Journal of Artificial Intelligence for Automation
https://www.sciltp.com/journals/jaia

Check for updates

Review

Learning-Based Optimization for Vehicle/Robot Routing Problems: A Survey

Jun Li

The Ministry of Education Key Laboratory of Measurement and Control of CSE, Southeast University, Nanjing 210096, China; j.li@seu.edu.cn

How To Cite: Li, J. Learning-Based Optimization for Vehicle/Robot Routing Problems: A Survey. *Journal of Artificial Intelligence for Automation* **2026**, *1*(2), 11. https://doi.org/10.53941/jaia.2026.1000011

Received: 19 January 2026
Revised: 22 May 2026
Accepted: 25 May 2026
Published: 11 June 2026

Abstract: Learning-based Neural Combinatorial Optimization (NCO) is an emerging paradigm for various vehicle/robot routing problems. It transitions solution strategies from manual heuristics to data-driven learning. This paper presents a systematic survey of deep learning-based approaches for route optimization. We first unify classical routing models and formulate reinforcement learning methods within a Markov Decision Process (MDP) framework. Existing literature is primarily classified into two categories: (1) end-to-end neural solvers, encompassing constructive and improvement-based methods, constraint-handling techniques, and various encoder–decoder or generative training schemes; and (2) scalability-oriented solvers, which leverage divide-and-conquer strategies to address large-scale routing problems. Finally, we discuss vital future research directions, including the integration of heuristic knowledge into NCO, large-scale multi-objective optimization, and automated modeling/solving. This survey offers a structured taxonomy of learning-based route optimization methods and discusses the potential for extending them to a broader class of combinatorial optimization problems.

Keywords: vehicle routing problem; combinatorial optimization; machine learning; neural network; learning-based optimization

1. Introduction

Vehicle/robot routing problems arise from transportation, manufacturing, and logistics [1]. Their representative formulations include Traveling Salesman Problem (TSP) [2], Multiple Traveling Salesman Problem (MTSP) [3], Vehicle Routing Problem (VRP) [4], and Colored Traveling Salesman Problem (CTSP) [5], all characterized by a discrete solution space.

Learning-based optimization algorithms have emerged as a promising approach to a variety of routing problems [6,7], propelled by advances in computing, communications, networking, and artificial intelligence. Conventional learning-based ones leverage reinforcement learning to optimize policies or utilize attention mechanisms to capture underlying problem features [6,8]. End-to-end deep learning-based optimization methods typically exploit encoder-decoder architectures, often in combination with graph neural networks or diffusion models [9,10]. These approaches learn direct mappings from problem instances to solutions and employ such strategies as masking and divide-and-conquer to tackle complex constraints and large-scale problems [11,12]. Recent developments in Large Language Models (LLMs) further extend these capabilities by enhancing semantic understanding and symbolic reasoning, providing new tools for the automated modeling and solving of complex vehicle/robot routing problems [13–17].

While some reviews have examined learning-based routing optimization methods, they often lack a cohesive structure [18]. Specifically, existing studies rarely address the connections between classical mathematical modeling, neural solver architectures, and scalable execution mechanisms. This results in a persistent gap in understanding the relationships between problem formulation and research needs. To address the mentioned

limitations, this study presents a unified perspective that links these paradigms, thereby providing a more systematic synthesis of the methods in this important field. We begin by presenting the mathematical programs for classic vehicle/robot routing problems in a unified notation and introduce a Markov Decision Process (MDP) framework as the theoretical basis for reinforcement learning-based approaches. Subsequently, we categorize existing learning-based methods into two paradigms: (1) End-to-end deep learning-based optimization one that utilizes encoder-decoder or generative models to map problem instances directly to solutions; and (2) Scalability-oriented one that employs learning-based strategies to tackle them via a divide-and-conquer approach. Finally, we examine key research questions for learning-based routing algorithms and outline directions for future investigations.

By systematically classifying and integrating the currently dispersed learning-based methods for vehicle/robot routing problems, this survey provides, for the first time, a comprehensive overview of the field and discusses its research/development trends. We aim to make the following contributions:

(1) A unified framework is established, integrating mathematical formulations, neural architectures, and execution strategies to address the current lack of a cohesive perspective in learning-based route optimization, to facilitate more consistent and scalable research for the community.
(2) A systematic taxonomy of deep learning approaches is proposed, providing a multi-dimensional analysis of model architectures and training algorithms to guide algorithm selection for different vehicle/robot routing problems.
(3) Key open challenges and future directions are identified, such as heuristic-enhanced NCO, large-scale multi-objective optimization, and automated modeling/solving, thereby offering insights into the future research/development of learning-based routing.

The remainder of this paper is organized as follows. Section 2 presents the mathematical formulation of vehicle/robot routing problems. Section 3 provides a taxonomy of main learning-based algorithms and elucidates their underlying principles. Section 4 discusses future research/development directions, and Section 5 concludes this survey paper.

2. Theoretical Foundations of Vehicle/Robot Routing Problems

2.1. Mathematical Programming Models

The vehicle/robot routing problems investigated mainly include TSP, MTSP, VRP, and their variants. To establish a unified theoretical basis for the learning-based methods reviewed later, this section first introduces the integer linear programming formulations of TSP and VRP. Several practical constraints are further incorporated in them.

Vehicle/robot routing problems can generally be defined on a graph $G = (V, E)$, where $V = \mathbb{Z}_n = \{0,1,2,\dots,n\}$ denotes the set of nodes (cities or customers) and E represents the connections between nodes. For symmetric routing problems such as the TSP, the graph can be treated as undirected. For VRP and its variants, the formulation can be naturally extended to directed graphs without loss of generality. $W = \left(w_{ij}\right)_{n\times n}$ represents the matrix of distance among nodes. Let $K = \mathbb{Z}_{m-1} = \{0,1,2,\dots,m-1\}$ denotes the set of salesmen (vehicles), each starting and ending at the depot, node 0. The decision variables x_{ij} or x_{ijk} indicate whether a salesman (vehicle) $k \in K$ travels from node i to node j, $i,j \in V$.

TSP seeks a minimum-cost Hamiltonian tour visiting each node exactly once. Its well-known Miller–Tucker–Zemlin (MTZ) formulation [2,19] is expressed as follows:

$$f = Min\sum_i\sum_j w_{ij}x_{ij} \tag{1}$$

$$\text{s.t. } \sum_i x_{ij} = 1, \forall j \in V, \tag{2}$$

$$\sum_j x_{ij} = 1, \forall i \in V, \tag{3}$$

$$u_i - u_j + (n-1)x_{ij} \le n-2, \forall i,j \in V\backslash\{0\}, \tag{4}$$

$$x_{ij} \in \{0,1\}, \forall i,j \in V, \tag{5}$$

where (1) is the objective function subject to visiting constraints (2) and (3) and subtour elimination constraints (4) and (5). (5) defines the range of decision variables. Let u_i denote an auxiliary ordering variable used in the Miller–Tucker–Zemlin (MTZ) formulation to eliminate subtours.

VRP generalizes TSP to a multi-vehicle setting and minimizes total travel cost subject to route, capacity, and depot constraints:

$$f = min\sum_{k}\sum_{i}\sum_{j} w_{ij}x_{ijk} \tag{6}$$

$$\text{s.t. } \sum_{i}\sum_{k} x_{ijk} = 1, \forall j \in V, \tag{7}$$

$$\sum_{j}\sum_{k} x_{ijk} = 1, \forall i \in V, \tag{8}$$

$$\sum_{j} x_{ijk} = \sum_{j} x_{jik}, \forall i \in V, \forall k \in K, \tag{9}$$

$$u_{ik} - u_{jk} + (n-1)x_{ijk} \leq n-2, \forall u_{ik} \in \mathbb{Z}_n, \forall i,j \in V\backslash\{0\},\ \forall k \in K, \tag{10}$$

$$\sum_{i\in V\backslash\{0\}} d_i \sum_{j\in V} x_{ijk} \leq Q_k, \forall k \in K, \tag{11}$$

$$\sum_{i\in V\backslash\{0\}} x_{i0k} = \sum_{j\in V\backslash\{0\}} x_{0jk} \leq 1, \forall k \in K, \tag{12}$$

$$x_{ijk} \in \{0,1\}. \tag{13}$$

In practical scenarios, vehicle/robot routing problems must not only adhere to basic routing connectivity and capacity constraints, but also real-world requirements such as customer service priorities and time windows, i.e.,

$$u_i \leq u_j + M\big(1 - x_{ijk}\big), \forall i,j \in V\backslash\{0\}, \forall k \in K, \forall p_i < p_j, \tag{14}$$

$$t_j \geq t_i + s_i + \tau_i - M\big(1 - x_{ijk}\big), \forall i,j \in V, \forall k \in K, \tag{15}$$

where M is a sufficiently large constant, p_i denotes the priority level of customer i, t_i is the service start time at customer i, s_i denotes the service time, and τ_i represents the travel time from i to j.

CTSP further extends VRP by leveraging colors to specify each city's accessibility by individual salesmen. A color matrix $C = \{c_{ij} = 0,1 | i \in \mathbb{Z}_{m-1}, j \in \mathbb{Z}_n\}$ defines such accessibility, where $c_{ij} = 1$ if salesman i can visit city j, and $c_{ij} = 0$ otherwise. Each city can be visited by the salesmen in its color set $S_i = \{k | c_{ik} = 1, k \in K\}$. The CTSP model inherits the VRP constraints in (6)–(11) and incorporates the following additional color (salesman-city matching) constraint to ensure that each city is visited by eligible salesmen only.

$$\sum_{j} x_{ijk} = \sum_{j} x_{jik} = 0, \forall i \in V, \forall k \notin c(i). \tag{16}$$

These models underpin exact solution methods for vehicle/robot routing problems and ensure that heuristic designs remain aligned with the original objective. Moreover, this unified mathematical framework provides the necessary groundwork for the MDP modeling and neural solver analysis presented in Sections 2.2 and 3, respectively.

2.2. Learning Policy Models

In contrast to traditional optimization algorithms, learning-based approaches aim to identify a parameterized policy model that defines a probabilistic mapping from a problem instance H to its solution space [20].

For a TSP instance with n nodes, a solution $\pi = \{\pi_0, \pi_1, \dots, \pi_{n-1}\}$ is a permutation of the nodes. The objective can be formulated as:

$$\underset{p}{Min}\ \mathbb{E}_{\pi\sim p(H)}[f(\pi)], \tag{17}$$

where $p(\cdot)$ defines the probability of sampling a candidate solution conditioned on *H*. $f(\cdot)$ is the tour length of the solution, i.e.,

$$f(\pi) = \|\pi_0 - \pi_{n-1}\| + \sum_{i=0}^{n-2} \|\pi_i - \pi_{i+1}\|. \tag{18}$$

When solving TSP via reinforcement learning, an agent typically constructs a tour by selecting cities one at a time. The problem can thus be modeled as a Markov Decision Process (MDP), where a state represents the

current partial tour, an action corresponds to choosing the next city, and reward is defined as the negative change in tour length, i.e.,

$$\min_{p} \mathop{\mathbb{E}}_{p(\pi|H)}[f(\pi)] = \mathop{Min}_{p} \mathop{\mathbb{E}}_{H}\left[f(\pi)\prod_{i=0}^{n-1} p(\pi_i|H,\pi_{0:i-1})\right], \tag{19}$$

where $p(\pi_i|H,\pi_{0:i-1})$ denotes an action policy model, representing the probability of selecting city π_i as the next node given the current state $\pi_{0:i-1}$.

Moreover, multi-routing problems can be formulated within the same sequential decision framework, where the insertion of depot nodes enables the division of multiple routes. The policy model simultaneously determines node order and route partitioning.

3. Learning-Based Optimization Methods for Vehicle/Robot Routing Problems

3.1. Taxonomy

Table 1 presents a taxonomy of deep learning-based algorithms, categorizing representative methods by their architectures, core features, problem applicability, computational efficiency, and training modes. This overview outlines the primary approaches detailed in subsequent sections. The taxonomy is structured around two main criteria. First, the ML vs. DL distinction reflects the evolution toward neural architectures with higher representational capacity. Second, the categorization into AR, NAR, and D&C highlights distinct solution generation and scalability strategies. Despite potential overlap, these categories, along with factors like training schemes and constraint handling, help clarify the underlying algorithmic behaviors.

Due to the heterogeneity in problem scales, benchmarks, experimental settings, and hardware platforms across various studies, a direct head-to-head comparison of the solving times and other performance metrics (e.g., gaps, stability, and generalization) reported in Table 1 is difficult. Therefore, Table 1 serves primarily as a summary of reported results from existing literature, rather than a definitive quantitative ranking of the methods.

Table 1. Taxonomy of Learning-Based Vehicle Routing Methods.

Taxonomy	Literature	Problem Size [1]	Model	Training Methods
ML	Ottoni et al. [8]	SOP-111 (53 min)	Q-table	Q-learning and SARSA
	Yang et al. [21]	MOTSP-200-1 (10 s)	Q-table	TD
DL Autoregressive	Bello et al. [22]	TSP-50-1 (-)	PointerNetwork + AR	SL
	Jin et al. [23]	TSP-500-1 (59.35 s)	Pointerformer + AR	REINFORCE
	Luo et al. [24]	TSP-1k-1 (1.6 min–7 h) CVRP-1k (1.6 min–8 h)	Light Encoder-Heavy Decoder + AR	DA-based SL
	Kool et al. [25]	TSP-100-1 (6 s–1 h) CVRP-100 (8 s–2 h)	AM + AR	REINFORCE
	Kwon et al. [26]	TSP-100-1 (2 s–1 min) CVRP-100 (3 s–2 min)	AM + POMO + AR	REINFORCE
	Kwon et al. [27]	ATSP-100-1 (34 s–1 h) FFSP-100 (27 s–23 min) TSP-100-1 (34 s–1 h)	MatNet + AR	REINFORCE
	Li et al. [28]	PDTSP-101-1 (5.7 min) m-PDTSP-101-1 (5.9 min)	MHSA + AR	REINFORCE
	Zhang et al. [29]	TSP-1k-1 (0.75 s–0.77 s) CVRP-1k (0.72 s–0.73 s)	GFlowNet + AR	Adversarial Training
DL Non-Autoregressive	Hudson et al. [30]	TSP-100-1 (10 s)	GNN + NAR	SL
	Sun et al. [10]	TSP-1k-1 (25 min–48 min) MIS-800 (26.7 min)	Graph-based Diffusion Solvers + NAR	SL
	Fu et al. [31]	TSP-10k-1 (1.7 h)	Att-GCRN + MCTS + NAR	SL + RL
	Kool et al. [32]	TSP-100-1 (1 h–2.5 h) VRP-100 (48.5 h)	GCN + DP + NAR	REINFORCE
	Chalumeau et al. [33]	TSP-200-1 (70 min) CVRP-200 (100 min) JSSP-20×15 (8 h)	Transformer + Latent Space Search + NAR	REINFORCE
	Lu et al. [34]	CVRP-1k (-)	AM + MLP + NAR	REINFORCE
	Choo et al. [35]	TSP-1k-1 (14 min–30 h) CVRP-1k (9 min–50 h)	AM + SGBS + NAR	REINFORCE
	Wang et al. [36]	TSP-100-1 (8 min–38 min)	GAN + NAR	Actor-Critic

Table 1. *Cont.*

Taxonomy	Literature	Problem Size [1]	Model	Training Methods
DL-Based Divide-and-Conquer	Pan et al. [37]	TSP-10k-1 (0.72 s–6.88 s)	CNN + Transformer + D&C	PPO and REINFORCE
	Zong et al. [38]	CVRP-2k (90 s)	LSTM + AM + D&C	REINFORCE
	Pan et al. [39]	CVRP-10k (3.4 min–5.1 min)	GNN + Transformer + D&C	REINFORCE and SL
	Ye et al. [9]	TSP-10k-1 (2.8 min) CVRP-7k (2.4 s–5.8 s) PCTSP-5k-1 (3.7 min–16 min) ATSP-1k-1 (15 s)	GNN + AM + D&C	REINFORCE
	Ma et al. [40]	TSP-1k-1 (6.55 min–38 min)	GNN + LSTM+ AM + D&C	REINFORCE

Notes: [1] This column details the problem type, scale, the number of salesmen or vehicles (as reported in the original papers), and the actual solving time. For example, "MTSP-111-3 (10 s)" indicates an MTSP instance with 111 nodes and 3 salesmen, solved in 10 s. Q-learning, SARSA, PPO, REINFORCE, and Actor-Critic are typical RL algorithms.

3.2. Neural Combinatorial Optimization Algorithms

Learning-based combinatorial optimization methods fall generally into two classes: constructive and improvement-based. The former generates solutions from scratch by progressively outputting node sequences. Bello et al. [22] first applied pointer networks (PtrNet) to TSP and the knapsack problem (KP), which was later extended by Nazari et al. [41] with separate encoders for static and dynamic inputs. As shown in Figure 1, PtrNet utilizes an encoder for structural feature extraction and a decoder that diverges into two main paradigms: autoregressive (AR) models that sequentially construct tours node-by-node, and non-autoregressive (NAR) models that output the entire solution in a single parallel forward pass.

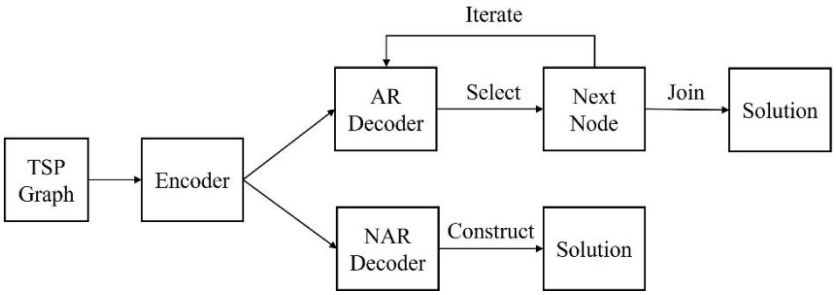

Figure 1. AR vs. NAR decoding in PtrNet.

AR variants include (a) a graph pointer network [30] that solves TSPs with tens of thousands of nodes, and (b) a Transformer-based PtrNet [23] with high scalability. Another AR variant is the Attention Method (AM) [23], which replaces recurrence with attention and is trained using REINFORCE. It achieves near-optimal TSP solutions and generalizes to VRP, the ordering problem (OP), and the prize-collecting TSP. To improve exploration efficiency, Kwon et al. [26] proposed Policy Optimization with Multiple Optima (POMO), which extends AM to a multi-optima framework. This work motivated subsequent AR and NAR methods such as Matrix-Encoded Network [27], heterogeneous attention for PDP [28], and a reinforced NAR method using a heavy encoder [24]. Another branch of AR methods utilizes a step-wise deep learning model [42], which improve constructive routing solvers by re-embedding only the remaining relevant nodes at each decoding step, thereby enhancing TSP and CVRP solution quality for both PtrNet- and attention-based models. Chalumeau et al. [33] further proposed a meta-policy for online adaptation. Meanwhile, the attention-based autoregressive paradigm is expanding into arc-routing problems such as the Chinese Postman Problem [43]. Related explorations include direction-aware attention mechanisms [44] and deep reinforcement learning methods integrating adjacency attention [45], demonstrating the applicability of this paradigm to diverse routing problems.

NAR approaches bypass sequential decoding by directly predicting structural information. Fu et al. [31] fused graph convolutions with attention to generate edge-probability heatmaps that guide Monte Carlo Tree Search (MCTS). Ye et al. [9] proposed Global and Local Optimization Policies (GLOP), a divide-and-conquer VRP solver utilizing heatmap-guided decomposition. Other NAR models include Generative Inverse Reinforcement Learning (GIRL) [36] for TSP and CVRP, Attention Feature Guided Network (AFGN) in an adversarial Generative Flow Network (GFlowNet) framework [29], and Diffusion Solvers for Combinatorial Optimization (DIFUSCO) [10]. As the first graph diffusion solver for combinatorial optimization problems (COPs), DIFUSCO demonstrates robust cross-scale generalization. Namely, the model trained on TSPs with 50 cities remains highly effective on instances with up to 10k cities.

Improvement-based neural solvers iteratively refine an initial solution and often outperform constructive approaches when provided with enough computational time [30,32,34,46]. Lu et al. [34] introduced the Learn-to-Improve (L2I) framework, exploiting an RL-based controller to coordinate improvement and perturbation operators. L2I can outperform baseline heuristics such as LKH-3 [40]. Kool et al. [32] combined GNNs with dynamic programming through heatmap-guided pruning. Hudson et al. [30] predicted global edge regret by using GNNs and embed this prediction into Guided Local Search (GNN-GLS) to escape local optima. Choo et al. [35] developed Simulation-Guided Beam Search (SGBS) to rank candidates effectively.

While some ML-based methods leverage classical reinforcement learning, they fundamentally rely on iterative optimization over the cost matrix. Consequently, they resemble traditional heuristics and often face computational bottlenecks. For instance, certain approaches require nearly 30 min to solve an SOP instance with only 111 nodes [8]. In contrast, AR models construct solutions incrementally, maintaining well computational efficiency for problems with up to approximately 1000 nodes [29]. NAR models generate complete solutions in a single forward pass, thereby avoiding the compounding errors inherent in AR decoding [30]. However, the high memory footprint of NAR models limits their scalability to large-scale instances [10,35]. Furthermore, while post-processing techniques like Beam Search and MCTS enhance solution quality, the resulting computational overhead can offset the inherent speed advantage of NAR architectures [31,35].

3.3. Constraint Handling Methods

Real-world vehicle/robot routing problems are often more complex than their idealized counterparts due to various practical constraints. In neural combinatorial optimization (NCO), masks are commonly used to enforce such constraints during solution construction. For instance, in TSP, visited nodes are masked to ensure each city is selected only once [6]. In CVRP, feasibility is usually maintained by combining visited-node masks and capacity masks through a logical AND operation [23,26,47].

However, stepwise masking may not guarantee global feasibility for more complex constraints, as a locally valid selection can lead to infeasible solutions later. Ensuring feasibility through masking requires exploring all subsequent states, making masking process itself NP-hard. Traditional methods, e.g., Lagrangian relaxation and penalty-based approaches, remain effective for COPs. Learning-based enhancements include NeuOpt [48], which adds a constraint regularization term to the reward function, outperforming mask-based methods for TSP and CVRP. Tang et al. [11] combined Lagrangian relaxation with constrained policy optimization to solve different VRP variants. Bi et al. [49] proposed a proactive framework to avoid infeasible decisions, using Lagrange multipliers to guide the construction process. This method has been applied to the TSP with Time Windows (TSPTW) and the TSP with Deadlines (TSPDL).

3.4. Training Methods

The choice of training strategy is critical to model performance and generalization in deep learning-based combinatorial optimization. Current approaches include supervised learning, unsupervised learning, RL, transfer learning, and meta-learning, which differ in data requirements, generalization behavior, and optimization objectives.

3.4.1. Supervised Learning (SL)

SL trains a neural solver using labeled instances to imitate optimal or near-optimal solutions. PtrNet [6] is an early representative work that formulates combinatorial optimization problems as sequence-generation tasks trained on reference TSP tours. Subsequent studies have focused on improving the generalization of supervised solvers. For example, Luo et al. [24] proposed LEHD, which uses an enlarged training set with mixed instance sizes and distributions, incorporating curriculum learning through resampling. These designs aim to reduce overfitting and enhance robustness for larger instances. To reduce dependence on expensive labels, Prates et al. [50] developed a lightly supervised GNN framework that constructs weak labels from dual instances with different cost thresholds based on optimal objective values, rather than requiring complete optimal tours.

Although SL can achieve strong performance, its heavy dependence on high-quality labels poses significant challenges in terms of computational cost and scalability. Furthermore, existing datasets are often limited in scale and diversity. To address this, Milan et al. [51] introduced a weak-supervision strategy that incorporates the objective function cost into the training process, allowing the model to differentiate solution quality. Kotary et al. [52] further improved training stability by selecting instances with consistent solution structures, thereby enhancing model generalization. Despite these advancements, the inherent cost of obtaining optimal labels has shifted the research focus toward label-free training methods, particularly unsupervised and reinforcement learning.

3.4.2. Unsupervised Learning (UL) and Reinforcement Learning (RL)

UL methods learn optimization strategies by exploring structural patterns in unlabeled data. Karalias and Loukas [53] proposed an unsupervised framework inspired by the Erdős probabilistic method. They designed a differentiable probabilistic penalty loss to address the non-differentiability of many combinatorial objectives. In their method, a GNN learns a probability distribution over node subsets by minimizing a continuous loss, and feasible solutions are then obtained through deterministic decoding based on conditional expectation.

Wang et al. [54] further studied unsupervised training from the perspective of objective relaxation. Their framework relaxes hard combinatorial constraints and builds a continuous loss that remains close to the original objective, enabling gradient-based training without labeled data. In a follow-up study, they introduced meta-learning into unsupervised combinatorial optimization [55]. By constructing diverse meta-task distributions, the model can learn more transferable heuristics and mitigate the overfitting issues commonly observed in conventional unsupervised methods.

As an alternative label-free paradigm, RL learns a policy through repeated interactions with the environment and updates the model using reward signals, unlike UL methods that rely on continuous relaxations. In routing optimization, RL is often used in constructive solvers, where the policy learns to select nodes or edges step by step. Bello et al. [22] proposed an end-to-end policy-gradient method with an RNN encoder, eliminating the need for labeled solutions. Kool et al. [12] developed a self-attention encoder-decoder model trained by REINFORCE, where greedy rollout baselines are used to reduce variance. Kwon et al. [25] proposed POMO, which performs multiple rollouts from different starting points and uses their average reward as a baseline, enhancing training stability. Kim et al. [56] further exploited problem symmetry in Sym-NCO. By introducing a symmetry-aware advantage into REINFORCE, the method improves sample efficiency and exploration without requiring additional data.

However, RL often suffers from sparse rewards and unstable convergence. Sun et al. [10] addressed this by developing DIFUSCO, a graph diffusion solver. Instead of optimizing a sparse reward directly, DIFUSCO learns to reconstruct perturbed graph structures, which reduces the impact of reward sparsity and demonstrates robust generalization. UTSP [57] adopts a proxy loss that combines path length and feasibility penalties over synthetic TSP instances. This allows the model to optimize more directly in the objective space, although additional heuristic correction is still required to ensure feasible tours.

By reducing the reliance on exact solvers, UL and RL offer scalable alternatives to supervised methods. Nevertheless, challenges such as sample inefficiency, training instability, and limited cross-distribution generalization remain. To address these issues, recent research has increasingly focused on transfer learning and meta-learning, where knowledge learned from one problem setting can be adapted to another.

3.4.3. Transfer Learning (TL) and Meta-Learning

Unlike UL and RL, TL and meta-learning aim to facilitate cross-problem transferability and rapid task adaptation. TL focuses on reusing knowledge from a source COP to accelerate learning on a related target problem. Souza et al. [58] designed a transfer reinforcement learning framework for vehicle routing problems. By transferring policy parameters or previous experience between related tasks, such as the asymmetric traveling salesman problem (ATSP), the method reduces retraining costs and improves learning efficiency on new COPs. In the context of knowledge distillation, Bi et al. [59] trained a teacher model on heterogeneous routing instances and then use it to guide a smaller student model by matching their output distributions. Consequently, the student model can inherit more general routing heuristics than those learned from standard RL training alone.

Meta-learning approaches generalization by focusing on how a model should adapt to new problem instances, rather than merely transferring parameters. Kanda et al. [60] applied this concept to algorithm selection, learning the relationship between TSP meta-features and solver performance to quickly identify a suitable algorithm for a new instance. Son et al. [61] proposed Meta-SAGE, which combines meta-learning with curriculum learning to support scale-adaptive guided search. Tested on several vehicle/robot routing problems, including TSP, CVRP, prize-collecting TSP (PCTSP), and the orienteering problem (OP), Meta-SAGE helps the model transfer from small instances to larger ones while maintaining a guided search process.

Recently, preference optimization (PO) has emerged as an alternative training scheme for neural combinatorial optimization. Instead of relying solely on scalar reward values, PO introduces preference information into policy learning. Pan et al. [62], for example, proposed a PO-based framework in which solution pairs are compared to train a reward model that reflects expert preferences. The learned reward model is then used to update the policy. This allows the solver to consider not only explicit optimization objectives but also higher-level preferences that are difficult to express as a single numerical reward.

While TL, meta-learning, and PO paradigms improve the generalization and practical applicability of learning-based COP solvers, they still face inherent limitations. Specifically, TL and meta-learning depend on task similarity, and meta-training can be computationally expensive [63], highlighting the need for more scalable training strategies.

3.5. Complex Large-Scale Optimization Problems

As instance sizes scale, the inherent NP-hardness of vehicle/robot routing problems makes direct exploration of the solution space computationally intractable. Consequently, classical algorithmic frameworks frequently adopt divide-and-conquer strategies to limit search space. Taillard et al. [64,65] established that optimizing decomposition granularity can restrict heuristic search complexity to an almost linear growth rate. The standard workflow, depicted in Figure 2, partitions a large-scale instance into structured subproblems, solves them independently, and merges these local solutions into a global route [9]. For multi-routing problems, a common strategy involves clustering the instance into parallel TSP-like components. Similarly, an individual TSP instance can be decomposed into smaller instances of Shortest Hamiltonian Path Problem (SHPP) to further accelerate computation.

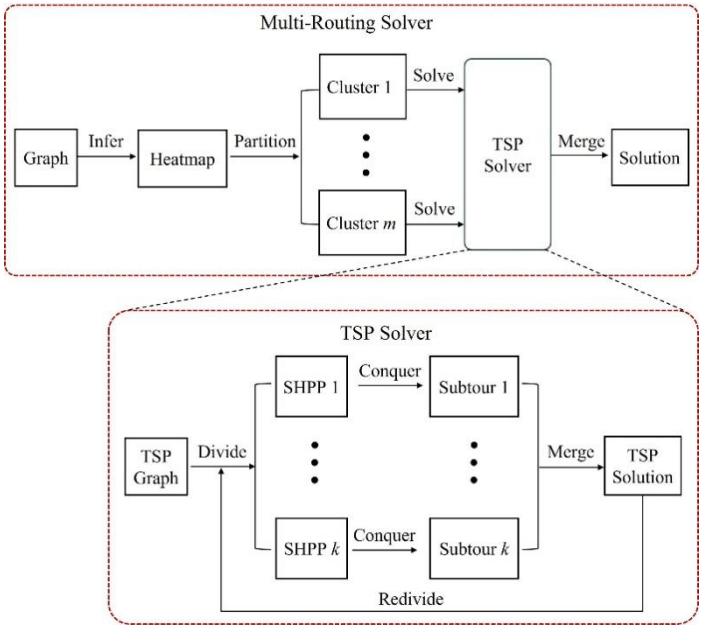

Figure 2. Deep learning-driven divide-and-conquer scheme.

This decomposition approach has been adapted for learning-based routing solvers. To handle VRP instances with up to 1000 nodes. Zong et al. [38] incorporated divide-and-conquer into a neural solver, executing a continuous partition-solve-merge pipeline. Pan et al. [37] proposed H-TSP, a hierarchical reinforcement learning framework for large-scale TSP. In this framework, a high-level policy divides the original TSP into several open-loop subproblems, while a pretrained Transformer-based low-level policy solves each subproblem before the partial solutions are merged. Zheng et al. [12] presented a heatmap-based global decomposition method for sequential optimization problems, including TSP, CVRP, MTSP, and OP. Coupled with a constructive neural solver for subproblems, their method can be applied to routing instances with more than 100,000 nodes. Similarly, Ye et al. [9] leveraged Graph Neural Networks (GNNs) to capture node correlations, extracting TSP sub-tasks via greedy or sampling strategies. Since errors in the partitioning stage may affect the final global solution, Pan et al. [39] further designed a hierarchical decomposition strategy for CVRP, aiming to mitigate local feature sensitivity and reduce error accumulation in large-scale settings.

In summary, learning-based routing methods have addressed vehicle/robot routing problems from several technical directions. AR models construct solutions step by step, which aligns with sequential decision-making and is useful for dynamic or online routing scenarios. NAR models and divide-and-conquer methods are typically employed to address scalability concerns, as they reduce sequential dependence through parallel decoding or problem decomposition. Beyond scalability, enforcing feasibility remains a critical challenge in applied contexts, particularly for problems involving time windows [66] and vehicle capacity constraints [67]. Consequently, many neural solvers incorporate constraint-handling components, such as masking mechanisms or feasibility-aware state representations. Furthermore, training remains a significant bottleneck for neural routing models, particularly on

complex or large-scale instances. To improve convergence stability and solution quality, researchers have explored reinforcement learning, supervised pretraining, and hybrid training schemes.

4. Future Directions

Learning-based optimization has shifted the focus of routing research toward data-driven pattern extraction rather than relying solely on manually designed heuristics. However, current NCO approaches still face clear limitations when they are applied to large-scale, highly constrained, or multi-objective vehicle/robot routing problems. In terms of solution quality, robustness, and generalization, they do not always match well-established heuristic algorithms. In large-scale scenarios, one important difficulty is to capture global structure. Although divide-and-conquer methods can reduce problem size, many of them still depend on heuristic partitioning rules. The relationships among subproblems are often not explicitly modeled, which may weaken the consistency of the final global solution. Moreover, many neural solvers lack efficient local search or repair mechanisms, so their ability to refine solutions at a detailed level remains limited. For vehicle/robot routing problems with complex constraints, masking mechanisms are effective means of guaranteeing solution feasibility. However, when constraints involve long-term dependency, such as time windows, a locally feasible action does not necessarily lead to a globally feasible route. In addition, the interaction among multiple constraints is often not fully represented, making it difficult for the model to learn the true boundary of the feasible solution space. In multi-objective routing, most NCO methods are still designed around single-objective formulations. This makes it difficult to model trade-offs among conflicting objectives or to generate diverse solutions along the Pareto front. Another practical issue is that model design, constraint representation, and strategy selection still require considerable expert experience. Consequently, future research should prioritize three directions: enhancing feasibility and robustness through heuristic-enhanced NCO, improving scalability and Pareto diversity in large-scale multi-objective optimization, and advancing automated problem modeling and solver construction with reduced expert intervention.

4.1. Heuristic-Enhanced NCO

Learning-based solvers often underperform compared to established classical heuristics in terms of accuracy and stability, particularly on large-scale instances. To bridge this performance gap, researchers are increasingly embedding heuristic knowledge across various stages of neural optimization, throughout model design, training, and inference. For example, population-based heuristics can effectively guide hyperparameter tuning [68], strategy selection [69–71], and reward shaping [72]. In addition, high-quality heuristic solutions provide robust supervisory signals for pre-training or fine-tuning neural architectures. The primary challenge lies in incorporating this knowledge without overfitting neural solvers to specific instance distributions or significantly increasing inference costs.

At the inference stage, deterministic search procedures, such as greedy search and tree search, can make better use of neural outputs, including attention scores, heatmaps, and node-selection probability matrices [32,35,73]. Compared with pure stochastic sampling, these procedures often provide a more controlled way to explore promising solution regions. In large-scale routing, neural divide-and-conquer methods also adapt ideas from classical decomposition. They reduce the original problem into smaller subproblems, while attempting to preserve enough global information for solution reconstruction. Further investigation is needed to determine how local search can be selectively applied to structurally critical parts of a solution, rather than uniformly increasing search effort across the entire route.

Given their theoretical maturity, heuristic algorithms remain valuable. Their combination with neural solvers may improve solution quality, scalability, and robustness for complex vehicle/robot routing problems. In VRP applications, this direction is especially relevant for constrained variants, such as VRP with time windows [74]. Local search operators, repair rules, or population-based mechanisms can help neural models maintain feasibility while searching in a large solution space. Nevertheless, achieving this integration remains highly non-trivial; handcrafted heuristics must complement neural policies without introducing excessive structural complexity or computational overhead. More specifically, for globally coupled constraints such as time windows, jointly learning feasibility-preserving heuristic operators with neural policies, instead of relying on them solely as post-processing repair tools, remains an open problem. In practice, designing effective coordination mechanisms between neural decision policies and heuristic operators is essential to balance computational efficiency and solution quality. For practical VRP deployment, improving feasibility and stability under complex constraints is a critical priority within heuristic-enhanced NCO.

4.2. Learning-Based Large-Scale Multi-Objective Optimization

Large-scale multi-objective routing optimization remains a significant challenge for learning methods. This difficulty primarily stems from the exponential growth of the solution space coupled with inherently conflicting objectives [75]. Divide-and-conquer provides a practical way to reduce this difficulty. An effective decomposition should divide the original problem into subproblems that can be solved in parallel, while still retaining the key structural information needed to construct a consistent global solution [76]. Solution-space reduction is another commonly used idea. By learning or refining the node connection graph, a neural model can focus more on potentially useful edges and avoid spending computation on clearly poor search regions.

Another approach to handle multi-objective optimization is scalarization. Weighted-sum methods [77], Chebyshev metrics [78], and penalty-based boundary intersection [79] transform a multi-objective problem into several scalar subproblems, which can then be solved by neural networks [21]. Neighborhood-based parameter transfer [80] can also be used to share useful information among related subproblems, improving generalization and reducing training cost. In this setting, each scalarized problem usually corresponds to one solution on the Pareto front, and a group of such solutions is used to approximate the front. However, whether neural models are able to directly approximate the Pareto front in a diverse and stable manner, rather than relying mainly on independently solving a set of scalarized subproblems, remains unresolved. Multi-agent reinforcement learning offers a framework for this, since the interaction among agents can naturally represent cooperation, competition, and trade-offs among objectives, helping generate diverse non-dominated solutions [81,82]. For high-dimensional and expensive multi-objective problems, surrogate-assisted methods based on autoencoders or inverse models can reduce the cost of solution evaluation [83–85]. Evolutionary multitask optimization provides another option by transferring knowledge across related tasks and improving robustness through domain adaptation [86,87].

Within VRP domains, large-scale multi-objective optimization is highly relevant to real logistics systems. Practical routing decisions often need to balance transportation cost, service time, energy consumption, and other operational factors. Learning-based methods can use flexible policy models to approximate a set of Pareto-optimal solutions [88]. Nevertheless, it remains difficult to coordinate several conflicting objectives while keeping the method scalable [89]. Such methods are particularly useful for large logistics networks, where decision makers may need multiple routing alternatives rather than a single solution. In practice, the design of objective aggregation strategies and stable training under multiple reward signals are both important. From a practical perspective, generating scalable and diverse trade-off solutions is often more valuable than pursuing exact reconstruction of the full Pareto front.

4.3. Automated Modeling and Solving

Automated modeling and solving are increasingly explored in routing optimization, because they may reduce the amount of manual work required in formulating problems and designing solvers. However, this direction is still sensitive to data quality and problem descriptions. Currently, designing robust modeling templates and configuring solver components still demands substantial domain expertise, which increases deployment costs and restricts adaptability to novel scenarios [90]. Therefore, the main challenge is not only automation itself, but also the correctness, consistency, and verifiability of automatically generated models and solver pipelines.

Routing problems inherently exhibit sequential planning and set-partitioning structures, making them well-suited for template-based modeling and modular algorithm design. First, based on optimization goals and common constraints, key decision variables and operation rules can be abstracted to form model templates and a standardized symbolic system. Second, vehicle/robot routing problems possess inherent structural characteristics, including permutation invariance and geometric regularities, which make them well suited for data augmentation. By generating diverse yet structurally consistent instances, data augmentation can effectively enrich the training distribution and reduce reliance on limited datasets. Then, with prompt engineering and a library of predefined templates, LLMs can translate natural-language descriptions into structured mathematical models, enabling standardized and automated modeling [91]. Recent applications in power system optimization [92] and multi-robot task allocation [93], as well as routing-specific studies such as *From Words to Routes* [14], have demonstrated this potential. Validating such automatically generated models prior to optimization, particularly regarding constraint completeness and logical consistency, remains a critical hurdle.

Automated solver design can also be developed in a modular way. Once a routing problem is expressed through a template-based model, the corresponding solver can be divided into several functional modules. Mathematical programming methods, heuristic operators, and learning-based components can then be organized as reusable solver blocks. Under this setting, LLMs act as high-level coordinators, dynamically selecting, synthesizing, or executing strategies tailored to specific instance configurations [81,94]. For example, ARS [15]

enables automatic construction of routing solvers, while LLM-powered neural solvers integrate LLMs with neural combinatorial optimization models to enhance generalization across diverse VRP variants [95].

Building upon these developments, recent studies have further explored LLM-enhanced routing from complementary perspectives. DROC [96] utilizes LLMs to retrieve and decompose problem-specific constraints, improving the handling of complex routing scenarios. LLM-A [16] explores the use of LLM-guided decision making to enhance heuristic search, collectively forming a semantic-driven, component-oriented automated solving system. Together with template-based modeling and modular solver construction, these studies illustrate the potential of LLMs to bridge problem modeling, constraint reasoning, and solver design, facilitating more adaptive routing solutions.

5. Conclusions

Learning-based optimization has driven the transition from manual designed heuristics to data-driven methods in vehicle/robot routing. This survey presents the field's theoretical foundations through unified formulations and an MDP-based reinforcement learning framework. We focus on two major classes of learning-based routing algorithms, i.e., end-to-end neural solvers and scalable divide-and-conquer frameworks, examining their architectures, core features, constraint management, and training schemes. Alongside our review of existing work, we identify critical challenges and future research, such as heuristic-enhanced NCO, large-scale multi-objective optimization, and automated modeling and solving. Ultimately, this unified perspective is intended to serves as reference for developing advanced routing solvers and to support the extension of learning-based techniques to a wider range of combinatorial optimization problems.

Funding

This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFF0500904.

Conflicts of Interest

The author declares no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the author used Grammarly and Gemini to polish the language. After using them, the author reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Toth, P.; Vigo, D. *Vehicle Routing: Problems, Methods, and Applications*, 2nd ed.; SIAM: Philadelphia, PA, USA, 2014; pp. 2–5.
2. Matai, R.; Singh, S.; Mittal, M.L. Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches. In *Traveling Salesman Problem: Theory and Applications*; Davendra, D., Ed.; InTech: Rijeka, Croatia, 2010; pp. 1–24.
3. Cheikhrouhou, O. A Comprehensive Survey on the Multiple Traveling Salesman Problem: Applications, Approaches and Taxonomy. *Comput. Sci. Rev.* **2021**, *40*, 100369.
4. Braekers, K. The Vehicle Routing Problem: State of the Art Classification and Review. *Comput. Ind. Eng.* **2016**, *99*, 300–313.
5. Li, J.; Zhou, M.; Sun, Q.; et al. Colored Traveling Salesman Problem. *IEEE Trans. Cybern.* **2015**, *45*, 2390–2401.
6. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer Networks. In *Advances in Neural Information Processing Systems 28, Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015*; Cortes, C., Lawrence, N., Lee, D., et al., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2015; pp. 2692–2700.
7. Zhou, F.; Lischka, A.; Kulcsár, B.Z.; et al. Learning for Routing: A Guided Review of Recent Developments and Future Directions. *Transp. Res. Part E Logist. Transp. Rev.* **2025**, *202*, 104278.
8. Ottoni, A.L.C.; Nepomuceno, E.G.; de Oliveira, M.S.; et al. Tuning of Reinforcement Learning Parameters Applied to SOP Using the Scott-Knott Method. *Soft Comput.* **2020**, *24*, 4441–4453.

9. Ye, H.; Wang, J.; Liang, H.; et al. GLOP: Learning Global Partition and Local Construction for Solving Large-Scale Routing Problems in Real-Time. In Proceedings of the 38th AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; pp. 16268–16276.
10. Sun, Z.; Yang, Y. DIFUSCO: Graph-Based Diffusion Solvers for Combinatorial Optimization. In Proceedings of the 37th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; pp. 3706–3731.
11. Tang, Q.; Kong, Y.; Pan, L.; et al. Learning to Solve Soft-Constrained Vehicle Routing Problems with Lagrangian Relaxation. *arXiv* **2022**, arXiv:2207.09860.
12. Zheng, Z.; Zhou, C.; Tong, X.; et al. UDC: A Unified Neural Divide-and-Conquer Framework for Large-Scale Combinatorial Optimization Problems. In Proceedings of the 38th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 9–15 December 2024; pp. 6081–6125.
13. Naveed, H.; Khan, A.U.; Qiu, S.; et al. A Comprehensive Overview of Large Language Models. *ACM Trans. Intell. Syst. Technol.* **2025**, *16*, 1–72.
14. Huang, Z.; Shi, G.; Sukhatme, G.S. From Words to Routes: Applying Large Language Models to Vehicle Routing. *arXiv* **2024**, arXiv:2403.10795.
15. Li, K.; Liu, F.; Wang, Z.; et al. ARS: Automatic Routing Solver with Large Language Models. *arXiv* **2025**, arXiv:2502.15359.
16. Meng, S.; Wang, Y.; Yang, C. F.; et al. LLM-A: Large Language Model Enhanced Incremental Heuristic Search on Path Planning. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, FL, USA, 12–16 November 2024; pp. 1087–1102.
17. Cao, L.; Wang, M.; Xiong, X. A Large Language Model-Enhanced Q-Learning for Capacitated Vehicle Routing Problem with Time Windows. *arXiv* **2025**, arXiv:2505.06178.
18. Bogyrbayeva, A.; Meraliyev, M.; Mustakhov, T.; et al. Learning to Solve Vehicle Routing Problems: A Survey. *arXiv* **2022**, arXiv:2205.02453.
19. Dantzig, G. Solution of a Large-Scale Traveling-Salesman Problem. *J. Oper. Res. Soc. Am.* **1954**, *2*, 393–410.
20. Zhang, J.; Liu, C.; Li, X.; et al. A Survey for Solving Mixed Integer Programming via Machine Learning. *Neurocomputing* **2023**, *519*, 205–217.
21. Yang, A.; Liu, Y.; Zou, J.; et al. Decomposed Multi-Objective Method Based on Q-Learning for Solving Multi-Objective Combinatorial Optimization Problems. In Proceedings of the International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2023), Singapore, 8–10 December 2023; pp. 59–73.
22. Bello, I.; Lee, C.K.; Tsang, Y.P. Neural Combinatorial Optimization with Reinforcement Learning. *arXiv* **2016**, arXiv:1611.09940.
23. Jin, Y.; Ding, Y.; Pan, X.; et al. Pointerformer: Deep Reinforced Multi-Pointer Transformer for the Traveling Salesman Problem. In Proceedings of the 37th AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–13 February 2023; pp. 8132–8140.
24. Luo, F.; Lin, X.; Liu, F.; et al. Neural Combinatorial Optimization with Heavy Decoder: Toward Large Scale Generalization. In Proceedings of the 37th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; pp. 8845–8864.
25. Kool, W.; Van Hoof, H.; Welling, M. Attention, Learn to Solve Routing Problems! In Proceedings of the 7th International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, 6–9 May 2019.
26. Kwon, Y.-D.; Choo, J.; Kim, B.; et al. POMO: Policy Optimization with Multiple Optima for Reinforcement Learning. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; pp. 21188–21198.
27. Kwon, Y.D.; Choo, J.; Yoon, I.; et al. Matrix Encoding Networks for Neural Combinatorial Optimization. In Proceedings of the 35th International Conference on Neural Information Processing Systems, Virtual, 6–14 December 2021; pp. 5138–5149.
28. Li, J. Heterogeneous Attentions for Solving Pickup and Delivery Problem via Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 2306–2315.
29. Zhang, N.; Yang, J.; Cao, Z.; et al. Adversarial Generative Flow Network for Solving Vehicle Routing Problems. In Proceedings of the 13th International Conference on Learning Representations (ICLR 2025), Singapore, 24–28 April 2025.
30. Hudson, B.; Li, Q.; Malencia, M.; et al. Graph Neural Network Guided Local Search for the Traveling Salesperson Problem. *arXiv* **2021**, arXiv:2110.05291.
31. Fu, Z.-H.; Qiu, K.B.; Zha, H. Generalize a Small Pre-Trained Model to Arbitrarily Large TSP Instances. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; pp. 7474–7482.
32. Kool, W.; van Hoof, H.; Gromicho, J.; et al. Deep Policy Dynamic Programming for Vehicle Routing Problems. In *Lecture Notes in Computer Science*; Volume 13292; Springer: Cham, Switzerland, 2022; pp. 190–213.

33. Chalumeau, F.; Surana, S.; Bonnet, C.; et al. Combinatorial Optimization with Policy Adaptation Using Latent Space Search. In Proceedings of the 37th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; pp. 7947–7959.
34. Lu, H.; Zhang, X.; Yang, S. A Learning-Based Iterative Method for Solving Vehicle Routing Problems. In Proceedings of the 7th International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, 6–9 May 2019.
35. Choo, J.; Kwon, Y.D.; Kim, J.; et al. Simulation-Guided Beam Search for Neural Combinatorial Optimization. In Proceedings of the 36th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; pp. 8760–8772.
36. Wang, Q.; Hao, Y.; Zhang, J. Generative Inverse Reinforcement Learning for Learning 2-Opt Heuristics Without Extrinsic Rewards in Routing Problems. *J. King Saud Univ. Comput. Inf. Sci*. **2023**, *35*, 101787.
37. Pan, X.; Jin, Y.; Ding, Y.; et al. H-TSP: Hierarchically Solving the Large-Scale Traveling Salesman Problem. In Proceedings of the 37th AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–13 February 2023; pp. 9345–9353.
38. Zong, Z.; Wang, H.; Wang, J.; et al. RBG: Hierarchically Solving Large-Scale Routing Problems in Logistic Systems via Reinforcement Learning. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 4648–4658.
39. Pan, Y.; Liu, R.; Chen, Y.; et al. Hierarchical Learning-Based Graph Partition for Large-Scale Vehicle Routing Problems. *arXiv* **2025**, arXiv:2502.08340.
40. Ma, Q.; Ge, S.; He, D.; et al. Combinatorial Optimization by Graph Pointer Networks and Hierarchical Reinforcement Learning. *arXiv* **2019**, arXiv:1911.04936.
41. Nazari, M.; Oroojlooy, A.; Snyder, L.; et al. Reinforcement Learning for Solving the Vehicle Routing Problem. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 9839–9849.
42. Xin, L. Step-Wise Deep Learning Models for Solving Routing Problems. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4861–4871.
43. Keskin, M.; Yılmaz, M. Chinese and windy postman problem with variable service costs. *Soft Comput*. **2019**, 23, 7359–7373.
44. Guo, R.; Xue, F.; Ming, A.; et al. An efficient learning-based solver comparable to metaheuristics for the capacitated arc routing problem. *arXiv* **2024**, arXiv:2403.07028.
45. Jia, Y.; Zheng, Q.; Wang, Y.; et al. A Neural Solver with Traversal-Based Feature Representation and Adjacent Attention for Capacitated Arc Routing Problem. *IEEE Trans. Intell. Transport. Syst*. **2025**, 26, 22329–22343.
46. Helsgaun, K. LKH-3. 2025. Available online: http://akira.ruc.dk/~keld/research/LKH-3 (accessed on 5 December 2025).
47. Wu, X.; Wang, D.; Wen, L.; et al. Neural Combinatorial Optimization Algorithms for Solving Vehicle Routing Problems: A Comprehensive Survey with Perspectives. *arXiv* **2024**, arXiv:2406.00415.
48. Ma, Y.; Cao, Z.; Chee, Y.M. Learning to Search Feasible and Infeasible Regions of Routing Problems with Flexible Neural k-Opt. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Virtual, 6–12 December 2020; pp. 49555–49578.
49. Bi, J.; Ma, Y.; Zhou, J.; et al. Learning to Handle Complex Constraints for Vehicle Routing Problems. In Proceedings of the 38th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 9–15 December 2024; pp. 93479–93509.
50. Prates, M.; Avelar, P.H.; Lemos, H.; et al. Learning to Solve NP-Complete Problems: A Graph Neural Network for Decision TSP. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 4731–4738.
51. Milan, A.; Rezatofighi, S.; Garg, R.; et al. Data-Driven Approximations to NP-Hard Problems. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4083–4089.
52. Kotary, J.; Fioretto, F.; Van Hentenryck, P. Learning Hard Optimization Problems: A Data Generation Perspective. In Proceedings of the 35th International Conference on Neural Information Processing Systems, Virtual, 6–14 December 2021; pp. 24981–24992.
53. Karalias, N.; Loukas, A. Erdos Goes Neural: An Unsupervised Learning Framework for Combinatorial Optimization on Graphs. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Virtual, 6–12 December 2020, pp. 6659–6672.
54. Wang, H.; Wu, N.; Yang, H.; et al. Unsupervised Learning for Combinatorial Optimization with Principled Objective Relaxation. In Proceedings of the 36th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; pp. 31444–31458.
55. Wang, H.; Li, P. Unsupervised Learning for Combinatorial Optimization Needs Meta-Learning. *arXiv* **2023**, arXiv:2301.03116.

56. Kim, M.; Park, J.; Park, J. Sym-NCO: Leveraging Symmetricity for Neural Combinatorial Optimization. In Proceedings of the 36th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; pp. 1936–1949.
57. Min, Y.; Bai, Y.; Gomes, C.P. Unsupervised Learning for Solving the Travelling Salesman Problem. In Proceedings of the 37th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; pp. 47264–47278.
58. Souza, G.K.B.; Santos, S.O.S.; Ottoni, A.L.C.; et al. Transfer Reinforcement Learning for Combinatorial Optimization Problems. *Algorithms* **2024**, *17*, 87.
59. Bi, J.; Ma, Y.; Wang, J.; et al. Learning Generalizable Models for Vehicle Routing Problems via Knowledge Distillation. In Proceedings of the 36th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; pp. 31226–31238.
60. Kanda, J.; Carvalho, A.D.; Hruschka, E.; et al. Meta-Learning to Select the Best Meta-Heuristic for the Traveling Salesman Problem: A Comparison of Meta-Features. *Neurocomputing* **2016**, *205*, 393–406.
61. Son, J.; Kim, M.; Kim, H.; et al. Meta-SAGE: Scale Meta-Learning Scheduled Adaptation with Guided Exploration for Mitigating Scale Shift on Combinatorial Optimization. In Proceedings of the 40th International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 32194–32210.
62. Pan, M.; Lin, G.; Luo, Y.W.; et al. Preference Optimization for Combinatorial Optimization Problems. *arXiv* **2025**, arXiv:2505.08735.
63. Hospedales, T.; Antoniou, A.; Micaelli, P.; et al. Meta-Learning in Neural Networks: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5149–5169.
64. Taillard, É.D.; Helsgaun, K. POPMUSIC for the Travelling Salesman Problem. *Eur. J. Oper. Res.* **2019**, *272*, 420–429.
65. Taillard, É.D. A Linearithmic Heuristic for the Travelling Salesman Problem. *Eur. J. Oper. Res.* **2022**, *297*, 442–450.
66. Li, T.; Zou, H.; Wu, J.; et al. LMask: Learn to Solve Constrained Routing Problems with Lazy Masking. *arXiv* **2025**, arXiv:2505.17938.
67. Xu, Y. Reinforcement Learning with Multiple Relational Attention for Solving Vehicle Routing Problems. *IEEE Trans. Cybern.* **2021**, *52*, 11107–11120.
68. Bai, H.; Cheng, R.; Jin, Y. Evolutionary Reinforcement Learning: A Survey. *Intell. Comput*. **2023**, *2*, 0025.
69. Jaderberg, M.; Dalibard, V.; Osindero, S.; et al. Population Based Training of Neural Networks. *arXiv* **2017**, arXiv:1711.09846.
70. Hong, J.; Shen, B.; Pan, A. A Reinforcement Learning-Based Neighborhood Search Operator for Multi-Modal Optimization and Its Applications. *Expert Syst. Appl.* **2024**, *246*, 123150.
71. Zhou, J.; Wu, Y.; Cao, Z.; et al. Learning Large Neighborhood Search for Vehicle Routing in Airport Ground Handling. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 9769–9782.
72. Li, X.; Qin, Y.; Huo, J.; et al. Heuristically Assisted Multiagent RL-Based Framework for Computation Offloading and Resource Allocation of Mobile-Edge Computing. *IEEE Internet Things J*. **2023**, *10*, 15477–15487.
73. Joshi, C.K.; Cappart, Q.; Rousseau, L.M.; et al. Learning the Travelling Salesperson Problem Requires Rethinking Generalization. *Constraints* **2022**, *27*, 1–29.
74. Wang, C. Heuristic-Augmented Attentions for the Electric Vehicle Routing Problem with Time Windows. *IEEE Trans. Veh. Technol.* **2026**, *Early Access*.
75. Wu, Q. MOELS: Multiobjective Evolutionary List Scheduling for Cloud Workflows. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 166–176.
76. Lin, J.; Wang, X.; Niu, R.; et al. A Q-Learning-Based Hyper-Heuristic for Capacitated Electric Vehicle Routing Problem. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 15746–15757.
77. Miettinen, K. *Nonlinear Multiobjective Optimization*; International Series in Operations Research & Management Science; Volume 12; Springer: New York, NY, USA, 2012.
78. Wang, R.; Zhou, Z.; Ishibuchi, H.; et al. Localized Weighted Sum Method for Many-Objective Optimization. *IEEE Trans. Evol. Comput.* **2018**, *22*, 3–18.
79. Wang, R.; Zhang, Q.; Zhang, T. Decomposition-Based Algorithms Using Pareto Adaptive Scalarizing Methods. *IEEE Trans. Evol. Comput.* **2016**, *20*, 821–837.
80. Li, K. Deep Reinforcement Learning for Multiobjective Optimization. *IEEE Trans. Cybern.* **2020**, *51*, 3103–3114.
81. Li, J.; Chu, Y.; Sun, Y.; et al. AutoPBO: LLM-Powered Optimization for Local Search PBO Solvers. *arXiv* **2025**, arXiv:2509.04007.
82. Hsu, C.H.; Chang, S.H.; Liang, J.H.; et al. MONAS: Multi-Objective Neural Architecture Search Using Reinforcement Learning. *arXiv* **2018**, arXiv:1806.10332.
83. Fu, Y.; Zhou, M.; Guo, X.; et al. Multiobjective Scheduling of Energy-Efficient Stochastic Hybrid Open Shop with Brain Storm Optimization and Simulation Evaluation. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 4260–4272.

84. Cui, M.; Li, L.; Zhou, M.; et al. Surrogate-Assisted Autoencoder-Embedded Evolutionary Optimization Algorithm to Solve High-Dimensional Expensive Problems. *IEEE Trans. Evol. Comput.* **2022**, *26*, 676–689.
85. Zhou, M.C.; Cui, M.; Xu, D.; et al. Evolutionary Optimization Methods for High-Dimensional Expensive Problems: A Survey. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 1092–1105.
86. Wang, X.; Kang, Q.; Zhou, M.; et al. Domain Adaptation Multitask Optimization. *IEEE Trans. Cybern*. **2023**, *53*, 4567–4578.
87. Wang, X.; Kang, Q.; Zhou, M.; et al. Knowledge Classification-Assisted Evolutionary Multitasking for Two-Task Multiobjective Optimization Problems. *IEEE/CAA J. Autom. Sin.* **2025**, *12*, 1176–1193.
88. Deng, J.; Wang, J.; Wang, X.; et al. Multi-Task Multi-Objective Evolutionary Search Based on Deep Reinforcement Learning for Multi-Objective Vehicle Routing Problems with Time Windows. *Symmetry* **2024**, *16*, 1030.
89. Tian, Y.; Si, L.; Zhang, X.; et al. Evolutionary Large-Scale Multi-Objective Optimization: A Survey. *ACM Comput. Surv*. **2021**, *54*, 1–34.
90. Hottung, A.; Berto, F.; Hua, C.; et al. VRPAgent: LLM-Driven Discovery of Heuristic Operators for Vehicle Routing Problems. *arXiv* **2025**, arXiv:2510.07073.
91. Astorga, N.; Liu, T.; Xiao, Y.; et al. Autoformulation of Mathematical Optimization Models Using LLMs. *arXiv* **2024**, arXiv:2411.01679.
92. Hu, Y.; Zhao, T.; Yue, M. From Natural Language to Solver-Ready Power System Optimization: An LLM-Assisted, Validation-in-the-Loop Framework. *arXiv* **2025**, arXiv:2508.08147.
93. Peng, M.; Chen, Z.; Yang, J.; et al. Automatic MILP Model Construction for Multi-Robot Task Allocation and Scheduling Based on Large Language Models. *arXiv* **2025**, arXiv:2503.13813.
94. Huang, Z.; Wu, W.; Wu, K.; et al. CALM: Co-Evolution of Algorithms and Language Model for Automatic Heuristic Design. *arXiv* **2025**, arXiv:2505.12285.
95. Huang, Z.; Wu, W.; Wu, K.; et al. CALM: Co-Evolution of Algorithms and Language Model for Automatic Heuristic Design. *arXiv* **2025**, arXiv:2505.12285. Tran, C.D.; Nguyen-Tri, Q.; Binh, H.T.T.; et al. Large Language Models Powered Neural Solvers for Generalized Vehicle Routing Problems. In Proceedings of the 13th International Conference on Learning Representations (ICLR 2025), Singapore, 24–28 April 2025.
96. Jiang, X.; Wu, Y.; Zhang, C.; et al. DRoC: Elevating Large Language Models for Complex Vehicle Routing via Decomposed Retrieval of Constraints. In Proceedings of the 13th International Conference on Learning Representations (ICLR 2025), Singapore, 24–28 April 2025.