounds and sometimes equivalent value to a primal problem.

Entropy. Expected self-information of a probability distribution.

Expectation. Probability-weighted average of a random variable.

Feasible set. All decisions satisfying the stated constraints.

Functional. A mapping from a function space to numbers or another scalar-like codomain.

Functor. A mapping between categories that preserves identities and composition.

Gradient. The vector representing a scalar differential after an inner product is chosen.

Hessian. The matrix or bilinear form of second derivatives of a scalar function.

Inner product. A pairing that measures alignment and induces a norm.

Jacobian. The derivative matrix of a vector-valued function.

KKT conditions. Stationarity, primal feasibility, dual feasibility, and complementary slackness for constrained optimization.

KL divergence. A directed information discrepancy between probability distributions; not a metric.

Lagrangian. Objective plus constraint functions weighted by multipliers.

Likelihood. Observed-data probability or density viewed as a function of model parameters.

Limit. The value approached by a function or sequence under a specified mode of approach.

Markov property. Conditional independence of the future from earlier history given the present state.

Metric. A distance satisfying nonnegativity, identity, symmetry, and the triangle inequality.

Morphism. An arrow or structure-preserving process between objects in a category.

Natural transformation. A coherent family of morphisms comparing two functors.

Norm. A magnitude function obeying positivity, homogeneity, and the triangle inequality.

Objective. The quantity, vector, or ordering used to compare admissible consequences.

Pareto front (also Pareto frontier). Objective images of nondominated feasible alternatives.

Positive semidefinite. Of a symmetric matrix: having nonnegative quadratic form in every direction.

Posterior. A probability distribution after incorporating observed evidence.

Prior. A probability distribution before the current evidence is incorporated.

Projection. A nearest-point map under a specified geometry.

Random variable. A measurable numerical function of an uncertain outcome.

Residual. A discrepancy remaining after fitting, solving, or enforcing an equation.

Robust optimization. Optimization protecting against every modeled parameter in an uncertainty set.

Scalarization. A rule converting multiple objectives into one comparison.

Shadow value. A local sensitivity of optimal value to a constraint bound, often represented by a multiplier.

Stochastic optimization. Optimization involving a probability model for uncertainty.

Surrogate. A model used in place of an expensive or unavailable objective or constraint.

Topology. Neighborhood and continuity structure independent of a particular coordinate distance.

Trust region. A neighborhood in which a local approximation or model is treated as credible.

Uncertainty set. A set of parameter values protected against in robust optimization.

Variance. Expected squared deviation from a random variable’s mean.

Wasserstein distance. A distance between probability measures induced by optimal transport cost.

Part F.12 · Mathematical concept index

This index uses stable appendix unit locators rather than page numbers, so it remains valid under later repagination.

Part F.13 · Appendix references and validation ledger

Reference validation date: 16 August 2026. Entries reused from the main volume inherit its validated audit trail. New pedagogical entries were checked against official publisher, author, DOI, or approved-open-textbook records. Identifiers are included only when they resolved to the cited work; a missing DOI is not replaced by a guessed identifier.

[F1] Lynn Marecek, MaryAnne Anthony-Smith, and Andrea Honeycutt Mathis. Prealgebra 2e. OpenStax, 2020. ISBN 978-1-951693-19-0. https://openstax.org/details/books/prealgebra-2e *Validation:* official publisher page.

[F2] Jay Abramson. Algebra and Trigonometry 2e. OpenStax, 2021. https://openstax.org/books/algebra-and-trigonometry-2e/pages/1-introduction-to-prerequisites *Validation:* official publisher text and citation metadata.

[F3] Gilbert Strang and Edwin 'Jed' Herman. Calculus, Volumes 1–3. OpenStax, 2016. Volume 1 ISBN 978-1-947172-13-5; Volume 3 ISBN 978-1-947172-16-6. https://openstax.org/k12/calculus *Validation:* official publisher series page.

[F4] Richard Hammack. Book of Proof, 3rd ed. Virginia Commonwealth University, 2018. https://richardhammack.github.io/BookOfProof/ *Validation:* author/AIM approved-textbook record.

[F5] Gilbert Strang. Introduction to Linear Algebra, 6th ed. Wellesley-Cambridge Press, 2023. ISBN 978-1-7331466-7-8. https://epubs.siam.org/doi/abs/10.1137/1.9781733146678 *Validation:* SIAM bibliographic record.

[F6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, 4th ed. MIT Press, 2022. ISBN 978-0-262-04630-5. https://mitpress.mit.edu/9780262046305/introduction-to-algorithms/ *Validation: official MIT Press catalog record and ISBN checksum.*

[F7] Michael R. Garey and David S. Johnson. Computers and Intractability. W. H. Freeman, 1979. ISBN 978-0-7167-1045-5. *Validation: ISBN checksum and bibliographic identity.*

[F8] Barbara Illowsky and Susan Dean. Introductory Statistics 2e. OpenStax, 2023. ISBN 978-1-961584-32-7. https://openstax.org/details/books/introductory-statistics-2e *Validation:* official publisher page.

[F9] Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, 2004. doi:10.1017/CBO9780511804441. *Validation: DOI resolution and Cambridge University Press metadata.*

[F10] Jorge Nocedal and Stephen J. Wright. Numerical Optimization, 2nd ed. Springer, 2006. doi:10.1007/978-0-387-40065-5. *Validation: DOI resolution and Springer metadata.*

[F11] Miguel A. Hernán and James M. Robins. Causal Inference: What If. Chapman & Hall/CRC, 2020. https://miguelhernan.org/whatifbook *Validation:* author's official book page.

[F12] Patrick Billingsley. Probability and Measure, 3rd ed. Wiley, 1995. ISBN 978-0-471-00710-4. *Validation:* publisher bibliographic identity.

[F13] Martin L. Puterman. Markov Decision Processes. Wiley, 1994. doi:10.1002/9780470316887. *Validation: DOI resolution and Wiley metadata.*

[F14] David G. Luenberger. Optimization by Vector Space Methods. Wiley, 1969; reprint 1998. ISBN 978-0-471-18117-0. *Validation: official Wiley record and ISBN checksum; used for vector spaces, function spaces, duality, and adjoints.*

[F15] R. Tyrrell Rockafellar. Convex Analysis. Princeton University Press, 1970. ISBN 978-0-691-01586-6. The ISBN identifies the 1997 first paperback edition. *Validation: Princeton University Press catalog record and ISBN checksum.*

[F16] Sébastien Bubeck. Convex Optimization: Algorithms and Complexity. Foundations and Trends in Machine Learning 8(3–4), 2015. doi:10.1561/2200000050. *Validation: DOI resolution and journal metadata.*

[F17] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. Lectures on Stochastic Programming, 3rd ed. SIAM, 2021. doi:10.1137/1.9781611976595. *Validation: DOI resolution and SIAM metadata; used for stochastic, distributional, and risk foundations.*

[F18] Kaisa Miettinen. Nonlinear Multiobjective Optimization. Kluwer, 1998. doi:10.1007/978-1-4615-5563-6. *Validation: DOI resolution and Springer metadata.*

[F19] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. Robust Optimization. Princeton University Press, 2009. doi:10.1515/9781400831050. *Validation: DOI resolution and publisher metadata.*

[F20] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of Conditional Value-at-Risk. Journal of Risk 2(3), 2000. doi:10.21314/JOR.2000.038. *Validation: DOI resolution and journal metadata.*

[F21] Richard Bellman. Dynamic Programming. Princeton University Press, 1957. ISBN 978-0-691-14668-3. The ISBN identifies the 2010 Princeton Landmarks paperback edition. *Validation: Princeton University Press catalog record and ISBN checksum.*

[F22] Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction, 2nd ed. MIT Press, 2018. ISBN 978-0-262-03924-6. *Validation: MIT Press catalog record and ISBN checksum.*

[F23] Lev S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. The Mathematical Theory of Optimal Processes. Interscience, 1962. *Validation: publisher and library catalog metadata.*

[F24] Rudolf E. Kalman. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering 82(1), 1960, 35–45. doi:10.1115/1.3662552. *Validation: DOI resolution and ASME journal metadata.*

[F25] John F. Nash. Equilibrium Points in n-Person Games. PNAS 36(1), 1950, 48–49. doi:10.1073/pnas.36.1.48. *Validation: DOI resolution and PNAS metadata.*

[F26] John D. C. Little. A Proof for the Queuing Formula: L = λW. Operations Research 9(3), 1961, 383–387. doi:10.1287/opre.9.3.383. *Validation: DOI resolution and INFORMS journal metadata.*

[F27] John F. Shortle, James M. Thompson, Donald Gross, and Carl M. Harris. Fundamentals of Queueing Theory, 5th ed. Wiley, 2018. doi:10.1002/9781119453765. *Validation:* official Wiley DOI record.

[F28] James R. Munkres. Topology, 2nd ed., Classic Version. Pearson, 2023. ISBN 978-0-13-784866-9. *Validation:* official Pearson record.

[F29] Cédric Villani. Topics in Optimal Transportation. American Mathematical Society, 2003. doi:10.1090/gsm/058. *Validation: DOI resolution and American Mathematical Society metadata.*

[F30] Gabriel Peyré and Marco Cuturi. Computational Optimal Transport with Applications to Data Sciences. Foundations and Trends in Machine Learning 11(5–6), 2019. doi:10.1561/2200000073. *Validation: DOI resolution and journal metadata.*

[F31] Claude E. Shannon. A Mathematical Theory of Communication. Bell System Technical Journal 27, 1948, 379–423 and 623–656. doi:10.1002/j.1538-7305.1948.tb01338.x. *Validation: DOI resolution and Wiley journal metadata.*

[F32] Thomas M. Cover and Joy A. Thomas. Elements of Information Theory, 2nd ed. Wiley, 2006. doi:10.1002/047174882X. *Validation:* official Wiley DOI record.

[F33] Saunders Mac Lane. Categories for the Working Mathematician, 2nd ed. Springer, 1998. ISBN 978-0-387-98403-2. *Validation: Springer catalog record and ISBN checksum.*

[F34] Emily Riehl. Category Theory in Context. Dover, 2016. ISBN 978-0-486-80903-8. *Validation:* publisher/author bibliographic record.

[F35] Brendan Fong and David I. Spivak. An Invitation to Applied Category Theory. Cambridge University Press, 2019. doi:10.1017/9781108668804. *Validation: DOI resolution and Cambridge University Press metadata.*

Glossary

Abstraction. A representation that preserves selected structure while suppressing other detail. Its usefulness depends on what the decision requires.

Active constraint. An inequality constraint satisfied at equality at a candidate solution. Its multiplier may indicate marginal value.

Active learning. Sequential selection of data or experiments intended to improve a model efficiently.

Adjoint method. A technique for computing the derivative of a scalar output with respect to many inputs by solving a backward or dual system.

Admissible decision. A choice permitted by all hard constraints, rights, and governance requirements; often synonymous with feasible when those requirements are modeled.

Algorithm. A finite operational procedure transforming inputs into outputs. Optimization algorithms search, update, bound, or certify.

Approximation algorithm. An algorithm with a proved bound comparing its solution value with the optimum, typically for a difficult discrete problem.

Argmin / argmax. The set of decision values attaining the minimum or maximum. This differs from the attained objective value.

Bayesian optimization. Global optimization using a probabilistic surrogate and an acquisition rule to select expensive evaluations.

Bellman equation. A recursive relation expressing the value of a state as immediate reward or cost plus the optimized value of successor states.

Bilevel optimization. An optimization problem whose constraints include the solution of another optimization problem, often representing a follower or training process.

Bounded rationality. Decision-making under limited information, time, attention, and computation. It treats the decision procedure and its costs as part of the problem rather than assuming omniscient maximization.

Branch-and-bound. Exact search that partitions a feasible set and uses bounds to discard regions that cannot improve the incumbent.

Bregman divergence. A generally asymmetric discrepancy generated by a differentiable strictly convex function, used in mirror descent and information geometry.

Calibration. Agreement between predicted probabilities and observed frequencies in relevant groups and conditions.

Category. In category theory, a collection of objects and morphisms with associative composition and identity morphisms.

Certificate. Checkable evidence for a claim such as optimality, a bound, feasibility, infeasibility, or program correctness.

Chance constraint. A requirement that a constraint hold with at least a specified probability.

Co-design. Joint optimization of coupled layers such as hardware and software, product and process, or policy and implementation.

Combinatorial optimization. Optimization over discrete structures such as sets, graphs, assignments, sequences, or partitions.

Compositionality. The property that the behavior or guarantee of a whole can be constructed from components and their interfaces.

Condition number. A measure of sensitivity of a problem or computation to perturbations; large values indicate error amplification.

Constraint. A condition a decision must satisfy. Treating a requirement as a penalty changes its logical status.

Constraint qualification. A regularity condition under which multiplier or duality results such as KKT conditions hold.

Convex function. A function whose value at a mixture is no greater than the corresponding mixture of values. Local minima are global on convex feasible sets.

Convex set. A set containing every line segment between any two of its points.

Counterfactual. A claim about what would occur under an alternative action or condition, central to causal reasoning.

Critical path. A longest-duration precedence path determining a project's earliest completion in a deterministic network model.

Cross-validation. Repeated splitting of data for model selection and performance estimation; valid only when splits respect the deployment dependency structure.

CVaR (conditional value at risk). The mean loss in a specified upper tail, with conventions varying at discontinuities.

Decision variable. A quantity, structure, policy, or artifact the decision maker is permitted to choose.

Derivative-free optimization. Search that does not require analytic gradients, useful for black-box, discontinuous, or noisy objectives.

Digital twin. A maintained relationship among a physical or operational entity, data, models, and decisions; not merely a static simulation.

Discount rate. A parameter converting future costs or benefits into present value. It can encode opportunity cost and ethical assumptions about time.

Dominance. In multiobjective minimization, solution x dominates y if it is no worse in every objective and better in at least one.

Dual problem. A related optimization problem that provides bounds, sensitivities, decomposition, or alternative interpretation of the primal.

Duality gap. The difference between primal and dual objective values. A zero or small gap can certify optimality under suitable conditions.

Dynamic programming. Optimization by recursively solving overlapping subproblems indexed by a sufficient state.

Endogeneity. Dependence created within the system, such as behavior changing in response to a policy or metric.

Equilibrium. A state consistent with the decisions or dynamics of interacting agents. Equilibrium need not be efficient, fair, stable, or unique.

Estimand. The precise quantity an analysis intends to estimate, such as an average causal effect for a population.

Expected value of information. The expected improvement in decision quality made possible by obtaining additional information, before subtracting its acquisition cost or burden.

Externality. A cost or benefit imposed on parties not fully represented in the decision or transaction.

Feasible set. All decisions satisfying the modeled constraints.

Feedback. A loop in which outcomes affect later inputs or decisions. Feedback can stabilize, amplify, oscillate, or change the objective's meaning.

Fitness. In evolutionary biology, reproductive contribution relative to an environment and population context; not a general measure of worth or perfection.

Functor. A mapping between categories that maps objects and morphisms while preserving identities and composition.

Generalization. Performance of a learned model beyond its training observations under a specified target distribution or range of shift.

Global optimum. A feasible solution no worse than every other feasible solution under the objective ordering.

Goodhart's law. The recurring failure in which a measure loses reliability as a measure when it becomes a target.

Gradient. The vector representing the differential of a scalar function under a chosen inner product; in Euclidean space it points toward steepest local increase.

Hard constraint. A condition that may not be violated. It differs from a soft constraint or penalty, which permits violation at a cost.

Heuristic. A search or decision rule useful in practice without a complete guarantee of optimality.

Hyperparameter. A model or training choice set outside the immediate parameter-fitting problem, often selected in an outer optimization loop.

Identification. Conditions under which a causal or statistical quantity is uniquely determined by observed data and assumptions.

Incumbent. The best feasible solution currently known during a search.

Invariant. A property intended to remain unchanged under a transformation, composition, or optimization step.

KKT conditions. Stationarity, primal feasibility, dual feasibility, and complementary slackness conditions for constrained optimization.

Lagrangian. The objective augmented by constraint functions weighted by multipliers.

Latency. Time between initiating and completing an operation; averages and high percentiles answer different service questions.

Learning rate. The step-size parameter in gradient-based learning or stochastic approximation.

Lexicographic optimization. Optimization in strict priority order: improve a lower-priority objective only without worsening higher-priority objectives.

Lifecycle boundary. The stages and actors included in evaluation, such as extraction, manufacture, use, maintenance, and disposal.

Local optimum. A feasible solution no worse than nearby feasible solutions. It need not be globally best.

Loss function. A numerical representation of error or harm used in estimation or decision-making. It is a designed surrogate, not an objective fact.

Mechanism design. Design of rules or institutions to achieve properties when participants act strategically and hold private information.

Metric. A measure of performance; mathematically, also a distance satisfying specific axioms. The two senses should not be confused.

Mixed-integer program. An optimization model containing both continuous and integer-constrained variables.

Model predictive control. Feedback control that repeatedly solves a constrained finite-horizon problem, applies an initial action, observes, and replans.

Multiobjective optimization. Optimization with multiple criteria whose trade-offs may be represented by dominance, scalarization, goals, or preference interaction.

Nash equilibrium. A strategy profile in which no player can improve its payoff by changing strategy unilaterally. Every finite game has an equilibrium in mixed strategies, although a pure-strategy equilibrium need not exist.

No-free-lunch result. For the finite search setting analyzed by Wolpert and Macready, uniform averaging over a permutation-closed set of objective functions makes all non-revisiting black-box optimizers equivalent in aggregate. The conclusion depends on those conditions; useful search requires assumptions that privilege some problem structure (Wolpert and Macready 1997).

Normal cone. A set of vectors supporting a feasible set at a point, used to express constrained stationarity.

Objective function. A mapping used to rank or compare feasible decisions. It is an operational representation of purpose, not purpose itself.

Online optimization. Sequential decision-making in which data or losses arrive over time and future information is unavailable.

Optimal control. Optimization of trajectories and controls subject to dynamical equations and boundary or path constraints.

Optimal transport. Optimization over couplings that move one distribution to another under a cost.

Optimality gap. A bound on how far a feasible solution's objective may be from the global optimum.

Optimizer's curse. The systematic upward bias in the estimated value of an option selected as best from noisy estimates; stronger selection tends to select favorable estimation error as well as genuine value.

Overfitting. Exploiting accidental structure in training or evaluation data so that apparent performance fails to transfer.

Overoptimization. Search pressure so strong that it exploits proxy, model, or boundary defects and worsens true outcomes.

Pareto efficiency. A property of an allocation or decision for which no feasible change can make at least one party better off without making another worse off. It says nothing by itself about equality, rights, or legitimacy.

Pareto frontier (also Pareto front). The set of nondominated feasible outcomes in a multiobjective problem.

Penalty method. A method that adds a cost for constraint violation. Finite penalties generally do not preserve the logical force of every hard constraint.

Policy. A rule mapping observed information or states to actions; also an institutional course of action. Context should disambiguate.

Preconditioning. A transformation that improves numerical geometry or conditioning, often accelerating iterative methods.

Primal problem. The original or primary formulation paired with a dual problem.

Proxy. A measurable quantity used in place of a harder-to-observe value. Proxies require validation under optimization and adaptation.

Quality-adjusted life year (QALY). A measure combining time and health-state utility. Useful for analysis but ethically incomplete as a sole allocation rule.

Queueing discipline. The rule choosing which waiting job, patient, packet, or request is served next.

Randomized algorithm. An algorithm using randomness to sample, break symmetry, or obtain probabilistic guarantees.

Recourse. Actions available after uncertainty is observed; socially, also a person's ability to challenge and correct a decision.

Regret. The cumulative difference between a sequential decision rule's loss and the loss of a specified comparator, such as the best fixed action in hindsight. A regret guarantee is meaningful only relative to its comparator and information model.

Regularization. Structure added to control complexity, instability, or undesirable solutions, commonly through a penalty or constraint.

Relaxation. A tractable problem formed by enlarging the feasible set or weakening conditions, often providing a bound.

Resilience. Capacity to sustain essential function, degrade safely, recover, and adapt after disruption.

Robust optimization. Optimization against all values in a specified uncertainty set, often emphasizing worst-case feasibility or performance.

Safety margin. Deliberate distance from a predicted limit to protect against variation, uncertainty, degradation, and error.

Satisficing. Selecting an option that meets an aspiration level or is good enough given the cost of further search, information, and delay. It can be a rational policy under bounded resources.

Scalarization. Conversion of multiple objectives into one scalar, for example by weighted sum. It embeds preferences and may miss parts of a nonconvex frontier.

Sensitivity analysis. Study of how outputs, solutions, or recommendations change when inputs and assumptions vary.

Shadow price. A dual multiplier interpreted as the local marginal value of relaxing a constraint, under regularity and units.

Simulation. Computational execution of a model through scenarios or time. Fidelity should be judged by decision-relevant behavior.

Slack. Unused capacity or distance from a constraint. Slack can be waste in one model and resilience in a broader model.

Slater's condition. A constraint qualification for convex programs requiring a strictly feasible point for inequality constraints (with the usual relative-interior refinement). It is a standard sufficient condition for strong duality and KKT necessity.

Soft constraint. A preferred condition whose violation is permitted at a modeled cost or priority.

Stochastic gradient. A noisy estimate of an objective gradient, commonly computed from a sample or minibatch.

Stochastic optimization. Optimization involving random variables, samples, scenarios, or noisy observations.

Subgradient. A generalized slope for a convex nonsmooth function. The set of all subgradients is the subdifferential.

Submodularity. A diminishing-returns property for set functions that supports strong discrete algorithms and approximations.

Sunk cost. A cost already incurred that should not affect a forward-looking decision except through resulting constraints or information.

Surrogate model. A cheaper approximation to an expensive experiment, simulator, or objective.

Sustainability. Maintenance of ecological and social conditions across time, including distribution, resilience, and intergenerational consequence.

System boundary. The chosen extent of entities, processes, time, and effects included in a model.

Throughput. Completed work per unit time. Increasing throughput can worsen delay or quality if queues and rework are ignored.

Trade-off. A relation in which improving one valued outcome worsens another within the current feasible set.

Trust region. A neighborhood in which a local model is considered reliable; steps are accepted based on predicted versus realized improvement.

Uncertainty set. A collection of plausible parameter values used in robust optimization.

Utility. A representation of preference or value used for comparison. Utility is meaningful only relative to its axioms, scale, and decision context.

Validation. Evaluation of whether a model or system is adequate for its intended real-world use. It differs from verification, which checks conformance to specification.

Value of an option. Benefit of retaining the ability to choose later after more information arrives.

Verification. Evidence that an implementation, calculation, or artifact satisfies its specified model or requirements.

Worst-case analysis. A guarantee over the most adverse admitted instance or uncertainty realization. Its relevance depends on what the set admits.

Subject index

Page references identify substantive or representative occurrences; "passim" marks concepts used throughout the field guide.

G

H

I

L

M

N

O

P

Q

R

S

References

Aarseth, Espen J. 1997. *Cybertext: Perspectives on Ergodic Literature*. Johns Hopkins University Press. https://www.press.jhu.edu/books/title/2048/cybertext. ISBN 9780801855795.

Aho, Alfred V., Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. 2006. *Compilers: Principles, Techniques, and Tools*. 2nd ed. Addison-Wesley. ISBN 9780321486813.

Alexander, Christopher. 1964. *Notes on the Synthesis of Form*. Harvard University Press. https://www.hup.harvard.edu/books/9780674627512. ISBN 9780674627512.

Amdahl, Gene M. 1967. “Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities.” In *Proceedings of the AFIPS Spring Joint Computer Conference*, 483–85. https://doi.org/10.1145/1465482.1465560.

American Library Association. 2019. “Library Bill of Rights.” https://www.ala.org/advocacy/intfreedom/librarybill.

Aristotle. 2002. *Nicomachean Ethics*. Edited by Sarah Broadie. Translated by Christopher Rowe. Oxford University Press. https://global.oup.com/academic/product/nicomachean-ethics-9780198752714. ISBN 9780198752714.

Arrow, Kenneth J. 1951. *Social Choice and Individual Values*. Wiley. Original edition; published before an ISBN was assigned to this edition; later-edition identifiers are not substituted.

Åström, Karl J., and Richard M. Murray. 2008. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press. https://fbsbook.org/. ISBN 9780691135762.

Banister, Eric W., T. W. Calvert, M. V. Savage, and T. M. Bach. 1975. “A Systems Model of Training for Athletic Performance.” *Australian Journal of Sports Medicine* 7: 57–61.

Barocas, Solon, Moritz Hardt, and Arvind Narayanan. 2023. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press. https://fairmlbook.org/. ISBN 9780262048613.

Barroso, Luiz André, Urs Hölzle, and Parthasarathy Ranganathan. 2018. *The Datacenter as a Computer*. 3rd ed. Morgan & Claypool. https://doi.org/10.2200/S00874ED3V01Y201809CAC046. ISBN 9781681734330.

Bellman, Richard. 1957. *Dynamic Programming*. Princeton University Press. https://press.princeton.edu/books/paperback/9780691146683/dynamic-programming. ISBN 9780691146683. Original edition cited; identifier and link are for the 2010 Princeton Landmarks paperback edition.

Bendsøe, Martin P., and Ole Sigmund. 2004. *Topology Optimization: Theory, Methods, and Applications*. 2nd ed., corrected printing. Springer. https://doi.org/10.1007/978-3-662-05086-6. ISBN 9783540429920.

Ben-Tal, Aharon, Laurent El Ghaoui, and Arkadi Nemirovski. 2009. *Robust Optimization*. Princeton University Press. https://doi.org/10.1515/9781400831050. ISBN 9780691143682.

Bertsimas, Dimitris, and Melvyn Sim. 2004. “The Price of Robustness.” *Operations Research* 52 (1): 35–53. https://doi.org/10.1287/opre.1030.0065.

Biegler, Lorenz T. 2010. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. SIAM. https://doi.org/10.1137/1.9780898719383. ISBN 9780898717020.

Boden, Margaret A. 2004. *The Creative Mind: Myths and Mechanisms*. 2nd ed. Routledge. https://www.routledge.com/The-Creative-Mind-Myths-and-Mechanisms/Boden/p/book/9780415314534. ISBN 9780415314534.

Bottou, Léon, Frank E. Curtis, and Jorge Nocedal. 2018. “Optimization Methods for Large-Scale Machine Learning.” *SIAM Review* 60 (2): 223–311. https://doi.org/10.1137/16M1080173.

Boyd, Stephen, and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press. https://doi.org/10.1017/CBO9780511804441. ISBN 9780521833783.

Breck, Eric, Shanqing Cai, Eric Nielsen, Michael Salib, and D. Sculley. 2017. “The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction.” In *Proceedings of IEEE Big Data*, 1123–32. https://doi.org/10.1109/BigData.2017.8258038.

Brooks, Frederick P., Jr. 1995. *The Mythical Man-Month: Essays on Software Engineering*. Anniversary. Addison-Wesley. ISBN 9780201835953.

Brown, Sue A., Boris P. Kovatchev, Dan Raghinaru, John W. Lum, Bruce A. Buckingham, et al. 2019. “Six-Month Randomized, Multicenter Trial of Closed-Loop Control in Type 1 Diabetes.” *New England Journal of Medicine* 381: 1707–17. https://doi.org/10.1056/NEJMoa1907863.

Bryson, Arthur E., Jr., and Yu-Chi Ho. 1975. *Applied Optimal Control: Optimization, Estimation, and Control*. Taylor & Francis. ISBN 9780891162285.

Bubeck, Sébastien. 2015. “Convex Optimization: Algorithms and Complexity.” *Foundations and Trends in Machine Learning* 8 (3–4): 231–357. https://doi.org/10.1561/2200000050.

Campbell, Donald T. 1976. *Assessing the Impact of Planned Social Change*. Dartmouth College Public Affairs Center. https://doi.org/10.56645/jmde.v7i15.297. DOI and link identify the authorized 2011 reprint of the original 1976 working paper.

Canguilhem, Georges. 1991. *The Normal and the Pathological*. Zone Books. https://press.princeton.edu/books/paperback/9780942299595/the-normal-and-the-pathological. ISBN 9780942299595.

Card, Stuart K., Thomas P. Moran, and Allen Newell. 1983. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates. ISBN 9780898592436.

Cerezo, Marco, Andrew Arrasmith, Ryan Babbush, et al. 2021. “Variational Quantum Algorithms.” *Nature Reviews Physics* 3: 625–44. https://doi.org/10.1038/s42254-021-00348-9.

Chouldechova, Alexandra. 2017. “Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments.” *Big Data* 5 (2): 153–63. https://doi.org/10.1089/big.2016.0047.

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2022. *Introduction to Algorithms*. 4th ed. MIT Press. https://mitpress.mit.edu/9780262046305/introduction-to-algorithms/. ISBN 9780262046305.

Cuturi, Marco. 2013. “Sinkhorn Distances: Lightspeed Computation of Optimal Transport.” In *Advances in Neural Information Processing Systems 26*, 2292–2300. https://papers.nips.cc/paper/4927-sinkhorn-distances-lightspeed-computation-of-optimal-transport.

Dantzig, George B. 1963. *Linear Programming and Extensions*. Princeton University Press. https://press.princeton.edu/books/paperback/9780691059136/linear-programming-and-extensions. ISBN 9780691059136. Original edition cited; identifier and link are for the 1998 paperback edition.

Darwin, Charles. 1859. *On the Origin of Species*. John Murray. https://darwin-online.org.uk/converted/published/1859_Origin_F373.html.

Dean, Jeffrey, and Luiz André Barroso. 2013. “The Tail at Scale.” *Communications of the ACM* 56 (2): 74–80. https://doi.org/10.1145/2408776.2408794.

Deb, Kalyanmoy. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley. https://www.wiley.com/en-us/Multi+Objective+Optimization+using+Evolutionary+Algorithms-p-9780471873396. ISBN 9780471873396.

Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II.” *IEEE Transactions on Evolutionary Computation* 6 (2): 182–97. https://doi.org/10.1109/4235.996017.

Del Castillo, Enrique, and Douglas C. Montgomery. 1995. “A Kalman Filtering Process Control Scheme with an Application in Semiconductor Short Run Manufacturing.” *Quality and Reliability Engineering International* 11 (2): 101–5. https://doi.org/10.1002/qre.4680110205.

Dempe, Stephan. 2002. *Foundations of Bilevel Programming*. Kluwer Academic Publishers. https://doi.org/10.1007/b101970. ISBN 9781402006319.

Dennard, Robert H., Fritz H. Gaensslen, Hwa-Nien Yu, V. Leo Rideout, Ernest Bassous, and Andre R. LeBlanc. 1974. “Design of Ion-Implanted MOSFET’s with Very Small Physical Dimensions.” *IEEE Journal of Solid-State Circuits* 9 (5): 256–68. https://doi.org/10.1109/JSSC.1974.1050511.

Dorigo, Marco, and Thomas Stützle. 2004. *Ant Colony Optimization*. MIT Press. https://mitpress.mit.edu/9780262042192/ant-colony-optimization/. ISBN 9780262042192.

Duchi, John, Elad Hazan, and Yoram Singer. 2011. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization.” *Journal of Machine Learning Research* 12: 2121–59. https://jmlr.org/papers/v12/duchi11a.html.

Dwork, Cynthia, and Aaron Roth. 2014. *The Algorithmic Foundations of Differential Privacy*. Now Publishers. https://doi.org/10.1561/0400000042. ISBN 9781601988188.

Eco, Umberto. 1990. *The Limits of Interpretation*. Indiana University Press. ISBN 9780253318527.

Edgar, Thomas F., David M. Himmelblau, and Leon S. Lasdon. 2001. *Optimization of Chemical Processes*. 2nd ed. McGraw-Hill. ISBN 9780070393592.

Elenes, Alejandro G. N., Eric Williams, Eric Hittinger, and Naga Srujana Goteti. 2022. “How Well Do Emission Factors Approximate Emission Changes from Electricity System Models?” *Environmental Science & Technology* 56 (20): 14701–12. https://doi.org/10.1021/acs.est.2c02344.

Elgammal, Ahmed, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. 2017. “CAN: Creative Adversarial Networks, Generating Art by Learning about Styles and Deviating from Style Norms.” In *International Conference on Computational Creativity*. https://arxiv.org/abs/1706.07068.

Elsken, Thomas, Jan Hendrik Metzen, and Frank Hutter. 2019. “Neural Architecture Search: A Survey.” *Journal of Machine Learning Research* 20 (55): 1–21. https://jmlr.org/papers/v20/18-598.html.

Endsley, Mica R. 1995. “Toward a Theory of Situation Awareness in Dynamic Systems.” *Human Factors* 37 (1): 32–64. https://doi.org/10.1518/001872095779049543.

Endsley, Mica R., and Esin O. Kiris. 1995. “The Out-of-the-Loop Performance Problem and Level of Control in Automation.” *Human Factors* 37 (2): 381–94. https://doi.org/10.1518/001872095779064555.

Espeland, Wendy Nelson, and Michael Sauder. 2007. “Rankings and Reactivity: How Public Measures Recreate Social Worlds.” *American Journal of Sociology* 113 (1): 1–40. https://doi.org/10.1086/517897.

European Parliament and Council of the European Union. 2024. *Regulation (EU) 2024/1689 Laying down Harmonised Rules on Artificial Intelligence*. https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng.

Farhi, Edward, Jeffrey Goldstone, and Sam Gutmann. 2014. “A Quantum Approximate Optimization Algorithm.” https://arxiv.org/abs/1411.4028. arXiv preprint arXiv:1411.4028.

Fisher, R. A. 1930. *The Genetical Theory of Natural Selection*. Clarendon Press. https://archive.org/details/geneticaltheoryo031631mbp.

Fong, Brendan. 2015. “Decorated Cospans.” *Theory and Applications of Categories* 30 (33): 1096–1120. https://www.tac.mta.ca/tac/volumes/30/33/30-33abs.html.

Fong, Brendan, and David I. Spivak. 2019. *An Invitation to Applied Category Theory: Seven Sketches in Compositionality*. Cambridge University Press. https://doi.org/10.1017/9781108668804. ISBN 9781108482295.

Forsgren, Nicole, Jez Humble, and Gene Kim. 2018. *Accelerate: The Science of Lean Software and DevOps*. IT Revolution Press. https://itrevolution.com/product/accelerate/. ISBN 9781942788331.

Frid, Emma. 2019. “Accessible Digital Musical Instruments—a Review of Musical Interfaces in Inclusive Music Practice.” *Multimodal Technologies and Interaction* 3 (3): 57. https://doi.org/10.3390/mti3030057.

Gale, David, and Lloyd S. Shapley. 1962. “College Admissions and the Stability of Marriage.” *American Mathematical Monthly* 69 (1): 9–15. https://doi.org/10.1080/00029890.1962.11989827.

Garey, Michael R., and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman. ISBN 9780716710455.

Genette, Gérard. 1980. *Narrative Discourse: An Essay in Method*. Cornell University Press. https://www.cornellpress.cornell.edu/book/9780801492594/narrative-discourse/. ISBN 9780801492594.

Ghani, Neil, Jules Hedges, Viktor Winschel, and Philipp Zahn. 2018. “Compositional Game Theory.” In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, 472–81. https://doi.org/10.1145/3209108.3209165.

Gigerenzer, Gerd, and Reinhard Selten, eds. 2002. *Bounded Rationality: The Adaptive Toolbox*. MIT Press. https://mitpress.mit.edu/9780262571647/bounded-rationality/. ISBN 9780262571647.

Goemans, Michel X., and David P. Williamson. 1995. “Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming.” *Journal of the ACM* 42 (6): 1115–45. https://doi.org/10.1145/227683.227684.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. https://www.deeplearningbook.org/. ISBN 9780262035613.

Goodhart, Charles A. E. 1975. “Problems of Monetary Management: The u.k. Experience.” In *Papers in Monetary Economics*. Reserve Bank of Australia. https://www.rba.gov.au/publications/confs/1975/goodhart.pdf.

Gould, Stephen Jay, and Richard C. Lewontin. 1979. “The Spandrels of San Marco and the Panglossian Paradigm: A Critique of the Adaptationist Programme.” *Proceedings of the Royal Society of London B* 205 (1161): 581–98. https://doi.org/10.1098/rspb.1979.0086.

Gustafson, John L. 1988. “Reevaluating Amdahl’s Law.” *Communications of the ACM* 31 (5): 532–33. https://doi.org/10.1145/42411.42415.

Haasnoot, Marjolijn, Jan H. Kwakkel, Warren E. Walker, and Judith ter Maat. 2013. “Dynamic Adaptive Policy Pathways: A Method for Crafting Robust Decisions for a Deeply Uncertain World.” *Global Environmental Change* 23 (2): 485–98. https://doi.org/10.1016/j.gloenvcha.2012.12.006.

Hardt, Moritz, Eric Price, and Nati Srebro. 2016. “Equality of Opportunity in Supervised Learning.” In *Advances in Neural Information Processing Systems 29*, 3315–23. https://arxiv.org/abs/1610.02413.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. 2nd ed. Springer. https://doi.org/10.1007/978-0-387-84858-7. ISBN 9780387848570.

Hennessy, John L., and David A. Patterson. 2019. “A New Golden Age for Computer Architecture.” *Communications of the ACM* 62 (2): 48–60. https://doi.org/10.1145/3282307.

Hino, Miyuki, Christopher B. Field, and Katharine J. Mach. 2017. “Managed Retreat as a Response to Natural Hazard Risk.” *Nature Climate Change* 7: 364–70. https://doi.org/10.1038/nclimate3252.

Hoffmann, Jordan, Sebastian Borgeaud, Arthur Mensch, et al. 2022. “Training Compute-Optimal Large Language Models.” arXiv:2203.15556. https://arxiv.org/abs/2203.15556.

Holland, John H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Original edition; published before an ISBN was assigned to this edition; later-edition identifiers are not substituted.

Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. “The Curious Case of Neural Text Degeneration.” In *International Conference on Learning Representations*. https://iclr.cc/virtual_2020/poster_rygGQyrFvH.html.

Horowitz, Mark. 2014. “1.1 Computing’s Energy Problem (and What We Can Do about It).” *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 10–14. https://doi.org/10.1109/ISSCC.2014.6757323.

Humble, Jez, and David Farley. 2010. *Continuous Delivery*. Addison-Wesley. ISBN 9780321601919.

Hunink, M. G. Myriam, Milton C. Weinstein, Eve Wittenberg, et al. 2014. *Decision Making in Health and Medicine: Integrating Evidence and Values*. 2nd ed. Cambridge University Press. https://doi.org/10.1017/CBO9781139506779. ISBN 9781107690479.

Hutter, Frank, Lars Kotthoff, and Joaquin Vanschoren, eds. 2019. *Automated Machine Learning: Methods, Systems, Challenges*. Springer. https://doi.org/10.1007/978-3-030-05318-5. ISBN 9783030053178.

Intergovernmental Panel on Climate Change. 2022a. “Cities and Settlements by the Sea.” Cambridge University Press. https://www.ipcc.ch/report/ar6/wg2/chapter/ccp2/.

———. 2022b. “Climate Change 2022: Mitigation of Climate Change.” Cambridge University Press. https://doi.org/10.1017/9781009157926.

International Federation of Library Associations and Institutions, and UNESCO. 2022. “The IFLA–UNESCO Public Library Manifesto 2022.” International Federation of Library Associations and Institutions. https://www.ifla.org/public-library-manifesto/.

International Organization for Standardization. 2023. *ISO/IEC 25010:2023 Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – Product Quality Model*. https://www.iso.org/standard/78176.html.

Jones, Donald R., Matthias Schonlau, and William J. Welch. 1998. “Efficient Global Optimization of Expensive Black-Box Functions.” *Journal of Global Optimization* 13: 455–92. https://doi.org/10.1023/A:1008306431147.

Jouppi, Norman P., Cliff Young, Nishant Patil, et al. 2017. “In-Datacenter Performance Analysis of a Tensor Processing Unit.” In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 1–12. https://doi.org/10.1145/3079856.3080246.

Kahneman, Daniel, and Amos Tversky. 1979. “Prospect Theory: An Analysis of Decision Under Risk.” *Econometrica* 47 (2): 263–91. https://doi.org/10.2307/1914185.

Kairouz, Peter, H. Brendan McMahan, Brendan Avent, et al. 2021. “Advances and Open Problems in Federated Learning.” *Foundations and Trends in Machine Learning* 14 (1–2): 1–210. https://doi.org/10.1561/2200000083.

Kalman, Rudolf E. 1960. “A New Approach to Linear Filtering and Prediction Problems.” *Journal of Basic Engineering* 82 (1): 35–45. https://doi.org/10.1115/1.3662552.

Kant, Immanuel. 1785. *Groundwork of the Metaphysics of Morals*. Johann Friedrich Hartknoch.

Kantorovich, L. V. 1960. “Mathematical Methods of Organizing and Planning Production.” *Management Science* 6 (4): 366–422. https://doi.org/10.1287/mnsc.6.4.366. Originally published in Russian in 1939.

Kaplan, Jared, Sam McCandlish, Tom Henighan, et al. 2020. “Scaling Laws for Neural Language Models.” arXiv:2001.08361. https://arxiv.org/abs/2001.08361.

Karmarkar, Narendra. 1984. “A New Polynomial-Time Algorithm for Linear Programming.” *Combinatorica* 4 (4): 373–95. https://doi.org/10.1007/BF02579150.

Kennedy, James, and Russell Eberhart. 1995. “Particle Swarm Optimization.” In *Proceedings of ICNN’95 - International Conference on Neural Networks*, 4:1942–48. https://doi.org/10.1109/ICNN.1995.488968.

Kingma, Diederik P., and Jimmy Ba. 2015. “Adam: A Method for Stochastic Optimization.” In *International Conference on Learning Representations*. https://arxiv.org/abs/1412.6980.

Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. 1983. “Optimization by Simulated Annealing.” *Science* 220 (4598): 671–80. https://doi.org/10.1126/science.220.4598.671.

Kleinberg, Jon, and Éva Tardos. 2005. *Algorithm Design*. Pearson. ISBN 9780321295354.

Knuth, Donald E. 1997. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. 3rd ed. Addison-Wesley. https://www-cs-faculty.stanford.edu/~knuth/taocp.html. ISBN 9780201896831.

Koopmans, Tjalling C., ed. 1951. *Activity Analysis of Production and Allocation*. John Wiley & Sons. https://cowles.yale.edu/sites/default/files/2022-09/m13-all.pdf.

Kuhn, Harold W., and Albert W. Tucker. 1951. "Nonlinear Programming." In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 481–92. University of California Press. https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Second-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Nonlinear-Programming/bsmsp/1200500249.

Kwakkel, Jan H., Marjolijn Haasnoot, and Warren E. Walker. 2015. "Developing Dynamic Adaptive Policy Pathways: A Computer-Assisted Approach for Developing Adaptive Strategies for a Deeply Uncertain World." *Climatic Change* 132: 373–86. https://doi.org/10.1007/s10584-014-1210-4.

Lawvere, F. William. 1973. "Metric Spaces, Generalized Logic, and Closed Categories." *Rendiconti Del Seminario Matematico e Fisico Di Milano* 43: 135–66. https://www.tac.mta.ca/tac/reprints/articles/1/tr1abs.html.

Lerdahl, Fred, and Ray Jackendoff. 1983. *A Generative Theory of Tonal Music*. MIT Press. https://mitpress.mit.edu/9780262621076/a-generative-theory-of-tonal-music/. ISBN 9780262621076.

Levin, Simon A. 1999. *Fragile Dominion: Complexity and the Commons*. Perseus Books. ISBN 9780738201115.

Little, John D. C. 1961. "A Proof for the Queuing Formula: $L = \lambda W$." *Operations Research* 9 (3): 383–87. https://doi.org/10.1287/opre.9.3.383.

Mac Lane, Saunders. 1998. *Categories for the Working Mathematician*. 2nd ed. Springer. ISBN 9780387984032.

Markowitz, Harry. 1952. "Portfolio Selection." *Journal of Finance* 7 (1): 77–91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x.

Martins, Joaquim R. R. A., and Andrew B. Lambe. 2013. "Multidisciplinary Design Optimization: A Survey of Architectures." *AIAA Journal* 51 (9): 2049–75. https://doi.org/10.2514/1.J051895.

Masanet, Eric, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. 2020. "Recalibrating Global Data Center Energy-Use Estimates." *Science* 367 (6481): 984–86. https://doi.org/10.1126/science.aba3758.

May, Gary S., and Costas J. Spanos. 2006. *Fundamentals of Semiconductor Manufacturing and Process Control*. Wiley-IEEE Press. https://doi.org/10.1002/0471790281. ISBN 9780471784067.

Maynard Smith, John. 1982. *Evolution and the Theory of Games*. Cambridge University Press. https://doi.org/10.1017/CBO9780511806292. ISBN 9780521288842.

McClean, Jarrod R., Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. 2018. "Barren Plateaus in Quantum Neural Network Training Landscapes." *Nature Communications* 9: 4812. https://doi.org/10.1038/s41467-018-07090-4.

McNeil, Barbara J., Stephen G. Pauker, Harold C. Sox, and Amos Tversky. 1982. "On the Elicitation of Preferences for Alternative Therapies." *New England Journal of Medicine* 306 (21): 1259–62. https://doi.org/10.1056/NEJM198205273062103.

Miettinen, Kaisa. 1998. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers. https://doi.org/10.1007/978-1-4615-5563-6. ISBN 9780792382782.

Mill, John Stuart. 1863. *Utilitarianism*. Parker, Son, and Bourn. https://www.gutenberg.org/ebooks/11224.

Mirhoseini, Azalia, Anna Goldie, Mustafa Yazgan, et al. 2021. "A Graph Placement Methodology for Fast Chip Design." *Nature* 594: 207–12. https://doi.org/10.1038/s41586-021-03544-w.

Moretti, Franco. 2005. *Graphs, Maps, Trees: Abstract Models for Literary History*. Verso. ISBN 9781844670260.

Muchnick, Steven S. 1997. *Advanced Compiler Design and Implementation*. Morgan Kaufmann. ISBN 9781558603202.

Muller, Jerry Z. 2018. *The Tyranny of Metrics*. Princeton University Press. https://press.princeton.edu/books/hardcover/9780691174952/the-tyranny-of-metrics. ISBN 9780691174952.

Myerson, Roger B. 1981. "Optimal Auction Design." *Mathematics of Operations Research* 6 (1): 58–73. https://doi.org/10.1287/moor.6.1.58.

Nash, John F. 1950. "Equilibrium Points in n-Person Games." *Proceedings of the National Academy of Sciences* 36 (1): 48–49. https://doi.org/10.1073/pnas.36.1.48.

National Institute of Standards and Technology. 2013. "Response Surface Designs." https://www.itl.nist.gov/div898/handbook/pri/section3/pri336.htm.

Nemhauser, George L., and Laurence A. Wolsey. 1988. *Integer and Combinatorial Optimization*. Wiley. https://www.wiley.com/en-us/Integer+and+Combinatorial+Optimization-p-9780471359432. ISBN 9780471359432.

Neumann, Peter J., Gillian D. Sanders, Louise B. Russell, Joanna E. Siegel, and Theodore G. Ganiats, eds. 2016. *Cost-Effectiveness in Health and Medicine*. 2nd ed. Oxford University Press. https://global.oup.com/academic/product/cost-effectiveness-in-health-and-medicine-9780190492939. ISBN 9780190492939.

Nocedal, Jorge, and Stephen J. Wright. 2006. *Numerical Optimization*. 2nd ed. Springer. https://doi.org/10.1007/978-0-387-40065-5. ISBN 9780387303031.

Norman, Donald A. 2013. *The Design of Everyday Things*. Revised and expanded. Basic Books. https://mitpressbookstore.mit.edu/book/9780465050659. ISBN 9780465050659.

Nussbaum, Martha C. 2011. *Creating Capabilities: The Human Development Approach*. Harvard University Press. https://www.hup.harvard.edu/books/9780674072350. ISBN 9780674072350.

Oppenheim, Alan V., and Ronald W. Schafer. 2010. *Discrete-Time Signal Processing*. 3rd ed. Pearson. ISBN 9780131988422.

Orth, Jeffrey D., Ines Thiele, and Bernhard Ø. Palsson. 2010. "What Is Flux Balance Analysis?" *Nature Biotechnology* 28: 245–48. https://doi.org/10.1038/nbt.1614.

Ostrom, Elinor. 1990. *Governing the Commons*. Cambridge University Press. https://doi.org/10.1017/CBO9780511807763. ISBN 9780521405997.

Papadimitriou, Christos H., and Kenneth Steiglitz. 1982. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall. ISBN 9780131524620.

Parasuraman, Raja, and Victor Riley. 1997. "Humans and Automation: Use, Misuse, Disuse, Abuse." *Human Factors* 39 (2): 230–53. https://doi.org/10.1518/001872097778543886.

Pareto, Vilfredo. 1896. *Cours d'économie Politique*. F. Rouge. https://archive.org/details/coursdconomiep01pare.

Parker, Geoffrey A., and John Maynard Smith. 1990. "Optimality Theory in Evolutionary Biology." *Nature* 348: 27–33. https://doi.org/10.1038/348027a0.

Pauker, Stephen G., and Jerome P. Kassirer. 1980. "The Threshold Approach to Clinical Decision Making." *New England Journal of Medicine* 302 (20): 1109–17. https://doi.org/10.1056/NEJM198005153022003.

Payne, John W., James R. Bettman, and Eric J. Johnson. 1993. *The Adaptive Decision Maker*. Cambridge University Press. https://doi.org/10.1017/CBO9781139173933. ISBN 9780521425261.

Peyré, Gabriel, and Marco Cuturi. 2019. “Computational Optimal Transport with Applications to Data Sciences.” *Foundations and Trends in Machine Learning* 11 (5–6): 355–607. https://doi.org/10.1561/2200000073.

Pontryagin, Lev S., Vladimir G. Boltyanskii, Revaz V. Gamkrelidze, and Evgenii F. Mishchenko. 1962. *The Mathematical Theory of Optimal Processes*. Interscience.

Preskill, John. 2018. “Quantum Computing in the NISQ Era and Beyond.” *Quantum* 2: 79. https://doi.org/10.22331/q-2018-08-06-79.

Propp, Vladimir. 1968. *Morphology of the Folktale*. 2nd ed. University of Texas Press. https://utpress.utexas.edu/9780292783768/morphology-of-the-folktale/. ISBN 9780292783768.

Puterman, Martin L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley. https://doi.org/10.1002/9780470316887. ISBN 9780471727828.

Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis. 2019. “Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations.” *Journal of Computational Physics* 378: 686–707. https://doi.org/10.1016/j.jcp.2018.10.045.

Rajkomar, Alvin, Jeffrey Dean, and Isaac Kohane. 2019. “Machine Learning in Medicine.” *New England Journal of Medicine* 380 (14): 1347–58. https://doi.org/10.1056/NEJMra1814259.

Rasmussen, Jens. 1983. “Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models.” *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13 (3): 257–66. https://doi.org/10.1109/TSMC.1983.6313160.

———. 1997. “Risk Management in a Dynamic Society: A Modelling Problem.” *Safety Science* 27 (2–3): 183–213. https://doi.org/10.1016/S0925-7535(97)00052-0.

Rawlings, James B., David Q. Mayne, and Moritz M. Diehl. 2017. *Model Predictive Control: Theory, Computation, and Design*. 2nd ed. Nob Hill Publishing. https://sites.engineering.ucsb.edu/~jbraw/mpc/.

Rawls, John. 1971. *A Theory of Justice*. Harvard University Press. https://www.hup.harvard.edu/books/9780674000780. ISBN 9780674000780. Original edition cited; identifier and link are for the 1999 revised edition.

Raworth, Kate. 2017. *Doughnut Economics*. Random House. https://www.kateraworth.com/doughnut/. ISBN 9781847941398.

Ristic, Bora, Kaveh Madani, and Zen Makuch. 2015. “The Water Footprint of Data Centers.” *Sustainability* 7 (8): 11260–84. https://doi.org/10.3390/su70811260.

Robbins, Herbert, and Sutton Monro. 1951. “A Stochastic Approximation Method.” *The Annals of Mathematical Statistics* 22 (3): 400–407. https://doi.org/10.1214/aoms/1177729586.

Rockafellar, R. Tyrrell. 1970. *Convex Analysis*. Princeton University Press. https://press.princeton.edu/books/paperback/9780691015866/convex-analysis. ISBN 9780691015866. Original edition cited; identifier and link are for the 1997 first paperback edition.

Rockafellar, R. Tyrrell, and Stanislav Uryasev. 2000. “Optimization of Conditional Value-at-Risk.” *Journal of Risk* 2 (3): 21–41. https://doi.org/10.21314/JOR.2000.038.

Rockström, Johan, Will Steffen, Kevin Noone, et al. 2009. “A Safe Operating Space for Humanity.” *Nature* 461: 472–75. https://doi.org/10.1038/461472a.

Sarter, Nadine B., and David D. Woods. 1995. “How in the World Did We Ever Get into That Mode? Mode Error and Awareness in Supervisory Control.” *Human Factors* 37 (1): 5–19. https://doi.org/10.1518/001872095779049516.

Schön, Donald A. 1983. *The Reflective Practitioner*. Basic Books. https://www.hachettebookgroup.com/titles/donald-a-schon/the-reflective-practitioner/9780465068784/. ISBN 9780465068784.

Schrijver, Alexander. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer. https://link.springer.com/book/9783540443896. ISBN 9783540443896.

Sculley, D., Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. 2015. "Hidden Technical Debt in Machine Learning Systems." In *Advances in Neural Information Processing Systems 28*, 2503–11. https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.

Selbst, Andrew D., danah boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. "Fairness and Abstraction in Sociotechnical Systems." In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 59–68. https://doi.org/10.1145/3287560.3287598.

SEMI. 2022. *SEMI E10: Specification for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM) and Utilization*. SEMI. https://www.semi.org/en/products-services/standards/step/equipment-performance-metrics.

Sen, Amartya. 1970. *Collective Choice and Social Welfare*. Holden-Day. ISBN 0816277656.

Sen, Amartya. 1985. *Commodities and Capabilities*. North-Holland. https://sen.scholars.harvard.edu/publications/commodities-and-capabilities. ISBN 0444877304.

Shahriari, Bobak, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. "Taking the Human Out of the Loop: A Review of Bayesian Optimization." *Proceedings of the IEEE* 104 (1): 148–75. https://doi.org/10.1109/JPROC.2015.2494218.

Shannon, Claude E. 1948. "A Mathematical Theory of Communication." *Bell System Technical Journal* 27 (3–4): 379–423, 623–56. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Shapiro, Alexander, Darinka Dentcheva, and Andrzej Ruszczyński. 2021. *Lectures on Stochastic Programming: Modeling and Theory*. 3rd ed. SIAM. https://doi.org/10.1137/1.9781611976595. ISBN 9781611976588.

Sheffi, Yosef. 1985. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall. https://sheffi.mit.edu/book/urban-transportation-networks. ISBN 9780139397295.

Sigmund, Ole, and Kurt Maute. 2013. "Topology Optimization Approaches: A Comparative Review." *Structural and Multidisciplinary Optimization* 48 (6): 1031–55. https://doi.org/10.1007/s00158-013-0978-6.

Silver, David, Aja Huang, Chris J. Maddison, et al. 2016. "Mastering the Game of Go with Deep Neural Networks and Tree Search." *Nature* 529: 484–89. https://doi.org/10.1038/nature16961.

Simon, Herbert A. 1955. "A Behavioral Model of Rational Choice." *Quarterly Journal of Economics* 69 (1): 99–118. https://doi.org/10.2307/1884852.

———. 1956. "Rational Choice and the Structure of the Environment." *Psychological Review* 63 (2): 129–38. https://doi.org/10.1037/h0042769.

———. 1996. *The Sciences of the Artificial*. 3rd ed. MIT Press. https://mitpress.mit.edu/9780262691918/the-sciences-of-the-artificial/. ISBN 9780262691918.

Sims, Karl. 1991. "Artificial Evolution for Computer Graphics." In *Proceedings of SIGGRAPH '91*, 319–28. https://doi.org/10.1145/122718.122752.

Smith, James E., and Robert L. Winkler. 2006. "The Optimizer's Curse: Skepticism and Postdecision Surprise in Decision Analysis." *Management Science* 52 (3): 311–22. https://doi.org/10.1287/mnsc.1050.0451.

Souppaya, Murugiah, Karen Scarfone, and Donna Dodson. 2022. "Secure Software Development Framework (SSDF) Version 1.1." NIST SP 800-218. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-218.

Spivak, David I. 2014. *Category Theory for the Sciences*. MIT Press. https://mitpress.mit.edu/9780262028134/category-theory-for-the-sciences/. ISBN 9780262028134.

Stahlberg, Felix, and Bill Byrne. 2019. "On NMT Search Errors and Model Errors: Cat Got Your Tongue?" In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 3356–62. https://doi.org/10.18653/v1/D19-1331.

Stephens, David W., and John R. Krebs. 1986. *Foraging Theory*. Princeton University Press. https://press.princeton.edu/books/paperback/9780691084428/foraging-theory. ISBN 9780691084428.

Sutton, Richard S., and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. 2nd ed. MIT Press. https://mitpress.mit.edu/9780262039246/reinforcement-learning/. ISBN 9780262039246.

Tabassi, Elham. 2023. "Artificial Intelligence Risk Management Framework (AI RMF 1.0)." NIST AI 100-1. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-1.

Topol, Eric J. 2019. "High-Performance Medicine: The Convergence of Human and Artificial Intelligence." *Nature Medicine* 25: 44–56. https://doi.org/10.1038/s41591-018-0300-7.

Tymoczko, Dmitri. 2006. "The Geometry of Musical Chords." *Science* 313 (5783): 72–74. https://doi.org/10.1126/science.1126287.

U.S. Food and Drug Administration and Health Canada and Medicines and Healthcare products Regulatory Agency. 2021. "Good Machine Learning Practice for Medical Device Development: Guiding Principles." https://www.fda.gov/medical-devices/software-medical-device-samd/good-machine-learning-practice-medical-device-development-guiding-principles.

Vickers, Andrew J., and Elena B. Elkin. 2006. "Decision Curve Analysis: A Novel Method for Evaluating Prediction Models." *Medical Decision Making* 26 (6): 565–74. https://doi.org/10.1177/0272989X06295361.

Villani, Cédric. 2003. *Topics in Optimal Transportation*. American Mathematical Society. https://doi.org/10.1090/gsm/058. ISBN 9780821833124.

von Neumann, John, and Oskar Morgenstern. 1944. *Theory of Games and Economic Behavior*. Princeton University Press. https://press.princeton.edu/books/paperback/9780691130613/theory-of-games-and-economic-behavior. ISBN 9780691130613. Original edition cited; identifier and link are for the 2007 60th Anniversary Commemorative Edition.

Wardrop, John Glen. 1952. "Some Theoretical Aspects of Road Traffic Research." *Proceedings of the Institution of Civil Engineers* 1 (3): 325–62. https://doi.org/10.1680/ipeds.1952.11259.

Weinstein, Milton C., Joanna E. Siegel, Marthe R. Gold, Mark S. Kamlet, and Louise B. Russell. 1996. "Recommendations of the Panel on Cost-Effectiveness in Health and Medicine." *JAMA* 276 (15): 1253–58. https://doi.org/10.1001/jama.276.15.1253.

Wessel, David, and Matthew Wright. 2001. "Problems and Prospects for Intimate Musical Control of Computers." In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 11–14. https://doi.org/10.5281/zenodo.1176382.

Wiggins, Geraint A. 2006. "A Preliminary Framework for Description, Analysis and Comparison of Creative Systems." *Knowledge-Based Systems* 19 (7): 449–58. https://doi.org/10.1016/j.knosys.2006.04.009.

Winner, Langdon. 1980. "Do Artifacts Have Politics?" *Daedalus* 109 (1): 121–36. https://www.jstor.org/stable/20024652.

Wolpert, David H., and William G. Macready. 1997. "No Free Lunch Theorems for Optimization." *IEEE Transactions on Evolutionary Computation* 1 (1): 67–82. https://doi.org/10.1109/4235.585893.

Woodbury, Robert. 2010. *Elements of Parametric Design*. Routledge. https://www.routledge.com/Elements-of-Parametric-Design/Woodbury/p/book/9780415779876. ISBN 9780415779876.

World Health Organization. 2021. "Ethics and Governance of Artificial Intelligence for Health." World Health Organization. https://www.who.int/publications/i/item/9789240029200. ISBN 9789240029200.

World Health Organization Regional Office for Europe. 2026. “Heat–Health Action Plans: Guidance.” 2nd ed. World Health Organization Regional Office for Europe. https://www.who.int/kazakhstan/publications/i/item/9789289062930. ISBN 9789289062930.

Xenakis, Iannis. 1992. *Formalized Music: Thought and Mathematics in Composition*. Revised. Pendragon Press. ISBN 9781576470794.

Optimization in Software Diagnostics, Observability, Memory Dump Analysis, Trace and Log Analysis, and Debugging

An Evidence-Governed Framework
for Better Diagnostic Decisions

Optimization in Software Diagnostics, Observability, Memory Dump Analysis, Trace and Log Analysis, and Debugging

An Evidence-Governed Framework for Better Diagnostic Decisions

Concept and editorial direction: Dmitry Vostokov, DumpAnalysis.org

Drafting and revision assistance: GPT-5.6 Sol Max

Independent editorial review: Fable 5 Extra

Standalone article | Version 10 | August 2026

Abstract

Optimization in software diagnostics is often reduced to making an analysis faster, collecting more telemetry, or automating more debugger commands. Those are useful improvements, but they are not the full problem. Diagnostic work is a sequential decision process under uncertainty. Engineers choose what to observe, which artifacts to acquire, which questions to ask, which hypotheses to test, when to stop investigating, and which intervention to attempt. Every choice shifts costs and risks across time, computation, storage, privacy, operational overhead, analyst attention, service availability, and the probability of an incorrect conclusion.

This article develops a unified framework for optimization across software diagnostics, observability, memory dump analysis, trace and log analysis, and debugging. It treats optimization as disciplined comparison among admissible diagnostic actions rather than as unconditional maximization. The framework combines multiobjective formulation, value of information, pattern-oriented analysis, preservation claims, controlled intervention, and post-change monitoring. It is presented in analysis-pattern form through Problem, Context, Forces, Solution, Resulting Context, Known Uses, and Related Patterns. It distinguishes optimization of the software system from optimization of the diagnostic process and shows why the two loops must remain coupled. A worked production-latency example demonstrates how targeted observability, artifact selection, hypothesis ranking, and reversible debugging can reduce total diagnostic loss without sacrificing

evidential quality. The resulting approach is evidence-governed: each recommendation carries its assumptions, provenance, uncertainty, authority, stopping rule, and revision trigger.

Keywords: software diagnostics; observability; memory dump analysis; trace analysis; log analysis; debugging; optimization; value of information; diagnostic patterns; software narratology; root cause analysis

__Attribution note:__ Dmitry Vostokov approved and assumes responsibility for the final text.

1. Problem

Software incidents create pressure for immediate action. A service is slow, a process crashes, memory consumption grows, a distributed transaction disappears, or a release changes behavior. The natural response is to collect data and search for a cause. Yet a modern system can emit more evidence than a person or team can inspect: metrics, traces, logs, profiles, dumps, events, network captures, configuration histories, source revisions, deployment records, and user reports. More evidence can improve diagnosis, but it can also create delay, cost, noise, privacy exposure, storage pressure, and false confidence.

The central problem is therefore not simply to analyze an artifact. It is to choose and order diagnostic actions so that the expected value of the eventual decision exceeds the total cost and risk of obtaining it. The analyst must answer questions such as these:

- Which observable represents the user-visible failure rather than a convenient internal proxy?
- Which artifact is sufficient for the current question, and which missing evidence could reverse the conclusion?
- Should the next action be another query, a targeted trace, a memory dump, a controlled reproduction, a comparison with a healthy system, or an immediate rollback?
- Which hypothesis deserves scarce analyst and production time?
- When does additional investigation have less value than its cost?
- What properties must a debugging change preserve?
- How will the team know that the intervention solved the intended problem without creating a different one?

Optimizing only analysis speed can reward premature closure. Optimizing only information volume can produce observability overload. Optimizing only runtime performance can damage reliability, security, and maintainability, which ISO/IEC 25010 treats as distinct product-quality characteristics (International Organization for Standardization 2023), as well as debuggability, which the standard approaches only indirectly through analysability. Optimizing only mean time to recovery can encourage risky restarts that erase evidence. Each local objective is legitimate in context, but none is identical to diagnostic success.

The desired result is not always a single root cause. It may be a ranked set of hypotheses, a bounded claim, a safe workaround, a request for a more discriminating artifact, a proof that a suspected mechanism is impossible, or a justified refusal to act on insufficient evidence. The output type must match the evidence.

2. Context

This pattern applies when a software construction or post-construction problem must be understood from incomplete evidence of execution. It covers live systems and postmortem artifacts; local processes and distributed services; functional failures, performance degradation, security anomalies, resource leaks, concurrency defects, and configuration problems.

The disciplines in the title occupy different but composable roles:

- **Software diagnostics** infers structure, behavior, and abnormality from software execution artifacts and related evidence.
- **Observability** designs and operates the mappings from hidden system state to inspectable outputs.
- **Memory dump analysis** reconstructs a dynamic system from one or more state snapshots, recognizing that acquisition may be partial or temporally inconsistent.
- **Trace and log analysis** reconstructs behavior from ordered, partially ordered, or weakly ordered message sequences and treats those sequences as software narratives.
- **Debugging** changes execution, code, configuration, or environment to test an explanation and remove or contain a defect.

These roles form a loop. Observability produces artifacts. Diagnostics turns artifacts into claims. Debugging uses claims to choose interventions. Interventions alter the system and its future artifacts. A change in instrumentation can change timing; a restart can erase state; a retry policy can change traffic; a new log statement can alter contention; a fix can invalidate an old baseline. The diagnostic process is inside the system it studies, not outside it. Treating execution message sequences as software narratives places trace and log analysis within **software narratology**: the application of narratological concepts to software behavior.

Systematic Software Diagnostics provides the methodological context for this loop. Its fundamental move is to treat artifacts as traces, logs, texts, and narratives and to apply reusable analysis patterns to both the evidence and the analysis process (Vostokov 2024). Memory snapshots and message sequences are not competing worlds. A dump is a state-oriented projection of execution; a trace is a sequence-oriented projection; debugger output is itself a log; stack collections can be treated as structured traces. The useful boundary is determined by the diagnostic question, not by the file extension.

Optimization supplies a common language for choosing among these projections. It begins before a solver and continues after a selected action. The formulation must name the decision space, objective or preference relation, constraints, uncertainty, evidence, affected parties, and revision rule. A perfectly executed sequence of debugger commands can still optimize for the wrong question.

3. Forces

Several forces make diagnostic optimization intrinsically multiobjective.

3.1 Information versus overhead

Higher-fidelity telemetry can reveal causal structure, but it consumes CPU, memory, bandwidth, storage, money, and human attention. It can change scheduling and timing enough to conceal or create concurrency behavior. The best instrumentation is not the maximum instrumentation; it is the least costly instrumentation that discriminates among decision-relevant hypotheses with an acceptable margin.

3.2 Speed versus evidential strength

Fast triage protects service, but early evidence is usually coarse. Deep analysis produces stronger claims, but it delays action. The rational balance depends on reversibility and consequence. A low-risk feature flag may be changed on provisional evidence. An irreversible data migration or safety-critical patch requires a stronger evidence packet.

3.3 Local visibility versus end-to-end consequence

CPU utilization, allocation rate, cache misses, queue length, and error counts are explanatory measures. Users experience completed transactions, tail latency, correctness, availability, and recovery. Long-tail behavior can dominate user-visible performance even when averages remain acceptable (Dean and Barroso 2013). Optimizing an internal counter can move work to another layer or improve a microbenchmark while worsening the service path.

3.4 Snapshot detail versus temporal explanation

A memory dump can expose object state, stacks, locks, heaps, modules, and corruption at a selected moment. A trace can expose order, duration, correlation, repetition, and absence over time. Neither projection dominates universally. The choice depends on whether the next uncertainty is primarily about state, history, mechanism, or interaction.

3.5 Standardization versus case sensitivity

Checklists and analysis patterns reduce omission, improve reproducibility, and make reports comparable. A rigid checklist can waste effort or hide a novel mechanism. Patterns should constrain and accelerate thought without replacing it. They are reusable morphisms between contexts, not scripts that guarantee identical outcomes.

3.6 Automation versus analyst judgment

Automation can reliably enumerate, filter, correlate, cluster, and repeat. Analysts can reframe the problem, recognize an unfamiliar context, judge data quality, and notice that the objective is wrong. Automation should externalize its selection rules and uncertainty so that judgment can challenge them.

3.7 Recovery versus evidence preservation

A restart, failover, scale-out, or rollback may restore service quickly while destroying the state needed for diagnosis. Conversely, preserving evidence too long can extend harm. Incident policy should pre-authorize evidence capture that fits within recovery deadlines and define cases in which service restoration takes priority.

3.8 Precision versus robustness

A detailed model can fit one incident closely and fail to transfer. A simpler model can be less precise but easier to test across versions, machines, and workloads. Diagnostic claims should state their scale: this process, this build, this workload, this environment, or a broader mechanism.

3.9 Search depth versus stopping

There is always another query, trace, dump, experiment, or code path to inspect. Search must stop when the expected benefit of the next action is lower than its cost, when the current evidence supports a sufficiently safe decision, or when the representation cannot support a legitimate conclusion. This is a satisficing decision under bounded rationality, not a claim that the search has found a global optimum (Simon 1955).

4. A General Model of Diagnostic Optimization

4.1 Diagnostics as a sequential policy

Let H be a set of possible hypotheses about the system, and let h ∈ H denote the hypothesis that is actually true. At time t, the analyst has evidence E_t and may choose an action a from an admissible set A_t. A decision d selects a diagnostic or corrective outcome; a policy π produces such a decision from the available evidence and context. An action may acquire an artifact, run a debugger query, calculate a measure, compare a baseline, reproduce a workload, change instrumentation, apply a workaround, or modify the system. The observation produced by an action changes the evidence state and may also change the system.

The object being optimized is therefore a policy, not a one-time command:

$$\pi^* = \arg\min_{\pi} \mathbb{E}[C_{\mathrm{acq}}(\pi) + C_{\mathrm{analysis}}(\pi) + C_{\mathrm{delay}}(\pi) + C_{\mathrm{perturb}}(\pi) + C_{\mathrm{intervention}}(\pi) + L(d_\pi, h)], \quad \text{subject to } g_k(\pi) \le 0. \qquad (1)$$

The policy maps the current evidence and operational context to the next diagnostic or corrective action. Its expected loss includes at least the acquisition cost, analysis cost, delay, operational perturbation, intervention risk, and the loss associated with an incorrect or late decision. In Equation (1), the loss term evaluates the decision produced by π against the true hypothesis h, and the expectation averages over uncertainty about h and future observations. Hard constraints can represent safety, privacy, legal duties, service limits, access rights, and evidence-preservation requirements. This constrained formulation uses standard optimization concepts of feasibility and admissibility (Boyd and Vandenberghe 2004; Nocedal and Wright 2006) and is consistent with the broader transdisciplinary optimization grammar described by Vostokov (2026).

This formulation makes an important distinction. A faster command is an improvement in one cost component. It is an optimization only if the total decision loss is reduced under the relevant constraints.

4.2 Value of information

The next diagnostic action should be selected for its ability to change a decision, not merely for its ability to produce data. Although EVI is decision-theoretic rather than a direct use of entropy, its treatment of observations against uncertainty has an information-theoretic background (Shannon 1948). Let y denote a possible observation produced by action a. The expected value of information (EVI) of a is the reduction in minimum expected decision loss after the possible values of y are considered:

$$\mathrm{EVI}(a \mid E_t) = \min_d \mathbb{E}_{h \mid E_t}[L(d,h)] - \mathbb{E}_{y \mid a, E_t}\left[\min_d \mathbb{E}_{h \mid E_t, y, a}[L(d,h)]\right]. \qquad (2)$$

An operational selection rule is to choose the action with the largest positive net value:

$$a_t^* = \arg\max_{a \in A_t} \{\mathrm{EVI}(a \mid E_t) - \lambda C(a)\}. \qquad (3)$$

Here C(a) is a deliberately constructed scalar triage estimate of elapsed time, compute, storage, production overhead, privacy exposure, and analyst attention after those components have been translated into a common cost scale. The multiplier λ is the organizational exchange rate that converts this cost into expected decision-loss units; it is a judgment encoded by policy, not a physical constant. Equation (3) is admissible only after the hard constraints g have removed unacceptable actions and only when no cost component carries an asymmetric or catastrophic consequence. Otherwise, the vector representation of Section 4.4 must be preserved, as discussed further in Section 8.4. A small probability of catastrophic data corruption cannot be traded casually against a few minutes of faster diagnosis.

EVI is often estimated qualitatively. The practical question is: which possible result would make us choose a different intervention? If no plausible result would change the decision, the action has little decision value, even if it generates an impressive artifact.

4.3 Stopping rules

Continue investigation while a feasible action has positive expected net value and the decision deadline has not been reached. Stop or switch modes when one of four conditions holds:

- **Decision sufficiency:** the current evidence supports an action whose residual risk is acceptable.
- **Marginal exhaustion:** no available diagnostic action is expected to change the decision enough to justify its cost.
- **Operational dominance:** ongoing harm makes containment or rollback more valuable than further evidence.
- **Representational failure:** available observations cannot distinguish the relevant mechanisms, so the correct output is a new instrumentation requirement or an explicitly bounded conclusion.

Stopping is not the same as certainty. It is a decision about the remaining value of search.

4.4 Multiobjective outputs and admissibility

Diagnostic optimization rarely has one natural scalar objective. A useful formulation preserves a consequence vector until a legitimate decision rule is available:

$$\mathbf{z}(a) = \begin{pmatrix} \text{diagnostic delay, acquisition cost, perturbation,} \\ \text{privacy exposure, decision risk, evidential deficit} \end{pmatrix}. \tag{4}$$

All six components in Equation (4) are oriented for minimization; evidential deficit is the shortfall from the evidential quality required for the decision. The feasible set removes actions that violate non-negotiable boundaries. Among the remaining actions, the team may use lexicographic priorities, thresholds, Pareto comparison, a transparent weighted model, or structured deliberation. For example, an incident policy may place safety and data integrity first, service restoration second, evidence preservation third, and cost fourth. A different incident class may reverse the middle priorities.

Constraints should not be disguised as very large weights without justification. A privacy boundary, a no-restart requirement, or a maximum tracing overhead expresses admissibility and authority, not merely preference.

4.5 A compositional view

Category-theoretic language clarifies what must be preserved when evidence is transformed. Raw execution state, a memory dump, debugger output, a normalized table, a pattern narrative, a hypothesis set, and an intervention recommendation are different objects. Acquisition, decoding, filtering, symbolization, aggregation, comparison, and interpretation are morphisms between them.

The chain is useful only when each morphism declares its preservation claim. A filter may preserve all messages for one correlation identifier while discarding cross-request context. A stack summary may preserve top frames while erasing recursion depth. A quotient trace may preserve message classes while discarding individual values. A scalar score may preserve a ranking under one weight vector while erasing disagreement among objectives.

The entire pipeline is not automatically a functor from world behavior to correct diagnosis. Composition can fail because timestamps are incomparable, symbols are wrong, acquisition is inconsistent, sampling is biased, or semantics changed between versions. Evidence governance consists partly of testing these composition boundaries.

5. Solution: Evidence-Governed Diagnostic Optimization

Apply the following circular process. Each stage produces inspectable artifacts. The main cycle is shown below, together with a representative feedback path in which analysis reopens artifact acquisition; other later stages may likewise reopen earlier ones.

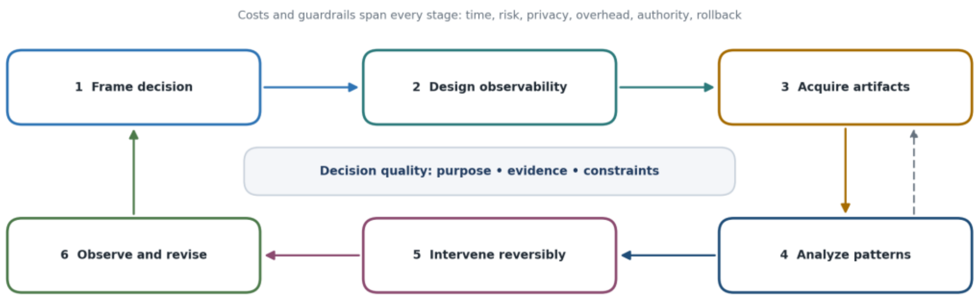


Figure 1. Evidence-governed diagnostic optimization is a control loop, not a waterfall. Observation can reopen framing, analysis can reopen acquisition, and constraints span every stage.

5.1 Frame the diagnostic decision

Begin with the decision that must be made, not with the available tool. State the user-visible or system-level consequence, the time horizon, affected components and people, and the cost of delay. Separate symptoms, observations, interpretations, and proposed causes.

A compact diagnostic charter should record:

- the incident or question and the decision deadline;
- the current baseline and the comparison population;
- the hypotheses that would lead to materially different actions;
- safety, privacy, access, performance, and recovery constraints;
- the authority to collect evidence, intervene, stop, and rollback;
- acceptable output types, including inconclusive or bounded results;
- the monitoring and revision conditions after action.

This charter prevents an available metric from silently becoming the objective. “Reduce CPU” is not a diagnostic purpose. “Restore the checkout journey below its latency objective without increasing failed orders or data risk” is closer to one.

5.2 Optimize observability for discriminating power

Observability should make important hidden distinctions recoverable from outputs. The design question is not “What can we log?” but “Which state distinctions and causal boundaries must an operator reconstruct within the incident budget?”

Use an observability budget across four dimensions:

- **Coverage:** which components, transitions, states, and failure paths are represented?
- **Resolution:** what temporal, spatial, and semantic detail is preserved?
- **Correlation:** can evidence be joined across requests, threads, processes, hosts, and versions?
- **Trust:** are clocks, schemas, symbol files, sampling rules, provenance, and integrity known?

Instrument stable identifiers, boundary crossings, significant state changes, causal parents, resource ownership, and failure results before adding high-volume commentary. Prefer structured fields with declared units and schemas. Preserve the ability to distinguish absence of activity from absence of instrumentation.

Adaptive instrumentation can improve efficiency: begin with low-cost signals, raise detail when triggers fire, and return to baseline after a bounded period. The trigger and decay rules must themselves be observable. Otherwise, the system may collect the richest data after the causal event has passed or remain in an expensive mode indefinitely.

Sampling should be aligned with failure semantics. Uniform sampling can miss rare errors and long tails. Tail-based, error-biased, stratified, or exemplar-linked sampling can preserve decision-relevant events, but each changes the population represented by the data. The analysis must retain the sampling policy as part of evidence provenance.

5.3 Optimize artifact acquisition

Choose artifacts as a portfolio. A small configuration file, deployment diff, or health state can dominate a multi-gigabyte dump when it directly separates the leading hypotheses. The Small DA+TA pattern captures this economy: small dump or trace artifacts may answer the question that a large execution artifact cannot (Vostokov 2023, 276–277).

For each candidate artifact, ask:

- What hypothesis distinction can this artifact resolve?
- What state or history will it omit?
- How much will acquisition perturb the system?
- Can it be collected before state disappears?
- What decoding dependencies are required, such as symbols, schemas, keys, or clocks?
- Can the artifact be reproduced, compared, and shared within policy?

Use staged acquisition when possible. Start with the smallest safe discriminator, then escalate. Preconfigure crash, hang, leak, and latency capture policies so that incident-time choices do not begin from zero. Capture related artifacts within a coordinated time window and record acquisition order; a dump and trace collected minutes apart may describe different systems.

5.4 Optimize the analysis path

Analysis is a search through representations and hypotheses. Pattern languages reduce search cost by naming recurrent transformations and findings. They also make the analysis path inspectable: another analyst can see not only the conclusion but the sequence of abstractions that produced it.

An effective path alternates broad orientation with discriminating tests:

1. Establish artifact validity, provenance, version, time range, completeness, and decoding dependencies.
2. Extract basic facts and construct a healthy or earlier baseline.
3. Identify significant events, state boundaries, discontinuities, dominant activities, and missing expected evidence.
4. Generate competing mechanism hypotheses rather than one story.
5. Select the cheapest high-value discriminator among those hypotheses.
6. Update the ranking, document contradictions, and stop when the decision rule is satisfied.

Avoid ranking hypotheses solely by familiarity. A hypothesis score should separate prior plausibility, evidential fit, consequence if missed, test cost, and reversibility of the associated action. An unlikely but catastrophic mechanism may deserve an early exclusion test.

Higher-order analysis is also optimizable. The sequence of analyst actions is itself a trace. Large time deltas, repeated queries, abandoned branches, and the late discovery of a decisive pattern reveal opportunities for better tooling, indexing, training, and checklists. The goal is not to enforce identical reasoning, but to reduce avoidable search while preserving the ability to reframe.

5.5 Optimize interventions, not only explanations

Debugging converts a diagnostic claim into an experiment or change. Candidate interventions include configuration changes, feature flags, rollbacks, traffic shifts, resource limits, instrumentation changes, code patches, data repair, and architectural redesign.

Select interventions by expected consequence and evidential value. A reversible workaround may both restore service and test a mechanism. A restart may restore service but supply weak causal evidence because it changes many state variables simultaneously. A minimal code change can provide a cleaner test but take too long for the incident horizon.

Every transformation needs a preservation claim. A compiler optimization should preserve chosen program semantics. A refactoring should preserve behavior. A rollback should preserve compatible data and interfaces. A cache change should preserve acceptable freshness. A logging change should preserve timing within a stated tolerance. Tests should target both the intended gain and the preservation claim.

Use canaries, shadow execution, fault injection, replay, and controlled workload comparison where appropriate (Humble and Farley 2010). Keep one mechanism change per experiment when feasible. When several changes must be bundled, state that causal attribution is limited.

5.6 Close the loop with monitoring and revision

The selected intervention changes the evidence-generating system. Monitor the intended outcome, guardrails, distributional effects, workarounds, and recurrence. Compare predicted with observed consequences, not merely pre-change with post-change metrics. This end-to-end view complements delivery-performance measures that connect lead time, reliability, and organizational outcomes (Forsgren, Humble, and Kim 2018).

Preserve a minimum evidence packet:

- the decision and rejected baselines;
- artifact, data, code, configuration, symbol, tool, and model versions;
- objective and constraint definitions with owners;
- hypothesis set, decisive evidence, contradictions, and uncertainty;
- sensitivity, failure, and rollback tests;
- intervention authority and implementation record;
- monitoring thresholds, response owners, and review date.

This packet is operational memory. It allows a later analyst to decide whether the original claim still holds after the software, workload, infrastructure, organization, or objective changes.

6. Domain-Specific Instantiations

The same optimization grammar applies across the five domains, but the variables and preservation claims differ.

Table 1. Diagnostic optimization across the five domains.

Domain	Primary decisions	Optimization focus	Constraints and preservation
Software diagnostics	Artifact requests; hypothesis order; analysis patterns; expert escalation	Decision utility; time to discrimination; reproducibility; bounded uncertainty	Evidence validity; deadline; access; service consequence; authority
Observability	Event placement; schemas; correlation; sampling; retention; alerts	Discriminating power; detection and localization delay; operator workload	Runtime overhead; privacy; storage; telemetry integrity and failure
Memory dump analysis	Dump scope; trigger; capture timing; snapshot sequence; debugger queries	State coverage; consistency; acquisition speed; analysis value	Pause time; size; privacy; symbols; capture semantics; missing pages
Trace and log analysis	Statements; fields; temporal resolution; sampling; filtering; aggregation	Temporal and causal coverage; comparability; signal-to-noise; search cost	Perturbation; clocks; schemas; provenance; cost; false absence
Debugging	Experiment; workaround; patch; rollout; rollback; architectural change	Restoration; mechanism discrimination; low residual risk; supportability	Preservation claim; reversibility; safety; compatibility; decision rights

6.1 Software diagnostics

The diagnostic objective is not simply to name a defect. It is to support the best admissible next decision while explicitly accounting for uncertainty. Variables include artifact requests, analysis patterns, hypothesis order, expert involvement, and report depth. Constraints include incident deadlines, access, evidence quality, service impact, and organizational responsibility.

Optimization improves software diagnostics when it reduces total search cost without prematurely shrinking the hypothesis space. Pattern-oriented diagnostic analysis helps because each pattern encapsulates a recurring diagnostic move in a specific context. A catalog is a search prior; a pattern sequence is a policy; a report records the path and its evidence. The catalog must remain revisable because new artifacts and mechanisms change the feasible analysis space.

6.2 Observability

Observability optimization chooses which hidden structure should be recoverable, at what cost, and with what confidence. Decision variables include event placement, metric definitions, trace context, log schemas, sampling, retention, aggregation, alert thresholds, and escalation rules.

A useful objective is a vector comprising incident-discriminating power, detection delay, localization time, causal coverage, overhead, storage, privacy exposure, operator workload, and resilience to telemetry failure. Tail events and correlated failures matter more than average volume. A telemetry pipeline that disappears during overload is least observable when it is most needed.

Alerts should be optimized as decisions, not message generators. Their purpose is to trigger an action whose expected benefit exceeds interruption cost. Measure precision, recall, delay, severity, actionability, and downstream workload together. Suppression and aggregation should preserve rare critical distinctions.

6.3 Memory dump analysis

Memory dump analysis models a dynamic computer system from a memory slice (Vostokov 2024). Optimization begins with dump policy. Full, kernel, process, heap, thread, or selective state captures differ in size, pause time, privacy exposure, and the hypotheses they can test. The largest dump is not always the best dump.

Acquisition may be temporally inconsistent because memory regions are copied over an interval while the system changes. The evidence packet should therefore record capture mechanism, duration, freeze semantics, missing pages, truncation, compression, and symbol availability. Multiple snapshots can turn a state problem into a short trace and reveal monotonic growth, repeated blocking, ownership transfer, or state transition.

Analysis order should maximize discrimination. Begin with dump integrity, platform and build identity, symbols, process and thread inventory, exception or stop reason, and system-wide context. Then choose specialized paths for crash, hang, leak, corruption, deadlock, or performance hypotheses. The number of debugger commands is a rough measure of analysis complexity, but fewer commands are beneficial only if they preserve decisive evidence. Macro-commands and Elementary Analysis Patterns should compress routine work while retaining the underlying observations.

Snapshot comparison is often more valuable than a deeper inspection of a single snapshot. Healthy versus failing, early versus late, before versus after, or repeated captures under a controlled workload can separate persistent structure from transient coincidence. The comparison must hold relevant conditions constant or explicitly record the differences.

6.4 Trace and log analysis

Trace and log optimization has two coupled problems: constructing the message sequence and navigating it. Message volume, semantic coverage, temporal resolution, correlation, and analyst cost must be traded without collapsing evidence into noise.

Several established patterns are directly interpretable as optimization operators:

- **Focus of Tracing** changes the semantic region of attention rather than merely selecting a time window (Vostokov 2023, 141).
- **Message Essence** searches for the most discriminating invariant form of a message for comparison and case retrieval (195).
- **Minimal Trace** provides a low-activity baseline against which interaction-induced behavior can be compared (206).
- **Measurement** names message-based quantities such as time delta, density, current, field values, and distance (183–184), while the related **Time Delta** pattern focuses specifically on intervals between significant events (298).
- **Relative Density** compares semantically related message types while canceling total trace size (253).

- **Silent Messages** turns absence at a chosen temporal resolution into analyzable evidence (272–273).
- **Small DA+TA** redirects attention from a large execution narrative to a small but decisive artifact (276–277).

For a message type m in a trace of N messages, statement density and current can be written as:

$$\rho_m = \frac{N_m}{N}, \qquad I = \frac{N}{\Delta t}, \qquad r_{(m,n)} = \frac{N_m}{N_n}. \qquad (5)$$

Density helps compare prominence, current separates volume from duration, and the ratio between related message types can reveal imbalance. These measures are diagnostic features, not objectives by themselves. A high error density may reflect a narrow trace filter; a low density may hide a serious event inside broad instrumentation.

The newer Trace Porosity idea extends this view by representing silent positions at a chosen time resolution:

$$\phi_{\text{porosity}} = \frac{N_s}{N_s + N_r}, \qquad (6)$$

where N_s positions are silent, and N_r positions contain recorded messages. The measure is meaningful only with the resolution, boundary rule, and treatment of leading and trailing silence stated explicitly. Its optimization value lies in comparing gap structure across like scenarios, not in maximizing or minimizing porosity universally.

6.5 Debugging

Debugging optimization chooses experiments and transformations. The objective is not the smallest patch or the fastest edit in isolation. It is the smallest admissible intervention that tests the mechanism, restores the intended function, preserves required invariants, and leaves the system supportable.

Debuggers make queries against live or captured state. A query plan can be optimized like a database plan: cheap orientation first, expensive expansion only when it separates hypotheses, cached intermediate results where valid, and explicit stopping. Yet debugging also changes the query target. Breakpoints, watchpoints, tracing, and timing alter execution. The observation model belongs in the claim.

Root cause analysis and repair should remain distinct outputs. One mechanism can admit several repairs; one repair can mask several mechanisms. A workaround can be optimal under the incident horizon and suboptimal as a permanent design. Record the horizon and schedule the transition from containment to correction.

7. Worked Example: Intermittent Tail Latency After a Release

Consider a distributed API whose median latency is stable but whose p99 latency and timeout rate increased after a release. CPU averages remain normal. Operators can restart instances, enable full debug logging, collect profiles, capture dumps, rollback, or alter retry behavior.

7.1 Frame

The purpose is to restore successful request completion within the service objective while preserving order correctness and avoiding retry amplification. The decision deadline is thirty minutes for containment and one business day for a supported causal claim. Tracing overhead must remain below the agreed production budget; payload data must not be captured; rollback must preserve schema compatibility.

The initial hypotheses are:

1. a new lock or queue creates intermittent serialization;
2. database waits increased for one request class;
3. retry behavior amplifies a downstream slowdown;
4. garbage collection pauses grew because of retention;
5. a telemetry or clock defect exaggerates the observed tail.

7.2 Select evidence by decision value

Full debug logging has high volume and introduces perturbation, and it does not guarantee cross-service causality. A targeted trace for slow and failed requests, correlated with deployment version and retry count, directly separates several hypotheses. A short CPU and allocation profile can be used to test compute and retention mechanisms. Coordinated thread snapshots during a latency trigger can expose lock ownership at lower cost than unconditional full dumps.

The first evidence portfolio is therefore targeted tail sampling, retry and queue fields, a bounded profile, and triggered thread snapshots. A heap dump is deferred unless allocation or retention evidence raises its expected decision value sufficiently to justify capture. Rollback remains prepared as the containment action.

7.3 Analyze patterns and competing mechanisms

The Time Delta pattern reveals that the longest delays occur between downstream timeout and local completion. Relative Density shows that retry messages rise faster than successful downstream responses in failing traces. Statement Current increases sharply during the tail window even though completed request current falls. Silent positions appear in one activity strand while retry activity continues elsewhere. Thread snapshots show many requests waiting behind the same bounded executor rather than a process-wide CPU shortage.

Together, these observations favor retry amplification plus queue saturation. They weaken the garbage collection hypothesis and contradict the assumption that normal average CPU implies spare capacity. Database spans for the affected request class remain near their baseline within the same windows, weakening the database-wait hypothesis while not excluding a database-specific interaction under unobserved workloads. Agreement among service-side monotonic clocks, causal trace ordering, and independent thread snapshots retires the telemetry-or-clock-defect hypothesis for the observed incident window. The targeted trace also shows that the new release shortened a retry interval.

7.4 Choose a reversible intervention

Three actions are admissible: roll back the release, restore the previous retry interval through configuration, or scale the executor. Scaling may hide the mechanism and increase downstream pressure. Rollback has broad effect but strong restoration value. Restoring the retry interval is the smallest mechanism-directed intervention and can be applied to a canary.

The team first restores the interval on a small traffic slice while monitoring successful completion, queue depth, downstream request rate, and error distribution. Tail latency and queue depth recover without increasing the number of failed orders. It then expands the change and keeps rollback ready.

7.5 Close and revise

The final claim is bounded: under the observed workload and release, shortened retries amplified a downstream delay and saturated a bounded executor. The evidence does not prove that all future tail latency has the same cause. A database-specific interaction remains a residual alternative outside the observed workload slice and is retained in the decision record. The follow-up work adds retry-budget instrumentation, a saturation alert tied to an action, a load test with injected, correlated downstream delay, with results segmented by database wait class, and a constraint that prevents the retry configuration from exceeding the downstream capacity envelope.

The optimized process did not maximize evidence. It selected a portfolio that changed the decision within the deadline, preserved privacy and service constraints, and produced a mechanism-directed reversible action.

8. Failure Modes and Countermeasures

8.1 Optimizing the proxy

Reducing log volume, debugger commands, alert count, or time to close can improve a metric while degrading diagnosis. When a measure becomes a target, search pressure finds the gap between the proxy and purpose (Goodhart 1975). Pair speed and volume measures with decision accuracy, reopen rate, incident recurrence, guardrail violations, and qualitative review.

8.2 Over-instrumentation

More instrumentation can create contention, cost, and analyst overload. It can also increase the number of apparent anomalies through multiple testing. Unused or weakly governed instrumentation can accumulate as a form of hidden technical debt (Sculley et al. 2015). Require every high-cost signal to support a named decision or hypothesis class, and review unused telemetry for removal or redesign.

8.3 Under-instrumentation and false absence

Missing evidence can mean no event, no coverage, sampling, truncation, a failed emitter, or a broken pipeline. Preserve instrumentation health and coverage metadata. Treat absence as a typed observation, not a universal negative.

8.4 Premature scalarization

A single incident score or hypothesis score can hide asymmetric risk and incomparable objectives. Preserve the vector long enough to separately expose safety, service, evidence, privacy, and reversibility. Use hard constraints for non-negotiable boundaries.

8.5 Local optimization

A team can reduce its analysis time by transferring work, retaining less evidence, or producing shallow reports that another team must reconstruct. Such a local improvement remains bounded by the untouched remainder of the end-to-end path (Amdahl 1967). Measure end-to-end decision lead time, handoffs, reopenings, and recovery, not local utilization alone. High utilization creates queues in diagnostic organizations just as it does in computing systems.

8.6 Tool-induced certainty

A solver, correlation engine, or debugger extension can produce precise output even from invalid symbols, incomplete data, incomparable clocks, or an incorrect model. Separate tool execution from claim validation. Record versions, assumptions, tolerances, and coverage.

8.7 Automation bias

Ranked hypotheses can become decisions by interface convenience. Show evidence for and against each hypothesis, uncertainty, missing discriminators, and the consequences of error. Permit “incomparable” and “need more evidence” outputs.

8.8 Destructive debugging

A restart, patch, or data correction can erase the mechanism or worsen the incident. Design capture-before-change rules, canaries, rollback assets, and post-change comparisons. An intervention that cannot be reversed requires stronger evidence and broader authority.

8.9 Search without a stopping rule

Open-ended investigation consumes scarce time and can become narrative overfitting. Define sufficiency, escalation, and refusal conditions in advance. Preserve unresolved alternatives rather than forcing a single cause.

8.10 Optimization theater

Formal scores and dashboards can legitimate a predetermined action. Warning signs include undocumented weights, no baseline, no contradictory evidence, no sensitivity analysis, unexplained exclusions, and no accountable owner. A small transparent model is often better than an unchallengeable one.

9. Resulting Context

After applying evidence-governed diagnostic optimization, the team has more than a technical conclusion. It has a typed decision record linking purpose, artifacts, transformations, claims, intervention, and revision.

Expected benefits include:

- faster discrimination among hypotheses without indiscriminate data growth;
- clearer trade-offs between recovery, evidence, risk, privacy, and cost;
- reproducible analysis paths expressed through patterns and provenance;
- safer debugging through explicit preservation and rollback claims;
- observability designed around decisions rather than dashboard accumulation;
- organizational learning from the trace of diagnostic work itself;
- bounded conclusions that remain useful when certainty is unavailable.

The approach also introduces costs. Teams must maintain diagnostic charters, evidence metadata, pattern catalogs, instrumentation health, and review rules. Value-of-information judgments can be wrong. Formal language can create false sophistication if it is not connected to actual decisions. These costs are justified when they replace hidden, repeated, and unreviewable choices already being made during incidents.

The optimized state is not a final optimum. Software, workloads, tools, teams, and values change. The result is a controlled loop with memory and a legitimate way to reopen itself.

10. Known Uses

Representative uses include:

- **Crash triage:** choosing dump type, symbol preparation, and early command sequence according to crash class and consequence.

- **Hang and deadlock analysis:** coordinating thread snapshots, wait-chain evidence, and time-bounded live inspection before restart.
- **Memory leak diagnosis:** scheduling repeated snapshots, comparing growth regions, and escalating to heap detail only after coarse evidence localizes the mechanism.
- **Performance regression:** profiling representative end-to-end workloads, protecting tails and failure cases, and guarding against benchmark selection bias.
- **Distributed incident response:** targeting traces by slow or failed outcomes, correlating retries and queues, and preserving sampling provenance.
- **Observability design:** allocating telemetry budgets across coverage, resolution, correlation, and trust rather than maximizing raw event count.
- **Alert engineering:** selecting thresholds and aggregation rules by action value, interruption cost, and residual risk.
- **Release debugging:** using canaries, feature flags, rollback, and versioned evidence to create reversible experiments.
- **Diagnostic automation:** turning repeated debugger and log-analysis operations into inspectable pattern sequences while retaining analyst override.
- **Post-incident learning:** treating the analysis narrative as data for improving tools, acquisition policies, training, and organizational interfaces.

11. Related Patterns and Concepts

Within pattern-oriented software diagnostics, this article is related to:

- A.C.P. Root Cause Analysis Methodology (Artifacts, Checklists, and Patterns);
- Pattern-Oriented Diagnostic Analysis Process;
- Software Diagnostics Canvas;
- Rapid Software Diagnostics Process;
- Right First Time Software Diagnosis;
- Elementary Analysis Patterns;
- Focus of Tracing;
- Measurement;
- Message Essence;
- Minimal Trace;

- Missing Component, Missing Data, and Missing Message;
- Relative Density;
- Silent Messages and Trace Porosity;
- Small DA+TA;
- Statement Density and Current;
- Systematic Software Diagnostics;
- Time Delta.

Related optimization concepts include multiobjective optimization, constrained optimization, robust decision-making, sequential experimental design, value of information, optimal stopping, queueing, active learning, causal inference, control, and reversible design.

12. Conclusion

Optimization in software diagnostics is not about pursuing the largest trace, the smallest dump, the fewest commands, the fastest closure, or the most automated workflow. It is disciplined comparison among diagnostic and corrective actions under explicit objectives, constraints, uncertainty, and authority.

The unifying object is a sequential evidence policy. Observability determines which distinctions can be recovered. Artifact acquisition selects projections of state and history. Pattern-oriented analysis transforms those projections into competing claims. Debugging turns claims into controlled interventions. Monitoring assesses whether the intervention delivered its intended gain while preserving the required invariants. Governance decides when the formulation, not merely a parameter, must change.

The practical test is simple: can another competent person reconstruct why this evidence was collected, why this hypothesis was preferred, why this action was authorized, what it was expected to preserve, and what would cause the decision to be revised? When the answer is yes, optimization has improved not only the software system but also the diagnostic system that keeps it understandable.

References

Amdahl, Gene M. 1967. “Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities.” In *Proceedings of the AFIPS Spring Joint Computer Conference*, 483–485. https://doi.org/10.1145/1465482.1465560.

Boyd, Stephen, and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press. https://doi.org/10.1017/CBO9780511804441.

Dean, Jeffrey, and Luiz André Barroso. 2013. “The Tail at Scale.” *Communications of the ACM* 56 (2): 74–80. https://doi.org/10.1145/2408776.2408794.

Forsgren, Nicole, Jez Humble, and Gene Kim. 2018. *Accelerate: The Science of Lean Software and DevOps*. IT Revolution Press.

Goodhart, Charles A. E. 1975. “Problems of Monetary Management: The U.K. Experience.” In *Papers in Monetary Economics*. Reserve Bank of Australia.

Humble, Jez, and David Farley. 2010. *Continuous Delivery*. Addison-Wesley.

International Organization for Standardization. 2023. *ISO/IEC 25010:2023 Systems and Software Engineering — Systems and Software Quality Requirements and Evaluation (SQuaRE) — Product Quality Model*. https://www.iso.org/standard/78176.html.

Nocedal, Jorge, and Stephen J. Wright. 2006. *Numerical Optimization*. 2nd ed. Springer. https://doi.org/10.1007/978-0-387-40065-5.

Sculley, D., Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. 2015. “Hidden Technical Debt in Machine Learning Systems.” In *Advances in Neural Information Processing Systems 28*, 2503–2511.

Shannon, Claude E. 1948. “A Mathematical Theory of Communication.” *Bell System Technical Journal* 27 (3–4): 379–423, 623–656. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Simon, Herbert A. 1955. “A Behavioral Model of Rational Choice.” *Quarterly Journal of Economics* 69 (1): 99–118. https://doi.org/10.2307/1884852.

Vostokov, Dmitry. 2023. *Trace, Log, Text, Narrative, Data: An Analysis Pattern Reference for Information Mining, Diagnostics, Anomaly Detection*. 5th ed., rev. 5.04. OpenTask. ISBN 978-1-912636-58-7.

Vostokov, Dmitry. 2024. *Theoretical Software Diagnostics: Collected Articles*. 4th ed., rev. 4.00. OpenTask. ISBN 978-1-912636-91-4.

Vostokov, Dmitry. 2026. *Optimization Across Disciplines: A Transdisciplinary Field Guide to Better Choices, Designs, Systems, and Works*. Version 33. OpenTask. ISBN 978-1-919135-00-7.