A Survey of Reinforcement Learning Combinatorial Optimization and Graph Neural Networks

www.surveyx.cn

Abstract

This survey provides a comprehensive overview of advanced computational techniques, specifically focusing on the integration of Reinforcement Learning (RL), Combinatorial Optimization (CO), and Graph Neural Networks (GNNs), aimed at addressing NP-hard problems and complex decision-making tasks. The paper explores the synthesis of these methodologies to develop solutions for intricate challenges such as scheduling and network interdiction, characterized by vague constraints and conflicting criteria. The survey highlights the convergence of RL with deep learning approaches to enhance algorithmic capabilities, particularly in multi-agent systems, and examines the role of constrained combinatorial optimization in identifying feasible solutions under probabilistic constraints. By investigating the integration of RL, CO, and GNNs, the survey uncovers innovative methodologies that leverage deep RL and learning-to-rank techniques, overcoming limitations of existing heuristic approaches. The significance of these techniques is underscored in real-world applications, including multi-objective optimization problems requiring simultaneous optimization of multiple objectives. The survey aims to bridge knowledge gaps and contribute to the development of efficient solutions for complex optimization tasks, providing insights into the core elements, mechanisms, and applications of deep reinforcement learning. Through a structured exploration of these techniques, the survey presents a coherent narrative that enhances understanding of their complexities and interrelations, ultimately advancing the field of computational optimization.

1 Introduction

1.1 Purpose and Scope

This survey provides an overview of advanced computational techniques, focusing on the integration of Reinforcement Learning (RL), Combinatorial Optimization (CO), and Graph Neural Networks (GNNs) in addressing NP-hard problems and complex decision-making tasks. The primary objective is to explore how GNNs and combinatorial optimization methodologies can yield innovative solutions for challenges such as scheduling and network interdiction, which often involve ambiguous constraints and conflicting criteria. By leveraging GNNs' inductive biases for relational input encoding and employing multi-task approaches to exploit inter-task similarities, this research aims to enhance problem-solving efficiency in scenarios where traditional exact solvers are inadequate, leading to improvements in computational speed and solution quality [1, 2, 3, 4].

The survey also examines the integration of RL with deep learning techniques to improve algorithmic capabilities, particularly in addressing sample efficiency and scalability challenges in multi-agent systems (MAS) [5]. Additionally, it investigates constrained combinatorial optimization, aiming to identify feasible solutions under specific probabilistic constraints while optimizing an objective function [6].

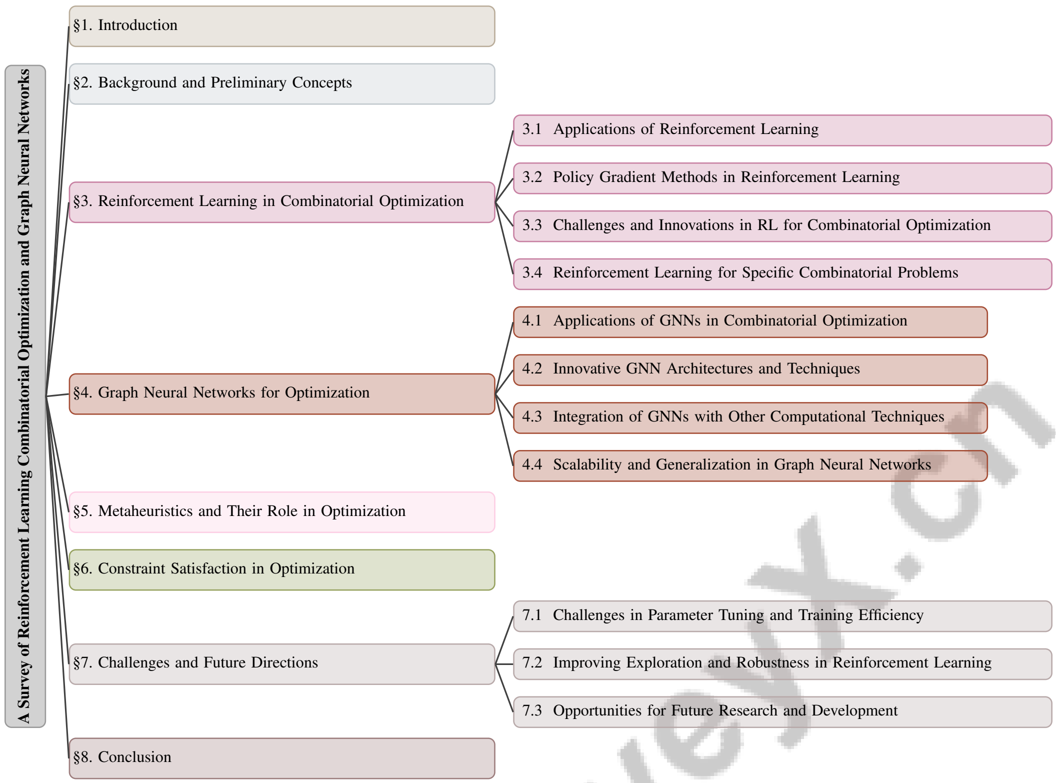


Figure 1: chapter structure

By exploring the convergence of RL, CO, and GNNs, this survey seeks to uncover methodologies that utilize deep RL and learning-to-rank techniques to surpass existing heuristic approaches. It emphasizes the relevance of these techniques in real-world applications, such as multi-objective optimization problems (MOPs), which require simultaneous optimization of multiple objectives [7]. Furthermore, it addresses knowledge gaps in optimization by synthesizing advanced computational techniques, particularly focusing on deep RL's core elements, including value functions and policy optimization, while examining its diverse applications across fields such as robotics, natural language processing, and combinatorial optimization tasks [8, 9, 10, 11].

1.2 Significance of Computational Techniques

The computational techniques discussed in this survey—Reinforcement Learning (RL), Combinatorial Optimization (CO), and Graph Neural Networks (GNNs)—are crucial for tackling the complexities of NP-hard problems. These methodologies are particularly relevant in domains like telecommunications and logistics, where sophisticated optimization strategies are required for tasks such as the Vehicle Routing Problem (VRP). The integration of machine learning with CO, especially in graph-based formulations, highlights the potential of these approaches to enhance operational efficiencies and decision-making processes [12].

Deep Reinforcement Learning (DRL) is notable for its ability to generalize across various applications, from robotics to complex multi-agent systems, thereby mitigating traditional issues such as prolonged computation times and the need for frequent re-training. DRL's capacity to facilitate multi-objective optimization is essential for addressing problems with conflicting criteria [13]. In combinatorial optimization, the development of graph representation learning methods has effectively addressed classical problems like the Traveling Salesman Problem (TSP) and Maximum Independent Set (MIS), showcasing GNNs' versatility and applicability in CO challenges [14]. The enhancement of large language model reasoning capabilities through combinatorial optimization strategies further underscores the importance of these techniques in advancing artificial intelligence, particularly in scenarios requiring minimal human intervention [1].

Additionally, the integration of RL with other computational techniques, such as the Reinforced LKH algorithm for TSP, illustrates how hybrid approaches can outperform traditional exact and heuristic algorithms, thereby expanding the boundaries of combinatorial optimization [15]. Collectively, these computational methodologies drive innovations in artificial intelligence and machine learning, enabling the resolution of large-scale, complex problems previously deemed insurmountable.

1.3 Motivation for Combined Study

The unification of Reinforcement Learning (RL), Combinatorial Optimization (CO), and Graph Neural Networks (GNNs) is motivated by the need to address the inherent complexity and NP-hardness of CO problems, where exact solutions are often impractical for large instances [14]. Traditional heuristic methods frequently struggle to generalize across diverse problem instances and adapt to varying user preferences, particularly in constrained combinatorial optimization scenarios. RL offers a robust alternative, leveraging its learning capabilities from interactions to address existing heuristics' limitations [13].

The motivation for integrating CO problems with neural networks, particularly GNNs, arises from the necessity to process matrix-style relationship data, which is crucial for many CO problems [16]. This integration aims to enhance the generalizability and efficiency of solutions while drawing inspiration from biological learning processes to improve adaptability in artificial systems. Such an approach addresses the constraints of current methodologies in managing complex decision-making tasks.

Moreover, exploring these computational techniques together is essential for overcoming the computational intractability of many combinatorial problems, which often require extensive domain knowledge for effective algorithm design [13]. By merging RL with CO, particularly through probabilistic constraints, this framework ensures decision feasibility and addresses the limitations of expected cost models while enhancing the modeling of solution dependencies. This unified approach facilitates the development of innovative solutions, advancing fields that demand precise decision-making and optimization, ultimately paving the way for breakthroughs that surpass traditional methodologies.

1.4 Structure of the Survey

This survey is structured to provide a comprehensive exploration of the intersection of Reinforcement Learning (RL), Combinatorial Optimization (CO), and Graph Neural Networks (GNNs), organized into several key sections. Initially, the survey introduces the purpose and scope of the study, highlighting the significance and motivation for examining these computational techniques in tandem. This is followed by a detailed background section that defines and explains core concepts, establishing a foundational understanding necessary for subsequent discussions.

The main body of the survey is divided into thematic sections focusing on the application and integration of these techniques. The section on Reinforcement Learning in Combinatorial Optimization delves into specific applications, policy gradient methods, challenges, and innovations in RL for CO problems. Complementing this is a section on Graph Neural Networks for Optimization, discussing applications, innovative architectures, integration with other techniques, and scalability challenges of GNNs. The survey further explores the role of Metaheuristics in optimization, including their concepts, applications, and enhancements for constraint satisfaction.

Additionally, the survey examines the role of Constraint Satisfaction in optimization, discussing decentralized models and specific applications such as shipping route design, alongside joint learning of constraints and objective functions. It concludes with a discussion of current challenges and future research directions, emphasizing parameter tuning, training efficiency, and improving exploration in RL applications. This structured approach aims to weave a cohesive narrative that enhances readers' understanding of the intricate complexities and interconnections inherent in advanced computational techniques, particularly in the context of combinatorial optimization, decision-focused learning, and graph learning applications, which are increasingly relevant across various domains such as artificial intelligence and machine learning [17, 1, 18, 19, 20].The following sections are organized as shown in Figure 1.

2 Background and Preliminary Concepts

2.1 Definitions and Core Concepts

Reinforcement Learning (RL), Combinatorial Optimization (CO), and Graph Neural Networks (GNNs) are essential methodologies for addressing complex computational challenges, each uniquely enhancing decision-making and optimization. CO seeks optimal solutions within discrete structures, tackling NP-hard problems like the Maximum k-Cut, Traveling Salesman Problem (TSP), and Multi-Objective Facility Location Problem (MO-FLP). The TSP exemplifies the need for sophisticated algorithms to manage intricate objectives and large datasets [16]. Additional graph-based CO challenges such as the Maximal Independent Set (MIS), Minimum Vertex Cover (MVC), and Maximal Clique (MC) necessitate advanced solutions [21].

In RL, agents adaptively learn from environmental interactions, which is vital for dynamic settings [13]. Despite low sample efficiency, requiring extensive interactions to reach optimal performance [22], RL's exploration and exploitation strategies make it a powerful tool for dynamic decision-making, particularly in complex multi-agent systems [7].

GNNs process graph-structured data, excelling in tasks involving relational data, such as identifying maximum independent sets and cliques in graphs [14]. Despite their promise, GNNs face challenges in scalability and generalization, especially in large-scale CO problems [3]. By exploiting relational data structures, GNNs extract crucial patterns for complex CO tasks [21].

Together, RL, CO, and GNNs offer robust tools for navigating NP-hard challenges. Their integration enables innovative strategies addressing modern optimization demands, enhancing solution generalizability and efficiency through the synthesis of learning and optimization methods [6]. This synergy incorporates insights from biological learning to improve adaptability in artificial systems [13].

2.2 Interrelation of Techniques

The synergy of RL, CO, and GNNs forms a cohesive framework for tackling complex computational challenges, particularly NP-hard problems. This integration is evident in methods utilizing graph structures to enhance optimal solution discovery, surpassing traditional heuristics through learning-based approaches [21]. The combination of RL with GNNs fosters scalable algorithms capable of processing graph-structured data, crucial for solving issues like the maximum clique and constrained minimum cut [5].

The GAT-PCM model exemplifies this integration, using directed acyclic graph representations to embed Distributed Constraint Optimization Problems (DCOPs), generating effective heuristics without full problem knowledge [22]. This highlights the potential of RL and GNNs to enhance scalability and learning efficiency in large state and action spaces.

Incorporating probabilistic constraints in optimization frameworks addresses feasibility and risk, offering a robust alternative to relying solely on expected costs [6]. This improves decision-making in CO by ensuring solutions are not only optimal but viable under uncertainty.

The Matrix Encoding Network (MatNet) demonstrates the interplay between traditional methods like Mixed Integer Programming (MIP) and modern neural network architectures for matrix-style data, broadening these techniques' applicability [16]. Modeling MO-FLP as bipartite graph optimization tasks and employing GNNs for implicit graph representations further illustrates these methodologies' integration in complex optimization problems [23].

Categorizing research into graph representation learning and CO problem-solving distinguishes between graph embedding and end-to-end learning methods, systematically deriving objectives and optimization processes to enhance solution quality and efficiency [14].

The integration of RL, CO, and GNNs enhances scalability, interpretability, and dynamic graph structure management, advancing computational optimization strategies. This interrelation fosters innovative solutions beyond traditional methodologies, providing novel insights into complex optimization tasks [13].

Category	Feature	Method
Applications of Reinforcement Learning	Optimization Strategies	MMILP-GNN[3]
Policy Gradient Methods in Reinforcement Learning	Policy and Parameter Optimization	RL-VRP[24]
	Hybrid Integration Approaches	PQN[25]
	Reward Adjustment Techniques	R2[26], NCO[27]
	Cooperative and Multi-Agent Strategies	MARL[28]
Challenges and Innovations in RL for Combinatorial Optimization	Graph-Based Strategies	HGRL[5], TCGRE[29], VSR-LKH[15], GCN-GTS[21]
	Parameter and Transfer Techniques	RLRD[30], DRL-MOA[7]
	Matrix and Encoding Methods	OTRL[31]
Reinforcement Learning for Specific Combinatorial Problems	Matrix-Based Decision-Making	MatNet[16]

Table 1: This table provides a comprehensive summary of the various methods employed in reinforcement learning (RL) for combinatorial optimization. It categorizes these methods into applications of RL, policy gradient methods, challenges, innovations, and specific problems, highlighting the diversity and adaptability of RL techniques in addressing complex optimization tasks.

3 Reinforcement Learning in Combinatorial Optimization

Reinforcement Learning (RL) significantly influences combinatorial optimization through its versatile applications and methodologies. This section examines RL's adaptability and efficacy in addressing complex optimization challenges. Table 1 presents a detailed classification of reinforcement learning methods applied to combinatorial optimization, illustrating the breadth of strategies and innovations in this domain. As illustrated in Figure 2, the hierarchical structure of RL applications in combinatorial optimization highlights key areas such as dynamic environment strategies, network interdiction, policy gradient methods, challenges, innovations, and specific problems. This figure underscores RL's capacity to navigate and solve intricate optimization issues, reinforcing the discussion of its impact in this field.

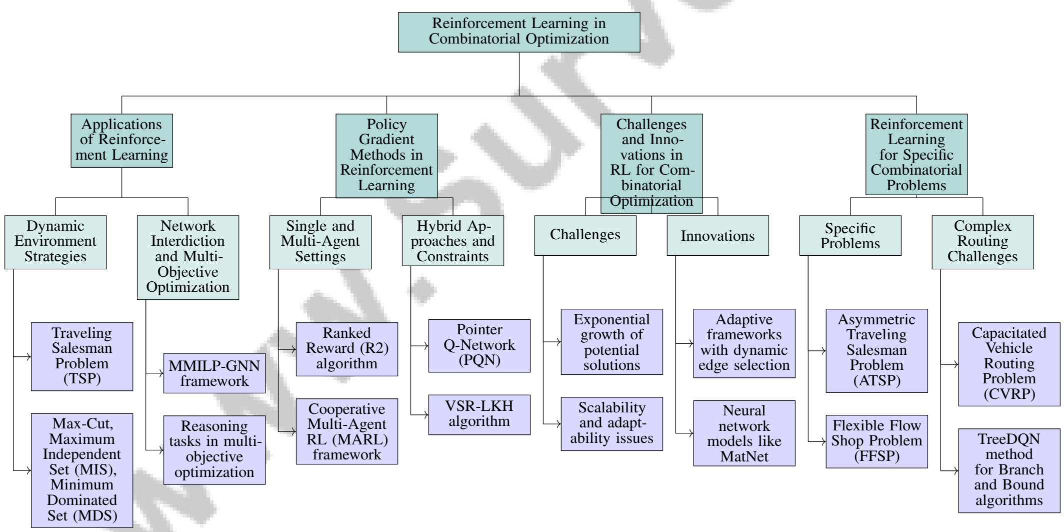


Figure 2: This figure illustrates the hierarchical structure of Reinforcement Learning (RL) applications in combinatorial optimization, highlighting key areas such as dynamic environment strategies, network interdiction, policy gradient methods, challenges, innovations, and specific problems. It emphasizes RL's adaptability and efficacy in addressing complex optimization challenges.

3.1 Applications of Reinforcement Learning

RL excels in combinatorial optimization by developing adaptive strategies in dynamic environments. In the Traveling Salesman Problem (TSP), RL techniques, such as the DRL-MOA method, surpass traditional heuristics by optimizing routes under multiple objectives [7]. RL's integration with graph-based methods extends its applicability to Max-Cut, Maximum Independent Set (MIS), and Minimum Dominated Set (MDS), leveraging machine learning to enhance solutions beyond handcrafted algorithms [14]. The GCN-GTS method exemplifies this by effectively exploring solution spaces and predicting optimal vertex participation [21].

In network interdiction, RL combined with neural networks, as seen in the MMILP-GNN framework, outperforms traditional solvers by enhancing the solution process [3]. Additionally, RL enhances reasoning in multi-objective optimization, outperforming conventional methods in reasoning tasks [13].

RL's adaptability to dynamic scenarios refines solutions across logistics, finance, and robotics, improving effectiveness in complex tasks. Techniques like learning-to-rank distillation develop efficient models for facility location and task scheduling [32, 30, 33]. By integrating RL with graph-based and neural network methods, researchers create innovative solutions that surpass traditional optimization techniques.

As illustrated in Figure 3, this figure showcases the diverse applications of reinforcement learning (RL) in solving complex problems, highlighting its use in combinatorial optimization, network interdiction, and dynamic scenarios. These examples underscore RL's versatility and efficacy in tackling combinatorial challenges [8, 11, 34].

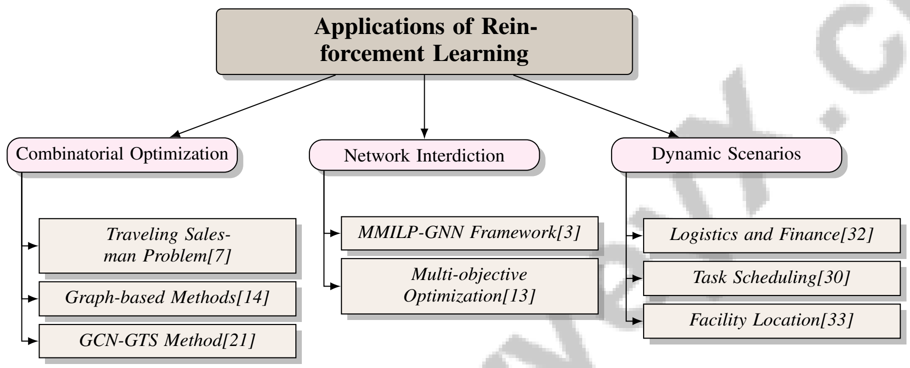


Figure 3: This figure illustrates the diverse applications of reinforcement learning (RL) in solving complex problems, showcasing its use in combinatorial optimization, network interdiction, and dynamic scenarios.

3.2 Policy Gradient Methods in Reinforcement Learning

Policy gradient methods optimize decision policies in RL, particularly for combinatorial optimization problems. These methods adjust policy parameters to maximize expected rewards, effective in continuous or large action spaces. The Ranked Reward (R2) algorithm enhances learning in single-player scenarios by reshaping rewards based on past performance [26].

In multi-agent settings, the Cooperative Multi-Agent RL (MARL) framework optimizes action plans under complex constraints, improving collective decision-making [28]. For the Capacitated Vehicle Routing Problem (CVRP), RL-CVRP formalizes the problem as a Markov Decision Process (MDP), refining policies through environmental interactions [24].

Incorporating penalties for constraint violations into the reward function, as seen in the NCO method, ensures solutions adhere to problem constraints [27]. Hybrid approaches like the Pointer Q-Network (PQN) combine Q-learning with Ptr-Nets for improved decision-making [25]. The VSR-LKH algorithm illustrates the synergy between policy gradient methods and heuristic optimization [15].

Policy gradient methods enhance RL applications by optimizing decision policies in complex environments. Techniques like Proximal Policy Optimization (PPO) facilitate efficient policy learning through environmental interaction, improving RL agents' efficiency in solving intricate tasks [35, 27, 36, 34].

3.3 Challenges and Innovations in RL for Combinatorial Optimization

RL faces challenges in combinatorial optimization due to problem complexity and discrete nature. The exponential growth of potential solutions in NP-hard problems like the TSP and Team Coordination Graphs Problem (TCGRE) complicates finding optimal solutions [29]. Multiple optimal solutions further complicate effective likelihood map generation [21].

Traditional methods struggle with scalability and adaptability, leading to inefficiencies in large state and action spaces. Frameworks like HGRL improve adaptability and performance through dynamic interventions [5]. Extensive iterations impose significant computational costs, making existing methods impractical for real-time applications [7].

Innovations like adaptive frameworks with dynamic edge selection mechanisms enhance efficiency and adaptability [15]. Neural network models like MatNet broaden machine learning applications to combinatorial optimization [16]. The RLRD framework offers efficient solutions for real-time applications by reducing computational intensity [30]. Leveraging historical data enhances performance across problem instances, offering robust alternatives to traditional methods [31].

These innovations demonstrate RL's potential to transcend traditional limitations through novel approaches integrating learning and optimization strategies, advancing combinatorial optimization.

3.4 Reinforcement Learning for Specific Combinatorial Problems

RL shows substantial potential in specific combinatorial optimization problems by leveraging adaptive learning capabilities. In the Asymmetric Traveling Salesman Problem (ATSP) and Flexible Flow Shop Problem (FFSP), MatNet encodes matrix relationships, enhancing decision-making [16]. In multi-objective optimization, RL predicts non-dominated solutions for the Multi-Objective Facility Location Problem (MO-FLP), reducing computational costs while maintaining performance [23].

In complex routing challenges like the CVRP, RL frameworks merge combinatorial action spaces with value-function-based methods, transforming action selection into a mixed-integer optimization problem. This approach constructs optimal routes for limited-capacity vehicles, enhancing efficiency and speed [35, 24, 37]. RL achieves near-optimal solutions with minimal gaps compared to state-of-the-art methods, providing rapid responses for unseen instances, advantageous for commercial applications.

The TreeDQN method illustrates RL's potential to enhance computational efficiency by optimizing branching heuristics in Branch and Bound (BB) algorithms, reducing training data and generating smaller BB trees [38, 39]. This method exemplifies RL's capacity to enhance optimization solutions' tractability and performance.

These advancements illustrate RL's transformative potential in specific combinatorial optimization problems. By integrating RL with computational techniques, researchers develop innovative solutions that enhance optimization processes' efficiency and effectiveness. The synergy between RL and traditional operations research methodologies fosters versatile frameworks that effectively tackle intricate optimization challenges across fields [40, 41, 42, 2, 43].

Figure 4 illustrates the application of Reinforcement Learning (RL) to specific combinatorial optimization problems, highlighting key methods and innovations, as well as challenges and future work directions. This visual representation complements the discussion by providing a concise overview of the landscape of RL applications in combinatorial optimization.

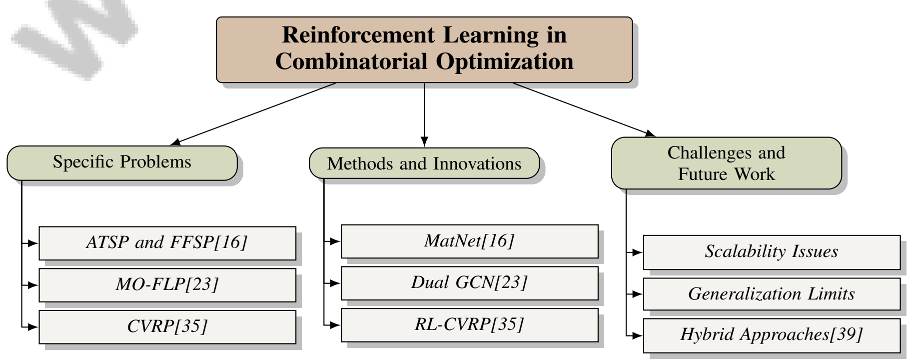


Figure 4: This figure illustrates the application of Reinforcement Learning (RL) to specific combinatorial optimization problems, highlighting key methods and innovations, as well as challenges and future work directions.

4 Graph Neural Networks for Optimization

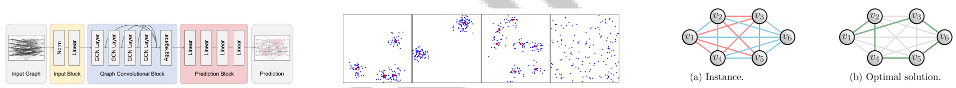
Graph Neural Networks (GNNs) have become vital in solving optimization problems involving graph-structured data. Table 4 presents a comprehensive comparison of Graph Neural Networks (GNNs) in combinatorial optimization, focusing on their applications, innovative architectures, and integration with other computational techniques. This section examines the role of GNNs in combinatorial optimization, emphasizing their ability to enhance decision-making and improve solution quality. The subsequent subsection details the diverse applications of GNNs in this field.

4.1 Applications of GNNs in Combinatorial Optimization

GNNs effectively address complex combinatorial optimization challenges by managing graph-structured data. The MatNet approach for the Asymmetric Traveling Salesman Problem (ATSP) and Flexible Flow Shop Problem (FFSP) highlights GNNs' capability to encode matrix relationships, improving decision-making through specialized neural architectures [16]. In sequential problems like the Traveling Salesman Problem (TSP), autoregressive methods leveraging data sequences outperform non-autoregressive ones, showcasing GNNs' potential in optimizing sequential decisions [14].

GNNs have been integrated with Distributed Constraint Optimization Problem (DCOP) algorithms to enhance scalability and learning efficiency. The GAT-PCM model exemplifies this by improving performance in large-scale graph data management [22]. This integration highlights GNNs' scalability advantages in combinatorial optimization.

The application of GNNs in combinatorial optimization demonstrates their transformative impact across various domains. By combining GNNs with other computational techniques, innovative solutions emerge, addressing modern optimization complexities and enhancing scalability, efficiency, and solution quality. GNNs' adaptability positions them as crucial components in evolving computational methodologies, effectively capturing relational information and improving generalization across diverse instances [44, 45, 4, 46].



(a) Graph Convolutional Network (GCN) Architecture[47]

(b) Clustering Analysis of Data Points[44]

(c) Comparison between an instance and an optimal solution in graph theory.[4]

Figure 5: Examples of Applications of GNNs in Combinatorial Optimization

As depicted in Figure 5, GNNs offer innovative solutions to complex combinatorial optimization problems. The GCN architecture demonstrates how input graphs are processed through convolutional operations to extract meaningful representations, forming the backbone of many optimization algorithms. The clustering analysis showcases GNNs' ability to categorize distinct groups within datasets, crucial for optimizing solutions by understanding data structures. Lastly, the comparison of instances and optimal solutions highlights GNNs' practical application in finding optimal configurations. Collectively, these examples illustrate the versatility and efficacy of GNNs in addressing combinatorial optimization challenges [47, 44, 4].

4.2 Innovative GNN Architectures and Techniques

Recent advancements in GNNs have introduced innovative architectures and techniques that significantly enhance their capabilities in solving complex combinatorial optimization problems. Graph embedding networks represent policies in greedy algorithms, improving generalization across problem instances and scales [48]. This adaptability allows GNNs to optimize scheduling decisions, enhancing efficiency in scheduling problems [41].

The MMILP-GNN model exemplifies integrating GNNs with traditional mathematical algorithms, enhancing compatibility and performance in network interdiction tasks [3]. This innovation underscores the potential of combining neural architectures with classical optimization techniques to address complex decision-making challenges effectively.

AI-generated, for reference only.

Method Name	Architectural Innovation	Integration with Traditional Methods	Application Versatility
S2V-DQN[48]	Graph Embedding Network	-	Various Optimization Problems
TSF[41]	New Gnn Structures	Reinforcement Learning Integration	Various Optimization Scenarios
MMILP-GNN[3]	Mmilp-GNN Model	Traditional Mathematical Algorithms	Network Interdiction Problems
GD[45]	-	-	-
RBM-EDA[49]	Generative Neural Networks	-	-
UGF-COP[46]	Unified Gnn-based Framework	Integrate With Existing	Different Optimization Scenarios
Meta-NCO[44]	Meta-learning Framework	-	Various Distributions
GNN[50]	New Gnn Structures	Classical Optimization Algorithms	Various Optimization Scenarios

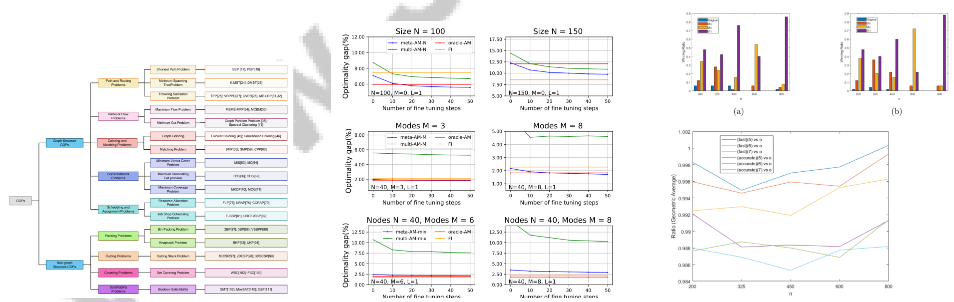
Table 2: Comparison of various GNN-based methods highlighting their architectural innovations, integration with traditional methods, and application versatility. The table provides insights into the adaptability and potential of these methods in addressing diverse optimization challenges.

GNNs have been applied to convert Multi-Objective Facility Location Problems (MO-FLP) into bipartite graphs, training Graph Convolutional Networks (GCNs) for node and edge predictions, showcasing versatility in multi-objective optimization tasks [23]. Despite these advances, GNNs face limitations due to their reliance on intricate structures, which do not always guarantee superior performance across all scenarios [45]. Continuous innovation in GNN architectures is essential for ensuring scalability and efficiency.

The RBM-EDA model illustrates how generative models can capture optimization problem structures more effectively than traditional methods, offering scalable solutions to complex challenges [49]. These innovations highlight the importance of integrating generative modeling techniques within GNN frameworks to enhance problem-solving capabilities.

Innovative GNN architectures significantly advance combinatorial optimization by providing a unified framework that captures relational information and transforms both graph-structured and non-graph-structured problems into solvable formats, demonstrating the profound influence of modern computational methodologies [4, 46]. By merging advanced neural designs with traditional optimization frameworks, researchers are developing robust solutions that tackle modern optimization complexities, enhancing scalability, efficiency, and solution quality.

Table 2 presents a comprehensive comparison of innovative GNN architectures and techniques, illustrating their integration with traditional methods and versatility in application to complex optimization problems.



(a) A Tree Diagram of Combinatorial Optimization Problems (COPs)[46]

(b) Meta-AM-N: A Novel Approach for Meta-Learning[44]

(c) Comparison of Winning Ratios for Different Number of Players in a Game[50]

Figure 6: Examples of Innovative GNN Architectures and Techniques

As shown in Figure 6, the exploration of GNNs for optimization has led to innovative architectures addressing complex combinatorial problems. The "Tree Diagram of Combinatorial Optimization Problems (COPs)" categorizes COPs into branches, distinguishing between Graph Structure and Non-graph Structure COPs, aiding in understanding GNN applications. The "Meta-AM-N: A Novel Approach for Meta-Learning" highlights performance comparisons among meta-learning algorithms, emphasizing GNNs' potential for enhancing learning efficiency. Lastly, the "Comparison of Winning Ratios for Different Number of Players in a Game" illustrates the effectiveness of strategies in dynamic environments, showcasing GNNs' adaptability. Collectively, these examples demonstrate innovation within GNN architectures, paving the way for efficient optimization solutions [46, 44, 50].

4.3 Integration of GNNs with Other Computational Techniques

Method Name	Integration Techniques	Optimization Enhancement	Adaptability and Scalability
CR[1]	Qubo Optimization	Combinatorial Optimization Approach	Diverse Domains
UPAF[51]	Max-SAT Bridge	Transferable Features Extraction	Generalizable Across Cos
SOLO[52]	Graph-based Representations	Reinforcement Learning Integration	Scalable And Generic
CKRF[53]	Pointer Network Integration	Deep Reinforcement Learning	Dynamic Keyword Selection
dNN[54]	Dataless Neural Networks	Eliminating Training Data	Broader Array Applications

Table 3: Summary of integration techniques, optimization enhancements, and adaptability features of various combinatorial optimization methods utilizing Graph Neural Networks (GNNs). The table outlines the methodologies employed, highlighting their unique contributions to optimization processes and scalability across diverse domains.

Integrating GNNs with other computational techniques has transformed approaches to complex combinatorial optimization problems. Combining GNNs with quantum annealing methods leverages quantum tunneling alongside GNNs' representation learning capabilities to efficiently explore solution spaces and discover near-optimal paths [55].

In reasoning tasks, integrating combinatorial optimization with large language models (LLMs) showcases GNNs' application in automating prompt generation, enhancing reasoning efficiency by optimizing reason selection for improved decision-making [1]. A unified pre-training and adaptation framework illustrates GNNs' ability to learn generalizable and transferable information, applicable to combinatorial optimizations on graphs, using Max-SAT as a bridge to enhance adaptability [51].

The synergy between GNNs and reinforcement learning, as demonstrated in the SOLO framework, enhances optimization processes by leveraging the strengths of each methodology, improving problem-solving capabilities in complex environments [52]. Additionally, combining pointer networks with deep reinforcement learning expands GNN applications for dynamic keyword selection [53].

Furthermore, integrating GNNs with differentiable neural network (dNN) methods presents a promising approach for solving NP-hard problems. By minimizing a loss function based on graph structure, this method enhances the tractability and scalability of complex optimization tasks, offering robust solutions to traditionally intractable challenges [54].

The integration of GNNs with various computational techniques has shown significant potential in enhancing combinatorial optimization and graph learning tasks, effectively encoding relational information, improving problem-solving capabilities, and providing scalable solutions across diverse application domains [45, 4, 46, 56, 20]. By leveraging diverse methodologies, researchers continue to develop innovative solutions that enhance the adaptability, efficiency, and problem-solving capabilities of modern optimization frameworks. Table 3 provides a comprehensive overview of the integration methodologies, optimization enhancements, and adaptability features of various approaches that combine Graph Neural Networks (GNNs) with other computational techniques to solve complex combinatorial optimization problems.

4.4 Scalability and Generalization in Graph Neural Networks

GNNs excel in processing graph-structured data but face challenges in scalability and generalization for large-scale combinatorial optimization problems. The computational overhead of extensive graphs can hinder GNNs' practical applications [3]. As graph size increases, model complexity grows, leading to longer training times and higher resource consumption, especially in NP-hard problems where graph size can exponentially increase with complexity [21].

Another critical challenge is GNNs' generalization ability across different problem instances and domains. While adept at capturing local graph structures, GNNs struggle to generalize to unseen data or varying graph configurations. This limitation is exacerbated in dynamic environments where graph topology changes, necessitating quick adaptation to new patterns [14]. The difficulty in generalizing across diverse structures often requires retraining the network for each new instance, which is computationally expensive.

Efforts to address these challenges include developing more efficient GNN architectures that reduce computational complexity without sacrificing performance. Techniques such as graph sampling and hierarchical pooling manage large graphs by focusing on informative substructures, thereby enhancing scalability [21]. Transfer learning approaches are also being explored to improve GNNs' generalization capabilities, enabling them to leverage knowledge from one domain to enhance performance in another [13].

Despite these advancements, achieving a balance between scalability and generalization remains a significant hurdle in deploying GNNs for large-scale combinatorial optimization tasks. Continued research into developing adaptable and efficient GNN models is crucial for addressing current limitations in performance. Existing GNN heuristics have shown only marginal improvements over traditional methods, such as greedy algorithms, in solving NP-hard problems like Max-Cut. As the field evolves, establishing a unified framework that enhances the scalability and generalizability of GNNs across various optimization tasks while effectively capturing relational structures is essential. This ongoing exploration aims to unlock GNNs' full capabilities, enabling them to navigate the intricacies of multimodal energy landscapes and improve their efficacy as solvers in complex optimization scenarios [45, 4, 46, 57, 51].

Feature	Applications of GNNs in Combinatorial Optimization	Innovative GNN Architectures and Techniques	Integration of GNNs with Other Computational Techniques
Optimization Focus	Combinatorial Challenges	Complex Combinatorial	Complex Integration
Scalability	Enhanced Scalability	Continuous Innovation	Significant Potential
Generalization	Improved Decision-making	Improved Generalization	Enhanced Adaptability

Table 4: This table provides a comparative analysis of the applications, architectures, and integration techniques of Graph Neural Networks (GNNs) in combinatorial optimization. It highlights the optimization focus, scalability, and generalization capabilities of GNNs across various methodologies, illustrating their potential in addressing complex optimization challenges. The table underscores the transformative impact of GNNs in enhancing decision-making, scalability, and adaptability in computational frameworks.

5 Metaheuristics and Their Role in Optimization

Metaheuristics are a cornerstone in optimization, offering flexible solutions to complex combinatorial problems. This section delves into the core principles and applications of metaheuristics, highlighting their iterative nature and adaptability across various domains. By analyzing their operational mechanisms and effectiveness in tackling optimization challenges, their significance in modern computational strategies is clarified. The subsequent subsection will detail the diverse types of metaheuristics, each tailored to address specific optimization tasks through unique methodologies.

5.1 Concept and Application of Metaheuristics

Metaheuristics are advanced algorithms designed to provide approximate solutions to intricate combinatorial problems, characterized by their iterative processes and cross-domain adaptability. Algorithms such as genetic algorithms, simulated annealing, and ant colony optimization excel in exploring expansive search spaces without necessitating detailed problem-specific insights [58]. Their versatility enhances solution quality through iterative refinement, proving invaluable when exact methods like Mixed Integer Linear Programming (MILP) encounter limitations due to complex objectives [59].

The integration of fuzzy logic with iterative optimization techniques exemplifies how metaheuristics address the complexities of combinatorial optimization, particularly under probabilistic constraints [60, 6]. Restart procedures, as demonstrated by Palmigiani, underscore metaheuristics' potential to systematically refine search strategies, enhancing performance in combinatorial tasks [58].

Frameworks like RALS, which leverage local search structures and adaptive strategies, highlight the adaptability of metaheuristics in identifying narrow admissible tuples in combinatorial optimization [17]. Understanding the relationship between objective correlation and the structure of the efficient set, as proposed by Verel, enhances metaheuristic designs for navigating the solution landscape more effectively [61].

Metaheuristics are also applied in designing approximation algorithms for diverse solutions, as illustrated by Hanaka's framework, which employs local search algorithms to systematically explore solution spaces [43]. This underscores their role in enhancing solution diversity and quality in optimization tasks.

Benchmarking is crucial for evaluating metaheuristic performance, especially in assessing machine learning models for tasks like the Traveling Salesman Problem (TSP) [62]. Such evaluations facilitate comparisons between metaheuristic approaches and traditional search methods, driving the development of more robust optimization strategies.

Metaheuristics provide a versatile toolkit for addressing complex optimization challenges, facilitating the discovery of diverse solutions that mirror real-world complexities. Their flexibility enables the development of approximation algorithms that efficiently handle multiple objectives, enhancing applicability across fields like operations research and artificial intelligence [63, 43]. By integrating advanced techniques like machine learning and probabilistic modeling, metaheuristics continue to evolve, enhancing their capacity to solve increasingly complex optimization tasks.

5.2 Types of Metaheuristics

Metaheuristics encompass a wide range of optimization algorithms, each characterized by specific strategies for effectively navigating search spaces. These algorithms identify near-optimal solutions for complex problems across various domains, including combinatorial optimization, artificial intelligence, and operational research. Techniques such as local search structures and approximation methods enable metaheuristics to address challenges like the Balanced Assignment Problem, enhancing the overall optimization process. Frameworks like Markov Decision Processes provide valuable insights into the exploration-exploitation tradeoff, guiding practitioners in selecting suitable metaheuristics for their specific optimization needs [63, 17, 64, 65, 43]. These algorithms are broadly categorized into population-based and single-solution-based methods, each offering distinct advantages in addressing combinatorial optimization challenges.

Population-based metaheuristics, such as Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), maintain a population of candidate solutions that evolve over time. These algorithms effectively explore large search spaces by leveraging collective intelligence and stochastic processes to converge toward optimal solutions. The Anytime Evolutionary DCOP (AED) exemplifies a population-based approach, utilizing evolutionary optimization to solve Distributed Constraint Optimization Problems (DCOPs) by maintaining a distributed set of candidate solutions among agents, enhancing solution quality through iterative refinement [66].

Single-solution-based metaheuristics, including Simulated Annealing (SA) and Tabu Search (TS), iteratively enhance a single candidate solution to tackle combinatorial optimization problems. Their effectiveness can vary significantly based on the specific problem context. Recent research introduces a theoretical framework utilizing Markov Decision Processes (MDP) to analyze these algorithms' behavior, providing insights into convergence properties and the exploration-exploitation tradeoff. This framework aids practitioners in systematically selecting the most suitable metaheuristic for their optimization challenges. New iterative restart procedures optimize performance by managing independent executions, demonstrating marked reductions in failure probability and improved solution quality in practical applications, such as the Traveling Salesman Problem [64, 58]. These methods adeptly exploit local search capabilities, making them suitable for problems requiring fine-tuning of solutions. Simulated Annealing employs a probabilistic acceptance criterion to escape local optima, while Tabu Search utilizes memory structures to avoid revisiting previously explored solutions, enhancing search efficiency.

Hybrid metaheuristics combine elements from both population-based and single-solution-based approaches to capitalize on their respective strengths. These hybrid methods often integrate machine learning techniques to adaptively guide the search process, improving convergence rates and solution quality. The integration of fuzzy logic with iterative optimization techniques exemplifies a hybrid approach that employs adaptive strategies to navigate complex solution landscapes effectively, particularly in combinatorial optimization problems where traditional exact methods often fall short due to vague constraints and uncertain data. This methodology, demonstrated in applications such as workforce scheduling and facility location problems, leverages approximate reasoning and intelligent search strategies to enhance solution efficiency and accuracy in challenging decision-making scenarios [17, 63, 67, 60, 33].

The diversity of metaheuristics provides a robust framework for addressing a wide array of optimization problems. By selecting and customizing appropriate metaheuristic techniques to align with specific optimization characteristics, researchers can create efficient solutions that effectively tackle the intricate challenges present in modern optimization tasks. This approach enhances algorithm performance in complex problems, such as the Balanced Assignment Problem and the Traveling Salesman Problem, while leveraging advancements in machine learning to automate the design of tailored heuristics, bridging the gap between traditional operations research methods and innovative algorithmic strategies [63, 62, 68].

5.3 Enhancing Metaheuristics and Constraint Satisfaction

Enhancing metaheuristics for improved performance in optimization tasks involves integrating advanced strategies and frameworks that address the complexities of combinatorial problems. One significant advancement is developing a framework for modeling local search metaheuristics, which systematically analyzes and compares various metaheuristic strategies, filling critical gaps in existing methodologies and offering insights into performance characteristics and potential improvements [64].

The introduction of the Restart Procedure (RP) exemplifies another enhancement in metaheuristic performance. By optimizing restart times based on performance feedback, RP improves convergence rates and reduces the probability of failure in reaching optimal solutions. This dynamic adjustment of search strategies enhances effectiveness in navigating complex solution landscapes [58].

In large neighborhood search, the Large Neighborhood Prioritized Search (LNPS) method demonstrates the potential of prioritizing search areas based on heuristic insights. However, a notable limitation of LNPS is the complexity involved in determining optimal heuristic parameter configurations, which can be both challenging and time-consuming to fine-tune [69]. Addressing this challenge is crucial for maximizing LNPS efficacy in solving intricate optimization problems.

The Anytime Evolutionary DCOP (AED) approach further highlights enhancements achievable through evolutionary mechanisms and cooperative agents. AED's superior solution quality and continuous improvement capabilities underscore the advantages of integrating evolutionary strategies with distributed optimization frameworks, facilitating robust solutions that adaptively refine themselves over time in dynamic and uncertain environments [66].

Collectively, these enhancements illustrate the transformative potential of advanced metaheuristic strategies in addressing the complexities of constraint satisfaction and optimization tasks. By utilizing frameworks that enhance search processes and integrate adaptive mechanisms, researchers can develop more efficient and effective solutions in combinatorial optimization. These approaches, demonstrated in studies exploring local problem structures and machine learning techniques, enable significant reductions in search space and improved strategies for overcoming local minima. This methodological integration facilitates discovering optimal solutions and expands the potential of combinatorial optimization by leveraging insights from artificial intelligence and data-driven decision-making [17, 70].

6 Constraint Satisfaction in Optimization

6.1 Decentralized Constraint Satisfaction in Smart Cities

Decentralized constraint satisfaction is essential for managing the intricate socio-technical networks of smart cities, where systems like energy distribution and transportation require efficient coordination among agents to optimize resource allocation and performance. Despite the complexity, current decentralized multi-agent methods often face challenges in effectively satisfying hard constraints, particularly in large-scale environments [71]. Integrating decentralized techniques enhances the adaptability and efficiency of urban systems, enabling agents to collaboratively address hard constraints while optimizing performance. This approach improves scalability and privacy, allowing for effective coordination with partial information and balancing hard constraint satisfaction with soft constraint optimization. Empirical results highlight the significance of these methods for policymakers and system operators committed to sustainable urban development [72, 32, 22, 52, 71]. These methods are crucial for real-time decision-making in dynamic urban environments.

In traffic management, decentralized algorithms dynamically adjust signals and reroute vehicles to minimize congestion, while in energy distribution, they balance supply and demand through coordinated resource operation, enhancing reliability and sustainability. These methods maintain optimal performance while adhering to stringent constraints, supporting policymakers and urban planners in creating resilient urban environments [71, 22]. By leveraging multi-agent systems, researchers develop solutions that enhance urban infrastructure efficiency and resilience, meeting hard constraints for net-zero targets. Innovative frameworks like Hierarchical Graph Reinforcement Learning foster cooperation among agents, contributing to smarter, more sustainable cities [71, 28, 32, 5].

6.2 Decentralized Hard Constraint Satisfaction Model (DHCSM)

The Decentralized Hard Constraint Satisfaction Model (DHCSM) significantly advances the optimization of multi-agent systems in smart cities. It enhances system efficiency by enabling agents to make coordinated decisions that adhere to hard constraints, crucial for optimal performance in applications like energy distribution networks, where resource allocation must comply with operational constraints [71]. By facilitating decentralized decision-making, DHCSM alleviates computational demands associated with centralized approaches, offering scalable and flexible solutions. It addresses the cold start problem of partial information during initialization, ensuring compliance with hard constraints while optimizing soft constraints subsequently. This dual focus enhances its applicability in complex infrastructures, making it valuable for policymakers and system operators aiming to create sustainable smart cities [22, 71, 73]. The model's integration with existing optimization algorithms further enhances its robustness in addressing urban challenges.

DHCSM fosters resilient urban environments by satisfying hard constraints, such as net-zero targets, while maintaining cost-effectiveness. It prioritizes privacy and scalability, enabling effective governance and promoting pro-social behavior among agents. Experimental results show that DHCSM significantly enhances decision-making processes and operational sustainability, providing valuable insights for urban planners and system designers [71, 5, 73].

6.3 Constraint Satisfaction in Shipping Route Design

Constraint satisfaction is crucial for designing efficient shipping routes, optimizing logistics operations while adhering to operational constraints such as travel time, fuel consumption, and environmental regulations. Recent advancements, including reinforcement learning and combinatorial optimization techniques, have improved maritime shipping route design by integrating heuristic-based solutions and dynamic decision-making to address real-time logistics complexities [74, 75, 76, 77]. These techniques develop adaptive routing strategies that handle modern shipping demands, including cost efficiency and responsiveness to real-time changes [75, 78, 77].

Integrating constraint satisfaction models with real-time data analytics allows for dynamic monitoring and adjustment of shipping routes, enhancing operational efficiency and cost-effectiveness. This approach utilizes reinforcement learning and constraint programming to optimize routing decisions in real-time, ensuring adaptability to variables such as weather changes, port congestion, and demand fluctuations [79, 77]. By incorporating real-time data, shipping companies can enhance decision-making processes, ensuring optimal routes compliant with all relevant constraints.

The use of constraint satisfaction in shipping route design optimizes maritime logistics operations. By exploring the intricate relationships among factors influencing shipping routes, researchers and practitioners leverage advanced methodologies, such as reinforcement learning and cooperative multi-agent systems, to devise innovative strategies that improve efficiency, safety, and sustainability in global shipping networks. These approaches optimize route design and resource allocation, leading to more resilient and cost-effective maritime operations [28, 73, 77].

6.4 Joint Learning of Hard Constraints and Objective Functions

Joint learning of hard constraints and objective functions is crucial in optimization, especially in multi-agent systems where balancing efficiency and fairness is essential. This approach involves simultaneously considering constraints and objectives, enabling systems to optimize performance while ensuring compliance with predefined limitations. In smart city applications, integrating hard constraints with objective functions facilitates solutions that are both efficient and equitable, addressing diverse urban needs [71].

Incremental satisfaction of hard constraints allows systems to adapt progressively to changing conditions without compromising adherence to constraints, particularly beneficial in dynamic environments where evolving constraints require continuous adjustment of optimization strategies [71]. Furthermore, recovering social capital within multi-agent systems seeks to balance efficiency with fairness, ensuring equitable resource allocation and decision-making, promoting social welfare alongside operational efficiency [71]. This approach emphasizes integrating social and economic factors into optimization frameworks, paving the way for holistic solutions addressing the interplay of constraints and objectives in urban systems.

Integrating hard constraints with objective functions in a joint learning framework enhances multi-agent systems' optimization, managing trade-offs between task performance and action costs in complex environments through techniques like differentiable submodular maximization and generalist combinatorial optimization models [80, 72]. By leveraging advanced learning techniques and social equity considerations, researchers develop innovative solutions that enhance both efficiency and fairness in optimization processes.

7 Challenges and Future Directions

Addressing parameter tuning and training efficiency is crucial in computational techniques for combinatorial optimization. These factors significantly impact model performance and the feasibility of deploying deep reinforcement learning (DRL) across various applications. The following subsections explore these challenges, focusing on the complexities of parameter tuning and its effects on training efficiency in combinatorial optimization problems.

7.1 Challenges in Parameter Tuning and Training Efficiency

Parameter tuning and training efficiency are significant hurdles in applying computational techniques to combinatorial optimization. The dependence on high-quality labeled training data, which may be scarce, limits the generalizability of learned policies without retraining. This issue is exacerbated by complex problem instances requiring extensive tuning of model architectures for optimal performance [13]. In large action spaces, parameter tuning complexity can severely affect training efficiency, where adaptive path selection is crucial. This complexity also hinders learning optimal policies, particularly in meta-learning frameworks, where overfitting can reduce training efficiency, making models overly specialized and less adaptable to new tasks. Effective meta-learning strategies are essential to balance task-specific performance with generalization across diverse problems, enhancing the model's ability to provide robust solutions in real-world applications [81, 82, 18]. The computational demands of reinforcement learning methods further complicate parameter optimization and training processes.

Advanced models strive to improve training efficiency for new subproblems, enhancing DRL's applicability across diverse contexts. However, performance gaps compared to classical solvers indicate a need for further research to optimize capabilities and parameter settings. Additionally, DRL models often face stability and interpretability issues, requiring significant computational resources for desired outcomes [13]. In combinatorial problems, precise parameter tuning, such as the influence of negative pheromones, is crucial for optimizing the search process, as demonstrated in swarm intelligence approaches. The specialized design of models for specific problems may limit broader applicability, where problem complexity often necessitates adaptable solutions. This underscores the need for developing flexible frameworks efficiently tuned for a wider array of tasks, such as leveraging Large Language Models (LLMs) that integrate domain knowledge and contextual reasoning to enhance solution diversity and optimization quality in engineering applications [83, 43].

Innovative strategies are required to enhance parameter tuning and training efficiency, ensuring computational models can effectively navigate the complexities of combinatorial optimization tasks across diverse applications. Future research in graph learning should prioritize methods for encoding global graph information, creating task-specific models, and incorporating traditional heuristics into machine learning frameworks. This approach aims to improve the generalization capabilities and scalability of graph-based solutions across diverse domains, including social networks, biological systems, and combinatorial optimization problems, where efficient analysis is essential due to the NP-hard nature of many tasks. Integrating these elements can enhance the effectiveness of machine learning techniques in tackling complex graph-related challenges [20, 14].

7.2 Improving Exploration and Robustness in Reinforcement Learning

Enhancing exploration and robustness in Reinforcement Learning (RL) is vital for effectively addressing the complexities of combinatorial optimization problems, especially in high-dimensional action spaces and dynamic constraints. A significant challenge is developing scalable RL methods that maintain performance across diverse problem instances and constraints, as seen in frameworks like MatNet, which require expansion to incorporate diverse constraints and improve generalizability [16].

Future research should refine coordination strategies and algorithmic efficiency to address scalability issues, particularly in large-scale team coordination problems [29]. Exploring advanced architectures, such as those used in GAT-PCM, is crucial for managing complex constraint types and asymmetric settings, enhancing RL models' robustness across various applications [22].

Integrating local search methods and utilizing weaker oracles can improve solution quality during learning, addressing challenges in parameter tuning and training efficiency [60]. Innovative restart strategies, demonstrated in traditional meta-heuristic algorithms, can enhance convergence speed and robustness by dynamically adjusting search processes [58].

The application of RL in network interdiction scenarios underscores the need for models with improved generalization capabilities, adaptable to a broader range of scenarios [3]. Furthermore, addressing scalability and generalization challenges in RL frameworks, such as those encountered in Pareto set prediction methods, is crucial for applying them to more complex real-world problems [23].

A comprehensive strategy is imperative to enhance exploration and robustness in RL, encompassing sophisticated exploration techniques, resilient model development, and the integration of traditional heuristic methods with contemporary learning approaches. Evidence from various applications, including robotics, finance, and combinatorial optimization, supports this strategy [32, 34, 36]. By focusing on these research directions, future advancements in RL can yield more robust and efficient solutions for complex optimization tasks, advancing the field of combinatorial optimization.

7.3 Opportunities for Future Research and Development

The intersection of combinatorial optimization, machine learning, and reinforcement learning offers numerous avenues for future research and development. Enhancing model interpretability and resolving stability issues in deep reinforcement learning could lead to more robust and reliable optimization frameworks [13]. Exploring different input representations and improving the solution distribution obtained by DRL-MOA are critical for advancing multi-objective optimization tasks [7].

Future research could extend stochastic combinatorial optimization algorithms to more general classes of graphs, broadening their applicability across various domains [6]. Enhancing model scalability and exploring heterogeneous input graph structures are essential for applying these techniques to a wider range of problems, thus improving their versatility and effectiveness [23].

Manipulating information flow as a subtle intervention tool offers a novel research direction for promoting cooperation and managing social learning dynamics more effectively [5]. This approach could lead to innovative strategies for influencing agent behavior in multi-agent systems, enhancing their adaptability and performance.

Developing frameworks that integrate traditional optimization methods with machine learning techniques can bolster models' generalization capabilities, leading to more robust and adaptable optimization solutions. This integration could address the complexities inherent in modern optimization tasks, offering new insights into problem-solving strategies [13].

By exploring these research directions, the computational optimization community can significantly enhance advanced strategy development to tackle modern optimization tasks' complexities. This involves integrating machine learning techniques to improve decision-making processes in combinatorial optimization, traditionally reliant on handcrafted heuristics. Recent studies highlight that this integration facilitates sophisticated algorithm creation and allows a deeper understanding of the data distributions associated with various optimization problems. Engaging in initiatives like the Machine Learning for Combinatorial Optimization Competition (ML4CO) enables researchers to refine state-of-the-art solvers, leading to more efficient solutions across practical applications, such as balanced item placement and maritime inventory routing [84, 85, 70].

8 Conclusion

The exploration of advanced computational techniques such as Reinforcement Learning (RL), Combinatorial Optimization (CO), and Graph Neural Networks (GNNs) underscores their transformative potential in solving complex optimization problems. By integrating deep learning with domain-specific knowledge, these methods have significantly enhanced algorithmic performance, as evidenced by improved convergence rates and solution quality. The fusion of deep learning models within RL frameworks has notably advanced the management of high-dimensional data, enhancing efficiency in dynamic settings.

Bi-level optimization frameworks have demonstrated superior performance and generalization over traditional heuristics, proving effective across various combinatorial tasks. Techniques like the SeqMO method leverage historical data to improve exploration in evolutionary algorithms, yielding superior results in multi-objective optimization scenarios. The NCO framework has shown its robustness and adaptability, outperforming conventional heuristic methods in practical applications. Additionally, the SPO method, by incorporating prediction errors into optimization, highlights its potential in decision-focused learning contexts.

The CR framework's capacity to augment large language model reasoning through automated prompt generation marks a significant advancement, achieving notable performance on reasoning tasks. Collectively, these innovations emphasize the critical role of integrating these methodologies to address complex decision-making challenges, setting the stage for future research focused on enhancing scalability, efficiency, and adaptability across various fields. The comprehensive taxonomy and identification of solver limitations provide a strategic foundation for future research endeavors aimed at improving solver performance and generalization.

References

[1] Mert Esencan, Tarun Advaith Kumar, Ata Akbari Asanjan, P. Aaron Lott, Masoud Mohseni, Can Unlu, Davide Venturelli, and Alan Ho. Combinatorial reasoning: Selecting reasons in generative ai pipelines via combinatorial optimization, 2024.

[2] Peng Li and Bo Liu. Multi-task combinatorial optimization: Adaptive multi-modality knowledge transfer by an explicit inter-task distance, 2023.

[3] Lei Zhang, Zhiqian Chen, Chang-Tien Lu, and Liang Zhao. Network interdiction goes neural, 2024.

[4] Quentin Cappart, Didier Chételat, Elias Khalil, Andrea Lodi, Christopher Morris, and Petar Veličković. Combinatorial optimization and reasoning with graph neural networks, 2022.

[5] Qiliang Chen and Babak Heydari. Adaptive network intervention for complex systems: A hierarchical graph reinforcement learning approach, 2024.

[6] Shipra Agrawal, Amin Saberi, and Yinyu Ye. Stochastic combinatorial optimization under probabilistic constraints, 2008.

[7] Kaiwen Li, Tao Zhang, and Rui Wang. Deep reinforcement learning for multi-objective optimization, 2020.

[8] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*, 2017.

[9] Kaiwen Li, Tao Zhang, Rui Wang Yuheng Wang, and Yi Han. Deep reinforcement learning for combinatorial optimization: Covering salesman problems, 2021.

[10] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[11] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[12] Natalia Vesselinova, Rebecca Steinert, Daniel F. Perez-Ramirez, and Magnus Boman. Learning combinatorial optimization on graphs: A survey with applications to networking, 2020.

[13] Shengbo Eben Li. Deep reinforcement learning. In *Reinforcement learning for sequential decision and optimal control*, pages 365–402. Springer, 2023.

[14] Yun Peng, Byron Choi, and Jianliang Xu. Graph learning for combinatorial optimization: A survey of state-of-the-art, 2021.

[15] Jiongzhi Zheng, Kun He, Jianrong Zhou, Yan Jin, and Chu-Min Li. Reinforced lin-kernighan-helsgaun algorithms for the traveling salesman problems, 2022.

[16] Yeong-Dae Kwon, Jinho Choo, Iljoo Yoon, Minah Park, Duwon Park, and Youngjune Gwon. Matrix encoding networks for neural combinatorial optimization, 2021.

[17] Xiao-Feng Xie and Zun-Jing Wang. Exploiting problem structure in combinatorial landscapes: A case study on pure mathematics application, 2018.

[18] Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1658–1665, 2019.

[19] Zhentao Tan and Yadong Mu. Learning solution-aware transformers for efficiently solving quadratic assignment problem, 2024.

[20] Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. Graph learning: A survey, 2021.

[21] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Combinatorial optimization with graph convolutional networks and guided tree search, 2018.

[22] Yanchen Deng, Shufeng Kong, and Bo An. Pretrained cost model for distributed constraint optimization problems, 2021.

[23] Shiqing Liu, Xueming Yan, and Yaochu Jin. End-to-end pareto set prediction with graph neural networks for multi-objective facility location, 2022.

[24] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V. Snyder, and Martin Takáč. Reinforcement learning for solving the vehicle routing problem, 2018.

[25] Alessandro Barro. Pointer networks with q-learning for combinatorial optimization, 2024.

[26] Alexandre Laterre, Yunguan Fu, Mohamed Khalil Jabri, Alain-Sam Cohen, David Kas, Karl Hajjar, Torbjorn S. Dahl, Amine Kerkeni, and Karim Beguir. Ranked reward: Enabling self-play reinforcement learning for combinatorial optimization, 2018.

[27] Ruben Solozabal, Josu Ceberio, and Martin Takáč. Constrained combinatorial optimization with reinforcement learning, 2020.

[28] Xihan Li, Jia Zhang, Jiang Bian, Yunhai Tong, and Tie-Yan Liu. A cooperative multi-agent reinforcement learning framework for resource balancing in complex logistics network, 2019.

[29] Yanlin Zhou, Manshi Limbu, Gregory J. Stein, Xuan Wang, Daigo Shishika, and Xuesu Xiao. Team coordination on graphs: Problem, analysis, and algorithms, 2024.

[30] Honguk Woo, Hyunsung Lee, and Sangwoo Cho. An efficient combinatorial optimization model using learning-to-rank distillation, 2021.

[31] Yong Liang Goh, Wee Sun Lee, Xavier Bresson, Thomas Laurent, and Nicholas Lim. Combining reinforcement learning and optimal transport for the traveling salesman problem, 2022.

[32] Yuxi Li. Reinforcement learning in practice: Opportunities and challenges, 2022.

[33] Andrea Lodi, Luca Mossina, and Emmanuel Rachelson. Learning to handle parameter perturbations in combinatorial optimization: an application to facility location, 2019.

[34] Mohamed-Amine Chadi and Hajar Mousannif. Understanding reinforcement learning algorithms: The progress from basic q-learning to proximal policy optimization, 2023.

[35] Leo Ardon. Reinforcement learning to solve np-hard problems: an application to the cvrp, 2022.

[36] Qingpeng Cai, Will Hang, Azalia Mirhoseini, George Tucker, Jingtao Wang, and Wei Wei. Reinforcement learning driven heuristic optimization, 2019.

[37] Arthur Delarue, Ross Anderson, and Christian Tjandraatmadja. Reinforcement learning with combinatorial actions: An application to vehicle routing, 2020.

[38] Dmitry Sorokin and Alexander Kostin. Treedqn: Learning to minimize branch-and-bound tree, 2023.

[39] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[40] Julien Baste, Michael R. Fellows, Lars Jaffke, Tomáš Masařík, Mateus de Oliveira Oliveira, Geevarghese Philip, and Frances A. Rosamond. Diversity of solutions: An exploration through the lens of fixed-parameter tractability theory, 2020.

[41] Yongming He, Guohua Wu, Yingwu Chen, and Witold Pedrycz. A two-stage framework and reinforcement learning-based optimization algorithms for complex scheduling problems, 2021.

[42] Ching Pui Wan, Tung Li, and Jason Min Wang. Rlor: A flexible framework of deep reinforcement learning for operation research, 2023.

[43] Tesshu Hanaka, Masashi Kiyomi, Yasuaki Kobayashi, Yusuke Kobayashi, Kazuhiro Kurita, and Yota Otachi. A framework to design approximation algorithms for finding diverse solutions in combinatorial problems, 2022.

[44] Sahil Manchanda, Sofia Michel, Darko Drakulic, and Jean-Marc Andreoli. On the generalization of neural combinatorial optimization heuristics, 2022.

[45] Stefan Boettcher. Inability of a graph neural network heuristic to outperform greedy algorithms in solving combinatorial optimization problems like max-cut, 2022.

[46] Yaochu Jin, Xueming Yan, Shiqing Liu, and Xiangyu Wang. A unified framework for combinatorial optimization based on graph neural networks, 2024.

[47] Haofeng Yuan, Peng Jiang, and Shiji Song. The neural-prediction based acceleration algorithm of column generation for graph-based set covering problems, 2022.

[48] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems*, 30, 2017.

[49] Malte Probst, Franz Rothlauf, and Jörn Grahl. Scalability of using restricted boltzmann machines for combinatorial optimization, 2014.

[50] Congsong Zhang, Yong Gao, and James Nastos. Learning branching heuristics from graph neural networks, 2022.

[51] Ruibin Zeng, Minglong Lei, Lingfeng Niu, and Lan Cheng. A unified pre-training and adaptation framework for combinatorial optimization on graphs, 2023.

[52] Joel Oren, Chana Ross, Maksym Lefarov, Felix Richter, Ayal Taitler, Zohar Feldman, Christian Daniel, and Dotan Di Castro. Solo: Search online, learn offline for combinatorial optimization problems, 2021.

[53] Zhipeng Li, Jianwei Wu, Lin Sun, and Tao Rong. Combinatorial keyword recommendations for sponsored search with deep reinforcement learning, 2019.

[54] Ismail R. Alkhouri, George K. Atia, and Alvaro Velasquez. A differentiable approach to combinatorial optimization using dataless neural networks, 2022.

[55] Haoqi He. Quantum annealing and graph neural networks for solving tsp with qubo, 2024.

[56] Humphrey Munn and Marcus Gallagher. Towards understanding the link between modularity and performance in neural networks for reinforcement learning, 2024.

[57] Maximilian Böther, Otto Kißig, Martin Taraz, Sarel Cohen, Karen Seidel, and Tobias Friedrich. What's wrong with deep learning in tree search for combinatorial optimization, 2022.

[58] Davide Palmigiani and Giovanni Sebastiani. A new restart procedure for combinatorial optimization and its convergence, 2017.

[59] Pierre-Cyril Aubin-Frankowski, Yohann De Castro, Axel Parmentier, and Alessandro Rudi. Generalization bounds of surrogate policies for combinatorial optimization problems, 2024.

[60] Andreas Raggl and Wolfgang Slany. A reusable iterative optimization software library to solve combinatorial problems with approximate reasoning, 1998.

[61] Sébastien Verel, Arnaud Liefooghe, Laetitia Jourdan, and Clarisse Dhaenens. Analyzing the effect of objective correlation on the efficient set of mnk-landscapes, 2012.

[62] Antoine François, Quentin Cappart, and Louis-Martin Rousseau. How to evaluate machine learning approaches for combinatorial optimization: Application to the travelling salesman problem, 2019.

[63] Anurag Dutta, K. Lakshmanan, A. Ramamoorthy, Liton Chandra Voumik, John Harshith, and John Pravin Motha. A review on optimality investigation strategies for the balanced assignment problem, 2023.

[64] Rubén Ruiz-Torrubiano. Modeling local search metaheuristics using markov decision processes, 2024.

[65] Phuong Le. A survey on combinatorial optimization, 2024.

[66] Saaduddin Mahmud, Moumita Choudhury, Md. Mosaddek Khan, Long Tran-Thanh, and Nicholas R. Jennings. Aed: An anytime evolutionary dcop algorithm, 2020.

[67] Jaynta Mandi, Emir Demirović, Peter. J Stuckey, and Tias Guns. Smart predict-and-optimize for hard combinatorial optimization problems, 2019.

[68] André Hottung, Yeong-Dae Kwon, and Kevin Tierney. Efficient active search for combinatorial optimization problems. *arXiv preprint arXiv:2106.05126*, 2021.

[69] Irumi Sugimori, Katsumi Inoue, Hidetomo Nabeshima, Torsten Schaub, Takehide Soh, Naoyuki Tamura, and Mutsunori Banbara. Large neighborhood prioritized search for combinatorial optimization with answer set programming, 2024.

[70] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon, 2020.

[71] Srijoni Majumdar, Chuhao Qin, and Evangelos Pournaras. Discrete-choice multi-agent optimization: Decentralized hard constraint satisfaction for smart cities, 2023.

[72] Darko Drakulic, Sofia Michel, and Jean-Marc Andreoli. Goal: A generalist combinatorial optimization agent learner, 2025.

[73] Cheikh Ahmed, Alexandre Forel, Axel Parmentier, and Thibaut Vidal. Districtnet: Decision-aware learning for geographical districting, 2024.

[74] Phillippe Samer, Evellyn Cavalcante, Sebastián Urrutia, and Johan Oppen. The matching relaxation for a class of generalized set partitioning problems, 2018.

[75] Toni Greif, Louis Bouvier, Christoph M. Flath, Axel Parmentier, Sonja U. K. Rohmer, and Thibaut Vidal. Combinatorial optimization and machine learning for dynamic inventory routing, 2024.

[76] Valentin Antuori and Florian Richoux. Constrained optimization under uncertainty for decision-making problems: Application to real-time strategy games, 2019.

[77] Utsav Dutta, Yifan Lin, and Zhaoyang Larry Jin. Liner shipping network design with reinforcement learning, 2024.

[78] Léo Baty, Kai Jungel, Patrick S. Klein, Axel Parmentier, and Maximilian Schiffer. Combinatorial optimization enriched machine learning to solve the dynamic vehicle routing problem with time windows, 2023.

[79] Imanol Echeverria, Maialen Murua, and Roberto Santana. Leveraging constraint programming in a deep learning approach for dynamically solving the flexible job-shop scheduling problem, 2024.

[80] Guangyao Shi, Chak Lam Shek, Nare Karapetyan, and Pratap Tokekar. Decision-oriented learning using differentiable submodular maximization for multi-robot coordination, 2024.

[81] Haoyu Wang and Pan Li. Unsupervised learning for combinatorial optimization needs meta-learning, 2023.

[82] Ruiqi He and Falk Lieder. What are the mechanisms underlying metacognitive learning?, 2023.

[83] Shuo Jiang, Min Xie, and Jianxi Luo. Large language models for combinatorial optimization of design structure matrix, 2024.

[84] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

[85] Maxime Gasse, Quentin Cappart, Jonas Charfreitag, Laurent Charlin, Didier Chételat, Antonia Chmiela, Justin Dumouchelle, Ambros Gleixner, Aleksandr M. Kazachkov, Elias Khalil, Pawel Lichocki, Andrea Lodi, Miles Lubin, Chris J. Maddison, Christopher Morris, Dimitri J. Papageorgiou, Augustin Parjadis, Sebastian Pokutta, Antoine Prouvost, Lara Scavuzzo, Giulia Zarpellon, Linxin Yang, Sha Lai, Akang Wang, Xiaodong Luo, Xiang Zhou, Haohan Huang, Shengcheng Shao, Yuanming Zhu, Dong Zhang, Tao Quan, Zixuan Cao, Yang Xu, Zhewei Huang, Shuchang Zhou, Chen Binbin, He Minggui, Hao Hao, Zhang Zhiyu, An Zhiwu, and Mao Kun. The machine learning for combinatorial optimization competition (ml4co): Results and insights, 2022.

AI-generated, for reference only.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.