Elite Pattern Reinforcement for Vehicle Routing Problems

Ning Li[1,2,†], **Peng Lin**[3,4,†], **Peng Zhang**[1,2,5,*], **Ruichen Tian**[2,5]

[1]College of Computer Science and Technology, Jilin University, Jilin, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, Jilin, China
[3]Key Laboratory of System Software, Institute of Software, Chinese Academy of Sciences, Beijing, China
[4]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China
[5]College of Software, Jilin University, Jilin, China
lining23@mails.jlu.edu.cn, peng.lin.csor@gmail.com, zhangpengccst@jlu.edu.cn, tianrc24@mails.jlu.edu.cn

Abstract

Machine learning methods have been increasingly applied to solve Vehicle Routing Problems (VRPs). A high-efficiency approach is to learn solution construction using deep neural networks. However, their tendency toward premature convergence is a critical barrier, severely hindering generalization across diverse distributions and scales. To overcome this, we introduce Elite-Pattern Reinforcement (EPR), a novel strategy designed to create a synergy between the diverse, exploratory nature of reinforcement learning and the high-quality, structured knowledge from classical heuristics. The strategy guides the learning process by reinforcing structural patterns from elite solutions, employing an elite-guided score modulation to integrate this external knowledge. The inherent symmetry of path patterns is also exploited to augment the structural information. This steers the policy away from premature convergence by enabling it to distinguish and favor elite path patterns over inferior ones. Integrating our strategy with four construction methods yields substantial performance improvements on the CVRPLIB and TSPLIB benchmarks. Furthermore, our approach outperforms state-of-the-art learning-based methods, demonstrating superior generalization across diverse distributions and scales.

Code — https://github.com/1477619915/EPR-POMO

Introduction

The Vehicle Routing Problem (VRP) is a classical NP-hard combinatorial optimization problem with significant practical relevance in transportation and logistics (Konstantakopoulos, Gayialis, and Kechagias 2022). Over the past few decades, VRPs have been extensively studied. Various methods have been proposed, including exact algorithms (Trick 2008) and heuristic methods such as LKH3 (Helsgaun 2017), SISA (Christiaens and Berghe 2020), and HGS (Vidal 2022). However, these methods exhibit notable limitations: their implementation relies heavily on domain-specific expertise, and they often struggle to generate high-quality solutions within acceptable timeframes for applications demanding real-time responsiveness (Matsuzaki et al. 2024).

Recently, the neural combinatorial optimization (NCO) (Bello et al. 2017) has emerged as a novel deep learning-based paradigm for solving VRPs (Bengio, Lodi, and Prouvost 2021). NCO employs deep neural networks to learn heuristic strategies directly from data, primarily through two learning paradigms: 1) Supervised learning (Christiaens and Berghe 2020; Chen et al. 2023), which requires large-scale optimal solutions as training labels and often difficult to fulfill in practical applications; 2) Reinforcement learning (Kool, van Hoof, and Welling 2019), which optimizes models end-to-end through environment interactions, using the objective function as a reward signal to guide training, thereby offering greater applicability (Zhang et al. 2020; Song et al. 2024).

NCO reinforcement learning methods are typically divided into two categories based on their solution-generation mechanisms: improvement and construction. Improvement methods learn to refine existing solutions iteratively (Lu, Zhang, and Yang 2020). While capable of incorporating domain knowledge, inefficient search processes and substantial inference latency often hinder these methods. In contrast, construction methods formulate the solving process as a Markov Decision Process, learning to build solutions sequentially, one decision at a time (Bello et al. 2017; Khalil et al. 2017). They demonstrate exceptional computational efficiency, often capable of processing numerous instances in a few seconds. However, these methods are significantly limited by their susceptibility to premature convergence to local optima, which constrains the model's exploratory capacity, leading to overfitting and poor generalization across diverse distributions and scales (Kool, van Hoof, and Welling 2019; Joshi et al. 2021).

Recent research in construction methods has made significant progress in addressing the problem of local optima and enhancing model generalization across diverse distributions and scales. To mitigate the local optima problem, efforts have focused on two main directions: 1) enhancing exploration strategies, such as curriculum learning mechanisms that gradually increase problem complexity (Li et al. 2020) and stochastic perturbation techniques to improve solution space exploration (Joshi et al. 2021); and 2) algorithmic framework innovations, including hybrid approaches incorporating Monte Carlo tree search (He and Bao

[†]These authors contributed equally.
[*]Corresponding author

2020) and distributed training paradigms with multi-agent collaboration (Gu, Sun, and Cai 2020). To tackle the cross-distribution challenge, solutions range from the straightforward approach of training on diverse node distributions to more sophisticated methods such as distributional robust optimization (Jiang et al. 2022), knowledge distillation (Bi et al. 2022), and graph contrastive learning (Jiang et al. 2023). Concurrently, cross-scale challenges have been investigated through innovative neural architectures like heavy decoders (Luo et al. 2023) and diffusion models (Sun and Yang 2023). A particularly influential work for VRPs is the POMO framework (Kwon et al. 2020), which enhances both cross-distribution and cross-scale generalization by leveraging problem symmetries and a multi-start inference strategy. Building upon this foundation, subsequent research has further improved POMO's performance through a variety of methods, including symmetry regularization (Kim, Park, and Park 2022), meta-learning (Zhou et al. 2023), and the integration of global-local policy (Gao et al. 2024).

In this paper, we focus on the local optima problem to enhance generalization on instances characterized by complex node distributions and large scales. Building upon the POMO framework, we propose a novel Elite-Pattern Reinforcement (EPR) strategy. Our approach is designed to create a synergy between the diverse, exploratory nature of reinforcement learning and the high-quality, structured knowledge from classical heuristics. Specifically, this strategy guides the training process by reinforcing superior path patterns from elite solutions, thereby improving the structural quality of the constructed solutions. Moreover, we exploit the inherent symmetry of these path patterns to augment the structural information. This structural information is progressively integrated into the training process via elite-guided score modulation, which enables the policy network to reinforce superior path patterns while suppressing inferior ones. This infusion of external knowledge prevents the model from converging prematurely based solely on its own limited exploration. By periodically integrating this guidance, our method remains computationally efficient while mitigating overfitting. Ultimately, this approach provides the model with a global optimization perspective, assisting it in escaping local optima and significantly enhancing its generalization capabilities.

We conduct experiments on the Traveling Salesperson Problem (TSP) and the Capacitated VRP (CVRP). We integrate and test our strategy with the POMO method (Kwon et al. 2020) and three of its variants. Our strategy is first trained on synthetic, small-scale instances with uniform node distributions, and then evaluated on the challenging TSPLIB (Reinelt 1991) and CVRPLIB (Uchoa et al. 2017) benchmarks to assess its generalization to instances with complex node distributions and larger scales. The integration yields significant performance improvements across all tested POMO-based methods, including the highly competitive ELG-POMO (Gao et al. 2024). Furthermore, our approach outperforms state-of-the-art learning-based methods, demonstrating superior generalization across diverse distributions and scales. Ablation studies further confirm the critical role of our guidance strategy in these performance gains.

Backgrounds

Vehicle Routing Problems

The Vehicle Routing Problems (VRPs) are classical NP-hard combinatorial optimization problems with extensive practical applications (Konstantakopoulos, Gayialis, and Kechagias 2022). This paper focuses on two of the most fundamental and widely studied problems in the VRP domain: CVRP and TSP.

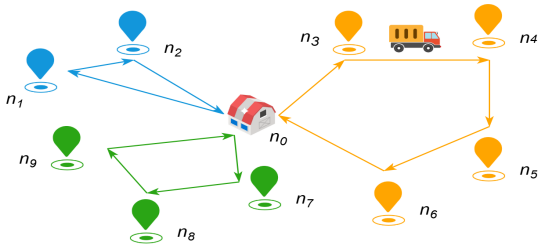

Figure 1: Example illustration of CVRP

CVRP: As illustrated in Figure 1, a typical CVRP instance consists of a depot (n_0) and N customer nodes $(n_1, n_2, \ldots, n_N)$. Each customer node (n_i) has fixed Euclidean coordinates (x_{n_i}, y_{n_i}) and a specific demand (d_i). A fleet of identical vehicles, each with a uniform capacity Q, is dispatched to service these customers to fulfill their demands. Departing from a central depot (n_0), the vehicle fleet services a set of distinct routes, ensuring that each customer node is visited exactly once across all routes. The primary **objective** of the CVRP is to minimize the total travel distance for all vehicles. This optimization is subject to two critical **constraints**:

- Each customer node must be visited exactly once.
- For each vehicle, the total demand of the customers on its route cannot exceed the vehicle's capacity Q.

A solution to the CVRP is a set of routes. For modeling purposes, a complete CVRP solution is linearized into a single sequence of nodes, $\boldsymbol{\tau}$. This sequence is formed by concatenating all vehicle routes, using the depot node (n_0) as the start and end of each individual route and as a delimiter between them. For instance, a solution consisting of two routes, $\{n_0 \to n_1 \to n_2 \to n_0\}$ and $\{n_0 \to n_3 \to n_4 \to n_5 \to n_6 \to n_0\}$, would be represented by the single sequence $\boldsymbol{\tau} = (n_0, n_1, n_2, n_0, n_3, n_4, n_5, n_6, n_0)$. Therefore, the objective is calculated as:

$$f(\boldsymbol{\tau}) = \sum_i \sqrt{(x_{\tau_i} - x_{\tau_{i+1}})^2 + (y_{\tau_i} - y_{\tau_{i+1}})^2},$$

TSP: As one of the most fundamental problems in combinatorial optimization, the TSP can be considered a simplified version of the CVRP, characterized by a single route and no capacity constraints.

Neural Construction Methods for VRPs

Neural networks have demonstrated significant potential in solving combinatorial optimization problems, particularly through the autoregressive generation of solution

sequences (Vinyals, Fortunato, and Jaitly 2015). These methods learn to model the probability distribution $P(\boldsymbol{\tau})$ of a solution sequence $\boldsymbol{\tau}$. The solution is constructed autoregressively, with the partial solution at step t being $(\tau_0, \tau_1, \ldots, \tau_{t-1})$. This sequential construction process can be naturally modeled as a Markov Decision Process (MDP) (Sutton and Barto 1998). In this MDP formulation, the partial solution $(\tau_0, \tau_1, \ldots, \tau_{t-1})$ serves as the state, and the next node, τ_t, is the action (Bello et al. 2017; Kool, van Hoof, and Welling 2019). In practice, the state representation is often simplified to include only the current and starting nodes.

Prevalent construction methods for VRPs now predominantly employ Transformer architectures (Vaswani et al. 2017), typically without positional encoding, to encode unordered nodes and decode solutions. Notable examples include the Attention Mechanism Model (AM) (Kool, van Hoof, and Welling 2019) and Multi-Optimal Policy Optimization (POMO) (Kwon et al. 2020). These models typically adopt an encoder-decoder architecture. Specifically, the encoder learns node embeddings from the complete problem graph. The decoder then computes compatibility scores between a query $\boldsymbol{q}$ and keys $\boldsymbol{K}$, i.e.,

$$\boldsymbol{u} = \frac{\boldsymbol{q}^\top \boldsymbol{K}}{\sqrt{d}}, \quad \boldsymbol{q} \in \mathbb{R}^{d\times 1},\ \boldsymbol{K} \in \mathbb{R}^{d\times N}$$

The scores for selecting actions are given by $\boldsymbol{u}$, where $\boldsymbol{q}$ represents the context embedding (the current state) encompassing the current node and loaded capacity, and $\boldsymbol{K}$ represents the candidate node embeddings (the possible actions). Here, N denotes the number of customer nodes, and d is the embedding dimension. Due to the multi-start parallel inference characteristic of POMO, it generates M ($M \leq N$) solutions for the same instance, each starting from a distinct initial node. For each solution, the corresponding action scores $\boldsymbol{u}^i$ are computed, where each node in the solution is associated with u^i_j, representing the action score associated with the j-th node of the i-th solution. To avoid violating constraints, $\boldsymbol{u}$ is masked, with the j-th dimension of the i-th solution of the masked scores $u^i_{\text{j-masked}}$ being:

$$u^i_{j\text{-masked}} = \begin{cases} C \cdot \tanh(u^i_j), & \text{if node } n^i_j \text{ is valid,} \\ -\infty, & \text{otherwise,} \end{cases}$$

Where $\tanh$ clips scores to be within [-1, 1], C is a parameter to control the scale, and the scores of invalid actions are set as $-\infty$. The i-th solution of the action probability $\boldsymbol{\pi}^i$ is computed as:

$$\boldsymbol{\pi}^i = \text{softmax}(\boldsymbol{u}^i_{\text{masked}}).$$

Where $\boldsymbol{u}^i_{\text{masked}}$ is the masked scores of the i-th solution. At each step, the decoder selects a node to extend the partial solution and transitions to a new state. This process repeats until a complete solution $\boldsymbol{\tau}^i$ is constructed. Once the construction is complete, the negative value of the objective function is computed as the reward $R(\boldsymbol{\tau}^i) = -f(\boldsymbol{\tau}^i)$. These models are typically trained using reinforcement learning algorithms such as REINFORCE (Williams 1992) with a greedy rollout baseline (Kool, van Hoof, and Welling 2019) or POMO with a shared baseline (Kwon et al. 2020).

Generalization Issue

Given the substantial data requirements for training, current reinforcement learning methods for combinatorial optimization are typically trained on small-scale instances with simple, uniform node distributions (Joshi et al. 2021; Bi et al. 2022; Liu et al. 2022). However, this training paradigm introduces a critical bottleneck: a tendency toward premature convergence, which severely restricts generalization across diverse data distributions and problem scales (Cappart et al. 2023). Their performance often deteriorates when applied to real-world problems, which feature complex, unknown distributions and larger scales. While many studies have attempted to address this generalization gap using advanced techniques such as stochastic perturbation strategies (Joshi et al. 2021), multi-agent collaborative distributed training (Gu, Sun, and Cai 2020), and meta-learning (Manchanda et al. 2022), substantial room for improvement remains, particularly for complex real-world problems.

In this paper, we introduce an Elite-Pattern Reinforcement (EPR) strategy that integrates external, high-quality knowledge to steer the learning process away from premature convergence to local optima. To comprehensively evaluate our method's performance, we selected the challenging public benchmarks CVRPLIB (Uchoa et al. 2017; Arnold, Gendreau, and Sörensen 2019) and TSPLIB (Reinelt 1991) for testing. These benchmarks comprise diverse instances, many from real-world scenarios, offering a more comprehensive evaluation than synthetic datasets with fixed distributions or limited scales, which cover only a narrow subspace of the problem landscape.

Method

Our approach treats the reinforcement learning (RL) agent and the heuristic solver as two complementary experts. The core challenge, which our Elite-Pattern Reinforcement (EPR) strategy solves, is to design a principled mechanism for fusing their distinct insights at the path pattern level, including the agent's diverse, exploratory attempts and the heuristic's high-quality, structured knowledge.

EPR actualizes this synergy through a periodic guidance mechanism that leverages structural information from the heuristic expert. This provides the learning agent with an external optimization perspective, improving its solution quality while remaining computationally efficient. As illustrated in Figure 2, our method operationalizes this fusion through three key components: 1) Heuristic Elite Solution Generation, 2) Path Pattern Identification via Elite Solution Filtering, and 3) an Elite-Guided Score Modulation.

Heuristic Elite Solution Generation

A common observation in VRPs is that construction models often exhibit myopic behavior during the decoding phase, such as prioritizing the nearest node. This local focus can prevent them from exploring solution spaces that contain

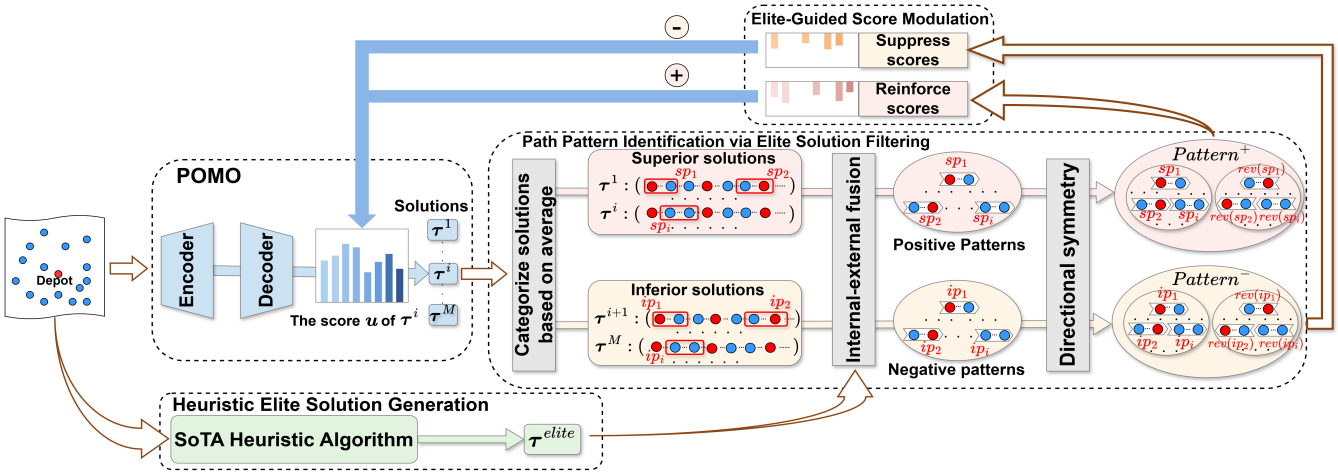

Figure 2: Framework of the EPR method, illustrating the synergy between the RL agent's internal exploration and the heuristic's external guidance.

globally superior paths. To counteract this myopia, we leverage elite solutions to guide the model's exploration. While using optimal solutions as guidance is ideal, acquiring them for numerous VRP instances is often computationally prohibitive. Therefore, we opt to generate high-quality elite solutions using powerful heuristic methods. Heuristic solvers currently represent the most effective and practical approach for VRPs, consistently delivering near-optimal solutions with high efficiency(Alqahtani and Kumar 2024).

To balance solution quality with computational efficiency, we introduce a periodic guidance mechanism. This is motivated by our key observation that for the relatively simple instances used during training, a sufficiently strong guidance solution can be generated within a very short time budget (e.g., one second) by SoTA heuristic algorithms. This mechanism activates the EPR strategy for only a predefined portion ($P\%$) of training instances in each epoch. We select the Hybrid Genetic Search (HGS) (Vidal 2022) algorithm to generate elite solutions for CVRP instances, a state-of-the-art heuristic solver for this problem. For TSP instances, we similarly employ LKH3 (Helsgaun 2017), another premier heuristic for solving the TSP. For an elite solution generated by a SoTA heuristic, we denote it as $\boldsymbol{\tau}^{\text{elite}}$. By periodically incorporating $\boldsymbol{\tau}^{\text{elite}}$, we can effectively guide the model's learning trajectory, providing a rich source of structural information for the elite-guided score modulation described in Section 3.3.

Moreover, our periodic guidance incurs only slight overhead: elites are generated once before training and can be parallelized, incurring no inference cost.

Path Pattern Identification via Elite Solution Filtering

The central principle of EPR is to provide guidance at the path pattern level, rather than the complete solution level. This is motivated by the observation that a solution's quality is determined by its constituent patterns, and that high-quality patterns are often shared building blocks across many distinct near-optimal solutions. Our identification process synergistically combines two sources of information: the internal diversity from the model's self-generated solutions and the external expertise embodied by a heuristically-generated elite solution.

During training, a POMO-based model generates M distinct solutions $\{\boldsymbol{\tau}^1, \boldsymbol{\tau}^2, ..., \boldsymbol{\tau}^M\}$ for each problem instance (Kwon et al. 2020). We first compute the average solution cost, which serves as a baseline for the model's current performance on the instance:

$$f_{\text{avg}} = \frac{1}{M} \sum_{i=1}^{M} f(\boldsymbol{\tau}^i)$$

where $f(\boldsymbol{\tau}^i)$ is the cost of the i-th solution, with lower values indicating better solution quality. Based on this baseline, we categorize the generated solutions into two disjoint sets:

- **Superior solutions set**:

$$\boldsymbol{S}_{\text{sup}} = \{\boldsymbol{\tau}^i \mid f(\boldsymbol{\tau}^i) \leq f_{\text{avg}}\}$$

- **Inferior solutions set**:

$$\boldsymbol{S}_{\text{inf}} = \{\boldsymbol{\tau}^i \mid f(\boldsymbol{\tau}^i) > f_{\text{avg}}\}$$

To formalize the guidance mechanism, we first define the concept of a path pattern. A key property of many VRPs is **directional symmetry**: traversing a path in either the forward or reverse direction yields an identical total cost. To leverage this property, our definition of a pattern set explicitly includes these reversed counterparts. Specifically, for any solution $\boldsymbol{\tau}$, we define its set of patterns, $\text{Pattern}(\boldsymbol{\tau}, K)$, as all unique contiguous subsequences of length K:

$$\text{Pattern}(\boldsymbol{\tau}, K) = \{\boldsymbol{p}' \mid \exists j,\ \boldsymbol{p} = (\tau_j, \ldots, \tau_{j+K-1}) \\ \wedge\ ((\boldsymbol{p}' = \boldsymbol{p}) \vee (\boldsymbol{p}' = \text{rev}(\boldsymbol{p})))\}$$

Our definition explicitly includes both a pattern $\boldsymbol{p}$ and its reverse $\text{rev}(\boldsymbol{p})$, thereby augmenting the structural information extracted from a single solution.

By fusing the two information sources from internal diversity solution sets and the external elite solution $\boldsymbol{\tau}^{\text{elite}}$, we define the sets of positive and negative patterns. The positive patterns are those found in both the superior solutions and the elite solution. The negative patterns are those present in the inferior solutions but absent from the elite solution. Formally:

- **Positive Patterns**:

$$\text{Pattern}^{+} = \bigcup_{\boldsymbol{\tau} \in \boldsymbol{S}_{\text{sup}}} \text{Pattern}(\boldsymbol{\tau}, K) \cap \text{Pattern}(\boldsymbol{\tau}^{\text{elite}}, K)$$

- **Negative Patterns**:

$$\text{Pattern}^{-} = \bigcup_{\boldsymbol{\tau} \in \boldsymbol{S}_{\text{inf}}} \text{Pattern}(\boldsymbol{\tau}, K) \setminus \text{Pattern}(\boldsymbol{\tau}^{\text{elite}}, K)$$

Elite-Guided Score Modulation

To effectively leverage the identified positive and negative patterns, we introduce a novel mechanism that directly modulates the model's action scores. The objective is to dynamically adjust the model's policy, enhancing its preference for superior patterns while suppressing the generation of inferior ones.

Specifically, for each solution $\boldsymbol{\tau}^i$ generated by the model, we first quantify its performance gap, $\boldsymbol{\Delta}_i$, relative to the elite solution $\boldsymbol{\tau}^{\text{elite}}$:

$$\boldsymbol{\Delta}_i = f(\boldsymbol{\tau}^i) - f(\boldsymbol{\tau}^{\text{elite}})$$

where a smaller $\boldsymbol{\Delta}_i$ value indicates a higher-quality solution.

For any path pattern $\boldsymbol{p}$ within a solution $\boldsymbol{\tau}^i$ that belongs to either the positive or negative sets ($\boldsymbol{p} \in \text{Pattern}^{+} \cup \text{Pattern}^{-}$), we compute an adjustment magnitude, $\boldsymbol{\Delta}_p$. This magnitude is inversely proportional to the solution's quality gap:

$$\boldsymbol{\Delta}_p(\boldsymbol{\tau}^i) = \frac{\eta}{\boldsymbol{\Delta}_i}$$

where η is a tunable coefficient controlling the adjustment sensitivity. This formulation ensures that solutions closer in quality to the elite solution (i.e., smaller Δ_i) receive a stronger corrective signal. The modulation strategy then operates as follows:

- For Positive Patterns ($\boldsymbol{p} \in \text{Pattern}^{+}$) found in a superior solution ($\boldsymbol{\tau}^i \in \boldsymbol{S}_{\text{sup}}$): The action scores of all nodes within $\boldsymbol{p}$ are reinforced:

$$u_j^{i\prime} = u_j^i + \boldsymbol{\Delta}_p(\boldsymbol{\tau}^i)$$

- For Negative Patterns ($\boldsymbol{p} \in \text{Pattern}^{-}$) found in an inferior solution ($\boldsymbol{\tau}^i \in \boldsymbol{S}_{\text{inf}}$): The action scores of all nodes within $\boldsymbol{p}$ are suppressed:

$$u_j^{i\prime} = u_j^i - \boldsymbol{\Delta}_p(\boldsymbol{\tau}^i)$$

where u_j^i is the original action score for the j-th node in the i-th solution $\boldsymbol{\tau}^i$.

In addition to these pattern-level adjustments, the final gradient update for the policy network $\boldsymbol{\pi}_\theta$ is modulated by the performance gap $\boldsymbol{\Delta}_i$. Specifically, we obtain a set of M trajectories in a single feedforward by executing rollouts from multiple start nodes, and utilize the REINFORCE (Williams 1992) algorithm with shared baseline (Kwon et al. 2020) to estimate the gradient of the expected return J. Building upon this, we introduce an adaptive weighting factor $(1 + \boldsymbol{\Delta}_i)$ to explicitly incorporate the performance gap, thereby robustly reinforcing superior solutions and strongly penalizing inferior ones. The resultant update rule is as follows:

$$\boldsymbol{\nabla} J(\theta) \propto \frac{1}{M} \sum_{i=1}^{M} (1 + \boldsymbol{\Delta}_i)(f(\boldsymbol{\tau}^i) - f_{\text{avg}}) \boldsymbol{\nabla}_\theta \log \boldsymbol{\pi}_\theta^i$$

where M denotes the number of solutions of instance, and $\boldsymbol{\pi}_\theta^i$ is the probability of selecting the entire solution sequence $\boldsymbol{\tau}^i$ under the current policy.

Experiments

We conduct experiments on two canonical VRPs: the Capacitated Vehicle Routing Problem (CVRP) and the Traveling Salesperson Problem (TSP). This section details our experimental setup and presents the results, with the primary aims of validating the effectiveness of our EPR strategy and evaluating its overall performance.

Experimental Settings

Here, we introduce the experimental settings, including the baseline methods, datasets, and performance metrics used for evaluation. We compare our method against several state-of-the-art baselines, categorized as follows.

Non-learning heuristics: We compare against two premier solvers: LKH3 (Helsgaun 2017) for TSP and HGS (Vidal 2022) for CVRP.

Learning-based methods:

- **POMO-based Methods:** This family of methods forms the primary baseline for our work, as our EPR strategy is designed to enhance them.
  - POMO (Kwon et al. 2020) is a foundational model that leverages problem symmetries and a multi-start inference strategy.
  - Sym-POMO (Kim, Park, and Park 2022) introduces a symmetry-regularized training framework to better exploit problem structure.
  - Omni-POMO (Zhou et al. 2023) tackles generalization by combining meta-learning with a hierarchical dispatcher.
  - ELG-POMO (Gao et al. 2024), the most powerful POMO-based method, enhances generalization by global and transferable local policies.
- **Cross-scale Methods:** We compare against two notable methods designed for cross-scale generalization: LEHD (Luo et al. 2023) and BQ (Drakulic et al. 2023).
- **Diffusion Model based Methods:** Recent methods based on diffusion models have shown strong generalization. We compare against DIFUSCO (Sun and Yang 2023) as a representative of this approach. DIFUSCO is designed for TSP.

All learning-based methods employ a greedy decoding strategy during inference (Kwon et al. 2020). Our primary experiments involve integrating the proposed EPR strategy with the POMO family of methods to demonstrate its effectiveness. Due to space constraints, a complete list of hyperparameter settings is detailed in the Appendix (Supplementary Material)

Datasets. For training, we follow the standard practice of using randomly generated instances of size $N = 100$ with a uniform node distribution, ensuring comparability with prior work (Gao et al. 2024). For evaluation, we use the challenging TSPLIB (Reinelt 1991) and CVRPLIB (Set X & XXL) (Uchoa et al. 2017; Arnold, Gendreau, and Sörensen 2019) benchmarks, grouping instances by scale: small ($N \leq 200$), large ($200 < N \leq 1002(1000)$), and very large ($N \geq 3000$). Additionally, to test robustness on complex structures, we use the synthetic CVRPLIB Set M (Christofides 1979), which features instances with high structural symmetry at scales of $N \in \{100, 200\}$. This comprehensive suite of benchmarks, featuring diverse distributions and a wide range of scales, enables a rigorous assessment of our model's generalization capabilities.

Performance Metrics. We evaluate generalization performance using the average optimality gap relative to the Best-Known Solutions (BKS), computed for each dataset as: $\text{Gap}\ (\%) = \frac{1}{n}\sum_{i=1}^{n}\frac{f_i - f_i^{\text{BKS}}}{f_i^{\text{BKS}}} \times 100$, where n is the number of instances, f_i is the objective value of the solution found by a method for instance i, and f_i^{BKS} is the objective value of the corresponding BKS. Additionally, we measure computational efficiency by recording the average runtime per test instance.

Effectiveness of the EPR Strategy

We evaluate our EPR strategy by integrating it in a non-invasive manner with four distinct POMO variants. The following analysis is based on results from three challenging benchmarks: CVRPLIB-M, CVRPLIB-X, and TSPLIB.

The results, summarized in Table 1, demonstrate EPR's widespread effectiveness. The strategy is particularly impactful for weaker baselines, dramatically reducing the optimality gaps for Sym-POMO and the original POMO on the CVRPLIB-X benchmark. Crucially, it also consistently improves the performance of the state-of-the-art ELG-POMO across all tested benchmarks—including CVRPLIB-M, CVRPLIB-X, and TSPLIB—proving its value extends beyond merely correcting flawed models. Furthermore, this performance gain is achieved without a significant increase in inference time. In some cases, such as ELG-POMO on TSPLIB, the guided model is even faster, suggesting that the structural guidance leads to more efficient solution discovery.

While minor performance degradation is observed in a few isolated cases, the overwhelming evidence confirms that EPR is a robust and broadly applicable enhancement. The minor inconsistencies are likely attributable to the fixed-length pattern mechanism not perfectly aligning with the specific structural properties of those instances.

Comparison with State-of-the-Art Methods

To ensure fairness, all baselines were evaluated using their officially provided, best-performing pre-trained models. In the following analysis, we denote our best-performing model, EPR-ELG-POMO, simply as **EPR** for brevity.

Performance on CVRPLIB. For the CVRP, our EPR model establishes a new state of the art among all tested learning-based methods. As shown in Table 2, it consistently outperforms all baselines on the CVRPLIB-X benchmark across both small ($N \leq 200$) and large ($200 < N \leq 1000$) scales. This superiority extends to the most challenging, very large-scale real-world instances from CVRPLIB-XXL (Table 4), where EPR achieves the best results in four out of five cases, significantly outperforming even specialized large-scale methods like LEHD and BQ.

Performance on TSPLIB. On the TSPLIB benchmark (Table 3), EPR demonstrates highly competitive, top-tier performance. It is the best-performing generalist method (non-specialized architecture) on large-scale instances ($200 < N \leq 1002$), and is negligibly behind the leader on small-scale instances (a 0.01% margin). While architectures tailored for TSP like LEHD and BQ show an advantage on larger scales (Luo et al. 2023), our model's strong all-around performance validates its robust generalization.

In summary, these comprehensive results validate our central claim. By integrating elite heuristic knowledge, EPR consistently advances the state of the art for the complex, multi-route CVRP and achieves highly competitive performance for the TSP, demonstrating a superior and robust generalization capability.

Ablation Studies

To dissect the contribution of each component in our strategy, we conducted an ablation study on the strong ELG-POMO baseline. The results, summarized in Table 5, clearly validate the effectiveness of both the core guidance mechanism and the symmetry feature.

The analysis reveals a two-step improvement. First, introducing the core EPR guidance alone (row 2) provides a consistent performance gain over the baseline (row 1), demonstrating that the external knowledge is inherently beneficial. Second, the subsequent addition of the symmetry mechanism (row 3) yields a further, more substantial improvement, reducing the CVRPLIB-X gap from 6.32% to 5.96%. This confirms that leveraging path symmetry is a critical component for maximizing the strategy's effectiveness.

Hyperparameter Sensitivity Analysis

A key advantage of our EPR strategy is its low sensitivity to its two main hyperparameters: the guidance proportion (P) and the path pattern length (K). This robustness alleviates concerns about costly parameter tuning and highlights the method's practical applicability.

The analysis presented in Table 6 confirms this stability. The model's performance remains remarkably robust across the tested ranges for both P and K. For instance, while performance peaks at $P = 0.1$, the results at other settings are highly comparable.

Method	CVRPLIB-M		CVRPLIB-X				TSPLIB			
	Gap(%) [100,200]	Time(s)	Gap(%) (0,200]	Gap(%) (200,1000]	Gap(%) Total	Time(s)	Gap(%) (0,200]	Gap(%) (200,1002]	Gap(%) Total	Time(s)
POMO	4.43	0.68	8.66	18.18	16.08	2.50	**2.29**	11.74	6.15	0.69
EPR-POMO	**3.41**	0.66	**4.94**	**12.12**	**10.54**	2.11	2.31	**11.56**	**6.09**	0.69
Omni-POMO	4.46	0.70	5.33	6.58	6.31	2.09	5.22	12.98	8.39	0.70
EPR-Omni-POMO	**4.25**	0.80	**5.03**	**6.52**	**6.19**	2.09	**4.37**	**8.49**	**6.05**	0.70
Sym-POMO	28.16	0.71	28.89	31.00	30.54	2.38	3.15	**15.51**	8.20	0.70
EPR-Sym-POMO	**27.29**	0.72	**20.62**	**25.73**	**24.61**	2.31	**3.00**	15.72	**8.11**	0.70
ELG-POMO	3.99	1.09	**4.71**	6.91	6.42	3.72	1.30	6.20	3.30	2.11
EPR-ELG-POMO	**3.72**	1.09	4.73	**6.31**	**5.96**	3.69	1.30	**5.80**	**3.13**	1.19

Table 1: Comprehensive Evaluation of the EPR Strategy's Effectiveness on POMO Variants across Multiple Benchmarks.

Method	(0,200]	(200,1000]	Total	Time
LKH3	0.47%	1.58%	1.34%	19m
HGS	0.02%	0.21%	0.17%	19m
HGS(10s)	0.28%	1.86%	1.51%	10s
POMO	6.55%	22.66%	19.12%	2.15s
Sym-POMO	42.30%	41.50%	41.68%	2.44s
Omni-POMO	5.64%	7.41%	7.02%	2.10s
ELG-POMO	4.51%	6.46%	6.03%	3.74s
LEHD	13.32%	16.48%	14.62%	2.16s
BQ	11.70%	12.23%	12.11%	3.86s
EPR	**4.24%**	**6.33%**	**5.87%**	3.72s

Table 2: Comparison with SoTA Methods on CVRPLIB-X.

Method	(0,200]	(200,1002]	Total	Time
LKH3	0.00%	0.00%	0.00%	31s
POMO	1.67%	15.56%	7.38%	0.69s
Sym-POMO	2.19%	16.55%	8.05%	0.69s
Omni-POMO	4.43%	8.68%	6.10%	0.69s
ELG-POMO	**1.18%**	5.94%	3.12%	1.19s
LEHD	2.19%	**3.29%**	**3.06%**	1.89s
BQ	1.96%	3.29%	2.50%	3.76s
DIFUSCO	1.96%	12.67%	6.33%	30.86s
EPR	1.19%	5.87%	3.10%	1.19s

Table 3: Comparison with SoTA Methods on TSPLIB.

Method	A1	A2	G	L1	L2
POMO	122.2%	99.1%	95.9%	154.3%	140.6%
Sym-POMO	96.0%	137.9%	122.5%	95.7%	150.2%
Omni-POMO	102.0%	28.5%	38.1%	16.0%	29.4%
ELG-POMO	12.3%	17.3%	13.4%	12.0%	21.0%
LEHD	21.6%	27.4%	19.8%	14.5%	25.9%
BQ	13.4%	**17.0%**	14.6%	14.6%	24.7%
EPR	**10.3%**	18.2%	**12.3%**	**10.9%**	**18.8%**

Table 4: Comparison with SoTA Methods on Very Large-Scale Real-World Instances (CVRPLIB-XXL)

Method Configuration	CVRPLIB-X	TSPLIB
ELG-POMO (Baseline)	6.42	3.30
+ Core EPR (w/o Symmetry)	6.32	3.23
+ Full EPR (with Symmetry)	**5.96**	**3.13**

Table 5: Ablation study of the EPR components' contributions, evaluated on CVRPLIB-X and TSPLIB test sets.

Setting	CVRPLIB-X	Setting	TSPLIB
$P = 0.05$	6.28	$P = 0.05$	3.45
$P = 0.1$	5.96	$P = 0.1$	3.13
$P = 0.2$	6.10	$P = 0.2$	3.23
$K = 2$	6.32	$K = 15$	3.26
$K = 3$	5.96	$K = 20$	3.13
$K = 4$	6.14	$K = 25$	3.25

Table 6: Sensitivity analysis for hyperparameters P and K.

The choice of pattern length K demonstrates exceptional robustness. Crucially, even the least optimal choice of K for each problem (6.32% for CVRP and 3.26% for TSP) still yields a result superior to the strong ELG-POMO baseline (6.42% and 3.30% respectively). This finding is significant, as it demonstrates that EPR provides a consistent benefit without requiring precise, problem-specific hyperparameter optimization.

Conclusion

We addressed the critical challenge of premature convergence that hinders the generalization of neural VRP solvers. We introduced the Elite-Pattern Reinforcement (EPR) strategy, a novel method that guides training by integrating structural patterns from high-quality heuristic solutions. Our extensive experiments on the CVRPLIB and TSPLIB benchmarks demonstrate that EPR significantly enhances the performance of state-of-the-art construction methods.

Acknowledgments

The work is funded by Jilin Provincial Science and Technology Development Plan Project No. 20240302084GX.

References

Alqahtani, H.; and Kumar, G. 2024. Efficient routing strategies for electric and flying vehicles: A comprehensive hybrid metaheuristic review. *IEEE Transactions on Intelligent Vehicles*.

Arnold, F.; Gendreau, M.; and Sörensen, K. 2019. Efficiently solving very large-scale routing problems. *Comput. Oper. Res.*, 107: 32–42.

Bello, I.; Pham, H.; Le, Q. V.; Norouzi, M.; and Bengio, S. 2017. Neural Combinatorial Optimization with Reinforcement Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Bengio, Y.; Lodi, A.; and Prouvost, A. 2021. Machine learning for combinatorial optimization: A methodological tour d'horizon. *Eur. J. Oper. Res.*, 290(2): 405–421.

Bi, J.; Ma, Y.; Wang, J.; Cao, Z.; Chen, J.; Sun, Y.; and Chee, Y. M. 2022. Learning Generalizable Models for Vehicle Routing Problems via Knowledge Distillation. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Cappart, Q.; Chételat, D.; Khalil, E. B.; Lodi, A.; Morris, C.; and Velickovic, P. 2023. Combinatorial Optimization and Reasoning with Graph Neural Networks. *J. Mach. Learn. Res.*, 24: 130:1–130:61.

Chen, W.; Liu, S.; Ong, Y.; and Tang, K. 2023. Neural Influence Estimator: Towards Real-time Solutions to Influence Blocking Maximization. *CoRR*, abs/2308.14012.

Christiaens, J.; and Berghe, G. V. 2020. Slack Induction by String Removals for Vehicle Routing Problems. *Transp. Sci.*, 54(2): 417–433.

Christofides, N. 1979. The vehicle routing problem. *Combinatorial optimization*.

Drakulic, D.; Michel, S.; Mai, F.; Sors, A.; and Andreoli, J. 2023. BQ-NCO: Bisimulation Quotienting for Generalizable Neural Combinatorial Optimization. *CoRR*, abs/2301.03313.

Gao, C.; Shang, H.; Xue, K.; Li, D.; and Qian, C. 2024. Towards Generalizable Neural Solvers for Vehicle Routing Problems via Ensemble with Transferrable Local Policy. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, 6914–6922. ijcai.org.

Gu, Y.; Sun, Q.; and Cai, X. 2020. Multiagent Reinforcement Learning for Combinatorial Optimization. In Zhang, H.; Zhang, Z.; Wu, Z.; and Hao, T., eds., *Neural Computing for Advanced Applications - First International Conference, NCAA 2020, Shenzhen, China, July 3-5, 2020, Proceedings*, volume 1265 of *Communications in Computer and Information Science*, 23–34. Springer.

He, Y.; and Bao, F. S. 2020. Circuit Routing Using Monte Carlo Tree Search and Deep Neural Networks. *CoRR*, abs/2006.13607.

Helsgaun, K. 2017. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. *Roskilde: Roskilde University*, 12: 966–980.

Jiang, Y.; Cao, Z.; Wu, Y.; and Zhang, J. 2023. Multi-view graph contrastive learning for solving vehicle routing problems. In Evans, R. J.; and Shpitser, I., eds., *Uncertainty in Artificial Intelligence, UAI 2023, July 31 - 4 August 2023, Pittsburgh, PA, USA*, volume 216 of *Proceedings of Machine Learning Research*, 984–994. PMLR.

Jiang, Y.; Wu, Y.; Cao, Z.; and Zhang, J. 2022. Learning to Solve Routing Problems via Distributionally Robust Optimization. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 9786–9794. AAAI Press.

Joshi, C. K.; Cappart, Q.; Rousseau, L.; and Laurent, T. 2021. Learning TSP Requires Rethinking Generalization. In Michel, L. D., ed., *27th International Conference on Principles and Practice of Constraint Programming, CP 2021, Montpellier, France (Virtual Conference), October 25-29, 2021*, volume 210 of *LIPIcs*, 33:1–33:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Khalil, E. B.; Dai, H.; Zhang, Y.; Dilkina, B.; and Song, L. 2017. Learning Combinatorial Optimization Algorithms over Graphs. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 6348–6358.

Kim, M.; Park, J.; and Park, J. 2022. Sym-NCO: Leveraging Symmetricity for Neural Combinatorial Optimization. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Konstantakopoulos, G. D.; Gayialis, S. P.; and Kechagias, E. P. 2022. Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. *Oper. Res.*, 22(3): 2033–2062.

Kool, W.; van Hoof, H.; and Welling, M. 2019. Attention, Learn to Solve Routing Problems! In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Kwon, Y.; Choo, J.; Kim, B.; Yoon, I.; Gwon, Y.; and Min, S. 2020. POMO: Policy Optimization with Multiple Optima for Reinforcement Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference*

on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Li, Y.; Fang, E. X.; Xu, H.; and Zhao, T. 2020. International Conference on Learning Representations 2020. In *International Conference on Learning Representations 2020*.

Liu, S.; Zhang, Y.; Tang, K.; and Yao, X. 2022. How Good Is Neural Combinatorial Optimization? *CoRR*, abs/2209.10913.

Lu, H.; Zhang, X.; and Yang, S. 2020. A Learning-based Iterative Method for Solving Vehicle Routing Problems. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Luo, F.; Lin, X.; Liu, F.; Zhang, Q.; and Wang, Z. 2023. Neural Combinatorial Optimization with Heavy Decoder: Toward Large Scale Generalization. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Manchanda, S.; Michel, S.; Drakulic, D.; and Andreoli, J. 2022. On the Generalization of Neural Combinatorial Optimization Heuristics. In Amini, M.; Canu, S.; Fischer, A.; Guns, T.; Novak, P. K.; and Tsoumakas, G., eds., *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2022, Grenoble, France, September 19-23, 2022, Proceedings, Part V*, volume 13717 of *Lecture Notes in Computer Science*, 426–442. Springer.

Matsuzaki, J.; Sakakibara, K.; Nakamura, M.; and Watanabe, S. 2024. Large neighborhood local search method with MIP techniques for large-scale machining scheduling with many constraints. *J. Supercomput.*, 80(9): 12297–12312.

Reinelt, G. 1991. TSPLIB - A Traveling Salesman Problem Library. *INFORMS J. Comput.*, 3(4): 376–384.

Song, W.; Mi, N.; Li, Q.; Zhuang, J.; and Cao, Z. 2024. Stochastic Economic Lot Scheduling via Self-Attention Based Deep Reinforcement Learning. *IEEE Trans Autom. Sci. Eng.*, 21(2): 1457–1468.

Sun, Z.; and Yang, Y. 2023. DIFUSCO: Graph-based Diffusion Solvers for Combinatorial Optimization. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press. ISBN 978-0-262-19398-6.

Trick, M. A. 2008. David L. Applegate, Robert E. Bixby, Vasek Chvátal , William J. Cook. The Traveling Salesman Problem: A Computational Study, Princeton University Press, Princeton, 2007, ISBN-13: 978-0-691-12993-8, 606 pp. *Oper. Res. Lett.*, 36(2): 276–277.

Uchoa, E.; Pecin, D.; Pessoa, A. A.; Poggi, M.; Vidal, T.; and Subramanian, A. 2017. New benchmark instances for the Capacitated Vehicle Routing Problem. *Eur. J. Oper. Res.*, 257(3): 845–858.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Vidal, T. 2022. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Comput. Oper. Res.*, 140: 105643.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer Networks. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2692–2700.

Williams, R. J. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.*, 8: 229–256.

Zhang, C.; Song, W.; Cao, Z.; Zhang, J.; Tan, P. S.; and Xu, C. 2020. Learning to Dispatch for Job Shop Scheduling via Deep Reinforcement Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhou, J.; Wu, Y.; Song, W.; Cao, Z.; and Zhang, J. 2023. Towards Omni-generalizable Neural Methods for Vehicle Routing Problems. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 42769–42789. PMLR.