Democratizing Optimization with Generative AI

David Simchi-Levi* Tinglong Dai† Ishai Menache‡ Michelle Xiao Wu§

*Institute for Data, Systems and Society, Operations Research Center, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, dslevi@mit.edu

†Carey Business School, Johns Hopkins University, Baltimore, Maryland 21202; Data Science and AI Institute, Johns Hopkins University, Baltimore, Maryland 21218, dai@jhu.edu

‡Machine Learning and Optimization, Microsoft Research, Redmond, Washington 98052, ishai@microsoft.com

§Purdue University, West Lafayette, Indiana 47907, michelle.xiao.wu@gmail.com

Abstract. Recent breakthroughs in generative artificial intelligence (GenAI) have captured public imagination and interest, while mathematical optimization remains largely underappreciated outside expert circles. In this article, we argue that GenAI can finally bridge the persistent gap between optimization's potent capabilities and its limited real-world uptake. We present the 4I framework—Insight, Interpretability, Interactivity, Improvisation—as a set of design principles for combining GenAI with mathematical optimization. Insight establishes a trusted, up-to-date view of the state; Interpretability explains model logic and trade-offs; Interactivity enables conversational what-if analysis; and Improvisation supports event-driven reoptimization. By making optimization tools more intuitive, explainable, and adaptable, we envision a future where frontline decision-makers are empowered to engage in rigorous decision-making. We discuss how GenAI complements, rather than replaces, optimization: GenAI lowers barriers to modeling and interpretation, while mathematical optimization reliably enforces business goals, rules, and hard constraints. We also address emerging concerns, from hallucinations to the risk of over-reliance, and outline research directions to ensure robust, ethical integration of GenAI and optimization. Ultimately, the GenAI boom gives the optimization community a historic opportunity to expand its impact, making decision-intelligence science more accessible and trustworthy to a wider audience while elevating human capabilities.

Key words: Generative AI, mathematical optimization, operations research, artificial intelligence.

History: This version: October 24, 2025

1. Introduction

Since its origins in wartime operations research and the subsequent development of linear programming, mathematical optimization has served as a foundational analytical tool across sectors such as manufacturing, healthcare, and finance (Encyclopaedia Britannica 2025; Levy 2005). Despite its broad applicability, it has received limited public attention. In contrast, artificial intelligence (AI), a field with similarly long-standing roots (Bringsjord and Govindarajulu 2024), has recently risen to prominence, particularly through the emergence of large language models (LLMs), attracting widespread interest from both the public and policymakers. A simple comparison highlights this

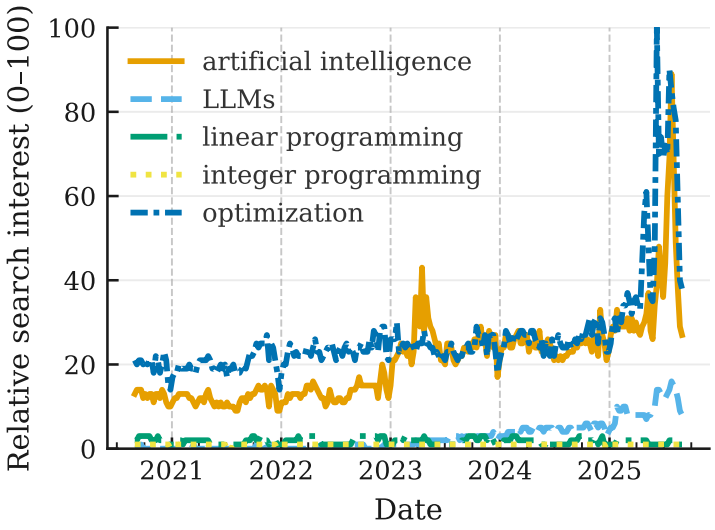

Figure 1 **Relative Google search interest for selected computational and artificial intelligence terms, August 2020–August 2025. Search queries included "artificial intelligence," "LLMs," "linear programming," "integer programming," and "optimization." Data were obtained from Google Trends for the United States and are presented as normalized values (0–100), where 100 indicates the maximum observed search interest during the study period.**

divergence: Google Trends data from the past five years show that terms such as "artificial intelligence" and "LLMs" have surged in their public interest, while precise optimization terms such as "linear programming" or "integer programming" remain close to baseline levels in relative search frequency (see Figure 1). The broader term "optimization" has attracted substantial attention, in a pattern roughly mirroring "artificial intelligence," but most often in a loose sense—applied to topics such as "search-engine optimization" or "productivity optimization"—rather than to the rigorous science of decision-making as understood within the OR/MS community.[1] Put differently, optimization's influence is pervasive, but its scientific foundations remain paradoxically hidden in plain sight.[2]

This disconnect is not for lack of technical success; the optimization community has made substantial progress in both theory and real-world applications. It simply falls short of sparking a broad managerial revolution in the way that recent AI advances have. In the 1990s, there were high hopes of democratizing optimization techniques as spreadsheets and solver add-ins brought

[1] Some professionals consider operations research (OR) to be a subfield of AI. Major AI conferences (e.g., AAAI, NeurIPS, and ICAPS) often have sessions on optimization and planning. We emphasize synthesis over separation, echoing Herbert Simon's view that "instead of differentiating between OR and AI, we need to confuse, blend, and synthesize them as much as possible" (Simon 1987, p. 11).

[2] In the rest of the article, unless otherwise specified, we use "optimization" to refer to "mathematical optimization" for conciseness.

linear programming to the desktop (Fylstra et al. 1998). These tools helped, but they did not fundamentally change how most managers think and work. The majority of business leaders continue to reason in narratives and trade-offs rather than mathematical models. Translating a messy, ill-structured problem into decision variables and constraints remained a high bar, so many would-be users fell back on intuition and heuristics. As Simon (1987) presciently observes, top executives deal in complex, knowledge-rich problems expressed in natural language, whereas optimization traditionally demands structured, quantitative formulations. He argued that to increase this highly technical field's visibility and influence, we must "incorporate the AI kit of tools" to tackle those ill-structured decision domains (Simon 1987, p. 8). In other words, optimization needs interfaces and intelligence that speak the language of its users.

Now, for the first time, generative AI (GenAI) offers a chance to truly change the game. The advent of LLMs means that anyone can describe their decision problem in English (or whatever language they speak) and receive back a first-cut optimization model, a status summary of current operations, minutes-fast what-if analyses, and quick model revisions when reality shifts. This modality shift—from coding algebra to conversing with an AI assistant—directly targets the longstanding barriers that have limited optimization's widespread adoption. We categorize these barriers as the four I's: the difficulty to *Inform* (establish a trusted view of "what's going on"), to *Interpret* (understand and reason about optimization model inputs and outputs), to *Interact* (explore scenarios and sensitivities), and to *Improvise* (adapt models rapidly when conditions change). GenAI has the potential to fundamentally reduce friction on all four fronts. Our perspective in this article is a *complementary* one: GenAI plus optimization, not one replacing the other[3]. LLMs serve as a bridge between people and mathematics, removing usability hurdles, while optimization engines supply the logical structure and optimality guarantees that a free-form AI on its own cannot provide.

Throughout our article, we use "GenAI" to denote generative models that produce samples conditioned on inputs. LLMs are GenAI models that focus on text. Multimodal AI combines different modalities, for example, text with code or images. We consider two complementary pathways: (i) LLM-assisted optimization interfaces that translate natural language to model checks, scenario edits, and solver runs; (ii) GenAI for Optimization that learns policies, heuristics, or model components. Our emphasis in this paper is on (i), as we already see viable impact in enterprise settings. We discuss how (ii) unlocks longer-term opportunities and outline where advances in verification will be required to support them.

The remainder of this article is organized as follows. We first examine the persistent gap between optimization's power and its limited practice, tracing the root causes to the four I's. We then

[3] Recent studies examine whether LLMs can be used to solve combinatorial optimization problem; results suggest that their reach is currently limited (see, e.g., Balachandran et al. 2025).

introduce the 4I framework—Insight, Interpretability, Interactivity, Improvisation—and show how GenAI can operationalize these principles to democratize optimization. We describe a real-world implementation in Microsoft cloud supply chain, where planners apply a GenAI-fused optimization technology to generate effective fulfillment plans. Then, we address emerging concerns such as whether ease-of-use might undermine rigor, how to ensure reliability and ethical use, and what this means for the future role of optimization experts. We conclude with a look at the road ahead: opportunities for research and steps the optimization community can take to harness the GenAI revolution.

2. The Persistent Gap Between Optimization and Practice

Why, after decades of advancements, has optimization not been embraced as universally as its merits warrant? This question has vexed optimization scholars and practitioners alike. The conversations we have had with diverse stakeholders led to a shared observation: while AI startups are attracting massive investment and hype, technologies marketed explicitly as "optimization" remain somewhat niche. In practice, companies that use optimization engines at the core of decision-making are the exception rather than the rule. Even within data-driven firms, it is common to find that planners default to spreadsheets or heuristic rules of thumb or—at an increasing rate—off-the-shelf GenAI tools such as ChatGPT; resorting to formal optimization models only for occasional large projects or not at all. There is a persistent gap between optimization's proven ability to find optimal solutions and its patchy adoption in everyday business planning.

We identify several longstanding factors behind this gap, distilled from both the literature and hard-won experience in industry projects.

First, organizations find it *hard to inform* optimization models with an up-to-date, trusted view of their business state. Before any decision tool can be applied, managers want to know, "What's the current situation? What did we do last week? Where are all the parts and orders right now?" Optimization projects often struggle here: data is siloed across systems, extract–transform–load (ETL) processes run in batch, and by the time a model is solved, the inputs may already be outdated. Most optimization models lack a real-time "state now" dashboard. Without a concise and credible summary of the relevant facts, decision makers hesitate to trust or even initiate an optimization effort.

Second, optimization is *hard to interpret* for the uninitiated. To many decision-makers, even a fully specified optimization tool feels like a black box. The model might spit out a recommendation to, say, source component B from supplier A instead of the usual supplier C, but if the advice contradicts a manager's intuition or experience, they will be reluctant to act on it without a clear explanation. Optimization experts understand concepts like binding constraints, dual prices, and

trade-off curves, but business users often do not. In the absence of transparent rationale, the safer choice for a manager is to stick with their gut or the status quo. This lack of interpretability and reasoning traceability in traditional optimization deployments has been a major adoption barrier (Kobbacy et al. 2007).

Third, most optimization tools are *hard to interact* with in the way managers need during planning cycles. Real-world planning is inherently iterative and scenario-based, but many optimization models are deployed as one-shot engines: one inputs a single dataset (often a deterministic forecast) and receives a single "optimal plan." In theory, the field has developed stochastic and robust optimization techniques for uncertainty, but in reality, firms rarely implement these sophisticated methods. Instead, planners cope with uncertainty by asking lots of "what if" questions. They need rapid answers to questions such as "If demand surges by 15%, or if the new supplier turns out to be unreliable, or if fuel prices double, how should our plan change?" Manually constructing and solving all these scenarios can take weeks and requires specialized analysts, meaning it happens infrequently if at all. The net effect is that optimization models, as traditionally used, often yield plans that are brittle—optimized for one narrow set of assumptions—rather than robust strategies that have been pressure-tested against multiple futures. This rigidity undermines trust: a manager will discount an optimal plan if they suspect it falls apart under slight changes in input assumptions.

Fourth, optimization tools are *hard to improvise* or adapt on the fly when business conditions change. In today's fast-paced environment, change is the norm—be it a sudden supply disruption, a new tariff policy, or shifting market demand. Yet, updating an optimization model to reflect a change (new constraints, parameters, or business rules) is typically a slow, manual process. One often must submit an IT ticket or ask a modeling expert to recode part of the model, then wait for testing and redeployment. By the time the revised model is ready, the window for decision may have passed. This lack of agility means that many firms forgo using optimization in rapidly evolving situations; they revert to simpler rule-based decisions that they can tweak in real time. In short, optimization models have traditionally been too static and too high-maintenance to keep up with the pace of business change.

These four challenges—inform, interpret, interact, and improvise—are not abstract theory but practical realities observed repeatedly in optimization applications. They help explain why optimization, despite its power, has not achieved the ubiquity of, say, spreadsheet analysis or machine learning prediction tools. Optimization methods themselves are effective; what has been missing is the supporting systems to make them accessible, transparent, and adaptable. Recognizing these pain points clarifies what a robust framework must deliver: sustainable data integration, interpretable model-grounded explanations, interactive scenario exploration with trustworthy feedback, and rapid adaptation as conditions change.

Related research already integrates AI and optimization in complementary ways. For example, OMLT operationalizes ML–OR couplings by embedding trained predictors as optimization constraints within Pyomo (Ceccon et al. 2022). Another body of work explores learning decision policies for hard combinatorial problems, such as reinforcement-learning approaches to job-shop or cluster scheduling (Zhang and Dietterich 1995; Mao et al. 2016). Our focus differs: a language-first interface that maps natural-language intent to solver-checked edits, explanations, and rapid reoptimization. In the next section, we propose a unifying set of design principles (the "4I" framework) that captures these requirements and argue that GenAI provides the catalyst needed to realize them in practice.

3. The 4I Framework: Insight, Interpretability, Interactivity, Improvisation

To bridge the gap between optimization's potential and its practical impact, we propose a 4I framework of design principles. Each "I" corresponds to a critical dimension in which traditional optimization workflows must improve to meet real-world decision-making needs: *Insight*, *Interpretability*, *Interactivity*, and *Improvisation*. These principles encapsulate the user-centric capabilities that an ideal decision-support system should have. They also directly align with the barriers discussed above, providing a structured agenda for modernization. We outline each element of the 4I framework below:

3.1. Insight: Inform the State

Before any optimization can happen, the user needs insight into the current state of the system. This principle is about providing an up-to-date, correct, and readily understandable summary of what's going on." In a traditional workflow, assembling this picture can be slow and siloed: data must be pulled from multiple sources, cleaned via ETL processes, and manually compiled into reports. By the time a planner sees the numbers, they may be out of date. Under the Insight principle, we envision a live integration of data streams into a coherent ops knowledge graph or digital twin, continually syncing with real-world status. Every time a decision-maker sits down to use the optimization tool, they should be greeted with a dynamic, conversational briefing; for example, "Inventory is 95% of where it was last week, backlog has grown by 10%, and there's an anomaly with supplier A's last shipment (it's delayed)." Achieving this kind of situational awareness means the optimization model always starts from a baseline of truth that the user trusts. Key metrics for Insight include data latency (how fresh is the data feeding the model), completeness (are we capturing all relevant state variables), uniqueness (is each entity represented only once in the dataset), and consistency (are there any conflicting data points that need reconciliation). Insight addresses the "hard to inform" barrier by making the state of the world transparent and readily available for decision modeling.

3.2. Interpretability: Understand the Logic

Even the best optimization result is of little value if the decision-maker cannot understand or justify it. Interpretability means the ability to explain model outputs, the reasoning behind recommendations, and the structure of the model itself in intuitive, business-friendly terms. Rather than treating the optimization model as an oracle that pronounces a solution, the system should engage in a dialog about *why* that solution makes sense (or why it is optimal). Concretely, the process could involve automatically generated explanations of which constraints were binding and how they drove the solution, or natural language justifications such as "Shipping from Warehouse Q to Region X is recommended because it balances lower transport cost against slightly longer delivery time, satisfying the service target at minimal total cost."

Interpretability also extends to surfacing the model's assumptions and inputs: for example, highlighting that the solution assumes a certain demand forecast or supplier capacity. Techniques like model-to-text generation, constraint provenance tracing, and sensitivity analysis visualizations all fall under this principle. For a diagnostics-oriented example, a planner might ask, "Why is the cheapest supplier A not used for demand D?" and the system would point to feasibility or compatibility constraints, binding capacities, or other trade-offs that preclude that assignment. For example, choosing the cheapest supplier results in long lead times, forcing higher inventory and supply chain costs. The success metrics might include an "explanation fidelity score" (how well the explanation matches the true logic of the model) or simply the percentage of decisions made with an accompanying rationale. Interpretability targets the "hard to interpret" barrier, turning the black box into a glass box and building user trust in the optimizer.

More broadly, we view interpretability as a bidirectional requirement: humans should be able to interpret machine outcomes, and machines should be able to capture human intent. For instance, machines could understand stakeholders' requirements when optimizing certain processes. Until recently, these requirements were captured through "standard" information exchanges (e.g., meetings, specification documents, and Excel spreadsheets), with data scientists and engineers responsible for translating them into code.

3.3. Interactivity: Explore What-ifs

Real decision-making is a conversation, not a one-shot query. The Interactivity principle is about enabling rapid, flexible exploration of scenarios and alternatives. Instead of a single static formulation tied to one dataset, the system should let users tweak assumptions on the fly, pose what-if questions, and compare outcomes across scenarios. Ideally, a planner could say to the system: "Show me what happens if demand is 15% higher and also if our cheapest supplier goes offline," and the system would quickly respond with results for those scenarios, perhaps side-by-side with the baseline plan. Achieving this result requires optimizers that can solve (or approximate) many variants

quickly, possibly through a combination of precomputation, warm-start heuristics, or on-the-fly model simplifications. It also benefits from interfaces that let users modify model parameters or constraints in plain language rather than editing code.

Interactivity addresses the reality that deterministic optimization on point forecasts leaves decision-makers uneasy; they much prefer to see that a plan holds up under a spread of futures. By making scenario analysis conversational and fast, we make it routine rather than a rare specialist exercise. Performance metrics may include the number of scenarios a user can iterate through per hour, or the time-to-first-feasible-solution for a what-if query (since an approximate answer delivered quickly is often more valuable than a perfect answer that arrives too late) or the ability to compare and contrast various scenarios side-by-side so decision makers understand the sensitivity of their plan to changes in the business environment. The above requirements directly address the "hard to interact" barrier, aiming to produce robust decisions through rapid scenario-based iteration, essentially bringing the spirit of robust/stochastic optimization to users without requiring them to know advanced math.

3.4. Improvisation: Iterate and Adapt

Finally, Improvisation refers to the system's ability to continuously adapt the model and solution as conditions change. Business environments are dynamic, and decision support tools must adapt (or *improvise*) in step with real-world changes. This principle goes beyond just scenario analysis (which can be viewed as offline exploration) to encompass real-time model updates and reoptimization. For instance, if a sudden disruption occurs—say a key supplier's plant is shut down—the system should detect this event (perhaps through monitoring data streams or news), alert the user, and assist in modifying the optimization model (e.g., automatically remove that supplier from the network, adjust capacities, and re-solve) within minutes or hours, not weeks. Improvisation thus implies a closed-loop workflow: monitor $\rightarrow$ detect change $\rightarrow$ adjust model $\rightarrow$ re-optimize $\rightarrow$ deploy new solution, all with minimal human friction. This may involve auxiliary processes (or 'agents') that can propose edits to constraints or parameters ("lead time for component Y has increased from 2 days to 5 days; shall I update the model accordingly?") and even automatically validate the new model in a sandbox before recommending the updated plan. It also implies keeping a change log and ensuring traceability, since frequent model tweaks must be governed responsibly. Key metrics could include time-to-reoptimize after a change, the percentage of changes that can be safely auto-handled without expert intervention, and the system's ability to detect drift or anomalies that warrant model revisions. These requirements address the "hard to improvise" barrier by making the optimization framework itself flexible and alive, rather than a static artifact that becomes obsolete when reality deviates from initial assumptions.

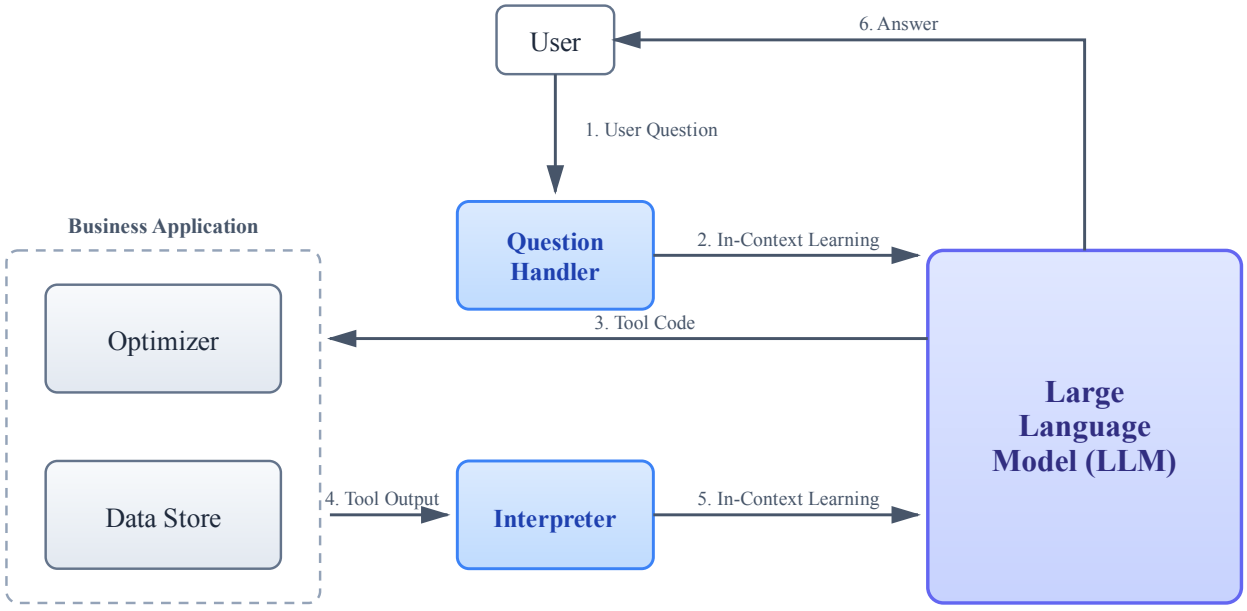

Figure 2 **System architecture for LLM-augmented optimization. The question handler routes a user query to the LLM for context (2), emits tool code (3), brings back tool output (4), updates the prompt (5), and returns an answer (6); (1) denotes the incoming question. The dashed block marks the business application, within which the optimizer and data remain secure.**

In summary, the 4I framework sets the stage for a user-centered, dynamic approach to optimization. Insight ensures the model starts with the right data and context, Interpretability ensures the results are understood and trusted, Interactivity ensures the decision space is fully explored, and Improvisation ensures the system stays relevant over time. Achieving all four in concert might sound ambitious, but we contend that the emergence of modern AI—especially GenAI—is precisely what makes it feasible. Next, we turn to how GenAI can be harnessed to fulfill these principles, effectively acting as the "glue" or interface that connects human decision-makers to optimization models in a far more accessible and powerful way than before.

4. GenAI as a Bridge Between Users and Optimization Models

GenAI, and LLMs in particular, are poised to be the enabling technology that bridges the gap between users and optimization models along each of the 4I dimensions. One key observation is that LLMs are extremely adept at understanding and generating natural language, which is the medium most managers think and communicate in. By placing an LLM layer atop our optimization tools, we can translate back and forth between the world of human context and the world of mathematical optimization.

Figure 2 illustrates this bridge in practice. It shows how a large language model sits between users and optimization services, translating natural language requests into structured solver queries

and returning results in conversational form. The architecture captures the emerging pattern for future systems: a secure enterprise enclave coupled with a stateless LLM interface that mediates human intent, data access, and decision execution. Specifically, in this system architecture, the data and optimization model remain within the firm's four walls, while the LLM interface can run in the public cloud. This design protects the firm's confidential information and provides a concrete blueprint to operationalize the 4I framework, making optimization easier to inform, interpret, interact with, and improvise.

We treat the 4I principles as testable design requirements. For each I, we specify a minimum viable capability and a measurable property, as summarized in Table 1. Recent efforts already implement pieces of the 4I design: natural-language to modeling and solver codes (e.g., OR-tuned language models and NL4Opt-style pipelines (Huang et al. 2025; Chen et al. 2025)); solver-grounded infeasibility and error diagnosis that reason about constraints in plain language (Chen et al. 2024); agentic loops that connect retrieval, data services, solvers, and visualization in production settings (Li et al. 2023; Menache et al. 2025; Simchi-Levi et al. 2025); ML models embedded as optimization constraints to couple prediction and prescription (Ceccon et al. 2022); and LLM-guided hyper-heuristics that propose warm starts or repair moves across instances (Ye et al. 2024). These systems map to 4I as follows: Insight (agentic retrieval and state summaries), Interpretability (solver-grounded narratives and diagnostics), Interactivity (conversational scenario edits with warm starts), and Improvisation (safe model edits with quick reoptimization). In the rest of the section, we elaborate on how GenAI can make optimization easier to inform, interpret, interact with, and improvise, thus operationalizing the 4I framework.

Table 1 The 4I principles as design requirements with example metrics.

Principle	Minimum capability	Example metric
Insight	Live, trusted summary of relevant state before any run	Data latency, completeness and uniqueness of required fields, number of unresolved data conflicts
Interpretability	Plain-language rationale tied to model structure	Percentage of decisions with an explanation, explanation fidelity checks against solver artifacts
Interactivity	Conversational what-if and side-by-side comparisons	Time-to-first-result for a scenario, scenarios iterated per hour, scenario coverage, ability to compare and contrast multiple scenarios
Improvisation	Safe model edits and quick re-optimization after change	Time from change detection to re-solve, percentage of edits auto-validated without expert review

4.1. Easier to Inform: Live Knowledge Graphs via AI Agents

GenAI can streamline the task of informing the optimization model with current, trustworthy data. One promising concept is the *GenAI-fused optimization agent* that sits on top of enterprise data sources and continuously curates a live summary of the operational state. Imagine a chatbot-style assistant that can connect to databases, APIs, and even documents or emails, using techniques like retrieval-augmented generation (RAG) to fetch relevant data on demand. When a user asks, "Where do we stand as of today?" the agent can compile an answer across systems: "We have 5,200 units in stock across all warehouses (down 5% from last week), outstanding orders for 1,100 units, and Supplier X has just reported a 2-day delay on shipments." Early versions of such "AI data concierges" are emerging. For example, Xu et al. (2025) describe an agentic digital twin for urban logistics that uses multiple AI agents to retrieve data, interface with simulation models, and update an optimization solver's inputs in real time. The key is that the GenAI agent can reconcile inconsistent data, fill in missing pieces by asking follow-up questions, and highlight anomalies, thereby producing a clean, integrated "state now" dataset for optimization. From the user's perspective, informing the model becomes as simple as having a conversation. No more hunting for the latest CSV export or worrying that the data is stale—the AI layer ensures the optimizer is fed with live, vetted information. This feature directly addresses the "Hard to Inform" barrier: the question of "what is true today?" becomes trivial to answer, which lowers the activation energy to then ask, "given this state, what should we do?" The synergy of an AI agent with an always-synchronized digital twin makes the optimization system feel less like a sporadic tool and more like a living, continuous decision companion.

4.2. Easier to Interpret: Bidirectional Natural Language Interfaces

GenAI can turn optimization into a two-way dialog, eliminating the black-box feeling. On the front end, an LLM-powered interface means a user can *speak* or write their intent and constraints in everyday language, and the system will translate that into a formal optimization model. For instance, a manager might say, "Optimize our inventory replenishment plan; I want no more than 5% stockouts on any product." Traditionally, that request would require optimization expertise to formulate (introducing binary variables for stockout occurrences, etc.). But current LLMs can generate the decision variables (e.g., order quantities), objective (e.g., minimize cost or maximize service level), and constraints (service level $\geq 95\%$, capacity limits, etc.), even writing solver-ready code. This capability is already on the horizon: recent research has introduced "Operations Research Language Models" (ORLMs) trained specifically to parse natural language into optimization formulations (Huang et al. 2025). These ORLMs have shown good performance on benchmarks, in some cases producing models and solver code that rival what human optimization experts or GPT-4 would

produce, despite having only around 7 billion parameters (i.e., relatively small by LLM standards) (Huang et al. 2025). LLMs are also capable of generating examples of optimization instances, which can in turn be used to improve the accuracy of LLMs' formulations through post-training techniques, such as supervised fine-tuning (Li et al. 2024b; Chen et al. 2025; Lu et al. 2025). With such tools, the modeling bottleneck can be tackled: we move closer to having non-optimization experts being able to outline a problem in prose and construct a formal mathematical program automatically. This feature is one half of the bidirectional interface. However, entirely new formulations proposed by an LLM must be verified with solver- and rule-based checks; today we lack automated guarantees for validity, so human-in-the-loop review and auditing remain essential (see Sections 6.1 and 6.2).

The other half is turning mathematical solutions back into plain-language explanations. After solving the model, a GenAI agent can provide a concise rationale; for example, the optimizer suggests, "Sourcing more from Supplier A and less from Supplier B because A has a lower cost and we met B's minimum order already. Warehouse Q is used to full capacity (constraint binding), which is why Region Z demand is served from a farther location at slightly higher transport cost." This kind of explanation might mention that the solution increases profit by 8% compared to current operations and highlight any unusual decisions ("Product X is not produced at plant M despite capacity, due to its higher unit cost"). AI can tailor the explanation to the audience: a high-level summary for executives, a detailed constraint analysis for analysts; for example, "there is no feasible solution, since the following set of constraints cannot be satisfied concurrently" (see, e.g., Chen et al. 2024). In effect, the generative model serves as the voice of the optimization model, able to justify and clarify the math in human terms. This approach greatly improves interpretability. Early signs of feasibility are evident in projects such as NL4Opt, where LLMs are used to explain solutions or even identify modeling errors by reasoning about constraints in natural language (Huang et al. 2025). LLMs can also be used to visualize optimization outcomes. For instance, they can be used to compare an optimized plan with the original plan based on various metrics. We note that ensuring the correctness of AI-generated explanations is an active research challenge; however, fine-tuning LLMs on the domain-specific contexts may reduce the hallucination issue and make explanations more accurate. As these interfaces mature, the optimization black box will gradually be demystified. Users will be able to interrogate the optimizer just as they would ask a human colleague to explain a recommendation. This transparency, powered by GenAI, lowers a key barrier to adoption by building understanding, increasing confidence, and fostering trust in the underlying technology.

4.3. Easier to Interact: Conversational Scenario Exploration and AI-Driven Solving

One of the most exciting contributions of GenAI is the potential to transform scenario analysis from a tedious process into an interactive conversation. With a GenAI optimization assistant, a planner

could simply *describe* a hypothetical scenario in natural language, and the system will handle the rest. For example, a supply chain manager might say, "Imagine demand is 15% higher next quarter, one of our suppliers (Supplier C) goes down for a month, and transportation costs double. How should our production and distribution plan change?" In a GenAI-fused optimization workflow, the AI agent would interpret this request, auto-modify the underlying optimization model (adjusting demand numbers, removing Supplier C's capacity, increasing transport cost coefficients, etc.), and then run the solver on this new scenario. Within a reasonable amount of time, it could return a proposed plan for that scenario, along with key performance metrics (e.g., total cost of operations and service levels) and even a comparison to the baseline scenario. The user can then iterate: "What if we also add overtime as an option at the plants?"—the agent in return updates the constraints to allow overtime, re-solves, and reports the outcome. This kind of back-and-forth makes exploring the space of possibilities more effective.

Under the hood, achieving rapid scenario analysis may require going beyond brute-force optimization solvers. This is where GenAI may assist not only in interfacing but also in assisting with the solving itself. LLMs and other AI techniques may act as *intelligent heuristics* to speed up scenario evaluations. For instance, an LLM may provide a good starting solution (feasible plan) for each scenario by leveraging statistical knowledge of past solutions, thereby warm-starting the solver and reducing solve time. In more complex problems, LLMs could even suggest a decomposition such as, "Given this scenario with Supplier C out, focus on reassigning its allocation among A and B while holding other decisions steady." The emerging topic of Language Model-based Hyper-Heuristics illustrates this potential. A recent study by Ye et al. (2024) introduced a method called Reflective Evolution (ReEvo), where an LLM generates heuristics for combinatorial optimization and then *reflects* on their performance to refine them. This approach yielded state-of-the-art or competitive heuristics across multiple problem types (Ye et al. 2024). In our context, one can imagine an AI agent that not only runs a solver but can also switch to a learned heuristic mode to get quick, feasible solutions for many scenario variants, then use the exact solver to polish a few top candidates. Similarly, GenAI could surface a Pareto frontier of solutions (e.g., cost vs. service level trade-off) for the user to consider, rather than a single point solution, thus embracing the multi-objective nature of real decisions.

By turning scenario planning into a fast, interactive loop, GenAI helps decision-makers achieve the kind of robustness and resilience that optimization has long promised but seldom delivered in practice. Instead of trusting a single optimal plan, the user can identify decisions that work well across a range of scenarios. In effect, a GenAI-based assistant can implement the spirit of robust optimization without the user needing to formally know robust optimization. The planner simply asks their "what if" questions in plain language, and the GenAI-fused optimization system does

the heavy lifting. The net result is a greatly enhanced level of Interactivity: iteration times shrink from weeks to hours or minutes, and exploring alternatives becomes a normal part of using the optimization tool rather than a specialist project. When everyday managers and operators, not just analysts, routinely engage with scenario-based planning through an AI conversational interface, we will have crossed a significant threshold in optimization's accessibility.

4.4. Easier to Improvise: Continuous Reoptimization and Model Evolution

In a GenAI-integrated environment, when the world changes, the optimization model can change with it—quickly and with minimal fuss. This feature may be the most transformative aspect of bringing together AI and optimization: the ability to *improvise* and adapt models on the fly. Consider how things work today: if a new constraint or business rule needs to be added to the model (say, product X cannot be shipped by air), it might take days for a modeler to implement and test that change. In contrast, a GenAI agent functioning as an "optimization co-pilot" could handle the first draft of this update. For instance, a user can simply tell the system, "We have a new policy: no air shipments for product X," and the GenAI agent, understanding this instruction, inserts the appropriate constraint into the formulation. The model is then re-solved with the new constraint, and the assistant presents the adjusted plan, noting the impact of the change ("This policy will increase logistics cost by 2% and delay delivery to region Y by one day for product X"). If the change caused infeasibilities, the assistant would highlight which constraints are now violated and possibly suggest remedies ("To meet demand for X without air, production at the closer plant Z must increase, but its capacity was binding; consider overtime or increasing capacity"). Through such dialogue, the AI can guide the user in improvising a new solution.

Furthermore, GenAI agents can be proactive in model adaptation. By monitoring data in real time, an agent can detect distribution shifts or trends that might warrant model updates. For example, if over several weeks an important commodity cost has consistently fallen by 10%, the agent might alert: "The price of alloy steel has declined by 10% over 8 weeks. Should we update the cost parameters in the model and re-optimize sourcing?" With one click, the user could accept, and the system would refresh the model with the new cost assumption and rerun, essentially doing a mid-cycle optimization update that traditionally almost never happens. In more autonomous setups, such as the agentic digital twin framework (Xu et al. 2025), the agent might even orchestrate the entire loop end-to-end: detect a change (from sensors, forecasts, or external news), retrieve relevant data, adjust the model's input files or formulation, run the solver or simulation, and then present the user with an updated action plan. Crucially, it would also explain what changed and why the new plan is recommended, to keep the human decision-maker in the loop.

This kind of continuous reoptimization enhances the agility of decision-making. It means the optimization model is never more than a few hours out of sync with reality. Businesses can respond

to disruptions or opportunities faster, potentially gaining a competitive edge. It also means that the model itself can evolve over time: as new product lines are introduced or new constraints (like sustainability goals) become relevant, the GenAI layer can help integrate them into the model rather than waiting for a big annual model overhaul by optimization consultants. Complementary to model changes, GenAI can be used for performance tuning of the underlying optimization solvers. For example, a user could say, "I now need faster solutions. I am willing to relax optimality by up to 5% as long as solution time decreases by 25% or more." This prompt can be translated by LLMs into parameter changes in the solver (e.g., optimality gap tolerance and time limits). The Improvisation principle, supercharged by GenAI, thus turns optimization from a static planning tool into a dynamic, always-adapting assistant. This democratizes optimization not only in terms of who can use it, but also when and how often it can be applied—moving toward a world where optimization shifts from an episodic tool to a continuous, adaptive capability.

Taken together, these patterns turn the 4I framework into implementable design. Section 5 demonstrates the application of the framework in a cloud supply chain setting.

5. Real-World Implementation: Microsoft Cloud Supply Chain

Demand for cloud computing has increased over the last few years. In this environment, users rent virtual machines (Hadary et al. 2020) and other compute resources and services and pay depending on their usage. To support the growing demand, cloud providers such as Amazon, Google, Microsoft, and Oracle manage a large network of data centers, each of which includes a large amount of servers and other equipment. This discussion is based on using GenAI for explainability in the context of Microsoft's cloud supply chain (Li et al. 2023, 2024a; Menache et al. 2025; Simchi-Levi et al. 2025). Managing Microsoft's cloud supply chain entails multiple activities, from demand forecasting to planning, sourcing, and execution. Our focus in this case is on execution. The input to the execution phase is Microsoft's internal demand. Demand is exposed through a demand plan, which summarizes important information about demand requests for the next few months (Li et al. 2023; Simchi-Levi et al. 2025). The information includes the Microsoft business group, the type of hardware required, the region where the hardware should be docked, and the ideal dock date.

Supply chain planners need to periodically generate a fulfillment plan that specifies which hardware to ship, from which warehouse (source) to which data center (destination), and where in the data center to dock the new equipment. We model the planner's problem as choosing a fulfillment plan that minimizes total supply chain costs (for example, shipping and hardware depreciation), subject to compatibility, capacity, and inventory constraints, while meeting growing demand for computing resources. A complex optimization problem thus underlies these routine decisions.

5.1. Problem at a Glance

Each week, planners generate a fulfillment plan that selects (i) which hardware to ship, (ii) from which supplier to which data center and row within the data center, and (iii) the shipping method and target arrival window. The plan minimizes total cost subject to capacity, compatibility, and service requirements. In production, the underlying model is a mixed integer linear program (MILP) with inputs on the order of hundreds of megabytes and it is solved periodically on a rolling horizon basis[4]. Planners are not necessarily optimization specialists, which motivates an interface that explains logic and supports quick what-if analysis.

5.2. Core Decisions and Constraints

We next present a partial and simplified formulation that captures the essence of the MILP that is used in production to assign and ship servers from the warehouse to the data centers.

5.2.1. Problem formulation Entities and capacities. Let $\mathcal{L}$ denote the set of data centers, $\mathcal{S}$ the set of suppliers, $\mathcal{D}$ the set of demands, and $\mathcal{T}$ the set of planning days. For a demand $d \in \mathcal{D}$, let $\mathcal{S}(d) \subseteq \mathcal{S}$ denote its compatible suppliers and $\mathcal{L}(d) \subseteq \mathcal{L}$ its compatible data centers. For simplicity, we assume each demand d corresponds to one unit of capacity (e.g., a rack of servers) and occupies exactly one row in a data center. Each data center $\ell \in \mathcal{L}$ has ρ_ℓ rows. We let σ_s denote the available inventory of supplier $s \in \mathcal{S}$, and $\delta_{\ell,t}$ the docking throughput at data center ℓ on day $t \in \mathcal{T}$.

Cost and penalties. We denote by $c_{d,t}$ the cost of deploying demand d on day t, which captures either the per-day idle cost or the delay cost, depending on the ideal dock date. If a demand d is not fulfilled, it incurs a penalty u_d. We further denote by $h_{d,s}$ the cost of shipping from supplier s to fulfill demand d.

Decision variables. Each decision variable $z_{d,\ell,t} \in \{0,1\}$ indicates whether demand d is docked in data center ℓ on day t, and $w_{d,s} \in \{0,1\}$ indicates whether demand d is fulfilled using supplier s.

The optimization problem is the following:

$$\min \quad \sum_{d\in\mathcal{D}} \Bigg[\underbrace{\sum_{\ell\in\mathcal{L}(d)} \sum_{t\in\mathcal{T}} c_{d,t} z_{d,\ell,t}}_{\text{delay/idle cost}} + \underbrace{u_d \left(1 - \sum_{\ell\in\mathcal{L}(d)} \sum_{t\in\mathcal{T}} z_{d,\ell,t}\right)}_{\text{cost of missed demand}} + \underbrace{\sum_{s\in\mathcal{S}(d)} h_{d,s} w_{d,s}}_{\text{shipping cost}} \Bigg] \tag{1}$$

$$\text{s.t.} \quad \sum_{\ell\in\mathcal{L}(d)} \sum_{t\in\mathcal{T}} z_{d,\ell,t} \le 1 \qquad \forall d \in \mathcal{D} \tag{2}$$

$$\sum_{\ell\in\mathcal{L}(d)} \sum_{t\in\mathcal{T}} z_{d,\ell,t} = \sum_{s\in\mathcal{S}(d)} w_{d,s} \qquad \forall d \in \mathcal{D} \tag{3}$$

$$\delta_{\ell,t} \ge \sum_{d\in\mathcal{D}} z_{d,\ell,t} \qquad \forall \ell \in \mathcal{L}, t \in \mathcal{T} \tag{4}$$

[4] More sophisticated optimization approaches—such as two-stage stochastic optimization—have been explored in Liu et al. (2025); however, their productization remains an avenue for future work.

Table 2 Examples of planner questions supported by the interface.

Natural-language question	Model edit and response sketch
"Can demand d dock at least one week earlier than the current plan?"	Add a corresponding constraint on the docking dates for d, re-solve, and report cost delta.
"What if supplier s is unavailable for a month?"	Zero out availability for s over that horizon, re-solve, report reassignment and cost impact.
"Can we still satisfy all demands if the throughput at data center ℓ drops by 50% next week?"	Set a new value for the corresponding $\delta_{\ell,t}$, resolve and check whether all demands are still satisfied.

$$\rho_\ell \geq \sum_{d\in\mathcal{D}:\ell\in\mathcal{L}(d)} \sum_{t\in\mathcal{T}} z_{d,\ell,t} \qquad \forall \ell \in \mathcal{L} \tag{5}$$

$$\sigma_s \geq \sum_{d\in\mathcal{D}} w_{d,s} \qquad \forall s \in \mathcal{S} \tag{6}$$

$$z_{d,\ell,t}, w_{d,s} \in \{0,1\} \tag{7}$$

Constraints (2) and (3) ensure a single data center, day, and supplier are chosen for the demands that dock successfully. The throughput constraints are enforced in Constraint (4), the row availability constraints in Constraint (5), and the supply availability in Constraint (6).

An agent front-end converts planner requests into checks or edits on the model and data, invokes the solver, and turns solver outputs into explanations and visuals. The loop supports: (i) a trusted snapshot of state to begin each session, (ii) explanations that cite constraints or costs, (iii) conversational scenarios with quick turnaround, and (iv) safe edits followed by fast re-optimization.

As an illustrative what-if vignette, a planner can ask: "demand d must be fulfilled by supplier s". The agent parses the question, invokes a function call that sets $w_{d,s} = 1$, re-solves, and returns a summary: revised fulfillment plan, cost increase, and any schedule impact. Table 2 provides several more examples of planner questions supported by the interface.

The system solves the MILP at a fixed cadence with updates to demand, suppliers and constraints.

5.3. Operational Context and Roles

Given the demand plan, Microsoft runs the optimization algorithm to produce the fulfillment plan. Microsoft planners responsible for the fulfillment plan are professionals with deep business context but they are not necessarily data scientists or optimization experts. Their tasks include confirming that the fulfillment plan meets business needs and ensuring execution of the underlying decisions within the fulfillment plan. This real-world setting therefore touches all four principles of the 4I framework.

When planners receive the outcome of an optimization tool, they can confirm that it meets business needs and ensure the execution of the decisions is completed as planned. However, the increased

complexity of the underlying optimization problems prevents immediate clarity for the reasoning behind each decision. Consequently, before GenAI technology was available, planners would often reach out to the engineers and data scientists who developed the optimization algorithms in order to obtain additional insights. Planners and engineers would have multiple rounds of interaction to understand an issue or explore what-if scenarios, which might have led to delays of several days before reaching a satisfactory solution. This experience emphasized the need for interpretability and a more interactive dialogue with optimization results. To address these barriers, Microsoft has connected the optimization tools with multiple GenAI-based assistants (or "agents"), which reflect the 4I framework (Menache et al. 2025; Simchi-Levi et al. 2025).

Insight: Inform the State. The integration begins by turning the demand plan and operational data into a concise picture of "what is going on." An assistant ingests the demand plan, including business group, hardware type, region, and ideal dock date, and synchronizes it with inventories, open orders, and lane availability. This allows the planner to start with a trusted overview rather than a stack of spreadsheets. In practice, the system summarizes the current load on sources and destinations, flags capacity and compatibility issues, and highlights noticeable patterns and trends. This feature provides the planner with a coherent, up-to-date view of the planning horizon. An agent is also used to track demand drift, that is, changes in the current demand plan and the reasoning behind them (Menache et al. 2025; Simchi-Levi et al. 2025).

Interpretability: Explain the Logic. Once a fulfillment plan is produced, an agent narrates the reasoning behind key assignments. Instead of a black-box shipment matrix, the planner sees why a warehouse to data center pair was chosen, for example a binding capacity at an alternative source, a compatibility restriction on certain racks, or the trade-off between shipping and depreciation costs. The agent may also provide explanations for why certain demands are docked after their ideal dock date. When the planner asks why another option was not selected, the answer identifies the relevant constraints and costs in the model, for example a specific supplier is incompatible with the demand. This restores clarity without requiring the involvement of data scientists and engineers.

Interactivity: Conversational What-Ifs. The natural-language questions that planners ask are converted into live experiments on the model. When a planner asks about docking a demand one week earlier or temporarily deactivating a warehouse, for example, the GenAI agent translates the request into a new constraint that is fed to the optimization algorithm. The algorithm is then invoked and returns the new plan, along with the change in total cost and any service-level impacts. The solver output is translated by the agent into a user-friendly summary that includes the potential cost change and comparisons with the original plan. This conversational loop replaces

the multi-day back-and-forth with engineers and incorporates the exploration of alternatives into everyday planning.

Improvisation: Continuous Re-Optimization. Similar to the what-if execution loop described above, a GenAI assistant can monitor and respond to changing conditions that affect the supply chain. For example, if a key supplier is down for five weeks, the planner can inform the GenAI assistant, which will update availability and other relevant parameters, re-solve the model, and present a refreshed plan with an explanation of the changes. By enabling continuous re-optimization as conditions change, the GenAI layer transforms optimization into an integral part of operations rather than functioning as a static tool.

5.4. Deployment

The AI-assistant system was piloted for Microsoft's cloud supply chain from March 2023 to October 2023 and then deployed broadly in November 2023. It supports a defined catalog of high-value queries and uses fallback messages when a query is out of scope. Planners received training and reference examples. The system has reduced the average response time to planners' questions from 2.5 days to near real-time, leading to a 23% reduction in the time spent by the fulfillment team. Accuracy for the current supported question set is reported around 90%, with gradual expansion based on observed usage and error reports (Menache et al. 2025; Simchi-Levi et al. 2025; Li et al. 2023). The periodic MILP that drives fulfillment has proven adequate for production needs.

The implementation keeps proprietary data in enterprise systems. The assistant translates language into model edits or data queries and calls the optimization and data services through documented interfaces. Results are then summarized back to the user. In this design, raw enterprise data need not be sent to the language model, which is significant from a security standpoint. Public artifacts, including prompts, question handlers, and example pairs, are available in the OptiGuide repository and related documentation (Li et al. 2023; Menache et al. 2025; Simchi-Levi et al. 2025).

6. The Road Ahead: Challenges and Opportunities

The fusion of GenAI with optimization opens up thrilling possibilities, but it also raises important challenges and sets a clear agenda for research and practice. In this concluding section, we outline key considerations for the road ahead. We first discuss technical challenges for optimization researchers and practitioners. We then highlight broader opportunities for future research and practice.

6.1. Methodological Challenges

The examples described in the previous sections motivate a set of research problems that translate practitioner needs into methodological challenges, which we detail below.

6.1.1. Validate LLM-Generated Model or Construct Counterexamples. Given a textual intent T (e.g., a problem description, a what-if question, or an algorithmic task) and a candidate formulation $\hat{M} \in \mathcal{F}$ within a model family $\mathcal{F}$, the task is to design a verifier that either (a) certifies $\hat{M}$ as valid or (b) constructs a counterexample showing a missing constraint. The counterexample is a feasible assignment satisfying $\hat{M}$ but violating one of the intended requirements. This is a formal verification problem within the optimization domain: does $\hat{M}$'s feasible region align with the specification implied by T? The ideal outcome is a polynomial-time verifier that inspects $\hat{M}$'s structure and either confirms soundness or identifies a violated condition. While general model verification is NP-hard, tractable subclasses may be found by exploiting structural properties of $\mathcal{F}$ (e.g., network flow, assignment, or convex constraints). Practical targets can include automated model auditors that validate typical LLM-generated linear or integer programs in a matter of seconds, and formal characterizations of constraint families for which correctness verification or counterexample generation can be done in polynomial time.

In fact, in current real-world deployments, organizations typically begin with valid formulations created by optimization specialists rather than allowing LLMs to generate formulations from scratch, precisely because reliable automated verification remains an open problem. Addressing this methodological challenge would eliminate a key bottleneck in democratizing optimization.

6.1.2. Real-Time Incremental Mixed-Integer Optimization. Given an optimal solution x^* for an instance of an optimization problem and a modified instance differing in only k input parameters (e.g., coefficients or bounds), the challenge is to compute the new optimal solution in time that scales with k rather than with the full problem size. For linear programs, reoptimization is often efficient—warm-starting simplex from the old basis—but for mixed-integer programs, even small perturbations can trigger exponential recomputation. The methodological question is whether proximity-based algorithms can exploit structural similarity to deliver real-time updates, perhaps by restricting the search to a neighborhood of x^* where at most d variables change. This leads naturally to a fixed-parameter tractability perspective: can we design algorithms whose runtime is polynomial in k even for NP-hard problems? Advances here would formalize what practitioners intuitively want—instantaneous adaptation to small input changes without restarting from scratch. Practical benchmarks include sub-second reoptimization for modest data updates or algorithms that provably recompute an exact or near-optimal solution in $O(\text{poly}(k))$ time for structured MILPs, enabling "always-on" optimization that evolves continuously with operations. An alternative direction would be to study quick approximate solutions, which can be specifically relevant for what-if analysis that may not require an exact solution. We briefly discuss this direction below.

6.1.3. Learning to Accelerate Optimization. Integrating learning and optimization offers the prospect of faster, more adaptive decision-making while retaining mathematical rigor. Exact solvers provide correctness and optimality guarantees, whereas learned models can supply heuristics, warm starts, or structure that reduces search effort. For example, an LLM can generate a good feasible solution for a mixed-integer program, suggest variable fixings or decompositions, or predict which constraints will bind at optimality, enabling the solver to focus on critical subsets. Recent studies on LLM–based hyper-heuristics show that such models can discover effective search strategies rivaling expert-designed ones (Yang et al. 2025; Ye et al. 2024). The challenge is to integrate these insights safely: the AI must never compromise feasibility or optimality, and its contributions must be verifiable. Hybrid solvers can maintain rigorous guarantees by allowing the AI to propose candidates that the solver then certifies. Practical targets can include, for instance, solvers that cut solution times by half on standard benchmarks through AI-generated hints, warm starts within 5% of optimal in seconds, and adaptive frameworks where AI guidance continuously improves as more instances are solved. Overall, we believe that using ML and AI techniques for accelerating optimization will remain a key area of investment, as interactivity and improvisation would often necessitate faster solution times.

6.1.4. Training for Explanation Fidelity and Generalization. Current GenAI assistants for optimization are often prompted or trained on fixed sets of question–answer pairs, which limits their ability to generalize and to provide explanations consistent with underlying logic. For instance, in production systems, a planner may routinely ask a defined set of questions (e.g., 1,000 queries about capacity constraints, lead times, or cost trade-offs), and the assistant is trained on expert-provided answers for these questions. One practice in some deployments is to block or redirect any query that falls outside this pre-defined set. The methodological challenge is to train models that can respond to unseen queries about optimization models without retraining, while ensuring their explanations remain faithful to true model reasoning. Formally, given a set of query types defined over decision problems, the task is to learn a mapping from model states to valid answers that generalize beyond the training distribution. This requires training paradigms that integrate symbolic reasoning, chain-of-thought traces, or optimization structure into the model so that explanations arise from internal logic rather than memorized patterns. Key questions include how to measure and enforce explanation fidelity and how to quantify generalization performance for unseen questions. Practical targets can include, for example, assistants that correctly explain or answer at least 90% of unseen queries, with verifiable alignment between their stated rationale and actual model behavior.

6.1.5. Scenario-Robust Policy Optimization without Distributional Information. In many planning problems, decision makers face a limited set of discrete scenarios (e.g., best-, worst-, and most-likely demand forecasts) but lack reliable probabilities for them. As we elaborated earlier, GenAI can facilitate exploring many different scenarios faster. However, a methodological question that still remains is how to eventually select a single policy that performs well across all scenarios without assuming a known distribution. Traditional robust optimization solves a min–max problem, minimizing the worst-case cost, but this approach is often too conservative: it protects against extremes at the expense of typical performance. Conversely, optimizing for a representative or "average" scenario can expose the system to large losses under unfavorable conditions. Intermediate formulations such as min–max regret provide a middle ground, minimizing the maximum deviation from each scenario's optimal value. This retains robustness while avoiding excessive conservatism. Alternative approaches include lexicographic minimax models, which prioritize improvement in the worst case up to a threshold, and scenario-weighted models, where implicit importance weights replace explicit probabilities. From a computational standpoint, scenario-robust optimization is challenging because it effectively embeds multiple optimization problems, often leading to NP-hardness even for moderate scenario counts. Promising directions include decomposition algorithms that solve scenario subproblems independently and coordinate their solutions, or approximation schemes that offer provable bounds on worst-case performance. The objective is to provide planners with tractable methods that yield a single, reliable policy with quantifiable guarantees. An example of practical targets is algorithms that can handle ten or more scenarios and thousands of variables, producing solutions within 10% of each scenario's optimum while solving in minutes.

6.1.6. Data Completeness and LLM-Based Imputation. Real-world datasets are often incomplete. For example, in pricing optimization, one may have demand observations but prices are missing for a large proportion of the dataset; as another example, in retail assortment planning, choice model estimation may be feasible with the majority but not all of the desired purchase records. A key question is whether generative models can fill these gaps and how to measure the reliability of resulting decisions. This challenge has two parts. First, large language models can impute missing values by leveraging unstructured context (e.g., historical records, textual descriptions, or domain patterns) to estimate plausible parameter values. However, these estimates may be inaccurate and their errors propagate through optimization. A promising approach is to treat imputed values as random variables with confidence intervals and embed them in robust formulations that hedge against estimation errors. The second part concerns data acquisition: determining which missing inputs most affect solution quality. One can frame this as an optimization problem over data collection, where the objective is to maximize the expected improvement in performance

for a given effort or cost. This links to value-of-information and experiment design principles, but here guided by algorithmic assessments of what data matter most. The methodological challenge is to quantify decision robustness as a function of data completeness: how much missing data can an optimization tolerate before performance deteriorates? Practical targets can include, for instance, decision systems that remain within a few percent of full-information optimality even with 10–20% missing data, and algorithms that automatically flag which additional data points would yield the largest improvement in decision reliability.

Beyond these six challenges, several emerging directions merit attention. One is determining how often to reoptimize as data and conditions evolve—too frequent updates waste resources, too infrequent ones yield outdated plans. Formal models could guide this balance between responsiveness and efficiency. Another is establishing principles for human–AI governance, defining when algorithmic recommendations should be reviewed or overridden by experts in high-stakes settings. Other fronts include learning from human feedback, such as users' adherence to or deviation from algorithmic recommendations and their ratings of optimization solutions. Addressing these questions will help advance the integration of GenAI and optimization to enable reliable, adaptive decision-making.

6.2. Broader Opportunities

The marriage of GenAI and optimization is a fertile ground for research, inviting contributions from both optimization and AI communities. We highlight several broader opportunities below:

6.2.1. Creating Benchmarks and Shared Tasks: To galvanize progress, the optimization community could spearhead the development of standard benchmarks that evaluate how well AI systems can formulate, solve, and explain optimization problems. For example, a benchmark suite might include a variety of problem descriptions (in natural language or data form) covering linear programming, integer programming, network flows, scheduling, etc., along with expected formulations and solutions. Early efforts in this vein are appearing: Huang et al. (2025) describe benchmarks like NL4Opt, MAMO, and a new Industry OR dataset for real-world problems. Building on this, we could create an open repository of problems and data, preferably extracted from real-world use cases, where any new GenAI-based optimization technique can be tested objectively. This would be analogous to the machine learning field's ImageNet or GLUE benchmarks, which drove rapid progress. A collective "Optimization Plus LLM Task Force" could encourage development and enable tracking of improvements over time. Ideally, the benchmarks should test the full loop: not only can the AI produce a correct model and solution, but can it also explain it accurately, and how does it perform when data is noisy or incomplete (testing robustness)? By defining concrete metrics (accuracy of formulations, optimality gap of solutions, explanation fidelity, etc.), we would provide clear goals that can be pursued by both researchers and practitioners.

6.2.2. Verifying and Validating GenAI–Optimization Models: As discussed in the pitfalls section, there is a broad research question around how to systematically verify an optimization model that was generated (at least in part) by an AI. Optimization researchers can bring expertise in constraint programming and formal methods to this problem. One could imagine a tool that takes an AI-generated model and a description of the real-world constraints and checks for any mismatch. Perhaps it can use satisfiability modulo theory (SMT) solvers such as Z3 (De Moura and Bjørner 2008) to test if the feasible region of the model admits solutions that violate known physical or logical requirements. Another approach is to generate test scenarios or edge cases and see if the model behaves as expected (a kind of stress testing harness for decision models). On the AI side, incorporating techniques such as chain-of-thought prompting and self-reflection (as in ReEvo) might reduce errors. For example, the AI could be prompted to double-check each constraint it produces (is this constraint linear? Does it reflect the user's intent or is something missing?"). Developing a library of common constraints and patterns that an AI can reference could also help. We anticipate research papers that blend formal verification with LLMs, essentially creating a new subfield of AI-assisted model auditing.

6.2.3. Human-AI Collaboration Paradigms in Optimization: Another promising area is studying and designing workflows where humans and AI collaborate on modeling and solutions. Instead of aiming for a fully automated optimization agent, we might find that a *mixed-initiative* approach yields the best outcomes. For example, a human might sketch a partial model, and the AI fills in the rest, or the AI proposes a model and the human revises it iteratively. What is the optimal division of labor? How can interfaces be designed so that human insight (such as domain knowledge of what constraints are truly important) is combined with AI speed and breadth? There may be parallels to pair programming in software development, where one party writes code and another reviews. Huang et al. (2025) hint at human-GenAI collaboration frameworks; those could be tested empirically. Controlled experiments could be conducted where some analysts use AI assistance and others do not, to compare the quality of models and decisions (keeping track of metrics such as solution quality, time taken, and user confidence). We might discover, for instance, that the best outcomes occur when humans focus on defining objectives and high-level structure while AIs handle low-level details and multiple scenarios, but oversight is indispensable at final validation. Insights from cognitive science and HCI (human-computer interaction) could guide these designs. The end goal is to create a synergy where the human's creativity and contextual judgment pair with the AI's computational brute force and memory.

6.2.4. Pilot Projects and Field Tests: Practically, there is a need for more demonstration projects in real organizational settings to validate the value of GenAI-integrated optimization. We encourage industry-academic partnerships to implement prototypes of the kind described in our examples and measure their impact. For instance, a pilot in supply chain planning could introduce an LLM assistant to a planning team and track metrics like planning lead time, service levels achieved, and user satisfaction compared to the prior process. Similarly, a revenue management pilot could see if AI-guided pricing decisions outperform historical manual decisions in terms of revenue or inventory outcomes. Collecting such evidence will be essential for persuading industry leaders and addressing skeptics who question whether GenAI offers real substance or merely hype. A particular area to test is the robustness of decisions made with AI assistance: do they handle variability better? One could simulate, via controlled trials or even using digital-twin environments, how a system combining GenAI and optimization deals with disruptions versus a traditional approach. Another angle is efficiency: does having an AI assistant reduce the need for specialist staff, or does it free those specialists to focus on more strategic analysis? The results of these pilots can feed back into refining the technology and addressing any shortcomings (for example, users found the AI explanations confusing in some cases, leading to improvements in that module).

In sum, the road ahead is both exciting and demanding. There are genuine technical hurdles to overcome and socio-technical questions to answer. Yet, the momentum is clearly there. Just as data science and machine learning have reinvigorated optimization—spurring the thriving analytics movement (Simchi-Levi 2014)—GenAI opens a new wave of innovation. Handled with care, namely ensuring robustness, ethics, and true user-centricity, it holds the promise of fulfilling Herbert Simon's long-standing vision: bringing the power of optimization into the broader, less structured domain of complex decision-making. The potential to amplify human decision intelligence has never been greater.

7. Conclusion

The GenAI era we are living in presents a unique opportunity to reimagine the research and practice of optimization. In this article, we argued that by integrating GenAI into optimization workflows, we can make the power of optimization accessible to a much broader audience of decision-makers. We introduced the 4I framework of Insight, Interpretability, Interactivity, and Improvisation as guiding principles to address the persistent barriers that have kept optimization in a somewhat niche role. GenAI, with its ability to understand natural language and generate context-specific content, serves as the critical enabler for each of these principles: It can synthesize data into insightful narratives, translate between natural language and mathematical formulations to enhance interpretability,

enable interactive what-if analyses through conversational interfaces, and iteratively adapt models and solutions as reality evolves.

Our vision is that in the coming years, advanced decision support will no longer be confined to specialists writing models on whiteboards or tinkering with solver code. Every frontline decision-maker, in effect, can become an optimizer—not in the sense of mastering the simplex or branch-and-bound methods, but in being able to harness optimization technology to inform their choices. This democratization does not diminish the role of optimization experts; on the contrary, it elevates it. Experts are needed to build and maintain these intelligent systems, to ensure their integrity, to tackle the complex problems that automated tools cannot handle alone and help to increase the trust in integrated GenAI and optimization systems. We believe that the application of optimization can be vastly broadened, much as calculators made arithmetic universal or spreadsheets made basic data analysis ubiquitous.

We have also emphasized that this transformation must be approached thoughtfully. Challenges around trust, correctness, bias, and human skill erosion are real. However, the optimization community has a history of rigorous thinking and can bring that to bear on AI integration. By collaborating with AI researchers, we can develop hybrid systems that marry AI's flexibility with optimization's structural genius. By educating the next generation in this blended paradigm, we ensure that the users of these tools remain critical thinkers rather than blind followers of AI. The end goal is not to cede decisions to machines, but to create a symbiosis where human judgment and algorithmic optimization inform each other.

In the grand arc of OR/MS, this moment feels like a new chapter echoing an old theme. Herbert Simon in 1987 called for combining optimization with AI's toolkit to tackle the messy problems of management (Simon 1987). At that time, AI was in its infancy, and the vision outpaced the reality. Today, AI has matured in astonishing ways, and the technology Simon alluded to is here on our desktops and in our pockets. There is a growing consensus that it is now up to us in the OR/MS community to seize this moment (Dai and Swaminathan 2025; Le et al. 2025; Wiberg et al. 2025). By doing so, we can vastly increase optimization's impact on business and society, ensuring that our hard-won modeling insights do not remain an underutilized "quiet giant" but actively help solve the complex, ill-structured problems that organizations and communities face.

Ultimately, the promise of integrating optimization and AI is a world where decisions are more data-driven, options are explored more thoroughly, and outcomes improve across the board—from more resilient supply chains to fairer resource allocations to more responsive public services. And perhaps just as importantly, it is a world where decision-making becomes more engaging for those involved. In place of black-box calculations or gut-feel guesses, we get a collaborative dialogue between humans and machines, each contributing what they do best. In that sense, democratizing

optimization is not about trading people for algorithms; it is about empowering people with better tools to optimize what they care about. The result could be truly transformative for the science and technology of decision-making and the many domains it empowers.

References

Balachandran V, Chen J, Chen L, Garg S, Joshi N, Lara Y, Langford J, Nushi B, Vineet V, Wu Y, et al. (2025) Inference-time scaling for complex tasks: Where we stand and what lies ahead. *arXiv preprint 2504.00294* .

Bringsjord S, Govindarajulu NS (2024) Artificial Intelligence. Zalta EN, Nodelman U, eds., *The Stanford Encyclopedia of Philosophy* (Metaphysics Research Lab, Stanford University), Fall 2024 edition.

Ceccon F, Jalving J, Haddad J, Thebelt A, Tsay C, Laird CD, Misener R (2022) OMLT: Optimization & machine learning toolkit. *arXiv preprint 2202.02414* .

Chen H, Constante-Flores GE, Li C (2024) Diagnosing infeasible optimization problems using large language models. *INFOR: Information Systems and Operational Research* 62(4):573–587.

Chen Z, Zhang X, Zope H, Barbalho H, Mellou K, Molinaro M, Kulkarni J, Menache I, Li S (2025) Optimind: Teaching llms to think like optimization experts. *arXiv preprint arXiv:2509.22979* .

Dai T, Swaminathan JM (2025) AI and operations: A foundational framework of emerging research and practice. Working paper, Johns Hopkins University, https://ssrn.com/abstract=5418934.

De Moura L, Bjørner N (2008) Z3: An efficient smt solver. *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 337–340 (Springer).

Encyclopaedia Britannica (2025) Operations research — history. URL https://www.britannica.com/topic/operations-research/History, article updated through 12 Mar 2025. Contributors include Morris Tanenbaum and William K. Holstein.

Fylstra D, Lasdon L, Watson J, Waren A (1998) Design and use of the microsoft excel solver. *Interfaces* 28(5):29–55.

Hadary O, Marshall L, Menache I, Pan A, Greeff EE, Dion D, Dorminey S, Joshi S, Chen Y, Russinovich M, et al. (2020) Protean: VM allocation service at scale. *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, 845–861.

Huang C, Tang Z, Hu S, Jiang R, Zheng X, Ge D, Wang B, Wang Z (2025) ORLM: A customizable framework in training large models for automated optimization modeling. *Operations Research* ePub ahead of print.

Kobbacy KA, Vadera S, Rasmy MH (2007) AI and OR in management of operations: history and trends. *Journal of the Operational Research Society* 58(1):10–28.

Le TV, Albert LA, Vidal T (2025) Making operations research more accessible: Insights from the rise of machine learning. *INFORMS Journal on Data Science* ePub ahead of print.

Levy D (2005) George B. Dantzig, operations research professor, dies at 90. URL https://news.stanford.edu/stories/2005/05/george-b-dantzig-operations-research-professor-dies-90.

Li B, Mellou K, Zhang B, Pathuri J, Menache I (2023) Large language models for supply chain optimization. *arXiv preprint 2307.03875* .

Li B, Zhang Y, Bubeck S, Pathuri J, Menache I (2024a) Small language models for application interactions: A case study. *arXiv preprint 2405.20347* .

Li S, Kulkarni J, Menache I, Wu C, Li B (2024b) Towards foundation models for mixed integer linear programming. *arXiv preprint 2410.08288* .

Liu RP, Mellou K, Gong EX, Li B, Coffee T, Pathuri J, Simchi-Levi D, Menache I (2025) Efficient cloud server deployment under demand uncertainty. *Manufacturing & Service Operations Management* Forthcoming.

Lu H, Xie Z, Wu Y, Ren C, Chen Y, Wen Z (2025) Optmath: A scalable bidirectional data synthesis framework for optimization modeling. *arXiv preprint 2502.11102* .

Mao H, Alizadeh M, Menache I, Kandula S (2016) Resource management with deep reinforcement learning. *Proceedings of the 15th ACM workshop on hot topics in networks*, 50–56.

Menache I, Pathuri J, Simchi-Levi D, Linton T (2025) How generative AI improves supply chain management. *Harvard Business Review* 104(1–2):86–95.

Simchi-Levi D (2014) OM research: From problem-driven to data-driven research. *Manufacturing & Service Operations Management* 16(1):2–10.

Simchi-Levi D, Mellou K, Menache I, Pathuri J (2025) Large language models for supply chain decisions. Cohen MC, Dai T, eds., *AI in Supply Chains: Perspectives from Global Thought Leaders*, chapter 7 (Springer).

Simon HA (1987) Two heads are better than one: The collaboration between AI and OR. *Interfaces* 17(4):8–15.

Wiberg H, Dai T, Lam H, Kulkarni R (2025) Synergizing artificial intelligence and operations research: Perspectives from INFORMS fellows on the next frontier. *INFORMS Journal on Data Science* ePub ahead of print.

Xu H, Sun Y, Tupayachi J, Omitaomu O, Zlatanova S, Li X (2025) Towards the autonomous optimization of urban logistics: training generative AI with scientific tools via agentic digital twins and model context protocol. *arXiv preprint 2506.13068* .

Yang X, Zhang L, Qian H, Song L, Bian J (2025) HeurAgenix: Leveraging LLMs for solving complex combinatorial optimization challenges. *arXiv preprint 2506.15196* .

Ye H, Wang J, Cao Z, Berto F, Hua C, Kim H, Park J, Song G (2024) ReEvo: Large language models as hyper-heuristics with reflective evolution. *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*.

Zhang W, Dietterich TG (1995) High-performance job-shop scheduling with a time-delay td(λ) network. *Advances in Neural Information Processing Systems 8*, 1024–1030.