STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **16**, October 2026, pp 3050–3064.
Published online in International Academic Press (www.IAPress.org)

Adaptive Operator Selection in ALNS Using Q-Learning: A Comparative Study of Roulette, Softmax, UCB, ϵ-Greedy and QTS Strategies for CVRP

Hajar Boualamia [1,*], Abdelmoutalib Metrane [2], Imad Hafidi [1]

[1] *University Sultan Moulay Slimane, Beni Mellal, Morocco*
[2] *Cadi Ayyad University, Marrakech, Morocco*

Abstract This paper presents a comprehensive comparative study of operator selection strategies within a previously proposed Q-learning-based Adaptive Large Neighborhood Search (ALNS) framework for solving the Capacitated Vehicle Routing Problem (CVRP). Unlike classical ALNS approaches, where operators are selected using predefined heuristic rules, the proposed framework dynamically learns effective destroy–repair operator pairs during the search process. In addition, a new Q-Value Thompson Sampling (QTS) action selection strategy is introduced and compared with the conventional Roulette Wheel Selection baseline as well as three widely used reinforcement learning policies, namely ϵ-greedy, Softmax, and Upper Confidence Bound (UCB). The five strategies are evaluated using representative benchmark instances from CVRPLIB under identical experimental conditions over 30 independent runs. The comparative analysis considers solution quality, computational time, convergence behaviour, robustness through boxplot analysis, and statistical significance using Friedman and Wilcoxon signed-rank tests. The results show that the investigated strategies exhibit complementary strengths. These findings provide new insights into the impact of action selection policies on adaptive operator selection and demonstrate that QTS constitutes a robust and competitive alternative within the ALNS framework.

Keywords Capacitated Vehicle Routing Problem, Metaheuristics, Adaptive Large Neighborhood Search, Reinforcement Learning, Q-Learning, Operator Selection

AMS 2020 subject classifications 90B06, 68T05, 90C59

DOI: 10.19139/soic-2310-5070-3956

1. Introduction

The Vehicle Routing Problem (VRP) has attracted continuous attention from researchers and remains one of the most active research areas in operations research. This is mainly due to its strong practical relevance, as transportation costs often represent a significant portion of logistics expenses in many companies. The VRP consists of serving a set of customers, each with a specific demand, from a central depot using a fleet of vehicles, while minimizing the total travel cost.

The Capacitated Vehicle Routing Problem (CVRP) extends the classical VRP by introducing vehicle capacity constraints, making it more suitable for real-world applications. Dantzig and Ramser [1] were the first to address this problem using an exact algorithm. However, since the CVRP is NP-hard, exact methods are limited to small-sized instances. For larger instances, metaheuristic approaches are widely used to obtain high-quality approximate solutions within reasonable computational times.

Various metaheuristics have been proposed to solve the CVRP. For example, [2] introduced an Artificial Bee Colony (ABC) algorithm, while [3] proposed a record-to-record travel (RRT)-based method with promising results for large instances. Tabu Search has also been applied successfully [4], along with Ant Colony Optimization (ACO)

*Correspondence to: Hajar Boualamia (Email: hajar.boualamia@usms.ac.ma), University Sultan Moulay Slimane, Beni Mellal, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

[5] and Active Guided Evolutionary Strategies (AGES) [6]. In addition, Variable Neighborhood Search (VNS) has proven to be an effective approach for solving VRP variants [7, 8].

Another important class of methods is adaptive metaheuristics, where operators are selected dynamically based on their past performance [9, 10]. Among these approaches, the Adaptive Large Neighborhood Search (ALNS), proposed by Ropke and Pisinger [11], is one of the most effective methods for solving VRP problems. Its success lies in its ability to iteratively apply destruction and repair operators while adapting their selection during the search. Several studies have demonstrated the efficiency of ALNS for solving VRP variants [12].

A key challenge in metaheuristic optimization is maintaining a balance between exploration and exploitation. Exploration allows the algorithm to investigate new regions of the search space, while exploitation focuses on improving promising solutions. An inappropriate balance may lead to premature convergence or inefficient search behavior, thus negatively affecting the solution quality.

In recent years, machine learning techniques have been increasingly integrated into metaheuristics to enhance their adaptability and performance. The search process of metaheuristics generates a large amount of data, such as solution trajectories, elite solutions, and local optima, which can be exploited to improve the search process. Several studies [13] have highlighted the potential of combining machine learning with metaheuristics.

Among these approaches, reinforcement learning (RL) has emerged as a powerful framework for guiding decision-making processes in optimization problems [14]. In RL, an agent interacts with its environment by selecting actions based on the current state and receiving feedback in the form of rewards. Unlike supervised learning, RL does not require labeled data, as the agent learns through trial and error. Q-learning [15] is one of the most widely used model-free RL methods, allowing the agent to learn an optimal policy based on experience.

In the context of ALNS, operator selection can be naturally modeled as a sequential decision-making problem. Each operator represents an action, and the quality of the resulting solution defines the reward. This perspective enables the integration of reinforcement learning to dynamically adapt operator selection during the search process.

Building upon our previous Q-learning-based ALNS framework [16], this work focuses on the comparative evaluation of different action selection strategies for adaptive operator selection. In addition, we introduce a Q-Value Thompson Sampling (QTS) strategy that combines tabular Q-learning with a Gaussian Thompson Sampling-inspired action selection mechanism.

The main contributions of this paper are as follows:

- We extend our previously proposed Q-learning-based ALNS framework by introducing a Q-Value Thompson Sampling (QTS) operator selection strategy based on Gaussian sampling around the learned Q-values.
- We conduct a comprehensive comparative study between the conventional Roulette Wheel Selection mechanism and four reinforcement learning-based action selection strategies, namely ϵ-greedy, Softmax, UCB and the proposed QTS strategy.
- We provide an extensive experimental evaluation on representative CVRPLIB benchmark instances, including convergence analysis, robustness analysis using boxplots, and statistical validation based on Friedman and Wilcoxon signed-rank tests.
- We provide an experimental analysis of the strengths and limitations of each operator selection strategy and provide practical insights into their exploration–exploitation behaviour within the ALNS framework.

The remainder of this paper is organized as follows. Section 2 presents the literature review. Section 3 describes the proposed method. Section 4 details the algorithmic framework. Section 5 reports the experimental results and analysis. Finally, Section 6 concludes the paper and outlines future research directions.

2. Literature Review

The Adaptive Large Neighborhood Search (ALNS), introduced by Ropke and Pisinger [11], is a powerful metaheuristic framework designed to explore large neighborhoods through iterative destruction and repair

mechanisms. At each iteration, a destruction operator removes part of the current solution, while a repair operator reinserts the removed elements to generate a new feasible solution.

In this framework, operator selection plays a central role in the effectiveness of the search process. The selected operators directly influence the balance between exploration and exploitation. Efficient operator selection promotes the exploration of promising regions of the solution space while improving solution quality. Conversely, inappropriate selection may favor poorly performing operators, slow down convergence, reduce search diversity, and lead to inferior final solutions. Therefore, operator selection is not merely a technical component of ALNS, but a key factor affecting its overall performance [11, 17]. According to the original formulation of ALNS [11], the algorithm relies on four main components: the set of destruction and repair operators, the operator selection mechanism (adaptation mechanism), the acceptance criterion, and the stopping criterion.

Several studies have proposed improvements for most of these components. In particular, various acceptance criteria have been investigated, including the Metropolis criterion, greedy acceptance, record-to-record travel, threshold acceptance, and Pareto-based mechanisms [30]. Similarly, different stopping criteria have been explored in the literature, such as a fixed number of iterations, a limit on iterations without improvement, or a maximum execution time.

However, despite these advances, the operator selection mechanism remains relatively less explored compared to other components. Most existing studies rely on simulated annealing as the primary adaptation mechanism in ALNS. Although several RL-based approaches have recently been proposed, comprehensive comparisons between action selection policies remain limited [18].

However, simulated annealing presents several limitations when used in this context. Its performance is highly sensitive to parameter settings, such as the initial temperature and the cooling schedule. Poor parameter calibration may result in excessive exploration or premature convergence, both of which negatively affect solution quality [**?**].

In addition, within the ALNS framework, each iteration involves computationally expensive destruction and repair operations. As a result, simulated annealing may require a large number of iterations to reach high-quality solutions, leading to increased computational time.

Although a few studies have proposed improvements, such as refined weighting schemes or enhanced reward mechanisms [19], these contributions remain limited and do not fundamentally address the limitations of the adaptation mechanism. Due to the limitations of traditional adaptation mechanisms and the increasing complexity of combinatorial optimization problems, machine learning (ML) has emerged as a promising approach to enhance metaheuristics.

ML enables algorithms to learn from past search experiences in order to improve their performance and robustness. Previous studies have shown that ML techniques can be used to dynamically select appropriate heuristics based on problem characteristics [31, 20]. In addition, supervised learning approaches have been proposed to predict solution quality or automatically adjust algorithm parameters [21, 22].

Among ML techniques, reinforcement learning (RL) has attracted particular attention for improving decision-making in optimization algorithms. RL allows an agent to learn optimal actions through interactions with the environment, without requiring labeled data. Several works have demonstrated its effectiveness in solving combinatorial optimization problems such as the TSP and VRP [23, 24].

More recently, hybrid approaches combining RL and ALNS have been proposed, where operator selection is modeled as an agent-environment interaction, with operators representing actions and solutions representing states [16, 25, 26]. These approaches show that reinforcement learning can significantly improve the adaptability of metaheuristics by automating operator selection and dynamically adjusting the search process. In this context, a key aspect of integrating reinforcement learning into ALNS lies in the action selection strategy, which determines the operator applied at each iteration.

Within the Q-learning framework, each destruction or repair operator can be modeled as an action, while the quality of the resulting solution defines the reward. The learning process aims to estimate the expected effectiveness of each operator based on past interactions.

To maintain a balance between exploration and exploitation, the choice of an appropriate action selection strategy is essential. Several methods have been proposed in the reinforcement learning literature, including ε-greedy, Softmax (Boltzmann exploration), Upper Confidence Bound (UCB), and Thompson Sampling (QTS) [32, 28, 29].

Each strategy provides a different trade-off between exploring new operators and exploiting those already identified as effective.

Despite their importance, these strategies have not been extensively compared within the ALNS framework. Therefore, this work focuses on a comprehensive experimental evaluation of different action selection strategies integrated into Q-learning-based ALNS, with the aim of improving operator selection and enhancing overall algorithm performance.

3. Proposed Method

The approach described and explained in this research is based on integrating machine learning into the ALNS metaheuristic. More precisely, Q-Learning will be included in the ALNS operator selection process. The primary goal is to make operator selection more dynamic and intelligent. This study aims primarily to analyze and compare various integrated operator selection strategies in the context of Q-learning applied to large-scale adaptive search (ALNS). The selection of destruction and repair operators has a significant impact on the effectiveness of ALNS, particularly with regard to striking the right balance between exploration and exploitation. The classic ALNS relies on a weighted selection mechanism, which undergoes changes based on observed performance after a certain number of iterations. This mechanism, although effective and simple, nevertheless has several limitations, including limited responsiveness and heavy reliance on predefined parameters. To overcome these limitations, we suggest conceptualizing this process as a sequential decision-making problem. In this approach, the choice of operators is acquired through interactions between an agent (Q-learning) and an environment (the ALNS process). In this context, Q-learning evaluates and modifies each operator's value according to how it affects the solution's quality. From a set of available operators, the agent chooses one at each cycle, applies it to the current solution s_t, and then observes a new solution, s_{t+1}, and a reward, r_t. The goal is to learn an action value function, the quality of the action in state s_t. Therefore, Q-learning aims to gradually discover which operators work best in different search settings through iterations. The action selection mechanism (ε-greedy, Softmax, UCB, or QTS) controls the balance between exploration (trying out novel operators) and exploitation (using operators that have previously been proven to work). As a result, the ALNS-Q-learning combination creates an adaptive optimization framework where the agent is always learning to focus the search on elements of the solution space that show promise. In this context, several selection approaches are examined, including ε-greedy, Softmax, Upper Confidence Bound (UCB), and Q-Value Thompson Sampling (QTS). Each of these methods represents a different approach to balancing exploration (testing new operators) and exploitation (using the most effective operators). This comparative study aims to examine the effect of these strategies on convergence speed, the quality of final solutions, and the consistency of the learning process in the context of Q-learning–ALNS. As a result, the agent gradually acquires the ability to determine the most appropriate operators based on the search context. The integration of Q-learning and ALNS leads to a self-adaptive optimization model in which search behavior adjusts according to the results obtained.

4. Architecture of the Q-learning-based ALNS Approach

This section details the structure of the proposed approach, which seeks to replace the classical operator adaptation mechanism of ALNS with dynamic selection based on Q-learning. In addition, several selection tactics are tested to conduct a comparative study and identify the optimal strategy, such as ε-greedy, Softmax, UCB, and Q-Value Thompson Sampling, offering a satisfactory balance between exploration and exploitation.

4.1. Overall Algorithm

The overall structure of the algorithm follows the following logic:

- Initialization: We begin by creating a random initial solution to subsequently determine the rate of progression. We then define all the destruction and repair operators. In this research, we use repair operators used in [16],

Algorithm 1 ALNS with Q-learning for a given selection strategy

Require: Learning rate α, discount factor γ, selection strategy π
Ensure: Best solution s^*
Initialize the Q-table
$s \leftarrow \text{INITIALSOLUTION}()$
$s^* \leftarrow s$
while the stopping criterion is not satisfied **do**
    Select action a_t according to strategy π
    Generate candidate solution $s' \leftarrow \text{DESTROYREPAIR}(s, a_t)$
    Compute reward r_t
    Update $Q(s_t, a_t)$ using the Q-learning rule
    if $f(s') < f(s^*)$ **then**
        $s^* \leftarrow s'$
    end if
    Accept or reject s' according to simulated annealing
    Update the current solution and the next state
end while
return s^*

such as the basic greedy heuristic, the regret heuristic, and destruction operators such as random removal, worst removal, related removal, and cluster removal. We assign initial values of zero to Q(a), then opt for the e-greedy selection strategy as a first approach.

- ALNS loop: At each iteration, the operator pair chosen according to the selection strategy implemented on the current solution is used to create a new solution. The quality of the solution produced is then evaluated. The solution is accepted or rejected according to the simulated annealing acceptance criterion. Next, the reward associated with the operator is determined based on the improvement of the solution. Finally, the Q(s,a) values are updated.

- Stop criterion: A straightforward acceptance criterion would be to only accept solutions that are superior to the current one. However, it seems sensible to occasionally accept solutions that are worse than the current answer because such a heuristic is prone to getting trapped in a local minimum. We have selected simulated annealing as our local search framework at the master level in order to accomplish this. The objective value is denoted by $f(s)$. The solution s' is always accepted if $f(s') \leq f(s)$. Otherwise, the solution may be accepted with a probability function that is determined by the temperature parameter τ. The probability function is

$$\exp\left(-\frac{f(s') - f(s)}{\tau}\right)$$

after each repetition; the temperature decreases using the formula $\tau = c \cdot \tau$.

4.2. Operator–State–Action–Reward Modeling

The integration of Q-learning into the ALNS framework is formulated as a sequential decision-making process. At each iteration t, the reinforcement learning agent observes the current state s_t, selects an action a_t, receives a reward r_t, and transitions to a new state s_{t+1}. The interaction between the learning agent and the search process is therefore represented by the tuple (s_t, a_t, r_t, s_{t+1}).

In the proposed framework, the state is defined as the pair of destruction and repair operators applied during the previous iteration. Formally, the state at iteration t is represented as

$$s_t = (d_{t-1}, r_{t-1}), \tag{1}$$

where d_{t-1} and r_{t-1} denote the destruction and repair operators selected at iteration $t-1$, respectively.

This low-dimensional state representation was intentionally adopted to reduce the complexity of the learning problem while capturing the short-term dependencies between successive operator pairs. The underlying assumption is that the effectiveness of a given operator may depend on the operator applied during the previous iteration. Consequently, the learning mechanism aims to identify beneficial operator sequences without introducing a high-dimensional state space.

An action corresponds to the selection of a pair of destruction and repair operators to be applied during the current iteration:

$$a_t = (d_t, r_t). \tag{2}$$

The set of destruction operators considered in this study includes Random Removal, Worst Removal, Related Removal, and Cluster Removal, whereas the repair operators consist of Greedy Insertion and Regret Insertion. Since four destruction operators and two repair operators are considered, the cardinalities of the state and action spaces are given by

$$|\mathcal{S}| = |\mathcal{A}| = 4 \times 2 = 8. \tag{3}$$

After applying the selected action a_t, a new solution is generated and the next state is updated according to the selected operator pair:

$$s_{t+1} = a_t. \tag{4}$$

The reward function quantifies the effectiveness of the selected operator pair according to the relative improvement of the objective value. Let $f(s_t)$ denote the objective value of the current solution before applying action a_t, and let $f(s_{t+1})$ denote the objective value of the resulting solution. The reward is computed as

$$r_t = \frac{f(s_t) - f(s_{t+1})}{f(s_t)}. \tag{5}$$

A positive reward indicates an improvement in solution quality, whereas a negative reward corresponds to a deterioration of the objective value. A reward equal to zero indicates that the solution quality remains unchanged. This normalized formulation ensures that reward values remain comparable across instances with different objective scales.

The Q-values are updated according to the standard Q-learning rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right], \tag{6}$$

where α denotes the learning rate and γ represents the discount factor. In this study, the parameter values are set to $\alpha = 0.1$ and $\gamma = 0.9$.

The learned Q-values are exploited through four action selection mechanisms, namely ϵ-greedy, Softmax, Upper Confidence Bound (UCB), and Q-learning Thompson Sampling (QTS). Each strategy defines a specific exploration-exploitation trade-off while relying on the same Q-learning update mechanism.

4.3. Selection Strategies

In the Q-learning–ALNS approach, the choice of operators is a determining factor that influences search performance. It provides the option to choose the destruction-repair operator pair for the current solution at each cycle. The behavior of this selection determines the balance between exploration and exploitation. This research examined four selection approaches drawn from reinforcement learning: ε-greedy, Softmax, Upper Confidence Bound (UCB), and Q-Value Thompson Sampling (QTS).

4.3.1. ϵ-greedy

The ε-greedy strategy [32] is based on a basic rule: at each cycle, the agent opts for the operator with the highest Q value with a probability of $1-\varepsilon$, and chooses an operator at random with a probability of ε. Formally:

$$a_t = \begin{cases} \arg\max_a Q(s_t, a), & \text{with probability } 1-\varepsilon, \\ \text{a random action from the action set } A, & \text{with probability } \varepsilon. \end{cases}$$

This tool prioritizes simplicity and efficiency; however, it may be subject to suboptimal exploration if the ε parameter is not set appropriately. In the context of ALNS, this method showed excellent results initially [16], but sometimes converged late in cases of high operator diversity.

4.3.2. Softmax

The Softmax approach [27] is based on a probabilistic perspective, according to which the probability of choosing an operator pair is related to its Q value, adjusted by a temperature parameter τ. Formally:

$$P(a_i \mid s_t) = \frac{\exp\left(\dfrac{Q(s_t, a_i)}{\tau}\right)}{\sum\limits_{a_j \in A} \exp\left(\dfrac{Q(s_t, a_j)}{\tau}\right)},$$

When τ is large, selection is similar to random selection (exploration), whereas a small τ favors the most effective operators (exploitation). Thus, this approach offers constant monitoring of the trade-off between exploration and exploitation, unlike ε-greedy. In the context of ALNS, Softmax offers more precise adjustment and diversification of the search process [19].

4.3.3. Upper Confidence Bound

The UCB strategy [28] aims to automate the balance between exploration and exploitation by incorporating the concept of uncertainty into the choice. A value is associated with each operator:

$$Q(s_t, a_i) + c\sqrt{\frac{\ln t}{n_i}},$$

where n_i symbolizes the number of times the operator has been selected, t the total number of iterations, and c a confidence-related parameter. This approach favors the selection of less investigated operators while taking into account their past performance. UCB has the advantage of not requiring an arbitrary exploration parameter such as ε, while guaranteeing proven theoretical convergence [28]. In the context of ALNS, it offers a more balanced adjustment between diversification and intensification.

4.3.4. Q-Value Thompson Sampling

The proposed Q-Value Thompson Sampling (QTS) strategy combines the standard Q-learning update mechanism with a Gaussian Thompson Sampling-inspired action selection policy. First, the Q-values associated with each destroy–repair operator pair are updated using the classical Q-learning update rule after each iteration. During the operator selection phase, instead of directly selecting the action with the highest Q-value, a random sample is drawn for each candidate action from a Gaussian distribution centered on its current Q-value:

$$\tilde{Q}(s, a) \sim \mathcal{N}\left(Q(s, a), \sigma^2\right), \tag{7}$$

where $Q(s, a)$ denotes the learned Q-value of action a in state s, and σ is a fixed standard deviation controlling the exploration level. The selected action is then determined as:

$$a_t = \arg\max_a \tilde{Q}(s, a). \tag{8}$$

This mechanism introduces stochastic exploration while preserving the exploitation of high-value actions. Actions with larger Q-values have a higher probability of being selected, whereas actions with lower Q-values still retain a non-zero probability of being explored through Gaussian sampling. Consequently, the proposed QTS strategy can be viewed as a hybrid approach in which Q-learning estimates the long-term utility of each operator pair, while Gaussian Thompson Sampling governs the action selection process. This combination provides a dynamic balance between exploration and exploitation and improves the robustness of operator selection, particularly in search spaces where operator performance changes over time.

5. Experimental Setup and Results

Each operator selection strategy, namely the conventional Roulette Wheel Selection baseline, ε-greedy, Softmax, UCB, and the proposed QTS (Q-Value Thompson Sampling) strategy, was evaluated independently using the same ALNS framework described in Algorithm **??**. For each strategy, 30 independent runs were performed on every benchmark instance under identical experimental conditions. To ensure a fair comparison, all algorithmic parameters, stopping criteria, cooling schedule, and initial solutions were kept identical across all experiments. The reported results correspond to aggregated statistics over these independent runs, including the mean solution cost, standard deviation, best solution cost, computational time, convergence behaviour, robustness analysis using boxplots, and statistical validation based on the Friedman and Wilcoxon signed-rank tests.

5.1. Test Instances

The efficacy of the proposed methodology was examined using a set of benchmark datasets often used in academic literature concerning vehicle routing problems (VRP), specifically for the evaluation of ALNS-type metaheuristics. The chosen case studies include many features and levels of complexity to provide a complete and representative study.

Representative benchmark instances from the CVRPLIB library were selected to cover different problem sizes and complexity levels, which include the following:

Instances of small size, as introduced by Augerat et al. (1995), simplify the analysis of initial behavior and convergence speed of the strategies.

Instances of intermediate dimensions, as proposed by Rochat and Taillard (1995), include cases with 75 to 380 customers, reflecting realistic scenarios commonly observed in industrial applications.

Large-scale instances introduced by Golden et al. and Li et al. aim to evaluate the robustness and scalability of operator selection strategies.

5.2. Parameter Settings

The parameters for the reinforcement learning part and the operator selection processes were established before the studies and stayed the same for all test cases. These values were chosen based on generally established standards in the literature and the initial tests of calibration.

The learning rate α was set to 0.1 in the Q-learning mechanism. This allowed the algorithm to slowly change the estimated utility of operators without causing unstable oscillations. The discount factor γ was set to 0.9, which means that long-term returns are more important than short-term ones, although short-term improvements are also taken into account.

In the Softmax technique, the temperature parameter τ was set to 0.5. This puts moderate selection pressure on the Q-values by smoothing down their probability distribution and stopping them from converging too quickly.

In Q-based Thompson Sampling (QTS), the sampling variance was set to 1.0, which allowed for some uncertainty at the start of the search but gradually reduced exploration as more information was collected.

The UCB technique set the exploration coefficient c to 2, which is a common quantity used to balance the employment of high-reward operators with the search for less common options.

For the ϵ-greedy strategy, the exploration probability ϵ was set to 0.1, ensuring a reasonable balance between exploration and exploitation throughout the search process.

All parameters were kept the same during the tests to make sure that the various operator selection procedures could be compared with equal efficacy. The table below shows the values used for each parameter.

Table 1. Experimental parameter settings.

Description	Parameter	Value
Q-learning learning rate	α	0.1
Discount factor	γ	0.9
Exploration probability in ϵ-greedy	ϵ	0.1
Softmax temperature	τ	0.5
UCB exploration coefficient	c	2.0
QTS sampling variance	σ^2	1.0

It should be noted that the primary objective of this study is to compare the effectiveness of different operator selection strategies under identical experimental conditions rather than to optimize the hyperparameters of each strategy individually. Therefore, the selected parameter values were kept fixed throughout all experiments to ensure a fair comparison and to isolate the impact of the action selection mechanisms on the overall performance. Although a comprehensive sensitivity analysis could provide additional insight into the influence of strategy-specific parameters, such an investigation is beyond the scope of the present work and will be considered in future research.

5.3. Comparative Results and Analysis

Table 2. Comparison of the five operator selection strategies on the selected CVRPLIB benchmark instances (30 independent runs) in terms of mean cost and standard deviation. The Best Known Solution (BKS) is reported for each instance. The best mean solution cost is highlighted in bold.

Instance	BKS	Roulette	ϵ-greedy	Softmax	UCB	QTS
A-n45-k6	944.00	963.40±16.30	962.03±8.25	962.13±8.63	960.73±10.78	**958.27±11.36**
B-n50-k7	741.00	**741.00±0.00**	**741.00±0.00**	**741.00±0.00**	**741.00±0.00**	**741.00±0.00**
B-n50-k8	1312.00	1323.43±11.06	1329.80±21.12	**1322.57±8.97**	1325.90±13.51	1322.60±9.05
B-n52-k7	747.00	753.03±15.72	751.30±12.67	749.97±10.72	**747.13±0.43**	752.97±15.66
B-n57-k9	1598.00	**1621.90±15.76**	1622.60±16.32	1627.00±14.67	1630.70±12.54	1627.20±12.25
B-n63-k10	1496.00	1538.87±21.22	1538.80±19.67	1537.07±17.96	1540.57±20.60	**1533.93±17.69**
B-n64-k9	861.00	**865.40±10.20**	868.33±12.70	866.20±9.69	866.47±17.90	867.20±11.75
B-n68-k9	1272.00	1293.30±8.60	1293.13±9.50	1295.30±15.18	1299.13±18.23	**1290.03±8.45**
B-n78-k10	1221.00	**1261.37±15.98**	1261.50±17.44	1265.43±16.61	1271.73±16.86	1267.60±16.67
Golden 1	5623.47	**5692.93±81.54**	5867.70±155.37	5712.70±117.49	5706.53±100.70	5705.27±109.92
Golden 2	8404.61	8882.80±245.71	9123.30±270.24	8861.90±219.82	**8795.90±211.45**	8865.83±248.32
Tai75a	1618.36	1643.27±24.62	**1636.57±19.83**	1647.10±21.02	1642.37±19.25	1649.10±22.48
Tai75d	1365.42	1386.97±19.77	1396.43±24.55	**1381.90±13.44**	1391.90±20.70	1382.03±13.58
Tai100d	1581.25	1618.27±15.07	1626.27±33.64	1626.70±28.34	1633.53±25.92	**1613.03±21.66**
Tai150c	2341.84	2512.57±75.20	2544.67±84.22	2495.33±72.96	**2475.40±58.41**	2516.30±80.55
Tai150d	2645.39	2801.03±81.42	2804.97±78.41	**2766.90±56.54**	2776.03±63.45	2776.10±66.43

The results reported in Table 2 show that the five operator selection strategies exhibit different behaviours across the benchmark instances. No single strategy dominates all evaluation criteria across all benchmark instances,

Table 3. Global summary over the 16 benchmark instances.

Method	Avg. mean	Avg. std	Avg. best	Avg. time	Avg. gap	Avg. rank	Wins
Roulette	2181.22	41.14	2120.63	**122.15**	2.45	2.88	5
ϵ-greedy	2210.53	49.00	2136.25	539.47	2.99	3.56	2
Softmax	2178.70	39.50	2120.00	164.10	**2.36**	**2.63**	4
UCB	**2175.31**	**38.17**	**2117.38**	190.18	2.39	3.19	4
QTS	2179.28	41.61	2121.94	447.69	2.37	2.75	5

Table 4. Global convergence indicators. Lower values of Iter90, Iter95, and normalized AUC indicate faster convergence.

Method	Iter90	Iter95	Norm. AUC	Improvement (%)
Roulette	89.66	**229.32**	0.0033	67.80
ϵ-greedy	108.18	257.27	0.0037	68.53
Softmax	94.09	248.41	0.0033	68.33
UCB	**85.57**	231.70	**0.0032**	**68.91**
QTS	149.32	327.95	0.0039	67.53

confirming that the performance of ALNS strongly depends on both the operator selection mechanism and the characteristics of the considered problem instances. This observation is consistent with the statistical validation presented in Section 5.5, which confirms significant differences among the investigated strategies while highlighting their complementary behaviours.

Table 3 summarizes the overall performance of the five operator selection strategies across all benchmark instances. In addition to the average solution cost, the average rank is reported to provide a global comparison of the investigated methods. The average rank corresponds to the mean position achieved by each strategy over all benchmark instances according to the mean solution cost, where rank 1 indicates the best-performing strategy for a given instance.

Roulette obtains the best mean cost on five benchmark instances, including one shared tie, confirming that this simple selection mechanism remains competitive despite its non-adaptive nature. Its main strength lies in its very low computational cost, as shown in Table 3, where it achieves the lowest average running time. This confirms that Roulette is simple, fast and competitive on several benchmark instances. However, its limitation is that it may stagnate early because the selection probabilities mainly depend on accumulated operator scores. Consequently, once a few operator pairs become dominant, the search diversity progressively decreases.

The ϵ-greedy strategy exhibits the weakest overall performance. It records the highest average solution cost, the largest average standard deviation and the worst average rank. These results indicate that ϵ-greedy exhibits the lowest robustness among the investigated strategies. Its main advantage is its simplicity and its ability to encourage exploration during the early stages of the search. However, as the search progresses, the algorithm tends to exploit the currently best operator pair too aggressively, which may lead to premature convergence and increased variability.

Softmax provides one of the best overall compromises. It achieves the lowest average optimality gap and the best average rank while maintaining a relatively low standard deviation. This demonstrates that Softmax offers consistent performance across different benchmark instances. Its probabilistic selection mechanism avoids abrupt decisions and allows multiple promising operator pairs to remain active during the search. Its main limitation is that the selection probabilities become less discriminative when the learned Q-values are very close, making it more difficult to distinguish between competing operators.

UCB achieves the lowest average solution cost, the lowest average standard deviation and the best average best solution. These results indicate that UCB provides the strongest overall solution quality according to the considered absolute performance indicators. Its exploration term encourages the selection of underexplored operator pairs, improving the balance between intensification and diversification. The convergence indicators reported in Table 4 further show that UCB reaches 90% of its final improvement earlier than the other strategies and achieves the lowest normalized AUC, indicating strong overall convergence behaviour. The normalized AUC corresponds to the normalized area under the convergence curve computed over the optimization process. Lower values indicate faster

convergence toward high-quality solutions. However, UCB does not systematically achieve the best rank on every benchmark instance, suggesting that its exploration mechanism may occasionally allocate excessive effort to less promising operator pairs.

QTS obtains the best mean cost on five benchmark instances and achieves the second-best average rank after Softmax. This observation is particularly relevant because QTS is the proposed hybrid strategy combining Q-learning with Gaussian Thompson Sampling-inspired action selection. Its main strength lies in its ability to maintain probabilistic exploration around the learned Q-values. Unlike deterministic exploitation strategies, QTS assigns a non-zero selection probability to several promising operator pairs, helping reduce the risk of premature convergence while maintaining search diversity. This behaviour is particularly beneficial on instances such as A-n45-k6, B-n63-k10, B-n68-k9 and Tai100d, where QTS achieves the best average solution quality. Its principal limitation is its higher computational time compared with Roulette, Softmax and UCB. Moreover, the convergence indicators show that QTS converges more slowly during the early stages of the search, suggesting that it favours long-term exploration rather than very rapid initial convergence. Overall, these results demonstrate that QTS constitutes a competitive alternative for adaptive operator selection by maintaining effective probabilistic exploration throughout the search process.

Overall, the comparison demonstrates that reinforcement learning-based operator selection strategies significantly improve the adaptive behaviour of ALNS by providing more informed operator selection mechanisms. UCB achieves the best overall absolute solution quality, whereas Softmax provides the best average ranking together with consistently competitive performance across the benchmark set. Although QTS does not systematically outperform all competing strategies in terms of average solution quality, it demonstrates the effectiveness of the proposed hybrid operator selection mechanism. By combining Q-learning with Gaussian Thompson Sampling-inspired action selection, QTS maintains probabilistic exploration around the learned Q-values, helping reduce the risk of premature convergence while producing the best solutions on several benchmark instances. These findings confirm that the investigated reinforcement learning policies exhibit complementary strengths rather than a universally dominant behaviour, and that the proposed QTS strategy represents a valuable and competitive alternative for adaptive operator selection within the ALNS framework.

Table 5. Comparative strengths and limitations of the investigated operator selection strategies.

Method	Strengths	Limitations
Roulette	Simple implementation, lowest computational time, competitive on several instances	Early stagnation, limited exploration, reduced search diversity
ϵ-greedy	Simple implementation, effective initial exploration	Premature convergence, highest variability, weakest overall performance
Softmax	Best average rank, stable performance, good exploration–exploitation balance	Temperature-sensitive; discrimination decreases when Q-values are close
UCB	Best overall absolute solution quality, lowest average standard deviation, strong convergence behaviour	May over-explore less promising operator pairs
QTS	Competitive solution quality, effective probabilistic exploration, robust performance across benchmark instances	Higher computational time, slower early convergence

5.4. Robustness Analysis Using Boxplots

Figure 1 presents the distribution of the final solution costs obtained over 30 independent runs for four representative benchmark instances, namely A-n45-k6, B-n68-k9, B-n78-k10 and Tai150d. Unlike the average values reported in Tables 2 and 3, the boxplots provide additional insight into the robustness, dispersion and reproducibility of each operator selection strategy.

For the small instance A-n45-k6, all strategies exhibit relatively compact distributions, indicating stable behaviour over repeated executions. Nevertheless, the proposed QTS strategy achieves one of the lowest median solution costs while maintaining a narrow interquartile range. In contrast, Roulette presents two high-cost outliers, revealing

occasional failures caused by premature operator selection. These observations suggest that the Gaussian Thompson Sampling-inspired exploration mechanism adopted by QTS helps maintain robust search behaviour while preserving sufficient exploration.

A clearer distinction appears on the medium-sized instance B-n68-k9. Among the compared strategies, QTS exhibits one of the most compact distributions together with one of the lowest median solution costs. Although UCB and Softmax also produce competitive solutions, they occasionally generate higher-cost outliers. In contrast, QTS maintains limited variability across the independent runs. This behaviour suggests that sampling around the learned Q-values enables the algorithm to continue exploring promising operator pairs without introducing excessive randomness. Consequently, the proposed strategy achieves highly reproducible performance while maintaining competitive solution quality.

For the larger benchmark instance B-n78-k10, the performance gap between the reinforcement learning-based strategies becomes smaller. Softmax, UCB and QTS all produce comparable distributions, whereas Roulette and ϵ-greedy exhibit larger interquartile ranges and greater variability. Although no strategy clearly dominates this instance, the proposed QTS strategy remains highly competitive, confirming that its probabilistic exploration mechanism continues to provide effective operator selection even as the search space becomes considerably larger.

The behaviour observed on the largest benchmark instance Tai150d further confirms the robustness of the investigated reinforcement learning strategies. As the problem complexity increases, all learning-based strategies outperform the conventional Roulette strategy in terms of median solution quality. Softmax achieves the lowest median solution cost, closely followed by UCB and QTS. Furthermore, QTS maintains a relatively compact distribution with only a limited number of outliers, indicating that the Gaussian Thompson Sampling-inspired action selection policy provides a good compromise between exploration and exploitation throughout the search process. Rather than relying exclusively on a small subset of destroy–repair operator pairs, QTS continues to explore alternative combinations while progressively reinforcing those that consistently produce high-quality solutions.

Overall, the boxplot analysis complements both the convergence study and the statistical validation. While UCB achieves the best overall solution quality according to the absolute performance indicators and Softmax provides the best average ranking together with consistently stable behaviour, the proposed QTS strategy demonstrates competitive solution quality combined with strong robustness across independent runs. By maintaining probabilistic exploration through Gaussian sampling around the learned Q-values, QTS helps reduce the risk of premature convergence while improving the reproducibility of the search. These observations confirm that the proposed QTS strategy constitutes a competitive and reliable alternative for adaptive operator selection within the ALNS framework.

5.5. Statistical Validation

To statistically validate the observed differences among the five operator selection strategies, non-parametric statistical tests were conducted using the results obtained from the benchmark instances. Since the performance distributions of stochastic metaheuristics cannot generally be assumed to satisfy the normality assumption, the Friedman test was first employed to compare the five strategies globally. The null hypothesis assumes that all investigated strategies exhibit equivalent performance.

The Friedman test, computed over the 16 benchmark instances, rejects the null hypothesis at the 5% significance level ($\boldsymbol{p = 0.00032}$), indicating that at least one operator selection strategy performs significantly differently from the others. Consequently, pairwise Wilcoxon signed-rank tests were conducted as post-hoc analyses. To control the family-wise error rate resulting from multiple pairwise comparisons, the Holm correction was applied. Table 7 provides a more detailed interpretation of the pairwise comparisons through the adjusted p-values and the corresponding rank-biserial effect sizes ($|\boldsymbol{r_{rb}}|$).

The Wilcoxon post-hoc analysis shows that the ϵ-greedy strategy differs significantly from Roulette, Softmax, UCB and QTS after Holm correction. This result confirms that ϵ-greedy exhibits the weakest overall performance among the investigated strategies, which is consistent with its highest average solution cost, largest average standard deviation and poorest average ranking reported in Tables 2 and 3.

In contrast, no statistically significant differences are observed between Roulette, Softmax, UCB and QTS after Holm correction. This finding indicates that these four strategies achieve statistically comparable solution quality

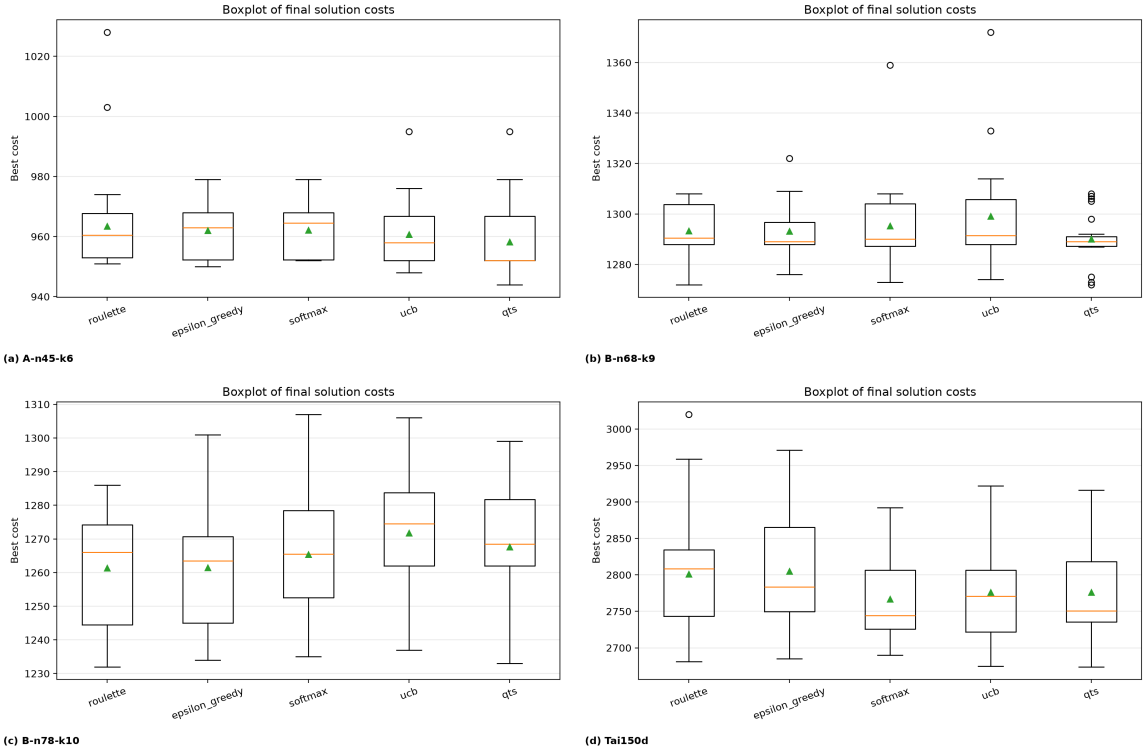

Figure 1. Distribution of final solution costs over 30 independent runs for representative small, medium, and large CVRPLIB instances: (a) A-n45-k6, (b) B-n68-k9, (c) B-n78-k10, and (d) Tai150d.

over the considered benchmark set, although they exhibit different practical characteristics. Roulette remains the fastest strategy in terms of computational time, Softmax achieves the lowest average optimality gap together with the best average ranking, UCB obtains the lowest average solution cost and the smallest average standard deviation, whereas the proposed QTS strategy provides competitive solution quality while maintaining probabilistic exploration through Gaussian Thompson Sampling-inspired action selection.

Overall, the Friedman and Wilcoxon analyses consistently indicate that ϵ-greedy performs significantly worse than the other strategies, whereas Roulette, Softmax, UCB and QTS exhibit statistically comparable performance despite their different practical characteristics. These results reinforce the conclusions drawn from the convergence analysis, the boxplots and the descriptive statistics.

Table 6. Friedman test results.

Test	Statistic	p-value	Decision
Friedman	20.9722	0.00032	Reject H_0

6. Conclusion

In this paper, we presented a comprehensive comparative study of five operator selection strategies within our previously proposed Q-learning-based ALNS framework for solving the Capacitated Vehicle Routing Problem

Table 7. Pairwise Wilcoxon signed-rank tests with Holm correction.

Comparison	p-value	Adjusted p-value	$\lvert r_{rb}\rvert$	Significant
Roulette vs ϵ-greedy	0.00022	0.00152	0.217	Yes
Roulette vs Softmax	0.74819	1.00000	0.019	No
Roulette vs UCB	0.85162	1.00000	0.011	No
Roulette vs QTS	0.27561	1.00000	0.064	No
ϵ-greedy vs Softmax	1.11×10^{-7}	1.11×10^{-6}	0.313	Yes
ϵ-greedy vs UCB	0.00007	0.00056	0.235	Yes
ϵ-greedy vs QTS	4.56×10^{-6}	0.00004	0.269	Yes
Softmax vs UCB	0.97016	1.00000	0.002	No
Softmax vs QTS	0.46242	1.00000	0.044	No
UCB vs QTS	0.62344	1.00000	0.029	No

(CVRP). In particular, we introduced the Q-Value Thompson Sampling (QTS) strategy and compared it with the conventional Roulette Wheel Selection baseline, ϵ-greedy, Softmax, and Upper Confidence Bound (UCB).

The experimental evaluation conducted on representative CVRPLIB benchmark instances demonstrates that the investigated strategies exhibit complementary strengths rather than a universally dominant behaviour. UCB achieves the best overall absolute solution quality, the lowest average standard deviation, and the strongest overall convergence behaviour. Softmax obtains the lowest average optimality gap together with the best average ranking, indicating consistently competitive performance across the benchmark set. Roulette remains the fastest strategy in terms of computational time while maintaining competitive performance on several benchmark instances. Although QTS does not systematically outperform all competing strategies, it consistently produces competitive solutions, achieves the best performance on several benchmark instances, and maintains effective probabilistic exploration through Gaussian Thompson Sampling-inspired action selection around the learned Q-values.

Overall, the results demonstrate that reinforcement learning significantly enhances adaptive operator selection within the ALNS framework. Rather than identifying a universally superior action selection policy, this study highlights the complementary characteristics of different reinforcement learning strategies and demonstrates that the proposed QTS strategy constitutes a competitive hybrid alternative for adaptive operator selection by combining the learning capability of Q-learning with probabilistic operator selection inspired by Gaussian Thompson Sampling.

Nevertheless, several limitations should be acknowledged. The performance of the proposed framework remains sensitive to the learning and exploration parameters associated with each operator selection strategy. Furthermore, the adopted state representation intentionally remains simple in order to preserve the tractability of tabular Q-learning, and richer state representations may further improve the learning process. Finally, the present study is limited to benchmark instances of the CVRP, and additional investigations are required to evaluate the generalization capability of the proposed framework on other classes of combinatorial optimization problems.

Future work will focus on extending this framework to other optimization problems, investigating richer state representations based on solution features, exploring deep reinforcement learning techniques, analysing the convergence behaviour of the learning process through the evolution of Q-values and policy stabilization, and developing adaptive parameter tuning mechanisms to further improve the robustness, scalability, and generalization capability of reinforcement learning-based ALNS.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability

No data availability statement is applicable.

REFERENCES

1. G. B. Dantzig and J. H. Ramser, *The truck dispatching problem*, Management Science, vol. 6, no. 1, pp. 80–91, 1959.
2. A. Gomez, A. Imran, and S. Salhi, *Solution of classical transport problems with bee algorithms*, International Journal of Logistics Systems and Management, vol. 15, no. 2–3, pp. 160–170, 2013.
3. F. Li, B. Golden, and E. Wasil, *Very large-scale vehicle routing: new test problems, algorithms, and results*, Computers & Operations Research, vol. 32, no. 5, pp. 1165–1179, 2005.
4. G. Barbarosoglu and D. Ozgur, *A tabu search algorithm for the vehicle routing problem*, Computers & Operations Research, vol. 26, no. 3, pp. 255–270, 1999.
5. B. Yu, Z.-Z. Yang, and B. Yao, *An improved ant colony optimization for vehicle routing problem*, European Journal of Operational Research, vol. 196, no. 1, pp. 171–176, 2009.
6. D. Mester and O. Bräysy, *Active guided evolution strategies for large-scale vehicle routing problems with time windows*, Computers & Operations Research, vol. 32, no. 6, pp. 1593–1614, 2005.
7. K. Fleszar, I. H. Osman, and K. S. Hindi, *A variable neighbourhood search algorithm for the open vehicle routing problem*, European Journal of Operational Research, vol. 195, no. 3, pp. 803–809, 2009.
8. O. Polat, C. B. Kalayci, O. Kulak, and H.-O. Günther, *A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit*, European Journal of Operational Research, vol. 242, no. 2, pp. 369–382, 2015.
9. I. Pereira and A. Madureira, *Self-optimization module for scheduling using case-based reasoning*, Applied Soft Computing, vol. 13, no. 3, pp. 1419–1432, 2013.
10. I. Pereira and A. Madureira, *Self-optimizing a multi-agent scheduling system: A racing based approach*, in Intelligent Distributed Computing IX, Springer, pp. 275–284, 2015.
11. S. Ropke and D. Pisinger, *An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows*, Transportation Science, vol. 40, no. 4, pp. 455–472, 2006.
12. S. Vonolfen and M. Affenzeller, *Distribution of waiting time for dynamic pickup and delivery problems*, Annals of Operations Research, vol. 236, no. 2, pp. 359–382, 2016.
13. M. Karimi-Mamaghan, M. Mohammadi, P. Meyer, A. M. Karimi-Mamaghan, and E.-G. Talbi, *Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art*, European Journal of Operational Research, vol. 296, no. 2, pp. 393–422, 2022.
14. H. Zhang and T. Yu, *Taxonomy of reinforcement learning algorithms*, in Deep Reinforcement Learning: Fundamentals, Research and Applications, Springer, pp. 125–133, 2020.
15. C. J. Watkins and P. Dayan, *Q-learning*, Machine Learning, vol. 8, no. 3, pp. 279–292, 1992.
16. H. Boualamia, A. Metrane, I. Hafidi, and O. Mellouli, *A new adaptation mechanism of the ALNS algorithm using reinforcement learning*, Operations Research Forum, vol. 6, no. 3, pp. 1–26, 2025.
17. D. Pisinger and S. Ropke, *A general heuristic for vehicle routing problems*, Computers & Operations Research, vol. 34, no. 8, pp. 2403–2435, 2007.
18. S. T. W. Mara, R. Norcahyo, P. Jodiawan, L. Lusiantoro, and A. P. Rifai, *A survey of adaptive large neighborhood search algorithms and applications*, Computers & Operations Research, art. 105903, 2022.
19. E. Demir, T. Bektaş, and G. Laporte, *An adaptive large neighborhood search heuristic for the pollution-routing problem*, European Journal of Operational Research, vol. 223, no. 2, pp. 346–359, 2012.
20. L. Kotthoff, *Algorithm selection for combinatorial search problems: A survey*, in Data Mining and Constraint Programming, Springer, pp. 149–190, 2016.
21. M. Mısır and E. Özcan, *Automated design of heuristic algorithms for combinatorial optimisation*, Journal of the Operational Research Society, vol. 64, no. 4, pp. 531–550, 2013.
22. J. Kanda, T. Vidal, and A. Subramanian, *Machine learning in heuristic search: Learning to guide local search*, European Journal of Operational Research, vol. 295, pp. 195–209, 2021.
23. M. Nazari, A. Oroojlooy, L. Snyder, and M. Takáč, *Reinforcement learning for solving the vehicle routing problem*, Advances in Neural Information Processing Systems, vol. 31, 2018.
24. Y. Bengio, A. Lodi, and A. Prouvost, *Machine learning for combinatorial optimization: a methodological tour d'horizon*, European Journal of Operational Research, vol. 290, no. 2, pp. 405–421, 2021.
25. A. Hottung and K. Tierney, *Neural large neighborhood search for the capacitated vehicle routing problem*, arXiv:1911.09539, 2019.
26. Q. Lu, Y. Zhang, and J. Yang, *Adaptive large neighborhood search with reinforcement learning for the vehicle routing problem*, Computers & Operations Research, vol. 110, pp. 1–12, 2019.
27. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.
28. P. Auer, N. Cesa-Bianchi, and P. Fischer, *Finite-time analysis of the multiarmed bandit problem*, Machine Learning, vol. 47, no. 2, pp. 235–256, 2002.
29. J. Vermorel and M. Mohri, *Multi-armed bandit algorithms and empirical evaluation*, in European Conference on Machine Learning, Springer, pp. 437–448, 2005.
30. S. T. Windras Mara, R. Norcahyo, P. Jodiawan, L. Lusiantoro, and A. P. Rifai, "A survey of adaptive large neighborhood search algorithms and applications," *Computers & Operations Research*, vol. 146, p. 105903, 2022. doi: 10.1016/j.cor.2022.105903.
31. K. A. Smith-Miles, "Cross-Disciplinary Perspectives on Meta-Learning for Algorithm Selection," *ACM Computing Surveys*, vol. 41, no. 1, Article 6, pp. 1–25, 2009. doi: 10.1145/1456650.1456656.
32. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.