向楚阳
kroxxcy777
kroxxcy777@sjtu.edu.cn

AIH4CO: AI-Assisted Heuristics to Solve Combinatorial Optimization Problems

Chuyang Xiang

ABSTRACT

Combinatorial optimization problems are addressed on a daily basis. Nevertheless, optimal solutions remain challenging to find. In recent years, researchers have sought to enhance traditional heuristic techniques through the incorporation of AI methodologies, thereby proposing novel approaches to the delivery of optimal or near-optimal solutions. However, there is currently no up-to-date review of this field. The objective of this survey is to fill this gap in order to stimulate new ideas on this combination. Firstly, the key concepts are explained. Then, the collected methods are categorised based on a proposed taxonomy, and some analysis of current challenges is provided. Finally, promising future research directions are proposed. Relevant papers up-to-date in this field can be found via the following link: https://github.com/KROX777/Awesome-AIH4CO.

1 Introduction

Combinatorial optimization problems (COP) form the foundation of numerous practical problems in real-life scenario such as vehicle routing and scheduling, electronic design automation, resource allocation, etc. These problems typically aim to minimize a specific cost function within discrete structures, often under *complex constraints*. Since they're often NP-Hard problems (Karp., 1972), loads of heuristic (Boussa¨ıd et al., 2013; Farhi E., 2014) algorithms are developed to offer more cost-efficient solutions. In the past decade, as AI-driven methods are becoming ubiquitous, methods like Machine learning (ML), Graph Neural Networks (GNN), Reinforcement Learning (RL), etc. have emerged as state-of-the-art techniques in CO problem-solving while still having room for improvement. On the other hand, researchers are also trying to utilize modern AI methods to refresh traditional heuristics given the superiority of well-established, highly optimized heuristics over ML, RL methods. (Darvariu Victor-Alexandru and Mirco, 2024)

However, while research on integrating AI with heuristics for CO (AIH4CO) has increased, comprehensive and systematic surveys specifically focusing on this integration are limited. Lots of surveys have been done on CO problem-solving, but only cover one method in isolation (Boussa¨ıd et al., 2013; Quentin Cappart et al., 2021) or the multiple solutions to one specific COP (Jiayi Zhang and Yan, 2022). E.-G. et al. (2020) have explored how researchers have combined ML and heuristics to solve general optimization problems. Maryam Karimi-Mamaghan et al. (2021); dos Santos et al. (2014) have done surveys on similar topics but fail to be up-to-date, only including the integration of heuristics and traditional ML algorithms.

The major contributions of this survey lie in three main aspects:

1. **Up-to-date overview:** The survey provides a focused overview on the work to-date on AIH4CO, addressing the gap mentioned above.
2. **Taxonomy optimization:** The survey updates the taxonomy proposed by Maryam Karimi-Mamaghan et al. (2021) and E.-G. et al. (2020) to better conclude recent models and guide future research.
3. **Re-evalution of recent ideas:** The survey collates novel benchmark methods for several COPs and re-evaluates the value of several models.

2 Preliminaries

2.1 Combinatorial Optimization Problems (COP)

As mentioned above, COPs are a family of problems aiming to minimize a specific cost function within discrete structures under complex constraints. Here we listed some representative COPs. Additionally, we denote an undirected and weighted graph $G = (V, E)$.

1. **Traveling Salesman Problem (TSP):** Given a set of n cities represented as vertices $V = \{1, 2, \dots, n\}$ and a distance matrix $D = (d_{ij})$ where d_{ij} is the distance between city i and city j, the goal is to find a permutation π of V that minimizes the total tour length:

$$\min_{\pi} \sum_{i=1}^{n} d_{\pi(i)\pi(i+1)},$$

where $\pi(n+1) = \pi(1)$ to ensure the tour returns to the starting city.

2. **Maximum Cut Problem (MCP):** The goal is to find a partition of the vertex set $V = S \bigsqcup T$ that maximizes the number of edges between S and T, i.e., $|\{\{v_i, v_j\} \in E : v_i \in S, v_j \in T\}|$

3. **Minimum Dominating Clique Problem (MDCP):** The goal is to find the smallest possible subset S of V such that S is a clique and dominates $V \setminus S$, i.e., $\forall u \in V \setminus S$, $\exists v \in S$ such that $\{v, u\} \in E$.

4. **Maximum Minimal Cut Problem (MMCP):** The goal is to find a partition of the vertex set V into two disjoint subsets V_1 and V_2 such that the minimum cut between V_1 and V_2 is maximized.

5. **Maximum Independent Set Problem (MISP):** The goal is to find the largest possible subset of vertices $S \subseteq V$ such that no two vertices in S are adjacent (i.e., for any $u, v \in S$, $\{u, v\} \notin E$).

6. **Mixed Integer Programming (MIP):** The goal is to optimize a linear objective function subject to linear constraints, with some variables constrained to be integers. Formally:

$$\min_{x,y} c^T x + d^T y$$

subject to

$$Ax + By \leq b,$$

where $x \in \mathbb{Z}^p$ (integer variables), $y \in \mathbb{R}^q$ (continuous variables), and A, B, c, d, and b are matrices and vectors of appropriate dimensions.

Only a few problems are listed. For more problem descriptions, please refer to Maryam Karimi-Mamaghan et al. (2021).

2.2 Meta-Heuristics (MH)

Meta-Heuristics is a broad concept encompassing all high-level heuristic algorithms designed to find near-optimal solutions across a wide range of COPs. Different from ML, which utilizes data-driven, generalizable approaches that learn patterns , MH are simpler, rule-based methods designed for specific tasks, thus providing acceptable solutions in reasonable computational time (Hertz A., 1990). Typical MH algorithms include Simulated Annealing (SA), Tabu Search (TS), Ant Colony Optimization (ACO), etc. They are primarily dedicated to directly solving problems. However, they often require manually designed and tuned strategies, making them less robust in front of new situations.

2.3 Hyper-Heuristics (HH)

Different from MH, HH is a kind of search techniques that can generate solutions for a wide range of problem domains instead of using specific technique to each problem instance thanks to the fact that HH works in the heuristic search space instead of a solution space (Ahmed Kafafy et al., 2022). Fig. 1 below presents two forms of HH (Edmund K. Burke et al., 2010). Given this advantage, HH already has remarkable performance in certain CO problems and presents a higher level of generality (Gabriel Duflo et al., 2019). However, they are still limited by the heuristic space predefined by human experts, calling for AI-driven models to automate this process (Ye H. et al., 2024).

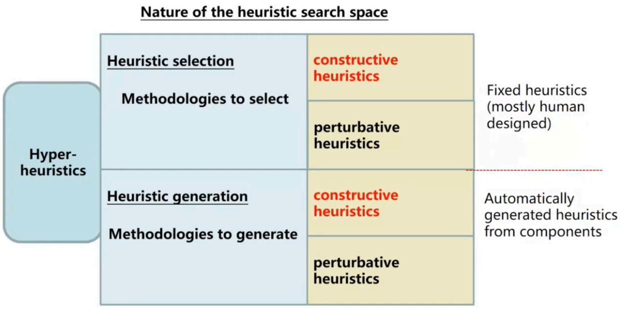

Figure 1: The Classification of Hyper-Heuristic

3 AIH4CO Techniques and their Analysis

This survey restructured the taxonomy method for how AI is used in the heuristics that was utilized in Maryam Karimi-Mamaghan et al. (2021) to make it clearer as well as to introduce new ideas like algorithm generation and scheduling. Fig. 2 shows the method. It's noteworthy that the focus of the survey is mainly on *AI-in-Heuristics* and *AI-and-Heuristics*.

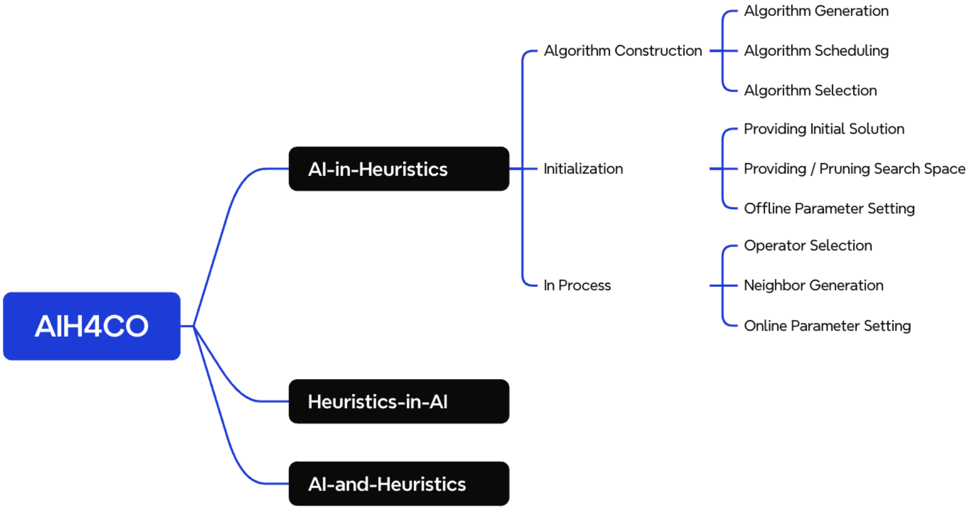

Figure 2: Taxonomy of AIH4CO

3.1 AI-in-Heuristics

3.1.1 Algorithm Selection & Generation

Algorithm selection and generation are two major research directions. The former concept is more established, with the Algorithm Search Problem (ASP) having been proposed and researched for a considerable period of time (Maryam Karimi-Mamaghan et al., 2021) . In comparison, the latter concept gains much development recently with the help of AI. A considerable number of implementations of reinforcement learning (RL) on CO problems have adopted the concept of algorithm generation (Nina Mazyavkina and Burnaev, 2021). However, pure RL is beyond the scope of this survey. The present survey focuses on RL for heuristics and heuristic generation HH as major methods in algorithm generation. Fig 3 illustrates how RL can be used for heuristics generation (Ye H. and Y., 2023). This survey presents recent research employing HH, as well as the conventional ASP-oriented meta-learning approach, as illustrated in Table 2.

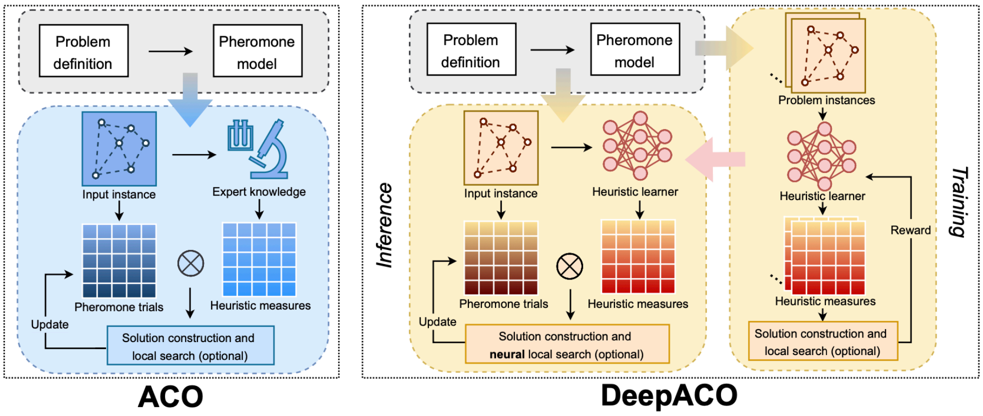

Figure 3: RL for Heuristics Generation

The approach of heuristic generation often stands out as it does not necessitate the prior construction of manually crafted algorithms. (Edmund K. Burke et al., 2010) This makes heuristic generation HH more general and robust. Nevertheless, when a substantial number of well-designed candidate algorithms are available, meta-learning and heuristic selection HH are optimal choices thanks to their better performance and low computational costs.

It is noteworthy that GNNs have emerged as a prevalent tool for learning algorithmic tasks, which are a crucial aspect of both algorithm selection and generation (Elias B. Khalil et al., 2017; Shen Yunzhuang and Xiaodong, 2021). However, Nath and Kuhnle (2024) proposed a benchmark standard and put forward the notion that GNNs do not actually perform as well as it's expected since linear regression even outperforms GNNs in some highly cited models (Elias B. Khalil et al., 2017), which is alarming and disappointing. The notion corresponded with what Angelini and Ricci-Tersenghi (2023) has expressed.

Table 1: Methods on algorithm selection & generation

Method	Heuristic Strategy	AI Model	CO Problems	Additional Techniques	Selection or Generation
Elias B. Khalil et al. (2017)	Forward Greedy	RL,GNN	COP over graphs	/	Generation
de la Rosa-Rivera et al. (2021)	MH	Linear Regression	CB-CTT	Meta-learning Framework	Selection
Shen Yunzhuang and Xiaodong (2021)	MH	GCN	MIP	PB-DFS	Selection
Ye H. and Y. (2023)	MH(ACO)	DRL	General	Local Search	Generation
Bernardino Romera-Paredes et al. (2023)	HH	LLM	General	Evolutionary Framework	Both
Ye H. et al. (2024)	HH,GLS	LLM	General	Gene Language Hyper-Herusitic	Both
F. Liu et al. (2024)	HH	LLM	General	Evolutionary Computation(EC)	Both

3.1.2 Algorithm Scheduling

Algorithm scheduling refers to utilizing AI methods to decide which heuristics to run and/or for how long (Scavuzzo et al., 2024). This is similar to one function of heuristic selection HH. Two advantages can be identified. The first is the combination of different heuristics to achieve optimal performance. The second is the ability to exert greater control over the computational budget. The process is illustrated in Fig 4, and the techniques are listed below on Table 2.

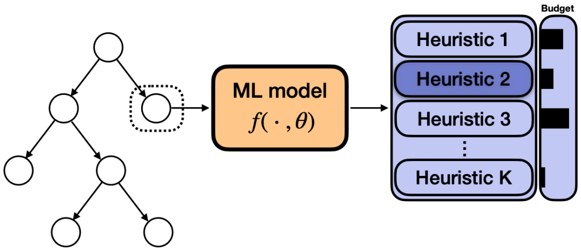

Figure 4: Heuristics Scheduling

Table 2: Methods on algorithm scheduling

Method	Heuristic Strategy	AI Model	CO Problems	Additional Techniques
Elias B. Khalil and Shao (2017)	B&B	Logistic Regression	MIP	/
Antonia Chmiela and Pokutta (2021)	B&B,LNS,Diving	ML	MIP	/
Antonia Chmiela and Sebastian (2023)	B&B,LNS,Diving	ML	MIP	ϵ-greedy bandit algorithm

3.1.3 Initialization

Any heuristics starts its search process from an initial solution or a population of solutions (Maryam Karimi-Mamaghan et al., 2021). Fortunately, it's easy for machine learning to identify elements to either be or not to be in the optimal solution based on available information (Lauri Juho and Deepak, 2023). So the AI-based initialization can provide good initial solutions (Zhang Jiwei and Deepak, 2024), provide search space or prune search space (Liang Xin et al., 2021; Congsong Zhang et al., 2022; Lauri Juho and Deepak, 2023; Zhang Jiwei and Deepak, 2024) through learning. Initial parameters, which are essential to MH, can also be learned. Concrete methods comparison are given below on Table 3.

Initialization has long been shown to be the single most impactful component of CO solvers (Achterberg and Wunderling, 2013), making it a promising research direction. Among these reviewed essays, Lauri Juho and Deepak (2023) stands out for its unique self-designed heuristics ALTHEA, which combines exact algorithms with light-weighted machine learning methods to prune search space.

Table 3: Methods on initialization

Method	Heuristic Strategy	AI Model	CO Problems	Additional Techniques
Zhuwen Li and Koltun (2018)	Branching Heuristic	GNN	MISP	/
Liang Xin et al. (2021)	LKH	SGN	TSP	/
Lauri Juho and Deepak (2023)	Self-Designed	Gradient Boosted Trees	General	Chi-square Statistics
Congsong Zhang et al. (2022)	Branching Heuristic	GNN	MDCP	/
Huaiyuan Liu et al. (2024)	Heuristic Tree Transformation	GNN	MMCP	Relaxation-plus-Rounding
Zhang Jiwei and Deepak (2024)	Local Search	SVM	Steiner Tree Problem	/

3.1.4 In Process

AI methods can also help heuristics either decide upon the selection of operators at each step of the search process based on the status of the search and the operators' performance history (Maryam Karimi-Mamaghan and Meyer, 2023) or identify the common characteristics that are often present in effective solutions and use this knowledge to generate new solutions with the same characteristics by fixing specific solution characteristics (Fairee S. and B., 2019; Taoan Huang et al., 2023). On the other hand, many heuristics rely heavily on online parameter setting, which can also be done by AI methods (Öztop H. and Q.-K., 2020). Here are some implementations on Table 4.

Generalization is the hugest challenge in this area given the need of executing (extremely) fast primal heuristics in process (Scavuzzo et al., 2024).

Table 4: Methods in process

Method	Heuristic Strategy	AI Model	CO Problems	Additional Techniques
Benlic Una and K. (2017)	Breakout Local Search	RL	Vertex separator problem	/
Fairee S. and B. (2019)	Artificial Bee Colony Optimization	RL	TSP	/
Mosadegh H. et al. (2020)	MH(SA)	RL	Assembly line sequencing	/
Öztop H. and Q.-K. (2020)	MH(Variable Neighborhood Search)	RL	Flowshop scheduling problem	/
Maryam Karimi-Mamaghan and Meyer (2023)	MH(Iterated Greedy)	RL(Q-Learning)	Permutation flowshop scheduling	/
Tönshoff et al. (2023)	Local Search	GNN, RL	MCP	/
Taoan Huang et al. (2023)	LNS	Contrastive Learning (CL)	MIP	Graph Attention Networks

3.2 AI-and-Heuristics

Here, heuristics and AI methods perform their own functions independently to solve CO problems. These efforts are listed on Table 5.

Table 5: Methods on AI-and-Heuristics

Method	Heuristic Strategy	AI Model	CO Problems	How they Cooperate
Maryam Karimi-Mamaghan and Ahmadi (2020)	Iterated Local Search	K-Means	IPP	K-Means to link multi-objective population-based MHs with single-solution based MHs
Hou Qingchun and Yuming (2023)	MH	Transformer	VRP	Transformer to divide VRPs into sub-VRPs and MH to solve sub-VRPs

4 Novel Benchmark Standards

The survey also gathers some novel benchmark standards to help evaluate future work. Lu Han et al. (2023) designed RL-based attacker and proposed a general framework for evaluating robustness of CO solvers on graphs. More specifically, Angelini and Ricci-Tersenghi (2023) put forward some hard benchmarks on MCP and MISP and Nath and Kuhnle (2024) also proposed a strict benchmark for MCP in terms of three dimensions: objective value, generalization, and scalability.

5 Conclusions & Outlooks

The survey looks at multiple ways that heuristics can be combined with AI methods to solve CO problems. With regard to the future research directions, Maryam Karimi-Mamaghan et al. (2021) has proposed utilizing clustering ML methods to pre-process data and using AI for multiple purposes with most models proposed so far focusing on specific aspects of heuristic processes, which still applies to the present. In addition to them, other five future research directions are provided based on the findings:

1. **Make Full Use of LLM:** As LLM begins to support code generation and optimisation, new avenues for research on this topic emerge. Research such as Ye H. et al. (2024) remains constrained by a relatively

naive idea. The field is in need of more ambitious approaches that leverage LLM and heuristics to solve CO problems.

2. **Towards Interpretable AI:** CO problems are calling for more interpretable methods. The recently-proposed Kolmogorov-Arnold Network (KAN) (Liu et al., 2024) is interpretable to some extent and excels at learning combinatorial structures, which has the potential to be integrated into heuristics and solve CO problems.

3. **AI for Quantum Heuristics:** The implementation of quantum-related techniques holds considerable promise for the future, with a number of quantum heuristics (such as QAOA) already being developed for use in CO problems (Farhi E., 2014; Tongyang Li et al., 2024). Quantum Neural Network (QNN) was also proposed to solve CO problems (Xinyu Ye and Yan, 2023). However, the combination of AI and quantum heuristics remains an area that is yet to be fully explored.

4. **Explore More Out-of-box Combination:** Most of the works surveyed can be categorized into the four branches AI-in-Heuristics. New combination ideas are too rare in AI-and-Heuristics. Groundbreaking improvements may emanate from completely new ideas.

5. **Unified Benchmark Standards:** As mentioned in Section 3.1.1, Nath and Kuhnle (2024) sharply pointed out the significant shortcomings of GNN-assisted heuristics compared with traditional heuristics and linear-regression-assisted ones. This underscored the limitations of benchmark standards employed by researchers.

References

Achterberg, T. and Wunderling, R. (2013). Mixed integer programming: Analyzing 12 years of progress. In *Facets of Combinatorial Optimization: Festschrift for Martin Grötschel*, pages 449–481. Springer.

Ahmed Kafafy, Asmaa Awaad, N. E.-H. et al. (2022). Multi-level hyper-heuristic for combinatorial optimization problems. *The Intelligent Networks and Systems Society.*

Angelini, M. C. and Ricci-Tersenghi, F. (2023). Modern graph neural networks do worse than classical greedy algorithms in solving combinatorial optimization problems like maximum independent set. *Nature Machine Intelligence*, 5(1):29–31.

Antonia Chmiela, Elias B. Khalil, A. G. A. L. and Pokutta, S. (2021). Learning to schedule heuristics in branch-and-bound. *arXiv, 2021.*

Antonia Chmiela, Gleixner Ambros, L. P. and Sebastian, P. (2023). Online learning for scheduling mip heuristics. In Cire, A. A., editor, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 114–123, Cham. Springer Nature Switzerland.

Benlic Una, E. M. G. and K., B. E. (2017). A hybrid breakout local search and reinforcement learning approach to the vertex separator problem. *European Journal of Operational Research*, 261(3):803–818.

Bernardino Romera-Paredes, Mohammadamin Barekatain, A. N. et al. (2023). Mathematical discoveries from program search with large language models. *Nature.*

Boussa¨ıd, J. Lepagnot, P. S. et al. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237:82–117.

Congsong Zhang, Y. G. et al. (2022). Learning branching heuristics from graph neural networks. *arXiv, 2022.*

Darvariu Victor-Alexandru, H. S. and Mirco, M. (2024). Graph reinforcement learning for combinatorial optimization: A survey and unifying perspective. *Transactions on Machine Learning Research.*

de la Rosa-Rivera, F., Nunez-Varela, J. I., Ortiz-Bayliss, J. C., and Terashima-Marín, H. (2021). Algorithm selection for solving educational timetabling problems. *Expert Systems with Applications*, 174:114694.

dos Santos, J. P. Q., de Melo, J. D., Neto, A. D. D., and Aloise, D. (2014). Reactive search strategies using reinforcement learning, local search algorithms and variable neighborhood search. *Expert Systems with Applications*, 41(10):4939–4949.

E.-G., T. et al. (2020). Machine learning into metaheuristics: A survey and taxonomy of data-driven metaheuristics. *Working paper.*

Edmund K. Burke, Matthew Hyde, G. K. et al. (2010). A classification of hyper-heuristic approaches. 146:449–468.

Elias B. Khalil, Bistra Dilkina, G. L. N. S. A. and Shao, Y. (2017). Learning to run heuristics in tree search. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Elias B. Khalil, Dai Hanjun, Z. Y. et al. (2017). Learning combinatorial optimization algorithms over graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

F. Liu, X. Tong, M. Y. X. L. et al. (2024). Evolution of heuristics: Towards efficient automatic algorithm design using large language mode. *International Conference on Machine Learning (ICML)*.

Fairee S., Khompatraporn C., P.-o. S. and B., S. (2019). Combinatorial artificial bee colony optimization with reinforcement learning updating for travelling salesman problem. In *2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 93–96. IEEE.

Farhi E., Goldstone J., a. o. (2014). A quantum approximate optimization algorithm. *arXiv, 2014*.

Gabriel Duflo, Emmanuel Kieffer, M. R. B. et al. (2019). A gp hyper-heuristic approach for generating tsp heuristics. *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*.

Hertz A., . d. W. D. (1990). The tabu search metaheuristic: How we used it. *Annals of Mathematics and Artificial Intelligence*, 1.

Hou Qingchun, Yang Jingwei, S. Y. W. X. and Yuming, D. (2023). Generalize learned heuristics to solve large-scale vehicle routing problems in real-time. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Huaiyuan Liu, Xianzhang Liu, D. Y. et al. (2024). An unsupervised learning framework combined with heuristics for the maximum minimal cut problem. *Knowledge Discovery and Data Mining (KDD ’ 24)*.

Jiayi Zhang, Chang Liu, X. L. H.-L. Z. M. Y. Y. L. and Yan, J. (2022). A survey for solving mixed integer programming via machine learning. *Neurocomputing*.

Karp., R. M. (1972). Complexity of computer computations. *Springer*, pages 85–103.

Lauri Juho, Dutta Sourav, G. M. and Deepak, A. (2023). Learning fine-grained search space pruning and heuristics for combinatorial optimization. *Journal of Heuristics*, 29:313–347.

Liang Xin, W. S. et al. (2021). Neurolkh: Combining deep learning model with lin-kernighan-helsgaun heuristic for solving the traveling salesman problem. *Advances in Neural Information Processing Systems (NeurIPS)*.

Liu, Z. M., Ma, P. C., Wang, Y. X., et al. (2024). Kan 2.0: Kolmogorov-arnold networks meet science.

Lu Han, Li Zenan, W. R. et al. (2023). Roco: A general framework for evaluating robustness of combinatorial optimization solvers on graphs. In *International Conference on Learning Representations (ICLR)*.

Maryam Karimi-Mamaghan, Mehrdad Mohammadi, B. P. and Meyer, P. (2023). Learning to select operators in meta-heuristics: An integration of q-learning into the iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 304(3):1296–1330.

Maryam Karimi-Mamaghan, Mehrdad Mohammadi, P. J.-A. P. and Ahmadi, H. (2020). A learning-based metaheuristic for a multi-objective agile inspection planning model under uncertainty. *European Journal of Operational Research*, 285(2):513–537.

Maryam Karimi-Mamaghan, Mehrdad Mohammadi, P. M. et al. (2021). Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research*, 296.

Mosadegh H., G. S. F. et al. (2020). Stochastic mixed-model assembly line sequencing problem: Mathematical modeling and q-learning based simulated annealing hyper-heuristics. *European Journal of Operational Research*, 282(2):530–544.

Nath, A. and Kuhnle, A. (2024). A benchmark for maximum cut: Towards standardization of the evaluation of learned heuristics for combinatorial optimization. *arXiv, 2024*.

Nina Mazyavkina, Sergey Sviridov, S. I. and Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. *Computers Operations Research*, 134:105400.

Öztop H., Tasgetiren M. F., K. L. and Q.-K., P. (2020). A novel general variable neighborhood search through q-learning for no-idle flowshop scheduling. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE.

Quentin Cappart, Didier Ch´etelat, E. B. K. et al. (2021). Combinatorial optimization and reasoning with graph neural networks. *International Joint Conference on Artificial Intelligence.*

Scavuzzo, L., Aardal, K., Lodi, A., and Yorke-Smith, N. (2024). Machine learning augmented branch and bound for mixed integer linear programming.

Shen Yunzhuang, Sun Yuan, E. A. and Xiaodong, L. (2021). Learning primal heuristics for mixed integer programs. In *International Joint Conference on Neural Networks (IJCNN).*

Taoan Huang, Aaron M Ferber, Y. T.-B. D. et al. (2023). Searching large neighborhoods for integer linear programs with contrastive learning. *International Conference on Machine Learning (ICML).*

Tongyang Li, Yuexin Su, Z. Y. et al. (2024). Quantum approximate optimization algorithms for maxmimum cut on low-girth graphs. *arXiv, 2024.*

Tönshoff, J., Kisin, B., Lindner, J., and Grohe, M. (2023). One model, any csp: Graph neural networks as fast global search heuristics for constraint satisfaction. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Macau, SAR, China. International Joint Conferences on Artificial Intelligence Organization.

Xinyu Ye, G. Y. and Yan, J. (2023). Towards quantum machine learning for constrained combinatorial optimization: a quantum qap solver. *International Conference on Machine Learning (ICML).*

Ye H., Wang J., C. Z. F. B. et al. (2024). Reevo: Large language models as hyper-heuristics with reflective evolution. *Advances in Neural Information Processing Systems (NeurIPS).*

Ye H., Wang J., C. Z. L. H. and Y., L. (2023). Deep-aco: Neural-enhanced ant systems for combinatorial optimization. In *Advances in Neural Information Processing Systems (NeurIPS).*

Zhang Jiwei, Tayebi Dena, R. S. and Deepak, A. (2024). Learning to prune instances of steiner tree problem in graph. In *INOC.*

Zhuwen Li, Q. C. and Koltun, V. (2018). Combinatorial optimization with graph convolutional networks and guided tree search. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31.