Frontiers in Emerging Multidisciplinary Sciences (Open Access)

Volume 03, Issue 08, August 2026
Publish Date: 18-08-2026
https://doi.org/10.64917/fems/Volume03Issue08-07 PageNo.104-110

Combi Scale: A Large Language Model Framework for Scalable Constraint Reasoning and Optimization

Haruto Tanaka
Department of Computer Science and Artificial Intelligence Institute of Machine Learning Research Japan

Yuki Nakamura
Department of Computer Science and Intelligent Systems, Digital Technology Research Institute, Osaka, Japan

ABSTRACT

The increasing complexity of software-intensive systems has created a need for scalable reasoning mechanisms capable of identifying interacting constraints rather than evaluating isolated performance conditions. Conventional optimization approaches frequently treat constraints independently, although real-world systems exhibit dependencies among computational, architectural, resource, and operational variables. This paper proposes CombiScale, a conceptual large language model (LLM) framework for scalable constraint reasoning and optimization. The framework combines language-based semantic interpretation, dependency-aware constraint representation, combinatorial reasoning, risk propagation, optimization-oriented recommendation, and explainable decision support. Its theoretical foundation is informed by the combinatorial LLM approach presented in ScalePulse, which demonstrates the value of jointly analyzing software artifacts and architectural relationships for scalability assessment (Ramamurthy, Bellamkonda and Amanmadov, 2026). The proposed framework extends this reasoning paradigm by treating scalability optimization as a multi-constraint decision problem rather than solely as anomaly detection. The supplied literature also demonstrates the importance of structured sensing, resource allocation, environmental monitoring, and multidimensional system analysis, providing transferable principles for constraint-aware optimization (Alizadeh & Soltani, 2016; Almagrabi, 2020; Cao et al., 2022; Chaudhri et al., 2022). The resulting framework is organized into six functional layers: constraint extraction, semantic normalization, dependency graph construction, combinatorial reasoning, optimization, and explainability. Analytical findings indicate that the principal value of CombiScale lies in connecting local constraints with their systemic consequences, enabling scalable reasoning across heterogeneous system representations. The framework nevertheless remains constrained by LLM inference cost, dependency-quality requirements, uncertainty propagation, and the absence of a universal optimization objective. Future validation should therefore combine controlled benchmark datasets with real-world workload and architectural telemetry.

KEYWORDS: Large Language Models, Constraint Reasoning, Scalability Optimization, Combinatorial Reasoning, Software Architecture, Dependency Graphs, Explainable AI, Intelligent Optimization.

INTRODUCTION

1.1 Background

Scalability is not simply a property of increasing computational capacity; it is a system-level characteristic determined by the interaction of resources, dependencies, workloads, algorithms, and architectural decisions. A system may perform adequately under normal conditions while exhibiting severe degradation when several constraints become active simultaneously. This creates a fundamental challenge for intelligent optimization: identifying a single bottleneck is often insufficient because the observed performance limitation may be produced by a combination of interdependent constraints.

The emergence of large language models creates an opportunity to address this problem through semantic and contextual reasoning. LLMs can process heterogeneous representations such as source code, architectural descriptions, technical documentation, configuration information, and natural-language requirements. The ScalePulse study provides a particularly relevant foundation by demonstrating a combinatorial LLM

architecture in which code representations and dependency structures are analyzed jointly for scalability-constraint detection (Ramamurthy, Bellamkonda and Amanmadov, 2026). Its central implication is that scalability reasoning can move beyond isolated static indicators toward integrated representations of system structure.

However, detecting a constraint is different from optimizing a system under multiple constraints. A practical optimization framework must determine which constraints interact, which are dominant, what downstream effects they create, which interventions are feasible, and how competing objectives should be balanced. CombiScale is therefore proposed as a framework that extends constraint detection toward constraint reasoning and optimization.

1.2 Problem Statement

Existing constraint-oriented systems may identify individual anomalies, inefficient components, or resource limitations, but complex systems frequently exhibit nonlinear interactions. For example, increasing computational capacity may not resolve a bottleneck if communication dependencies, energy allocation, sensor coverage, or architectural coupling remain restrictive. Research on wireless rechargeable sensor networks demonstrates that energy division must account for network-wide operational requirements rather than treating energy distribution as an isolated decision (Almagrabi, 2020). Similarly, gas-sensor research illustrates how sensing performance depends on combinations of sensor characteristics, pattern-recognition requirements, and environmental conditions (Alizadeh & Soltani, 2016; Chiu & Tang, 2013).

The research problem addressed by this paper is therefore: How can an LLM-based architecture represent, reason over, prioritize, and optimize interacting scalability constraints while maintaining explainability and computational feasibility?

1.3 Research Relevance

The problem is relevant because scalable systems increasingly operate across heterogeneous computational and sensing environments. Optimization decisions may involve multiple objectives, including performance, resource consumption, robustness, fairness, latency, and operational cost. The literature supplied by the user demonstrates that such multidimensional optimization problems occur across different technological domains, including energy allocation, gas sensing, environmental monitoring, and resource-constrained IoT systems (Almagrabi, 2020; Chaudhri et al., 2022; Chen et al., 2007).

CombiScale positions LLM reasoning as a coordination mechanism capable of translating heterogeneous technical information into a common constraint representation.

1.4 Objectives

The objectives of the proposed framework are to:

1. establish a semantic representation of heterogeneous scalability constraints;
2. identify interactions and dependencies among constraints;
3. apply combinatorial reasoning to determine high-impact constraint combinations;
4. generate optimization alternatives according to explicit system objectives;
5. provide interpretable rationales for optimization recommendations; and
6. support scalable integration with development, monitoring, and decision-making workflows.

1.5 Scope and Significance

The framework is primarily conceptual and focuses on system-level constraint reasoning rather than a specific programming language, hardware platform, or application domain. Its significance lies in providing an architecture through which LLM reasoning can be connected to structured optimization mechanisms. The approach is particularly relevant to systems in which constraints are heterogeneous and interconnected.

LITERATURE REVIEW

2.1 Constraint Representation and Intelligent Sensing

Alizadeh and Soltani (2016) investigated a reduced graphene oxide-based gas sensor array for DMMP vapor pattern recognition. The study demonstrates the importance of combining multiple sensor signals to distinguish patterns rather than relying on a single measurement. This principle is transferable to scalable constraint reasoning because system-level optimization similarly requires interpreting multiple indicators collectively.

Chiu and Tang (2013) examined chemiresistive sensor-integrated electronic noses and emphasized integrated sensing architectures. From a CombiScale perspective, integrated sensing provides a useful conceptual analogy:

optimization requires multiple observations to be transformed into a coherent representation before reasoning can occur. The framework therefore treats constraints as a structured collection of signals rather than isolated observations.

Cao et al. (2022) examined TiO2 nanostructures with different crystal phases for sensitive acetone gas sensing. Their work illustrates how changes in underlying system characteristics can alter sensing behavior. For computational optimization, this supports the proposition that constraint models should preserve contextual information rather than reducing every observation to a single generic score.

2.2 Resource Allocation and Network-Level Optimization

Almagrabi (2020) investigated fair energy division for wireless rechargeable sensor networks. The relevance to CombiScale is the recognition that local resource allocation can influence broader network operation. An optimization framework must consequently distinguish between local improvements and system-level improvements.

Chaudhri et al. (2022) addressed gas sensor node optimization in a resource-constrained 6G-IoT environment using zero-padding and spatial augmentation. This study is particularly relevant to the proposed framework because it explicitly combines sensing requirements with resource constraints. CombiScale adopts a similar multidimensional perspective by representing constraint relationships and evaluating candidate interventions according to system-level objectives.

2.3 Environmental and Health-Related Multivariable Systems

Chen et al. (2007) analyzed health effects associated with outdoor air pollutants, including nitrogen dioxide, sulfur dioxide, and carbon monoxide. Their work highlights that complex environmental outcomes may be associated with multiple pollutant variables rather than one independent factor. Balakrishnan et al. (2019) further demonstrated the broad health burden associated with air pollution across Indian states, while Chakraborty and Basu (2021) connected air-quality conditions with environmental injustice.

These studies are not direct LLM or software-optimization studies, but they reinforce an important theoretical principle: complex outcomes require multidimensional reasoning. CombiScale transfers this principle to computational systems by modeling interacting constraints and their downstream effects.

2.4 LLM-Based Combinatorial Reasoning

The most directly relevant supplied reference is the ScalePulse framework proposed by Ramamurthy, Bellamkonda and Amanmadov (2026). ScalePulse combines LLM-based analysis with dependency graphs, risk propagation, forecasting, and explainability to identify emerging scalability constraints. Its reported framework architecture demonstrates how language-based reasoning can be combined with structural system representations.

CombiScale builds conceptually on this direction but changes the primary objective from scalability-risk detection to constraint reasoning and optimization. Rather than asking only whether a component represents a scalability risk, CombiScale asks how multiple constraints interact and which intervention provides the most favorable trade-off.

2.5 Research Gap

The supplied literature collectively demonstrates strong capabilities in sensing, resource allocation, network optimization, environmental analysis, and LLM-supported scalability assessment. Nevertheless, these perspectives remain largely domain-specific. A gap exists in a unified architecture that represents heterogeneous constraints, reasons about their combinations, evaluates competing optimization actions, and communicates the rationale in human-readable form.

This gap motivates CombiScale. The framework treats an LLM not as an isolated prediction model but as a semantic reasoning layer connected to structured graphs and optimization mechanisms. This conceptual direction is consistent with the combinatorial architecture of ScalePulse (Ramamurthy, Bellamkonda and Amanmadov, 2026).

METHODOLOGY

3.1 Research Design

This study adopts a framework-development methodology. Instead of claiming experimental validation not contained in the supplied references, the research develops a conceptual architecture derived from cross-domain synthesis. The methodology consists of six stages: constraint extraction, normalization, dependency modeling, combinatorial reasoning, optimization, and explanation.

The fundamental abstraction is a constraint vector:

[

C={c_1,c_2,\ldots,c_n}

]

where each (c_i) represents a measurable or semantically inferred restriction. Each constraint is associated with attributes such as severity, confidence, affected component, resource dimension, temporal behavior, and dependency relationships.

3.2 Constraint Extraction Layer

The first layer converts heterogeneous inputs into structured constraint candidates. Inputs may include source code, architectural diagrams, system logs, configuration files, resource metrics, natural-language requirements, or sensor observations.

The LLM performs semantic extraction rather than relying exclusively on numerical thresholds. For example, a textual architectural description stating that several services depend on a single processing component can be converted into a potential coupling constraint.

This approach reflects the broader integrated-sensing principle found in the supplied literature (Chiu & Tang, 2013; Alizadeh & Soltani, 2016).

3.3 Semantic Normalization

Raw constraints may have different units and meanings. CombiScale therefore maps them into a normalized representation:

[

c_i=(x_i,d_i,s_i,q_i)

]

where (x_i) represents the measured state, (d_i) represents constraint dimension, (s_i) denotes severity, and (q_i) indicates confidence.

Normalization prevents the optimization layer from treating heterogeneous variables as directly comparable without contextual interpretation.

3.4 Constraint Dependency Graph

The framework constructs a directed graph:

[

G=(V,E,W)

]

where nodes represent constraints or system components, edges represent dependency relationships, and weights indicate interaction strength.

For example, a computational constraint may increase queue length, which may increase latency and consequently affect service availability. Instead of representing these as independent observations, CombiScale represents them as connected causal or dependency pathways.

This graph-based perspective extends the system-level reasoning demonstrated by ScalePulse (Ramamurthy, Bellamkonda and Amanmadov, 2026).

3.5 Combinatorial Reasoning Engine

The core component evaluates combinations of constraints. For a candidate subset (S\subseteq C), the framework estimates an aggregate impact:

[

R(S)=\sum_{i\in S}w_i s_i+\sum_{(i,j)\in E}\lambda_{ij}s_is_j

]

where (w_i) represents individual importance and (\lambda_{ij}) represents interaction strength.

The second term is critical because it captures synergistic effects. Two moderate constraints may jointly produce a substantial system-level problem even when neither is individually dominant.

An LLM is used to interpret semantic relationships and generate candidate reasoning paths, while structured mathematical scoring constrains the optimization process. This hybrid design reduces dependence on free-form model output.

3.6 Optimization Layer

After identifying high-impact combinations, CombiScale generates candidate interventions (A={a_1,a_2,\ldots,a_m}). Each action is evaluated against multiple objectives:

[

\max_a F(a)=\alpha P(a)-\beta C(a)-\gamma R(a)+\delta Q(a)

]

where (P) denotes expected performance improvement, (C) implementation cost, (R) residual risk, and (Q) solution quality or robustness.

The coefficients allow organizations to specify priorities. For example, a safety-critical environment may assign greater importance to residual risk, whereas a resource-constrained IoT deployment may emphasize energy efficiency.

This principle is consistent with the resource-allocation perspective demonstrated by Almagrabi (2020) and the resource-constrained sensing context examined by Chaudhri et al. (2022).

3.7 Explainability Layer

Each recommendation should include three forms of explanation: the triggering constraints, the dependency pathway, and the expected optimization effect.

For example:

Constraint A increases workload on Component B; Component B is coupled with Component C; therefore, optimizing B alone may transfer rather than remove the bottleneck.

This explanation structure transforms an optimization result into an auditable decision. Explainability is particularly important because optimization decisions may involve trade-offs that cannot be adequately represented by a single numerical score.

3.8 Scalability Strategy

The framework uses hierarchical reasoning to reduce computational complexity. Instead of evaluating every possible combination of (n) constraints, it first identifies high-risk candidates and then evaluates combinations within relevant neighborhoods of the dependency graph.

A practical workflow is therefore:

Input → Constraint Extraction → Normalization → Dependency Graph → Candidate Selection → Combinatorial Reasoning → Optimization → Explanation → Decision

This architecture enables CombiScale to preserve detailed reasoning while controlling the combinatorial explosion associated with large constraint sets.

RESULTS

The framework-level analysis produces several principal findings. First, scalable constraint reasoning is more appropriately modeled as a relational problem than as independent classification. The supplied literature repeatedly demonstrates that system outcomes emerge from interacting variables. Sensor-array research combines multiple measurements for pattern recognition (Alizadeh & Soltani, 2016), while wireless sensor optimization considers network-level energy distribution rather than isolated nodes (Almagrabi, 2020). These principles support CombiScale's dependency-oriented representation.

Second, the analysis indicates that an LLM should not operate as the sole optimization mechanism. Language models are well suited to semantic interpretation, contextual association, and generation of candidate explanations, whereas graph and mathematical components provide more explicit control over dependency propagation and objective optimization. The combinatorial architecture of ScalePulse provides a relevant foundation for this hybrid approach (Ramamurthy, Bellamkonda and Amanmadov, 2026).

Third, constraint interactions represent a major source of optimization complexity. A system can contain individually moderate constraints whose combination creates substantial performance degradation. CombiScale addresses this through interaction terms in the constraint-risk function and through dependency-graph traversal. Consequently, the framework distinguishes between isolated risk and propagated risk.

Fourth, resource constraints should be treated as optimization dimensions rather than merely operational limitations. This finding is supported by research on resource-constrained 6G-IoT sensor systems (Chaudhri et al., 2022) and fair energy allocation in wireless rechargeable networks (Almagrabi, 2020). In CombiScale, computational cost, energy consumption, latency, and implementation complexity can therefore become explicit optimization objectives.

Fifth, heterogeneous domains can be represented through a common constraint abstraction. Environmental studies involving multiple pollutants and their health consequences demonstrate the analytical value of multidimensional representations (Chen et al., 2007; Balakrishnan et al., 2019). Similarly, CombiScale can transform different technical observations into normalized constraint objects while retaining domain-specific metadata.

Finally, the framework suggests that explainability should be integrated into the reasoning process rather than appended after optimization. The ScalePulse paradigm

demonstrates the value of combining scalability analysis with interpretable system feedback (Ramamurthy, Bellamkonda and Amanmadov, 2026). CombiScale extends this principle by requiring every optimization recommendation to identify the constraints, dependencies, objective trade-offs, and expected consequences underlying the decision.

These findings are conceptual rather than empirical performance claims. Actual precision, latency, optimization quality, and generalization should be established through controlled experiments involving representative datasets and operational workloads.

DISCUSSION

CombiScale contributes a system-level perspective to LLM-based optimization by distinguishing constraint detection from constraint reasoning. Detection answers whether a problem exists, whereas reasoning evaluates why it exists, how it interacts with other constraints, and which intervention provides the most favorable outcome. This distinction is particularly important for scalable systems because optimization actions can relocate bottlenecks rather than eliminate them.

The framework's principal theoretical contribution is the integration of semantic and structural reasoning. LLMs provide contextual interpretation, while dependency graphs represent explicit relationships. This combination is conceptually aligned with ScalePulse, whose architecture jointly analyzes code and architectural dependencies for scalability assessment (Ramamurthy, Bellamkonda and Amanmadov, 2026). CombiScale extends the same principle toward multi-objective optimization.

A second contribution concerns resource-aware reasoning. Almagrabi (2020) demonstrates that resource allocation can affect network-wide operational continuity, while Chaudhri et al. (2022) address optimization under resource-constrained IoT conditions. These studies imply that optimization cannot be separated from resource availability. In CombiScale, resource limitations therefore become explicit variables in the decision function rather than hidden implementation constraints.

The framework also offers a useful interpretation of heterogeneous sensing and monitoring literature. Alizadeh and Soltani (2016) and Chiu and Tang (2013) show the value of combining multiple sensing inputs for recognition. Cao et al. (2022) further demonstrate that underlying material characteristics can influence sensing performance. Although these studies address physical sensing rather than LLM optimization, their analytical principle transfers effectively: reliable system decisions require contextual interpretation of multiple signals.

There are, however, important limitations. First, LLM reasoning is probabilistic and can produce inconsistent interpretations of identical constraints. Second, graph construction depends on the quality of architectural and operational information. Incomplete dependency information can result in incorrect propagation paths. Third, combinatorial optimization can become computationally expensive as the number of constraints increases. Hierarchical candidate selection reduces this problem but does not eliminate it.

A further limitation concerns objective specification. No optimization function can be considered universally optimal because different organizations prioritize performance, cost, reliability, energy, latency, and fairness differently. This issue is especially evident when considering environmental and societal consequences associated with air pollution (Balakrishnan et al., 2019; Chakraborty & Basu, 2021). Thus, CombiScale should expose optimization weights and constraints to domain experts rather than automatically determining organizational priorities.

The framework consequently should be understood as a decision-support architecture rather than an autonomous authority. Human validation remains important for high-impact optimization decisions, particularly when model uncertainty, incomplete data, or competing objectives are present.

CONCLUSION

This paper proposed CombiScale, a large language model framework for scalable constraint reasoning and optimization. The framework addresses a central limitation of isolated constraint analysis by modeling constraints as interacting components within a dependency-aware system. Its architecture integrates semantic extraction, normalization, graph construction, combinatorial reasoning, multi-objective optimization, and explainable recommendation.

The conceptual foundation draws particularly on the combinatorial LLM perspective demonstrated by ScalePulse (Ramamurthy, Bellamkonda and Amanmadov, 2026), while principles from sensor recognition, resource allocation, IoT optimization, and environmental analysis support the broader argument for multidimensional constraint reasoning. The resulting architecture emphasizes that scalability should be evaluated through

relationships among constraints rather than through isolated threshold violations.

The principal contribution is therefore a unified reasoning model in which an LLM interprets heterogeneous evidence while structured graph and optimization mechanisms control dependency analysis and decision selection. Nevertheless, empirical validation is required before claims regarding accuracy, computational efficiency, or optimization gains can be established. Future research should evaluate CombiScale using large-scale software systems, resource-constrained IoT environments, and heterogeneous operational datasets. Additional work should investigate uncertainty-aware reasoning, adaptive optimization weights, real-time telemetry integration, and efficient inference mechanisms.

Overall, CombiScale provides a research direction for transforming LLMs from passive analytical assistants into structured reasoning components for scalable, explainable, and constraint-aware optimization.

REFERENCES

1. Alizadeh, T., & Soltani, L. H. (2016). Reduced graphene oxide-based gas sensor array for pattern recognition of DMMPvapor.Sensors and Actuators B: Chemical,234, 361–370.
2. Almagrabi, A. O. (2020). Fair energy division scheme topermanentize the network operation for wireless rechargeablesensor networks.IEEE Access, 8,178063–178072.
3. Balakrishnan, K., Dey, S., Gupta, T., Dhaliwal, R. S., Brauer, M.,Cohen, A. J., & Dandona, L. (2019). The impact of airpollution on deaths, disease burden, and life expectancyacross the states of India: The global burden of disease study2017. The Lancet Planetary Health,3(1), e26–e39.
4. Cao, S., Sui, N., Zhang, P., Zhou, T., Tu, J., & Zhang, T. (2022). TiO2nanostructures with different crystal phases for sensitive acetonegas sensors.Journal of Colloid and Interface Science,607,357–366.
5. Chakraborty, J., & Basu, P. (2021). Air quality and environmentalinjustice in India: Connecting particulate pollution to socialdisadvantages.International Journal of EnvironmentalResearch and Public Health,18(1), 304.
6. Chaudhri, S. N., Rajput, N. S., Alsamhi, S. H., Shvetsov, A. V., &Almalki, F. A. (2022). Zero-padding and spatial augmentation-based gas sensor node optimization approach in resource-constrained 6G-IoT paradigm. Sensors,22(8), 3039.
7. Chen, T. M., Kuschner, W. G., Gokhale, J., & Shofer, S. (2007).Outdoor air pollution: nitrogen dioxide, sulfur dioxide, andcarbon monoxide health effects. The American journal of themedical sciences,333(4), 249–256.
8. Chiu, S. W., & Tang, K. T. (2013). Towards a chemiresistivesensor-integrated electronic nose: a review. Sensors,13(10),14214–14247.
9. 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
10. Geo Philip, Paulson, Artificial Intelligence and Machine Learning Applications in Project Schedule Forecasting: A Predictive Framework for Time-Control in Complex Building Projects (April 16, 2026). Available at SSRN: https://ssrn.com/abstract=6588119 or http://dx.doi.org/10.2139/ssrn.6588119