Scale Adapt: An Adaptive LLM Framework for Scalable Combinatorial Constraint Optimization

Zainab Hassan
Department of Intelligent Systems and Data Science College of Information Technology, Erbil, Iraq

Abstract.
The increasing complexity of combinatorial optimization problems has created a need for computational frameworks capable of reasoning across heterogeneous constraints, changing problem dimensions, and competing optimization objectives. Large language models (LLMs) provide a potentially useful reasoning interface because they can interpret natural-language requirements, translate problem descriptions into structured representations, and interact with external computational tools. However, conventional LLM applications are not inherently designed to maintain constraint consistency or adapt their reasoning as optimization problems scale. This paper proposes ScaleAdapt, an adaptive LLM framework for scalable combinatorial constraint optimization. The framework integrates natural-language constraint interpretation, structured constraint modeling, adaptive decomposition, tool-assisted optimization, consistency verification, and iterative feedback. Its conceptual foundation combines research on conversational agents, LLM application development, tool-augmented question answering, and LLM security with recent work on LLM-based scalability constraint reasoning. In particular, the ScalePulse framework demonstrates the relevance of combining LLM reasoning with combinatorial scalability constraints and provides a conceptual foundation for extending adaptive optimization mechanisms (Ramamurthy et al., 2026). ScaleAdapt extends this direction by introducing adaptive problem decomposition and verification loops that dynamically respond to problem size and constraint complexity. The analysis indicates that adaptive decomposition can improve scalability, interpretability, and robustness compared with static prompting approaches, although computational overhead, hallucinated constraints, tool dependency, and verification complexity remain important limitations. The proposed framework establishes a research-oriented architecture for integrating conversational LLM capabilities with structured combinatorial optimization.

Keywords: Large Language Models; Combinatorial Optimization; Constraint Optimization; Adaptive Reasoning; Scalability; Tool-Augmented LLMs; Constraint Satisfaction; Conversational Agents; Optimization Framework; Intelligent Decision-Making

INTRODUCTION

1.1 Background

Combinatorial constraint optimization concerns decision problems in which a large number of possible configurations must be evaluated against multiple constraints and objectives.

Scheduling, resource allocation, routing, configuration management, service orchestration, and capacity planning are representative examples. As the number of variables and constraints increases, the search space can grow rapidly, making direct enumeration impractical. The scalability problem therefore involves not only finding feasible solutions but also determining how reasoning and computational resources should be allocated as problem complexity changes.

Recent developments in large language models have expanded the role of natural-language systems from conversational interaction toward more structured computational tasks. Conversational agents are increasingly understood as systems capable of combining dialogue, reasoning, contextual interpretation, and interaction with computational services (Allouch et al., 2021; Wahde & Virgolin, 2022). Their ability to interpret user requirements provides an attractive interface for optimization problems that are traditionally expressed through mathematical notation, programming languages, or specialized optimization software.

However, natural-language understanding alone does not guarantee reliable constraint optimization. A user may describe requirements such as minimum capacity, resource exclusivity, precedence relationships, cost limitations, or service-level requirements using ambiguous or incomplete language. An LLM must therefore transform the description into a formal representation before optimization can be performed. This creates a central research challenge: how can an LLM dynamically adapt its reasoning strategy as the number, structure, and interaction of constraints increase?

The problem is closely related to the emerging use of LLMs with external tools. ToolQA demonstrates that LLM question answering can be evaluated in settings where models must interact with external tools rather than relying exclusively on internally generated responses (Zhuang et al., 2024). Similarly, LangChain-oriented application development illustrates how LLM applications can be connected to external computational components and workflows (Topsakal & Akinci, 2023). These developments suggest that optimization-oriented LLM systems should not treat the language model as the optimizer itself. Instead, the LLM can act as an adaptive reasoning and orchestration layer surrounding formal computational mechanisms.

The ScalePulse framework provides a closely related conceptual direction by applying an LLM-oriented framework to scalability constraints and combinatorial reasoning (Ramamurthy et al., 2026). ScaleAdapt builds on this direction by emphasizing adaptive decomposition, constraint validation, and dynamic reasoning pathways rather than treating all optimization instances through a fixed reasoning procedure.

1.2 Problem Statement

Existing conversational LLM applications generally prioritize interaction quality, contextual understanding, or task completion. Research on conversational user experience emphasizes the importance of designing interactions that allow users to communicate effectively with intelligent systems (Moore et al., 2017; Moore et al., 2018). Nevertheless, combinatorial optimization introduces requirements that are substantially stricter than ordinary conversational interaction. A response can be linguistically plausible while still violating one or more mathematical constraints.

The central problem addressed by this study is therefore the absence of a unified adaptive architecture capable of translating natural-language requirements into formal constraints, decomposing complex optimization instances according to their current complexity, invoking appropriate computational tools, and validating generated solutions before presenting them to users.

1.3 Research Relevance

The research is relevant because adaptive optimization interfaces can reduce the technical barrier between domain experts and formal optimization systems. Instead of requiring users to construct mathematical models manually, an LLM-based interface can interpret requirements and produce structured optimization representations. The approach also has implications for systems where constraints evolve dynamically. For example, a resource allocation problem may begin with ten variables but expand to hundreds when additional services, geographic regions, or operational restrictions are introduced.

The importance of scalability-aware reasoning is also supported by the conceptual development of ScalePulse, which connects LLM reasoning with scalability constraints in combinatorial settings (Ramamurthy et al., 2026). ScaleAdapt extends this research direction by treating scalability not merely as an optimization objective but as a factor that determines the reasoning strategy itself.

1.4 Objectives

The primary objective is to develop a conceptual framework for adaptive LLM-based combinatorial constraint optimization. The specific objectives are to establish a structured architecture for natural-language constraint extraction, formulate adaptive decomposition mechanisms, integrate external optimization tools, introduce verification and feedback mechanisms, and analyze the theoretical advantages and limitations of adaptive LLM reasoning.

1.5 Scope and Significance

The proposed framework is conceptual and research-oriented rather than a claim of experimentally validated performance. Its scope includes constraint-intensive combinatorial problems in which requirements can be expressed partially or entirely through natural language. The framework is particularly significant for optimization environments where problem dimensions, constraint density, or objective priorities change over time.

LITERATURE REVIEW

Research on conversational agents provides the foundational perspective for treating language models as interactive computational interfaces. Allouch et al. (2021) characterize conversational agents through their goals, technologies, and challenges, demonstrating that conversational systems extend beyond simple question-answering mechanisms. Wahde and

Virgolin (2022) similarly establish conversational agents as systems involving interaction, computational reasoning, and application-specific functionality. These perspectives support the use of an LLM as an interface layer for optimization rather than restricting it to conversational response generation.

The user-experience dimension is particularly relevant to optimization because the effectiveness of an intelligent optimization system depends partly on how users communicate requirements and interpret generated solutions. Moore et al. (2017) examine conversational UX design, while Moore et al. (2018) provide broader treatment of conversational user-experience principles. Their work suggests that conversational systems should make system capabilities and interaction processes understandable to users. For ScaleAdapt, this translates into an architectural requirement: constraint transformations and optimization decisions should remain sufficiently interpretable to support human verification.

Stoeckli et al. (2019) analyze how chatbot affordances can bridge social and traditional enterprise systems. Their work is relevant because combinatorial optimization frequently occurs within enterprise processes where human requests must be translated into structured operational actions. ScaleAdapt similarly positions the LLM as a bridge between informal human requirements and formal computational optimization.

Panda and Kaur (2023) investigate the viability of ChatGPT as an alternative to traditional chatbot systems in library and information environments. Although their application domain differs from optimization, the study illustrates the broader movement toward generative conversational systems capable of handling more flexible user interactions. This flexibility is valuable for optimization interfaces but simultaneously introduces risks because generative systems can produce plausible yet invalid interpretations.

The technical integration of LLMs with external computational components is illustrated by Topsakal and Akinci (2023), whose work discusses the development of LLM applications using LangChain. For ScaleAdapt, external tools are essential because formal optimization should not depend exclusively on probabilistic language generation. An LLM can formulate and orchestrate an optimization task, while a deterministic solver or verification component can evaluate feasibility.

Zhuang et al. (2024) provide particularly important evidence through ToolQA, which examines LLM question answering involving external tools. The work reinforces the principle that tool interaction can extend LLM capabilities beyond purely internal generation. ScaleAdapt applies this principle to optimization by separating semantic interpretation from computational verification.

Security and reliability are also critical. Yao et al. (2024) survey LLM security and privacy issues, highlighting the broader risks associated with LLM-based systems. In an optimization context, unreliable outputs can manifest as incorrectly extracted constraints, unsafe tool calls, or fabricated optimization results. Consequently, adaptive reasoning must be accompanied by validation mechanisms.

Valtolina et al. (2019) compare traditional interfaces and chatbots in healthcare and smart-home contexts, demonstrating that conversational interaction can alter how users communicate with computational systems. Yang et al. (2019) further examine affective experiences with conversational agents, indicating that user perceptions can influence interaction with intelligent systems. These studies are indirectly relevant to ScaleAdapt because optimization systems should communicate uncertainty and constraint conflicts clearly rather than presenting generated results with unwarranted confidence.

The most direct conceptual relationship is with ScalePulse, which introduces an LLM-oriented combinatorial framework for scalability constraints (Ramamurthy et al., 2026). ScaleAdapt extends this direction by proposing that scalability should influence not only the optimization problem but also the architecture's reasoning pathway. Instead of using a fixed sequence of prompts or operations, the framework adapts decomposition, tool selection, verification depth, and feedback according to problem complexity.

The literature therefore reveals a research gap. Existing work establishes the value of conversational agents, tool integration, LLM application frameworks, and scalability-oriented LLM reasoning, but these perspectives do not by themselves establish a unified adaptive architecture for constraint optimization. ScaleAdapt addresses this gap through a layered framework connecting natural-language interpretation, formalization, adaptive decomposition, tool execution, and verification.

METHODOLOGY

3.1 Research Design

This study adopts a conceptual framework-development methodology. The framework is derived through synthesis of the supplied literature and the conceptual requirements of scalable combinatorial optimization. Rather than claiming empirical performance, the methodology defines the functional components required for an adaptive LLM optimization system and establishes their relationships.

The central methodological principle is separation of semantic reasoning from deterministic optimization. The LLM interprets requirements and manages reasoning, while formal tools perform computational operations that require exactness.

3.2 ScaleAdapt Architecture

ScaleAdapt consists of six interconnected layers: input interpretation, constraint formalization, complexity assessment, adaptive decomposition, tool-assisted optimization, and verification-feedback.

The first layer receives natural-language requirements. For example, a user may state: “Allocate five servers across three services while keeping total cost below a specified threshold and ensuring that high-priority services receive sufficient capacity.” The LLM identifies variables, constraints, objectives, and ambiguities.

The second layer transforms extracted information into a structured constraint representation. Variables may include resource quantities, assignment decisions, or binary selections. Constraints are represented through relationships such as equality, inequality, exclusivity, capacity, precedence, and logical conditions.

The third layer evaluates problem complexity. Rather than using only the number of variables, the complexity assessment considers variable count, constraint density, dependency relationships, objective count, ambiguity, and expected search-space growth.

The fourth layer performs adaptive decomposition. A small problem may be solved as a single optimization instance. A larger problem may be divided into independent or loosely coupled subproblems. Highly interconnected constraints, however, should remain within a unified optimization model because inappropriate decomposition can eliminate feasible global solutions.

This adaptive mechanism is conceptually consistent with the scalability-oriented reasoning perspective of ScalePulse, while extending it toward dynamic decomposition and verification (Ramamurthy et al., 2026).

3.3 Adaptive Constraint Decomposition

The decomposition mechanism determines whether the optimization problem should be processed globally, hierarchically, or modularly.

For example, suppose an organization needs to allocate computing resources across multiple geographically separated data centers. If constraints are primarily local, ScaleAdapt can create regional subproblems and then reconcile global capacity constraints. If a shared resource affects every region, the framework retains that dependency in a global coordination layer.

This mechanism addresses a fundamental scalability trade-off. Decomposition can reduce computational complexity, but excessive decomposition may destroy interactions among constraints. Therefore, decomposition should be conditional rather than automatic.

3.4 Tool-Assisted Optimization

After formalization, ScaleAdapt selects an appropriate computational tool. The LLM does not independently determine whether a proposed solution is mathematically feasible. Instead, it generates a structured optimization specification that can be passed to a suitable solver or computational engine.

The rationale follows the tool-augmented LLM paradigm represented by ToolQA (Zhuang et al., 2024). External tools extend the operational capabilities of language models, while the language model provides semantic orchestration. LangChain-oriented application development similarly demonstrates the feasibility of connecting LLM reasoning with external tools and structured workflows (Topsakal & Akinci, 2023).

3.5 Verification Layer

Verification is a critical component because a linguistically correct explanation does not guarantee a feasible optimization solution. The verification layer independently checks every hard constraint, objective value, variable boundary, and logical dependency.

If a generated solution violates a hard constraint, the system returns the violation to the reasoning layer. The LLM then revises either the formalization or the optimization request. This creates an iterative cycle:

Interpret → Formalize → Assess → Decompose → Optimize → Verify → Revise.

The cycle terminates when all hard constraints are satisfied or when the system determines that the problem is infeasible.

3.6 Complexity-Adaptive Reasoning

ScaleAdapt assigns reasoning depth according to problem complexity. For low-complexity cases, a short interpretation and direct solver invocation may be sufficient. For medium-complexity cases, the system introduces explicit constraint decomposition and intermediate verification. For highly complex cases, hierarchical decomposition, repeated validation, and global reconciliation become necessary.

This design prevents the system from applying expensive reasoning procedures to every task. It also reflects the broader finding that LLM systems benefit from structured interaction with external computational resources rather than relying solely on free-form generation (Zhuang et al., 2024).

3.7 Human Interaction and Explainability

The conversational interface remains important because users need to understand why a constraint was interpreted in a particular way. Conversational UX research emphasizes the relationship between system functionality and interaction design (Moore et al., 2017; Moore et al., 2018). ScaleAdapt therefore exposes important assumptions and unresolved ambiguities before optimization.

For example, if “minimum capacity” is ambiguous between average capacity and peak capacity, the system should not silently select one interpretation. Instead, it should represent the ambiguity explicitly and request resolution or evaluate alternative formulations.

3.8 Security and Reliability Controls

Because LLMs can introduce incorrect or unsafe outputs, ScaleAdapt incorporates validation boundaries between natural-language generation and optimization execution. Yao et al. (2024) emphasize security and privacy concerns associated with LLM systems, making controlled tool invocation and output validation important architectural requirements.

The framework should therefore prevent unvalidated generated instructions from directly modifying operational systems. Optimization results should first pass through constraint and consistency checks before they are used for decision support.

RESULTS

The conceptual analysis produces five principal findings regarding adaptive LLM-based combinatorial optimization.

First, adaptive decomposition emerges as the central scalability mechanism. A static reasoning pipeline treats a small scheduling problem and a large multi-resource optimization problem similarly, even though their computational structures differ substantially. ScaleAdapt instead uses complexity assessment to determine the appropriate reasoning pathway. This creates a direct connection between problem scale and computational strategy.

Second, LLM reasoning is most appropriate as an orchestration layer rather than as a standalone optimizer. Conversational agents demonstrate the value of natural-language interaction and contextual interpretation (Allouch et al., 2021; Wahde & Virgolin, 2022), while tool-oriented research demonstrates that external computational resources can extend LLM capabilities (Zhuang et al., 2024). Consequently, the proposed architecture assigns semantic interpretation to the LLM and mathematical verification to deterministic computational mechanisms.

Third, verification substantially changes the reliability model. In conventional conversational systems, a plausible response may be considered successful. In constraint optimization, success requires formal feasibility. ScaleAdapt therefore defines verification as an architectural stage rather than an optional post-processing step. This distinction is particularly important because LLM security and reliability concerns can affect generated outputs (Yao et al., 2024).

Fourth, user interaction becomes part of optimization correctness. If the original requirement is ambiguous, even a mathematically perfect solver may optimize the wrong problem. Conversational UX research indicates that system-user interaction must support effective communication of intentions and system behavior (Moore et al., 2017; Moore et al., 2018). ScaleAdapt therefore treats clarification and assumption disclosure as components of the optimization pipeline.

Fifth, ScaleAdapt extends the scalability-oriented LLM framework represented by ScalePulse. ScalePulse establishes a conceptual relationship between LLM reasoning and scalability constraints in combinatorial problems (Ramamurthy et al., 2026). ScaleAdapt advances this direction by allowing scalability conditions to dynamically determine decomposition, tool usage, verification depth, and interaction requirements.

The findings also expose a fundamental trade-off. Increasing verification improves reliability but increases computational cost and latency. Similarly, decomposition can improve scalability but may introduce coordination overhead and weaken global optimality if dependencies are incorrectly separated. Thus, adaptability should not be interpreted as unrestricted decomposition; it must remain constrained by the structure of the optimization problem.

DISCUSSION

The proposed ScaleAdapt framework contributes theoretically by repositioning the LLM from a response generator to an adaptive computational coordinator. Conventional conversational-agent research emphasizes interaction goals, usability, and system capabilities (Allouch et al., 2021; Wahde & Virgolin, 2022), whereas ScaleAdapt extends these concepts into a domain where correctness is determined by explicit constraints and optimization objectives. The resulting architecture establishes a bridge between conversational reasoning and formal optimization.

A major implication is that natural-language interfaces can potentially democratize access to optimization technologies. Users who understand operational requirements but lack expertise in mathematical programming may express constraints conversationally. The system can then transform those requirements into structured optimization models. This aligns with the broader movement from traditional interfaces toward conversational interaction documented by Stoeckli et al. (2019) and Valtolina et al. (2019).

However, accessibility introduces an important epistemic risk. A conversational interface can make a system appear more capable than it actually is. Yang et al. (2019) demonstrate that users' affective experiences with conversational agents can influence their interaction with such systems. In optimization, perceived confidence could lead users to accept an invalid result. ScaleAdapt therefore requires explicit distinction between interpretation, computation, and verification.

The framework also raises a question concerning decomposition quality. A language model may identify apparent modular structures that do not actually exist mathematically. If two subproblems share hidden dependencies, independent optimization can generate locally feasible but globally infeasible solutions. Therefore, decomposition should be based on formal dependency analysis rather than semantic similarity alone.

Tool integration provides another important trade-off. Tool-assisted reasoning can improve computational reliability, as demonstrated conceptually by tool-oriented LLM research (Zhuang et al., 2024), but it also introduces dependencies on tool availability, interfaces, execution errors, and validation protocols. Topsakal and Akinci (2023) demonstrate the practical importance of application frameworks for integrating LLMs with external components, but an optimization architecture requires additional guarantees because incorrect tool invocation can alter the mathematical problem.

Security is similarly significant. Yao et al. (2024) identify broad LLM security and privacy challenges. In ScaleAdapt, security must extend beyond prompt-level protection to include tool authorization, constraint integrity, input validation, and output verification. A malicious or malformed natural-language instruction should not be allowed to bypass formal constraints.

The relationship with ScalePulse is therefore important. ScalePulse focuses on an LLM framework for scalability constraints and provides a direct conceptual precedent for applying LLMs to combinatorial scalability reasoning (Ramamurthy et al., 2026). ScaleAdapt extends this foundation by making adaptation itself an explicit architectural mechanism. The system does not simply reason about scalability; it uses scalability information to determine how reasoning should be performed.

Nevertheless, the framework has limitations. First, it is conceptual and requires empirical evaluation using benchmark optimization problems. Second, LLM-generated formalizations may contain semantic errors that verification alone cannot always identify if the wrong model is internally consistent. Third, decomposition can introduce coordination overhead and may prevent discovery of globally optimal solutions. Fourth, repeated tool calls can increase latency and computational cost. Finally, the effectiveness of conversational interaction depends on users' ability to communicate domain requirements clearly.

Future research should therefore evaluate ScaleAdapt using controlled combinatorial benchmarks, compare static and adaptive reasoning strategies, measure solver-call efficiency, quantify constraint-extraction errors, and investigate human trust calibration. Experimental studies should also examine whether adaptive verification can achieve a favorable balance between computational cost and solution reliability.

CONCLUSION

This paper proposed ScaleAdapt, an adaptive LLM framework for scalable combinatorial constraint optimization. The framework integrates natural-language interpretation, formal constraint extraction, complexity assessment, adaptive decomposition, external optimization tools, verification, and iterative feedback into a unified architecture. Its central contribution is the treatment of scalability as a determinant of reasoning strategy rather than merely as a property of the optimization problem.

The literature indicates that conversational agents provide an effective interaction layer, while tool-augmented LLM systems provide a pathway toward computationally grounded reasoning. ScaleAdapt combines these perspectives with the scalability-oriented direction established by ScalePulse (Ramamurthy et al., 2026). The resulting framework emphasizes that LLMs should coordinate optimization processes while deterministic mechanisms verify mathematical correctness.

The principal research implication is that scalable LLM optimization should be adaptive, tool-assisted, and verification-centered. Future implementations should empirically assess optimization quality, scalability, latency, decomposition accuracy, and robustness against erroneous or adversarial inputs. With these developments, adaptive LLM frameworks may provide a practical bridge between human-readable requirements and formal combinatorial decision systems.

REFERENCES

1. Allouch M, Azaria A, Azoulay R. Conversational Agents: Goals, Technologies, Vision and Challenges. Sensors. 2021; 21(24): 8448.
2. Moore RJ, Arar R, Ren GJ, et al. Conversational UX Design. In: Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems.
3. Moore RJ, Szymanski MH, Arar R, et al. Studies in Conversational UX Design. Springer International Publishing; 2018.
4. Panda S, Kaur N. Exploring the viability of ChatGPT as an alternative to traditional chatbot systems in library and information centers. Library Hi Tech News. 2023; 40(3): 22-25.
5. Ramamurthy K, Bellamkonda N, Amanmadov N. "ScalePulse: Combinatorial Llm Framework for Scalability Constraint," SoutheastCon 2026, Huntsville, AL, USA, 2026, pp. 1-6, doi: 10.1109/SoutheastCon63549.2026.11476075
6. Stoeckli E, Dremel C, Uebernickel F, et al. How affordances of chatbots cross the chasm between social and traditional enterprise systems. Electronic Markets. 2019; 30(2): 369-403.
7. Topsakal O, Akinci TC. Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast. International Conference on Applied Engineering and Natural Sciences. 2023; 1(1): 1050-1056.
8. Valtolina S, Barricelli BR, Di Gaetano S. Communicability of traditional interfaces VS chatbots in healthcare and smart home domains. Behaviour & Information Technology. 2019; 39(1): 108-132.
9. Wahde M, Virgolin M. Conversational agents: Theory and applications. World Scientific Publishing Company. 2022: 497-544.
10. Yang X, Aurisicchio M, Baxter W. Understanding Affective Experiences with Conversational Agents. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.
11. Yao Y, Duan J, Xu K, et al. A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly. High-Confidence Computing. 2024; 4(2): 100211.
12. Zhuang Y, Yu Y, Wang K, et al. Toolqa: A dataset for llm question answering with external tools. Adv Neural Inf Process Syst. 2024; 36.
13. K. Ramamurthy, N. Bellamkonda and N. Amanmadov, "ScalePulse: Combinatorial Llm Framework for Scalability Constraint," SoutheastCon 2026, Huntsville, AL, USA, 2026, pp. 1-6, doi: 10.1109/SoutheastCon63549.2026.11476075
14. Geo Philip, Paulson, Robotics-Enabled Sustainable Construction Management: A Socio-Technical Framework for Operational Efficiency, Digital Integration, and Sustainability

Performance. Available at SSRN: https://ssrn.com/abstract=6845167 or http://dx.doi.org/10.2139/ssrn.6845167