International Journal of Next-Generation Engineering and Technology (IJNGET)
eISSN: 3087-4319
Volume03 Issue08 | Page No: - 75-80

Received: 30 May 2026, Revised: 19 June 2026, Accepted: 23 July 2026, Published: 19 August 2026

DOI: https://doi.org/10.55640/ijnget-v03i08-07

CombiScale: A Large Language Model Framework for Scalable Combinatorial Constraint Solving

Rizky Pratama [1], **Siti Rahmawati** [2]

[1] Department of Artificial Intelligence and Informatics Institute of Digital Technology, Jakarta, Indonesia

[2] Department of Computer Science and Intelligent Systems National Technology University, Bandung, Indonesia

ABSTRACT: Combinatorial constraint solving requires the systematic identification of feasible configurations within large and often interdependent search spaces. As the number of variables, constraints, and contextual dependencies increases, conventional statistical and computational approaches may experience substantial difficulties in maintaining scalable reasoning, adapting to changing parameters, and integrating heterogeneous information. This paper proposes CombiScale, a conceptual Large Language Model (LLM) framework for scalable combinatorial constraint solving that combines language-based reasoning, structured constraint representation, heuristic search, causal feature analysis, and iterative solution verification. The framework is theoretically positioned at the intersection of constraint reasoning, data-driven prediction, and computational knowledge discovery. Evidence from the provided literature demonstrates the relevance of neural forecasting, deep learning, heuristic dynamic programming, causal discovery, and ecological informatics to the development of scalable decision architectures. In particular, ecological informatics emphasizes systematic data management and knowledge discovery, providing an important conceptual foundation for organizing the information required by complex reasoning systems (Recknagel & Michener, 2017). CombiScale therefore introduces a layered architecture in which natural-language problem interpretation is separated from constraint extraction, candidate generation, feasibility evaluation, and optimization. The proposed design is intended to improve scalability, interpretability, and adaptability while recognizing the limitations of LLM-based reasoning, including hallucination, computational overhead, and imperfect constraint consistency. The resulting framework provides a research-oriented foundation for integrating LLM capabilities with deterministic and heuristic constraint-solving mechanisms.

KEYWORDS: Large Language Models, Combinatorial Optimization, Constraint Solving, Scalable Reasoning, Heuristic Search, Causal Discovery, Knowledge Discovery, Deep Learning.

1. Introduction

1.1 Background

Combinatorial constraint solving concerns problems in which a solution must satisfy multiple conditions simultaneously while selecting among a potentially enormous number of alternatives. Such problems occur whenever variables are interdependent and feasible decisions are restricted by logical, numerical, temporal, or resource-related requirements. As problem dimensionality increases, exhaustive enumeration becomes computationally impractical, creating a need for intelligent mechanisms capable of reducing the search space while preserving solution quality.

The literature supplied for this study demonstrates several approaches relevant to this challenge. Adhikari, Agrawal, and Kant examined neural networks alongside traditional statistical models for seasonal time-series forecasting, demonstrating the broader value of computational learning for complex predictive relationships (Adhikari et al., 2013). Huo et al. investigated artificial neural networks for eutrophication prediction, while Lee and Lee applied deep learning to harmful algal bloom prediction, illustrating how learned representations can capture nonlinear relationships in environmental systems (Huo et al., 2013; Lee & Lee, 2018). Zhang et al. further demonstrated the use of action-dependent heuristic dynamic programming for prediction with time-varying parameters, providing a useful conceptual connection between adaptive learning and sequential decision processes (Zhang et al., 2020).

For scalable constraint solving, however, prediction alone is insufficient. A solver must interpret requirements, identify conflicts, construct candidate solutions, evaluate feasibility, and iteratively improve the result. Ecological informatics provides a particularly

relevant conceptual foundation because data management and knowledge discovery require systematic organization of information before meaningful analytical reasoning can occur (Recknagel & Michener, 2017). CombiScale extends this principle into an LLM-oriented constraint-solving architecture.

1.2 Problem Statement

Traditional computational approaches frequently require explicitly defined variables, constraints, objective functions, and search procedures. This creates a practical limitation when problem descriptions are expressed through natural language, incomplete specifications, heterogeneous documents, or changing operational conditions. A human decision-maker may describe a problem using qualitative and quantitative statements simultaneously, whereas a conventional solver generally requires formalized input.

The central problem addressed by CombiScale is therefore the development of a scalable architecture that can translate natural-language requirements into structured constraints while retaining a reliable computational mechanism for validating and optimizing candidate solutions. The framework must distinguish between linguistic reasoning and mathematical feasibility rather than assuming that an LLM alone can guarantee correctness.

1.3 Research Relevance

The relevance of CombiScale arises from the convergence of LLM-based reasoning and established computational optimization techniques. Neural and deep-learning approaches demonstrate the ability to model nonlinear p atterns (Adhikari et al., 2013; Huo et al., 2013; Lee & Lee, 2018), whereas heuristic and knowledge-discovery approaches provide mechanisms for structured decision-making and information organization (Zhang et al., 2020; Recknagel & Michener, 2017). Integrating these capabilities offers a potential pathway toward more flexible constraint-solving systems.

1.4 Objectives

The primary objective is to develop a conceptual architecture for scalable combinatorial constraint solving using an LLM as a semantic reasoning layer. The framework specifically aims to formalize natural-language requirements, reduce irrelevant search dimensions, generate candidate solutions, verify constraint satisfaction, and support iterative optimization.

1.5 Scope and Significance

The proposed framework is applicable to scheduling, resource allocation, configuration, planning, forecasting-related decision support, and other domains characterized by interacting constraints. Its significance lies not in replacing deterministic optimization algorithms, but in providing an intelligent interface and reasoning layer that can make complex constraint problems more accessible while maintaining computational verification.

2. Literature Review

The provided literature presents complementary perspectives on computational prediction, environmental modeling, heuristic reasoning, and knowledge discovery. Adhikari et al. (2013) compared particle-swarm-optimization-based neural networks with traditional statistical approaches, indicating the potential advantages of adaptive computational models when relationships within temporal data are complex. Their work is relevant to CombiScale because scalable constraint problems may similarly contain nonlinear relationships among variables.

Huo et al. (2013) demonstrated the application of artificial neural networks to eutrophication prediction. Their work indicates how learned models can represent relationships that may be difficult to formulate through simple deterministic rules. Similarly, Lee and Lee (2018) showed the applicability of deep learning to harmful algal bloom prediction, reinforcing the importance of learned representations for complex environmental prediction.

McCormick and Cairns (1994) approached algae as indicators of environmental change, while Radmer (1996) examined algal diversity and commercial algal products. These studies illustrate how heterogeneous domain information can be interpreted through structured analytical perspectives. Yang et al. (2008) provided a mechanistic examination of water eutrophication, demonstrating the importance of domain-specific relationships and interacting causes.

Wong, Lee, and Hodgkiss (2007) proposed a model for forecasting coastal algal blooms, highlighting the role of predictive modeling in systems affected by multiple environmental variables. Zhang et al. (2020), by contrast, introduced an action-dependent heuristic dynamic programming approach incorporating time-varying parameters. This is particularly relevant to CombiScale because scalable constraint solving may require adaptive decision policies rather than static evaluation.

Sun et al. (2023) addressed causal discovery for feature selection in multivariate numerical time series. The causal perspective is valuable for constraint reasoning because not every variable contributes equally to a decision. Identifying structurally important variables can reduce unnecessary search and improve computational efficiency.

Recknagel and Michener emphasize ecological informatics as a field concerned with data management and knowledge discovery (Recknagel & Michener, 2017). This perspective is foundational to CombiScale because an LLM-based solver must first organize heterogeneous information before reasoning over constraints. Knowledge discovery provides the conceptual bridge between raw information and structured decision support.

Despite these contributions, the provided literature does not directly establish an integrated LLM-based architecture for natural-language constraint extraction, candidate generation, deterministic verification, and scalable optimization. This gap motivates CombiScale. The proposed framework combines the representational flexibility suggested by neural learning with the structured reasoning requirements of constraint solving.

Recent work has also demonstrated the application of graph-based deep learning for identifying cyber threats in cloud platforms, highlighting the ability of deep models to capture complex relationships among interconnected entities (Marri et al., 2025). This perspective is relevant to CombiScale because graph-structured representations can support the identification of dependencies and relationships within complex constraint-solving environments.

3. Methodology

3.1 Research Design

This study adopts a conceptual framework-development methodology. Rather than presenting a claim of empirical superiority without experimental evidence, the methodology derives architectural principles from the provided literature and organizes them into an LLM-assisted constraint-solving pipeline. The framework treats the LLM as a semantic reasoning component rather than an autonomous mathematical authority.

3.2 CombiScale Architecture

CombiScale consists of six interconnected layers: problem interpretation, constraint extraction, knowledge organization, candidate generation, feasibility verification, and optimization feedback.

The problem interpretation layer receives natural-language requirements. For example, a user may state that several tasks must be scheduled across limited machines, that certain tasks cannot overlap, and that priority jobs should receive earlier processing. The LLM converts this description into explicit entities, variables, objectives, and constraints.

The constraint extraction layer classifies requirements into hard and soft constraints. Hard constraints must always be satisfied, whereas soft constraints represent preferences that may be optimized. This separation is essential because a candidate solution violating a hard constraint cannot be considered feasible even if it performs well on secondary objectives.

The knowledge organization layer stores relevant variables, relationships, assumptions, and contextual information. This layer reflects the broader principle that systematic data management is necessary for meaningful knowledge discovery (Recknagel & Michener, 2017). Without such organization, an LLM may repeatedly reinterpret the same information, increasing inconsistency.

The candidate generation layer transforms structured constraints into potential solution configurations. Rather than enumerating every possibility, the LLM can propose promising combinations based on semantic patterns and previously identified restrictions. Heuristic strategies can then guide the exploration of the remaining search space.

The feasibility verification layer is deliberately separated from free-form language generation. Candidate solutions are passed to deterministic constraint checks that calculate whether every hard constraint is satisfied. This architecture reduces the risk that persuasive but mathematically invalid LLM output will be accepted.

The optimization feedback layer evaluates feasible candidates against objective functions. The resulting scores are returned to the reasoning layer, which can generate improved candidates. This creates an iterative process in which language reasoning supports search rather than replacing it.

3.3 Constraint Representation

A generic combinatorial problem can be represented as:

[

\min_{x} F(x)

]

subject to:

[

g_i(x) \leq 0,\quad i=1,\ldots,m

]

and

[

h_j(x)=0,\quad j=1,\ldots,k

]

where (x) represents decision variables, (F(x)) represents the optimization objective, and (g_i) and (h_j) represent inequality and equality constraints.

CombiScale introduces an intermediate semantic representation between natural language and this formal structure. For example, "Task A must occur before Task B" is transformed into an ordering constraint, while "Machine 1 cannot process Tasks A and C simultaneously" becomes a resource-conflict constraint.

3.4 Search-Space Reduction

Scalability depends strongly on reducing unnecessary combinations. Causal feature selection provides a useful conceptual mechanism because causal relationships can help distinguish influential variables from less relevant ones (Sun et al., 2023). CombiScale can therefore use an LLM to identify candidate dependencies and then employ structured analytical procedures to determine whether those dependencies materially affect the search.

Heuristic dynamic programming provides another relevant principle. Zhang et al. (2020) demonstrated how heuristic decision mechanisms can accommodate changing parameters. CombiScale similarly treats optimization as an iterative process in which the current state influences subsequent candidate selection.

3.5 Learning and Prediction Integration

Prediction can support constraint solving when future states influence feasible decisions. Neural and deep-learning approaches reported in the supplied literature demonstrate the usefulness of learned models for complex nonlinear systems (Adhikari et al., 2013; Huo et al., 2013; Lee & Lee, 2018). Within CombiScale, such predictive components are treated as auxiliary models rather than substitutes for constraint verification.

For example, in production scheduling, a predictive component might estimate processing duration, while the constraint engine determines whether the resulting schedule violates machine availability or precedence requirements. This separation improves conceptual reliability.

3.6 Verification and Feedback

Every generated candidate should pass through a verification stage. A candidate satisfying all hard constraints receives a feasibility status, while violations are converted into structured feedback. The LLM can then revise the candidate based on the violated constraints.

This architecture produces a closed loop:

Natural-language problem → semantic parsing → formal constraints → candidate generation → verification → objective evaluation → feedback → improved candidate.

The separation of semantic reasoning from deterministic checking is the central methodological principle of CombiScale.

4. Results

The conceptual analysis indicates that CombiScale can be organized around three principal capabilities: semantic formalization, scalable search coordination, and independent solution verification. These capabilities address different weaknesses that arise when LLMs or conventional computational approaches are used independently.

First, semantic formalization provides an interface between human problem descriptions and structured computational models. Conventional optimization systems generally perform efficiently when variables and constraints have already been specified, but their usability can decline when requirements are distributed across natural-language descriptions. CombiScale addresses this limitation by using the LLM to identify entities, relationships, priorities, and restrictions before formal solving begins. This capability is conceptually consistent with the broader data-management and knowledge-discovery perspective of ecological informatics (Recknagel & Michener, 2017).

Second, the proposed architecture provides a mechanism for search-space reduction. The literature demonstrates several computational approaches capable of identifying patterns or useful decision structures. Neural networks can represent nonlinear relationships (Adhikari et al., 2013; Huo et al., 2013), deep learning can model complex predictive dependencies (Lee & Lee, 2018), and causal discovery can assist with feature selection (Sun et al., 2023). Within CombiScale, these capabilities can be used selectively to determine which variables or candidate configurations deserve further computational attention.

Third, independent verification represents a critical architectural finding. An LLM may produce a logically plausible solution that nevertheless violates a numerical, temporal, or resource constraint. CombiScale therefore does not define linguistic confidence as solution validity. Instead, candidate solutions are evaluated by explicit constraint functions. This separation creates a verification boundary between probabilistic generation and deterministic feasibility.

The literature also suggests that adaptive mechanisms are important when system parameters change. Zhang et

al. (2020) demonstrated an approach involving heuristic dynamic programming and time-varying parameters, while Wong et al. (2007) examined forecasting in a complex coastal environment. These findings support the use of iterative candidate generation in CombiScale rather than assuming that one static reasoning pass is sufficient.

The framework consequently produces a layered decision process rather than a single monolithic LLM operation. Its expected benefits include improved natural-language accessibility, modularity, support for heterogeneous constraints, and the ability to combine learned predictions with formal feasibility checks. However, these are architectural findings rather than experimentally validated performance claims. Actual improvements in computational cost, optimality, or accuracy require benchmarking against established solvers and controlled datasets.

5. Discussion

The primary theoretical contribution of CombiScale is the integration of probabilistic language reasoning with formal constraint verification. The literature supplied suggests that computational intelligence is particularly useful when systems contain nonlinear relationships, changing parameters, and large information spaces. Neural-network-based approaches have demonstrated predictive capabilities (Adhikari et al., 2013; Huo et al., 2013), while deep-learning approaches provide additional representation capacity (Lee & Lee, 2018). However, prediction and constraint satisfaction are fundamentally different tasks. A model can predict a likely outcome without guaranteeing that the resulting decision is feasible.

CombiScale addresses this distinction through architectural modularity. The LLM performs semantic interpretation and candidate reasoning, while formal constraint mechanisms determine feasibility. This creates an explicit division between "what might be a good solution" and "whether the solution satisfies the stated rules." Such separation is particularly important for combinatorial problems because a single violated hard constraint may invalidate an otherwise high-quality candidate.

The framework also extends the knowledge-discovery perspective described by Recknagel and Michener (2017). In a conventional information-management setting, knowledge discovery involves transforming organized data into useful insights. In CombiScale, the same conceptual progression becomes an input-to-decision pipeline: unstructured requirements are transformed into structured representations, relevant relationships are identified, candidate configurations are generated, and verified solutions are returned. Thus, knowledge organization is not treated as a peripheral preprocessing activity; it becomes an essential component of scalable reasoning.

The use of causal discovery also introduces an important opportunity and limitation. Sun et al. (2023) show the relevance of causal discovery to feature selection, suggesting that identifying meaningful dependencies can reduce irrelevant dimensions. However, causal relationships cannot simply be inferred from LLM-generated associations. CombiScale must therefore treat LLM-generated causal hypotheses as candidates requiring analytical validation.

The heuristic dimension creates another trade-off. Zhang et al. (2020) demonstrate the relevance of heuristic dynamic decision mechanisms in changing environments. Heuristics can substantially reduce search effort, but they do not necessarily guarantee global optimality. Consequently, CombiScale should be viewed as a coordination architecture capable of combining heuristic search with exact verification rather than as an inherently optimal solver.

Practical implementation also faces computational costs. Repeated LLM calls may introduce latency and resource consumption, especially when the search space requires thousands of candidate evaluations. A scalable implementation would therefore need caching, structured intermediate representations, candidate pruning, and selective invocation of the LLM. The architecture may also encounter hallucinated constraints, omitted requirements, or inconsistent interpretations. These limitations reinforce the importance of formal validation.

Overall, CombiScale should be understood as a hybrid architecture rather than an LLM-only optimization system. Its strongest contribution is the coordination of semantic flexibility with computational rigor. Future empirical studies should evaluate whether this architecture can reduce search complexity while maintaining feasibility and solution quality across benchmark combinatorial problems.

6. Conclusion

This paper introduced CombiScale, a conceptual Large Language Model framework for scalable combinatorial constraint solving. The framework integrates natural-language interpretation, structured constraint extraction, knowledge organization, candidate generation, feasibility verification, and iterative optimization. Its design is informed by the provided literature on neural prediction, deep learning, heuristic decision-making, causal discovery, environmental modeling, and ecological informatics.

The central contribution is the separation of probabilistic reasoning from deterministic constraint

validation. Rather than treating an LLM as a complete solver, CombiScale positions it as an intelligent coordination and semantic reasoning layer connected to formal computational mechanisms. This design can potentially improve accessibility to complex optimization problems while preserving explicit feasibility checks.

The theoretical foundation also emphasizes that scalable reasoning depends on effective organization and discovery of relevant information. The knowledge-management perspective of Recknagel and Michener (2017) is therefore directly relevant to the proposed architecture. By transforming unstructured requirements into structured decision representations, CombiScale establishes a pathway from human-oriented problem descriptions to computationally verifiable solutions.

Nevertheless, the framework remains conceptual. Its effectiveness, computational efficiency, robustness, and optimality require empirical validation using standardized combinatorial benchmarks. Future research should investigate hybrid LLM-exact solver architectures, automated constraint verification, adaptive search-space reduction, uncertainty-aware reasoning, and comparative evaluation against conventional optimization methods. Such studies would determine whether CombiScale can provide measurable scalability advantages while maintaining the reliability expected from formal constraint-solving systems.

References

1. Adhikari R, Agrawal RK, Kant L. PSO based neural networks vs. traditional statistical models for seasonal time series forecasting. In: Proceedings of the 2013 3rd I\EEE International Advance Computing Conference (IACC); 22 – 23 February 2013; Ghaziabad, India. pp. 719 – 725.

2. Anderson DM. Approaches to monitoring, control and management of harmful algal blooms (HABs). Ocean & Coastal Management 2009; 52 (7): 342 – 347.

3. Huo S, He Z, Su J, et al. Using artificial neural network models for eutrophication prediction. Procedia Environmental Sciences 2013; 18: 310 – 316.

4. Lee S, Lee D. Improved prediction of harmful algal blooms in four major South Korea's rivers using deep learning models. International Journal of Environmental Research and Public Health 2018; 15 (7): 1 – 15.

5. McCormick PV, Cairns J. Algae as indicators of environmental change. Journal of Applied Phycology 1994; 6(5 – 6): 509 – 526.

6. Radmer RJ. Algal diversity and commercial algal products. BioScience 1996; 46(4): 263 – 270.

7. Recknagel F, Michener WK. Ecological Informatics: Data Management and Knowledge Discovery. Springer; 2017.

8. Sun Y, Li J, Liu J, et al. Using causal discovery for feature selection in multivariate numerical time series. Computing and Artificial Intelligence 2023; 1 (1): 100. 21 Machine Learning 2015; 101(1 – 3): 377 – 395.

9. Wong KTM, Lee JHW, Hodgkiss IJ. A simple model for forecast of coastal algal blooms. Estuarine, Coastal and Shelf Science 2007; 74(1 – 2): 175 – 196.

10. Yang X, Wu X, Hao H, He Z. Mechanisms and assessment of water eutrophication. Journal of Zhejiang University SCIENCE B 2008; 9 (3): 197 – 209.

11. Zhang H, Hu B, Wang X, et al. An action dependent heuristic dynamic programming approach for algal bloom prediction with time-varying parameters. IEEE Access 2020; 8: 26235 – 26246.

12. 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

13. M. R. Marri, S. B. Kurada, S. Gupta, S. Dey, S. Kumar and R. Chauhan, "Graph-Based Deep Learning Model for Identifying Cyber Threats in Cloud Platforms," 2025 International Conference on Computational Intelligence, Security, and Artificial Intelligence (IntelliSecAI), Al-Khobar, Saudi Arabia, 2025, pp. 1-7, doi: 10.1109/IntelliSecAI66368.2025.11472831.