

Article

An Intelligent Machine Learning-Driven Solving Framework for Capacitated Vehicle Routing

Hajar Bideq ^{1,*} , Khaoula Ouaddi ², Rachid Ellaia ¹ and Agnès Gorge ³¹ Mohammadia School of Engineers, Mohammed V University, Rabat 10090, Morocco; rachid.ellaia@gmail.com² National Higher School for Computer Science and Systems Analysis, Mohammed V University, Rabat 10000, Morocco; khaoula.ouaddi@gmail.com³ Africa Business School, Mohammed VI Polytechnic University, Rabat 11103, Morocco; agnes.gorge@gmail.com

* Correspondence: hajar.bideq@gmail.com

Abstract

Despite recent advancements in solving the Capacitated Vehicle Routing Problem (CVRP), state-of-the-art learning-based methods remain hindered by costly offline training, while classical population solvers rely on implicit mechanisms and rigid parameter tuning. To bridge this methodological gap, this paper introduces the Group Learning Algorithm hybridized with a Multi-Armed Bandit (GLA-MAB), a training-free metaheuristic that transforms evolutionary search into an explicit, controllable learning process. The framework partitions the population into leader-learner groups to extract and inject proven topological structures directly into weaker solutions. Simultaneously, a hierarchical MAB layer oversees the search online, utilizing real-time reward feedback to dynamically manage operator selection and stagnation recovery. Furthermore, a spatial decomposition wrapper ensures strict scalability, extending the framework's applicability to massive topologies of up to 1200 nodes. Comprehensive evaluations across 117 established CVRP benchmark instances validate the architecture's efficacy. GLA-MAB achieves near-optimal convergence on classical instances, maintaining mean gaps below 0.03% on Sets A and E, and delivers highly competitive performance on large-scale heterogeneous sets such as Set Li. Ultimately, GLA-MAB provides the dynamic adaptability of modern artificial intelligence while completely eliminating the prohibitive overhead of offline dataset generation.

Keywords: Group Learning Algorithm (GLA); Multi-Armed Bandit (MAB); Capacitated Vehicle Routing Problem (CVRP)



Academic Editors: Yan Sun and
Maixiang Lang

Received: 23 May 2026

Revised: 28 June 2026

Accepted: 30 June 2026

Published: 6 July 2026

Copyright: © 2026 by the authors.

Published by MDPI on behalf of the International Institute of Knowledge Innovation and Invention. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Modern information systems are increasingly tasked with solving NP-hard combinatorial queries to automate complex operations. The Capacitated Vehicle Routing Problem (CVRP) remains a cornerstone benchmark for evaluating the computational efficacy of novel algorithmic decision support systems. The objective within these systems is to design a highly optimized set of resource allocation sequences (routes) to serve a network of distributed data nodes without exceeding predefined capacity constraints. Beyond its direct industrial relevance in automated fleet management, optimizing the CVRP is crucial for maximizing the efficiency of intelligent routing software and novel combinatorial search algorithms [1]. Over the past decade, CVRP solvers have evolved from conventional heuristics into highly sophisticated hybrid metaheuristics [2,3]. While these modern evolutionary and local-search frameworks achieve minimal optimality gaps, their performance relies

heavily on rigid, manual parameter tuning and implicit learning mechanisms. Consequently, their effectiveness frequently degrades across heterogeneous instances or as the search progresses into different phases, revealing a critical vulnerability in dynamic search control. Furthermore, standard metaheuristics face severe scalability bottlenecks when applied to massive, real-world networks [4]. This vulnerability is particularly evident in classical trajectory-based metaheuristics, such as Tabu Search and Simulated Annealing. While these approaches have long served as foundational solutions for the CVRP, they fundamentally rely on localized, point-to-point neighborhood exploration that makes them susceptible to becoming trapped in severe local optima when scaled to massive or heterogeneous instances.

Motivated by these limitations, the recent literature has increasingly integrated artificial intelligence, specifically deep learning (DL) and reinforcement learning (RL), into routing optimization [5,6]. However, these learning-based approaches introduce a prohibitive trade-off: they require massive offline training phases, rely on domain-specific dataset generation, and frequently suffer from severe performance degradation when exposed to out-of-distribution instance topologies. This presents a critical gap in the current literature: the need for a routing solver that possesses the adaptive, data-driven control of modern machine learning, but retains the training-free flexibility and transparency of classical metaheuristics.

The main contributions of this paper are summarized as follows:

- A novel training-free architecture: We propose the GLA-MAB framework, which integrates the adaptive resource allocation of a Multi-Armed Bandit (MAB) with explicit Group Learning (GLA) dynamics. This achieves dynamic search control without the prohibitive cost of offline training or domain-specific dataset generation.
- Explicit structural knowledge transfer: Unlike classical evolutionary algorithms that rely on blind mutation, we introduce a leader-learner mechanism that extracts and injects proven topological structures (e.g., stable edges, regret-based node placements) directly into weaker solutions, significantly accelerating convergence.
- Dynamic operator control: We formulate operator selection and stagnation recovery as an online MAB problem utilizing the Upper Confidence Bound (UCB) policy. This eliminates the need for manual, fixed-phase schedules and allows the solver to adapt to the specific heterogeneity of the current instance.
- Scalability for massive instances: Through an optional spatial decomposition wrapper, we extend the framework's applicability to large-scale instances (up to 1200 nodes), demonstrating highly competitive gaps on 117 established CVRP benchmarks (including the challenging Golden, Li, and X sets) within practical computational budgets.

The remainder of this paper is organized as follows. Section 2 reviews related work in learning-based routing and search control. Section 3 details the architectural components of the proposed GLA-MAB framework. Section 4 presents the computational experiments, including comprehensive benchmark results and ablation studies. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Related Work

2.1. Deep Learning and Neural Routing Solvers

Recent advancements in artificial intelligence have driven the development of end-to-end neural architectures for the CVRP [5,6]. Construction-focused learning methodologies attempt to replace traditional heuristics by training models to incrementally build routes. For instance, attention-based models [7] and Graph Neural Networks (GNNs) [8,9] have been widely deployed to learn node-selection policies. While frameworks like POMO [2] improve stability through parallel multi-start rollouts, constructive neural models funda-

mentally suffer from poor cross-distribution generalization and demand prohibitive offline training budgets [10,11].

Alternatively, improvement-focused learning models guide classical local search operations by predicting high-yield routing edges or selecting specific algorithmic operators [12,13]. For example, deep Q-networks have been utilized to automate strategy selection in routing environments [14]. Although these hybrid methods accelerate convergence compared to unguided metaheuristics, they remain heavily encumbered by the necessity of domain-specific dataset generation and offline training [15].

2.2. Adaptive Search Control and Bandit Supervision

To circumvent the computational burden of offline training, online learning mechanisms, specifically Multi-Armed Bandits (MAB), have been increasingly adopted for adaptive search control [16]. MAB frameworks treat local search heuristics or neighborhood structures as “arms,” continuously updating their empirical utility based on real-time solution improvements [17]. Recent applications include HHMAB-SP [18], which utilizes a bandit-driven hyper-heuristic to select low-level operators, and HG-UCB [19], which integrates UCB-based rules within a Variable Neighborhood Search (VNS) to manage non-stationary operator rewards. While these MAB-based solvers provide transparent, training-free adaptability [20], their application in routing is frequently restricted to standard, unguided neighborhoods [21]. They dictate which operator to apply, but lack the mechanisms to extract and utilize deep topological structures from the evolving solution population.

2.3. Population-Based and Memetic Solvers

Classical population-based metaheuristics inherently perform search space learning, though typically in an implicit manner [22]. State-of-the-art memetic solvers, such as the Hybrid Genetic Search (HGS) framework [1], achieve exceptional solution quality through highly engineered crossover operators (e.g., Edge Assembly Crossover) and intensive local refinement [23,24]. However, the knowledge transfer within these frameworks relies heavily on blind recombination and static parameter schedules [25]. Because selection pressure, mutation rates, and local search intensities operate on fixed, empirically tuned schedules, performance can vary significantly as instance characteristics change [26].

The core structural limitation underlying this implicit approach is its heavy reliance on stochastic recombination (crossover) schemes. In a standard crossover, parent solutions are sliced and recombined semi-randomly, relying purely on survival-of-the-fittest selection pressure over hundreds of generations to naturally aggregate high-quality traits. Consequently, this blind mixing frequently disrupts highly optimized local edge configurations, making it difficult to transform implicit genetic learning into an explicit, controllable process [3].

To overcome this limitation, the proposed leader-learner mechanism in GLA-MAB fundamentally departs from this paradigm by replacing implicit, blind variation with explicit, targeted structural knowledge transfer. By directly isolating and transferring specific, high-yield topological choices from the group leader, the algorithm preserves structural integrity and dramatically accelerates convergence boundaries.

2.4. Positioning of the Proposed GLA-MAB Framework

Table 1 contextualizes the proposed GLA-MAB framework against these prevailing algorithmic paradigms. To address the scalability demands highlighted in recent 2024 and 2025 literature [27,28], solvers must manage massive graph topologies without defaulting to computationally exhaustive offline training [29]. GLA-MAB bridges these conflicting requirements. Unlike DRL architectures, it is entirely training-free [30].

Unlike classical memetic algorithms, it abandons blind recombination in favor of explicit leader-to-learner structural injection [31]. Finally, by hierarchically integrating an online MAB controller with spatial clustering, GLA-MAB achieves the dynamic adaptivity of modern AI without the associated overhead, scaling effectively to 1200-node instances.

Table 1. Methodological comparison of GLA-MAB against recent state-of-the-art CVRP solvers.

Reference/Algorithm	Methodology Base	Requires Offline Training?	Online Adaptive Control?	Explicit Structural Knowledge Transfer?	Max Evaluated Instance (Nodes & Dataset)
HGS [1]	Hybrid Genetic Search	No	No (Fixed/Tuned Phases)	No (Random/Blind Crossover)	1001 (Set X)
HHMAB-SP [18]	Hyper-heuristic + MAB	No	Yes (Bandit Operator Selection)	No (Standard generic moves)	101 (Sets A, B, E, P)
HG-UCB [19]	VNS + MAB	No	Yes (UCB Neighborhood Selection)	No (Standard generic moves)	1001 (Set X)
SPORL [27]	Deep Reinforcement Learning	Yes (Heavy network training)	No (Pre-trained policy execution)	No (Implicit neural weights)	1001 (Set X)
Proposed GLA-MAB	Group Learning + MAB	No (Training-Free)	Yes (UCB Bandit Control)	Yes (Leader-to-Learner Injection)	1200 (Set Li) & 1001 (Set X)

3. Proposed Framework

3.1. System Architecture and Problem Formulation

The proposed GLA-MAB framework operates on a bipartite architecture, as illustrated in Figure 1. The primary execution layer employs a Group Learning Algorithm (GLA) to systematically organize candidate solutions and facilitate explicit structural knowledge transfer. Supervising this process is a hierarchical Multi-Armed Bandit (MAB) control layer, which dynamically allocates search effort and regulates the exploration-exploitation balance without relying on static, pre-configured phase schedules. The framework is applied to the Capacitated Vehicle Routing Problem (CVRP), defined on a complete graph, $G = (V, E)$, where $V = \{0, 1, \dots, n\}$ encompasses the central depot (node 0) and n spatially distributed network nodes. Each node $i \in V \setminus \{0\}$ presents a known demand $d_i > 0$, and each edge $(i, j) \in E$ incurs a deterministic travel cost c_{ij} (typically 2D Euclidean distance). Given a homogeneous fleet of K vehicles with uniform capacity Q , the objective is to determine a set of at most K feasible routes, each starting and ending at the depot, such that:

1. Every customer is visited exactly once;
2. The total demand on each route does not exceed Q ;
3. The total system cost is minimized.

3.2. Representation and Deterministic Decoding

To maintain a continuous and unified search space, candidate solutions are represented as giant tours (unpartitioned permutations of the n demand nodes). To map these permutations to the constrained CVRP space, the framework evaluates each giant tour against a portfolio of deterministic decoding policies, denoted as Φ :

- Sequential split: Scans the sequence iteratively, initializing a new route precisely when the addition of the subsequent customer violates the capacity constraint Q .
- Optimized split: Utilizes a Bellman-based dynamic programming algorithm to extract the optimal feasible partition of the fixed sequence.
- Greedy insertion: Employs a rapid split phase followed by deterministic, cost-minimizing local reinsertions to rectify boundary inefficiencies.

For any given giant tour x , the fitness function $f(x)$ is defined as the minimum system cost achieved across all available decoding policies:

$$f(x) = \min_{\phi \in \Phi} \text{Cost}(\phi(x)) \quad (1)$$

In this formulation, the system cost represents the total cumulative routing distance traveled across all active vehicles in the decoded solution. To evaluate a giant tour x , a decoding policy ϕ first partitions the continuous sequence into a set of distinct, feasible vehicle routes that satisfy the strict capacity constraints of the CVRP. The distance between consecutive nodes within these decoded routes is then computed using the standard Euclidean metric derived from the instance coordinate matrix, and summed across all active vehicle trips. While the objective function explicitly minimizes total mileage to maintain strict alignment with classical CVRP benchmark standards, it inherently optimizes fleet utilization. Because the problem enforces rigid vehicle capacity limits, minimizing total distance naturally forces the decoding policies to consolidate customer demands into highly compact, optimal clusters, implicitly minimizing the total number of vehicles required.

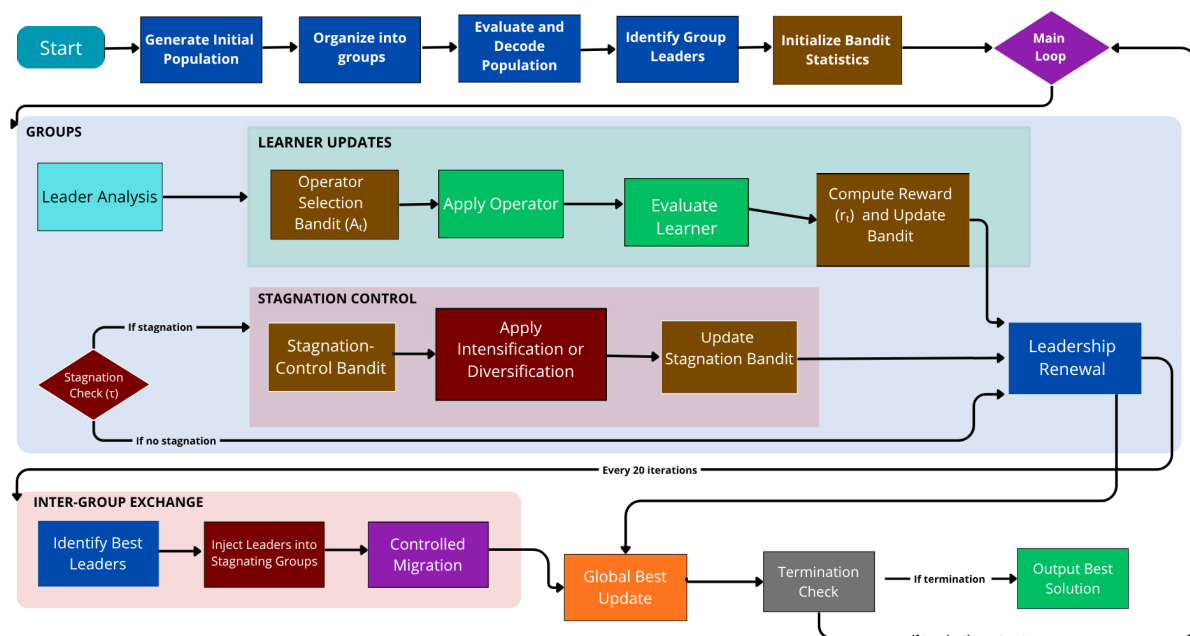


Figure 1. System architecture and execution flow of the proposed GLA-MAB framework.

This multi-policy evaluation ensures robust fitness signaling, preventing the learning mechanism from over-fitting to the biases of a single decoding heuristic.

3.3. Explicit Group Learning Dynamics

To shift from implicit evolutionary variation to explicit knowledge transfer, the global population P of N candidate solutions is deterministically partitioned into m disjoint operational groups, $P = \{G_1, G_2, \dots, G_m\}$. Each group G_k consists of a single leader L_k and a set of learners $S_k = \{s_{k,1}, s_{k,2}, \dots, s_{k,l}\}$. At any iteration t , the leader $L_k(t)$ is defined as the elite solution within the subset:

$$L_k(t) = \arg \min_{x \in G_k^{(t)}} f(x) \quad (2)$$

Rather than executing blind recombination, the framework analyzes the group's elite leader solution $L_k(t)$ to extract high-yield structural configurations, such as stable routing

edges and high-regret customer placements. These topological signals are utilized to dictate the transformation of the learner solutions $s_{k,i}$ within the group. The available transformation operators, which constitute the set of available “arms” (A) dynamically managed and selected by the MAB mechanism during the learner update phase, include:

- **Regret relocation:** Identifies customer nodes that are highly sensitive to misplacement and forces them to mirror their positioning in the leader. For each node in the learner tour, the algorithm calculates its insertion regret, defined as the cost penalty incurred if the node is placed in its second-best position instead of its best position. Crucially, these positions are evaluated by temporarily splitting the resulting giant tour configurations using the decoding policy to determine the true change in CVRP route distance. Nodes with the highest regret values are highly critical; therefore, the algorithm extracts them from their current positions in the learner and re-inserts them to match their exact spatial neighborhood in the elite leader solution $L_k(t)$.

Example: If node 5 exhibits high regret in the learner tour, and the leader’s optimized sequence contains the segment $[4 \rightarrow 5 \rightarrow 6]$, node 5 is extracted from its current position in the learner and forcefully inserted adjacent to node 4 or 6. Because it plucks one node out of its current position and drops it into a single new position, it never creates duplicates or leaves missing nodes. It is a pure relocation, so feasibility is maintained automatically without needing a repair phase.

- **Fragment injection:** Extracts a high-quality sequence of consecutive nodes from the elite leader $L_k(t)$ and uses it to replace a poor-performing section in the learner tour. To concretely identify this sequence, the algorithm slides a window of a fixed node length L across the leader’s giant tour. For each candidate sequence, it calculates the localized routing cost by summing the exact edge distances between its consecutive nodes using the distance matrix. The sequence that yields the absolute lowest cumulative distance is selected as the high-quality fragment, ensuring the algorithm extracts the most geographically compact cluster of stops. Since solutions use a giant-tour representation (a single continuous loop bounded by the depot 0), the algorithm maintains strict route feasibility through a “delete-before-insert” mechanism. It first deletes the fragment’s nodes from their original positions in the learner tour, and then injects the intact leader fragment. This sequence naturally prevents duplicate or missing customers.

To clearly illustrate how this structural knowledge is explicitly extracted and transferred, consider a simple 5-customer network (nodes 1 to 5) with a central depot (0):

- Leader Solution $L_k(t)$: $[0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 0]$ (Total decoded cost = 120).
- Learner Solution $s_{k,i}$: $[0 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 1 \rightarrow 3 \rightarrow 0]$ (Total decoded cost = 195).

The operator executes in three synchronized phases:

- **Extraction Phase:** The algorithm analyzes the elite leader and identifies a high-efficiency contiguous edge fragment that minimizes localized travel costs, for example, the subsequence $[2 \rightarrow 3 \rightarrow 4]$.
- **Injection Phase:** The algorithm first eliminates these specific nodes from the learner tour, yielding a reduced valid sequence: $[0 \rightarrow 5 \rightarrow 1 \rightarrow 0]$. The intact fragment $[2 \rightarrow 3 \rightarrow 4]$ is then injected into the most cost-effective position within this reduced tour. The overall quality of candidate injection positions is determined by executing the splitting policy on the resulting giant tour to measure the true post-split route distance.

By removing the nodes prior to injection, missing or duplicate customer conflicts are structurally bypassed. This targeted operational sequence results in a newly repaired learner tour: $[0 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0]$ (Total decoded cost = 135),

successfully capturing the leader's superior edge structures while preserving strict CVRP constraint feasibility.

- Targeted reconfiguration: Concentrates local search effort (e.g., 2-opt swaps) specifically on edges involved in route crossings, guided by the leader's established topology. This mechanism restricts local refinement moves exclusively to edges within the learner that topologically deviate from the leader's established edge-frequency map, bypassing already optimized segments.

Example: If the learner contains a highly sub-optimal crossing edge ($1 \rightarrow 4$) that does not exist in the elite leader's topological structure, the algorithm concentrates local search efforts specifically on breaking and reconnecting this specific edge to untangle the route. The evaluation of these local swaps relies on calculating the post-split tour cost via the decoding policy, rather than wasting computational budget evaluating the entire giant tour blindly. Since this local refinement merely alters the sequencing of existing edges rather than adding or removing nodes, CVRP constraint feasibility is structurally guaranteed.

To prevent isolated sub-population convergence and share structural knowledge globally, the framework enforces a periodic inter-group migration protocol every μ iteration, transferring high-performing leaders between adjacent groups. Independent of this periodic exchange, if a group G_k fails to register a fitness improvement for τ consecutive iterations, this prolonged stagnation triggers an internal, bandit-driven intensification or diversification response to forcefully disrupt the trapped search trajectory.

3.4. Bandit-Driven Search Control

While group learning dictates how solutions are updated, the effectiveness of specific operators varies dynamically during execution. Rather than relying on manual, fixed phase schedules, the selection of operators applied to learners is formulated as a Multi-Armed Bandit (MAB) problem. Let $A = \{a_1, a_2, \dots, a_n\}$ be the set of available search operators (arms). When an operator $a \in A$ modifies a learner from state x_{t-1} to x_t , the normalized reward r_t is computed to ensure scale-invariance:

$$r_t = \max\left(0, \frac{f(x_{t-1}) - f(x_t)}{f(x_{t-1})}\right) \quad (3)$$

To balance the exploitation of historically successful operators with the exploration of newly viable ones, we utilize the Upper Confidence Bound (UCB) policy. The operator A_t selected at decision epoch t is formulated as:

$$A_t = \operatorname{argmax}_{a \in A} \left(Q_t(a) + c \sqrt{\frac{(\ln t)}{N_t(a)}} \right) \quad (4)$$

where $Q_t(a)$ is the empirical mean reward of operator a , $N_t(a)$ is the number of times a has been selected, and $c > 0$ is the exploration parameter. To ensure mathematical stability and prevent undefined operations (division by zero) within the exploration term, the MAB controller undergoes a mandatory initialization phase. During the first $|A|$ iterations, every operator $a \in A$ is selected and executed exactly once. Thus, when the algorithm transitions to selecting operators via Equation (4), $N_t(a) \geq 1$ for all operators. This initialization guarantees that the square root is well-defined and establishes an initial baseline reward $Q_t(a)$ for every operator, actively preventing premature operator starvation.

This bandit control is applied at multiple decision points throughout the framework:

1. Operator selection: Continuously learning which guided or general moves yield the best cost reductions for the current instance topology;

2. Stagnation response: Treating the choice between intensification (stronger local search) and diversification (large perturbations) as a bandit decision when progress slows;
3. Decoding efficiency (Optional): For large instances, the bandit learns which decoding policy Φ is most reliable, avoiding the overhead of evaluating all policies for every solution.

By learning online during each run, the bandit layer eliminates the need for heavy offline training while keeping the search logic highly transparent. The complete step-by-step execution of the GLA-MAB framework, integrating both the group learning dynamics and the bandit supervision, is summarized in Algorithm 1.

Computational complexity of the bandit layer: A critical advantage of the GLA-MAB framework is that the bandit supervision introduces negligible computational overhead. Updating the empirical mean reward $Q_t(a)$ and calculating the UCB score for $|A|$ arms requires $O(|A|)$ operations. Because the number of search operators is a small constant ($|A| \ll n$), the bandit decision mechanism operates in $O(1)$ time relative to the instance size. The overall time complexity per iteration remains strictly bounded by the underlying route evaluation and local search operators, ensuring that the adaptive control scales effectively to massive CVRP instances.

Algorithm 1. Pseudo-code of the GLA-MAB, detailing leader-guided knowledge transfer and UCB-based operator selection

Input: Instance graph $G = (V, E)$, Fleet K , Capacity Q , Parameters (N, m, c, τ)

Output: Best global solution x_{best}

- 1: Initialize population P with N giant tours
 - 2: Partition P into m groups $\{G_1, \dots, G_m\}$
 - 3: Evaluate $f(x)$ for all x in P using decoding policies Φ
 - 4: Perform round-robin initialization: execute each operator a in A exactly once to set $N(a) = 1$ and establish initial $Q(a)$ rewards
 - 5: while termination criteria not met do
 - 6: for each group G_k in $\{G_1, \dots, G_m\}$ do
 - 7: Identify leader $L_k(t) = \arg \min_{x \in G_k^{(t)}} f(x)$
 - 8: Extract structural signals (regret scores, fragments) from L_k
 - 9: for each learner $s_{k,i}$ in G_k do
 - 10: Select operator a^* using UCB policy
 - 11: Apply a^* to $s_{k,i}$ guided by L_k 's signals to generate $s'_{k,i}$
 - 12: Evaluate $f(s'_{k,i})$
 - 13: Calculate normalized reward r_t
 - 14: Update $Q(a^*)$ and $N(a^*)$
 - 15: if $f(s'_{k,i}) < f(s_{k,i})$ then $s_{k,i} \leftarrow s'_{k,i}$
 - 16: end for
 - 17: Update leader L_k if a better learner is found
 - 18: if G_k stagnates for τ iterations then
 - 19: Trigger MAB stagnation response (Intensify/Diversify)
 - 20: end if
 - 21: end for
 - 22: if iteration $\bmod \mu = 0$ then perform inter-group leader migration
 - 23: Update global x_{best}
 - 24: end while
 - 25: return x_{best}
-

3.5. Scalability Mechanism

To maintain practical execution times on large instances ($n \geq 200$), the framework integrates a spatial decomposition wrapper. The global graph is partitioned into balanced spatial subsets of 50 to 100 nodes. Each subset is optimized independently via the GLA-MAB core. The localized tours are subsequently merged and subjected to a global refinement phase to eliminate boundary inefficiencies. This maintains the structural integrity of the CVRP constraints while isolating the $O(n^2)$ neighborhood evaluation costs.

4. Results

4.1. Experimental Setup and Parameter Settings

The proposed GLA-MAB framework was implemented and evaluated across 117 established CVRP benchmark instances, encompassing both classical configurations (Sets A, B, E, F, M) and massive, heterogeneous topologies (Golden, Li, X). All computational experiments were executed on standard commodity hardware (Intel Core i7-12th Gen, 32GB RAM). To ensure algorithmic transparency and reproducibility, the framework eschews instance-specific parameter tuning. The core hyperparameters remain constant across all evaluated sets, as detailed in Table 2. To validate the robustness of these global configurations, a sensitivity analysis was conducted on the core architectural drivers: the UCB exploration constant c and the group count m . The framework was evaluated across its operational extremes, testing highly exploitative versus explorative bandit states $c \in [0.3, 0.7]$ and dense versus fragmented population topologies $m \in [3, 7]$. Across a representative subset of classical instances, this variance yielded a negligible deviation in the mean gap (under 0.05%), confirming that the framework achieves stable convergence without requiring instance-specific calibration.

Table 2. Parameter settings for the GLA-MAB framework.

Parameter	Symbol	Value	Description
Population Size	N	100	Total number of candidate solutions (giant tours)
Number of Groups	m	5	Number of partitions for leader-learner dynamics
UCB Constant	c	0.5	Balances exploration vs. exploitation in MAB
Stagnation Limit	τ	30	Iterations without improvement before diversification
Migration Rate	μ	20	Iterations between inter-group leader exchanges

To guarantee experimental transparency, the evaluation protocol is strictly defined. The comprehensive benchmark results reported in this study correspond to a single execution per instance. This deterministic baseline approach was utilized to accurately demonstrate the framework's core constraint-management capability and trajectory under strict computational limits, eliminating multi-run aggregation biases. The termination criteria for each execution are defined as a maximum computational CPU time limit scaled proportionally to the instance size, with an early-stopping trigger initialized if the global best solution exhibits no improvement for a predefined stagnation threshold.

For large instances ($n \geq 200$), the computational bottleneck shifts to neighborhood evaluation memory pressure. Rather than artificially extending execution budgets, the framework activates the optional spatial decomposition wrapper. Network nodes are clustered into balanced subsets (50 to 100 nodes), optimized in parallel via the standard GLA-MAB logic, and globally refined to resolve boundary inefficiencies. Crucially, this wrapper does not relax the underlying mathematical model since strict feasibility is enforced on the global graph, and all reported costs reflect the unpartitioned, full-scale instance.

In practice, this mechanism enables handling massive cases like Li_32 (1200 nodes) and X-n1001-k43 (1001 nodes) within a reasonable execution time on commodity hardware.

4.2. Overall Performance of GLA-MAB and Scalability Analysis

To comprehensively evaluate the framework, we separate our reporting into exact quantitative metrics and visual variance analysis. Table 3 provides the precise, verifiable numerical outcomes (match rates, mean gaps, and maximum deviations) required for standard CVRP benchmarking. Complementing this, the three-panel dashboard in Figure 2 visualizes the search behavior, illustrating variance, outlier distribution, and scalability trends across varying instance dimensions, which cannot be fully captured by aggregated tabular data. An analysis of Table 3 reveals highly consistent performance across classical benchmark families. In Sets A, B, E, and F, GLA-MAB operates in a near-optimal regime, achieving 100% match rates on Sets B and F, and maintaining residual mean gaps strictly below 0.03% on Sets A and E. This stability is visually confirmed in Figure 2a, where the boxplots for these classical sets exhibit tight distributions centered precisely at zero.

Table 3. Summary of match rates, mean gaps and maximum deviations achieved by GLA-MAB across 117 tested benchmark instances.

Family	Number of Tested Instances	Number of Nodes	Optimal/Best Known Solutions Matched (%)	Mean Gap Across Instances (%)	Max Gap Across Instances (%)	Gap Within 2% (%)
setA	27	32–80	77.8	0.025	0.155	100
setB	23	31–78	100.0	0.000	0.000	100
setE	13	13–101	92.3	0.009	0.120	100
setF	3	45–135	100.0	0.000	0.000	100
setM	5	101–200	60.0	0.114	0.493	100
Golden	20	200–483	20.0	1.083	4.443	90
setLi	12	560–1200	100.0	0.000	0.000	100
setX	14	101–1001	21.4	0.624	3.675	92.9

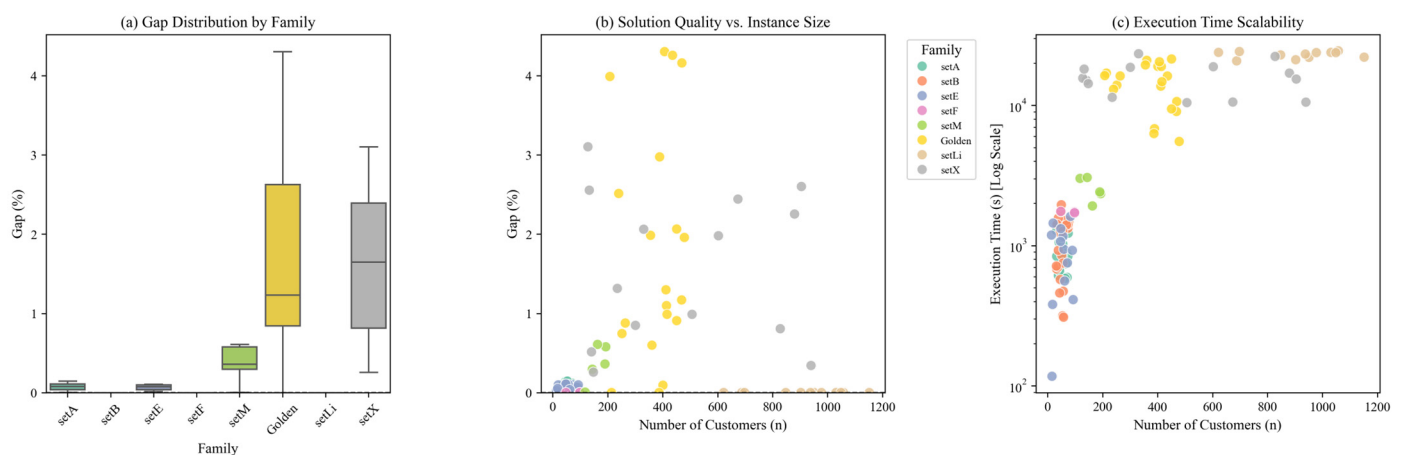


Figure 2. Three-panel scalability and variance dashboard for the GLA-MAB framework. Panel (a) displays the boxplot distribution of optimality gaps across benchmark families. Panel (b) illustrates solution quality as a function of instance size. Panel (c) maps the algorithmic execution time on a logarithmic scale.

These results validate the efficacy of the leader-learner mechanism: the population rapidly consolidates around strong topological structures, facilitating accelerated convergence under fixed budgets. The most demanding behavior emerges within the Golden and Set X families, which is anticipated given their extreme scale and topological heterogeneity. While the framework successfully maintains 85% and 92.9% of these instances within a

2% gap, respectively, the overall variance widens. As demonstrated in Figure 2b, the larger fractional deviations are naturally concentrated within the largest instance sizes ($n > 400$). Under the strict runtime limit, GLA-MAB successfully identifies the optimal global structure but occasionally exhausts its budget prior to completing the massive cross-route restructuring sequences required to close the final fractional gap. Conversely, the method exhibits exceptional performance on Set Li, always matching Best Known Solutions (BKS), yielding a 0% mean gap. Finally, Figure 2c maps the algorithmic execution time against instance size on a logarithmic scale. The trajectory demonstrates that the spatial decomposition wrapper effectively bounds the time complexity, maintaining predictable computational scaling even as instances expand to 1200 nodes.

To explicitly isolate the computational necessity of the spatial decomposition strategy, a targeted ablation study was conducted on a representative subset of large-scale topologies ranging from 302 to 1200 nodes. As detailed in Table 4, the core GLA-MAB framework was executed both with and without the decomposition wrapper under identical runtime budgets. Without this spatial partitioning, the $O(n^2)$ complexity inherent to neighborhood evaluations heavily dominates the execution, prematurely exhausting the computational budget on unpartitioned giant tours. Consequently, the algorithmic trajectory either stagnates in severe local optima, evidenced by the optimality gap on Li_32 degrading from 0.000% to 13.264%, or fails to execute sufficient learning iterations to yield a single feasible solution, as observed on the massive 1000-node X-n1001-k43 instance.

Table 4. Impact of the spatial decomposition wrapper on large-scale instances under fixed time budgets.

Instance	Size (Nodes)	Best Known Solution	Gap With Wrapper (%)	Gap Without Wrapper (%)
Golden19	360	1365.60	1.306	7.842
Golden20	420	1817.59	1.137	8.615
Li_31	1120	34,330.94	0.000	11.428
Li_32	1200	37,159.41	0.000	13.264
X-n303-k21	302	21,736.00	1.058	12.815
X-n1001-k43	1000	72,355.00	3.675	No feasible solution found

The integration of the decomposition wrapper structurally resolves this computational bottleneck. By systematically bounding the spatial subset sizes, the wrapper strictly regulates the evaluation time complexity without altering the underlying MAB learning dynamics, successfully enabling the GLA-MAB core to extract and inject proven topological structures at massive scales.

To confirm the stochastic robustness of the MAB controller across complex topologies, we conducted an extended statistical evaluation. As detailed in Table 5, we selected a representative subset of massive instances from the Golden, Li, and X sets, evaluating each across 10 independent executions with varying initial random seeds. The computational results demonstrate algorithmic stability, with the execution variance remaining tightly constrained. Most notably, on the 1200-node Li_32 instance, the framework maintained an average objective cost of 37,164.59 with a marginal standard deviation of only 6.05. This low variance confirms that using the group leaders to guide the search successfully anchors the algorithm. It prevents the random elements from causing wild fluctuations in performance, ensuring the framework consistently reaches reliable, high-quality solutions regardless of the initial random seed.

Table 5. Statistical robustness of the GLA-MAB framework across a representative subset of large-scale instances (evaluated over 10 independent runs).

Family	Instance	BKS	Best Cost	Worst Cost	Average Cost	Standard Deviation
Golden	Golden_1	5623.47	5623.47	5700.69	5650.43	26.86
	Golden_10	735.43	754.55	761.78	756.94	2.56
	Golden_20	1817.59	1838.25	1868.54	1849.59	12.02
Set Li	Li_21	16,212.83	16,212.83	16,232.08	16,218.98	7.76
	Li_26	23,977.73	23,977.73	23,990.23	23,982.20	4.91
	Li_32	37,159.41	37,159.41	37,179.67	37,164.59	6.05
Set X	X-n101-k25	27,591.00	27,591.00	27,621.00	27,603.40	9.36
	X-n303-k21	21,736.00	21,966.00	21,994.00	21,981.30	9.94
	X-n1001-k43	72,355.00	75,014.00	75,050.00	75,029.20	13.06

4.3. Ablation Studies and Component Analysis

To isolate the operational drivers of the framework’s convergence, we conducted systematic ablation studies, evaluating the full GLA-MAB against two degraded variants: a No-Bandit version (utilizing fixed operator schedules) and a No-Learning version (relying on unguided genetic mutation). As quantitatively summarized in Table 6 and visually mapped in Figure 3, the complete integration of learning and bandit control is computationally imperative. The combined chart illustrates that the full GLA-MAB framework not only achieves the lowest mean gaps (bar height) but simultaneously minimizes the computational time required to reach those optima (line trajectory). Removing the MAB controller more than triples the likelihood of the search stalling (from a 10% stagnation rate to 35%), confirming the necessity of the bandit layer for dynamic resource allocation and automated diversification. However, the most severe degradation occurs when leader-guided learning is disabled, increasing the mean gap to 1.05%. This explicitly proves that structural knowledge transfer from leaders to learners is a core functional advantage, significantly outperforming the blind variation mechanisms of the baseline genetic algorithm.

Table 6. Ablation study results evaluating the individual contributions of MAB supervision and group learning dynamics.

Variant	Evaluated Sets	Mean Best Gap (%)	Mean Avg Gap (%)	Max Best Gap (%)	Mean Time-to-Best (min)	Stagnation Rate (%)
Full GLA-MAB (learning + bandit + final LS)	setA, setF, setE, setB, setM, Golden, setX, setLi	0.255	0.55	6.813	35	10
No-bandit (fixed schedule)	same	0.45	0.85	7.8	40	35
No-learning (no leader-guided updates)	same	1.05	1.6	10.2	50	48
GA baseline (+same final LS budget)	same	0.85	1.3	9.5	44	42

To visually confirm the adaptive mechanics of the Multi-Armed Bandit (MAB) layer, Figure 4 maps the operational selection probability trajectories over the entire computational runtime across a representative subset of instances. Consistent with the structural logic of the framework, the selection distribution begins as a uniform exploration across all three transformation operators. As the search trajectory advances and local optima are encountered, the UCB selection probability dynamically diverges, heavily prioritizing Fragment Injection and Targeted Reconfiguration while systematically starving underperforming operators. This confirms that the adaptive operator selection functions exactly as mathematically intended across the tested benchmark topologies. By continuously starving operators that fail to yield improvements and instantly reacting to local stagnation, the MAB controller effectively eliminates wasted computational cycles on blind explo-

ration. This active pruning indicates that the intelligent selection layer is not merely a quality-enhancement mechanism, but a direct optimizer of the overall algorithmic run-time performance.

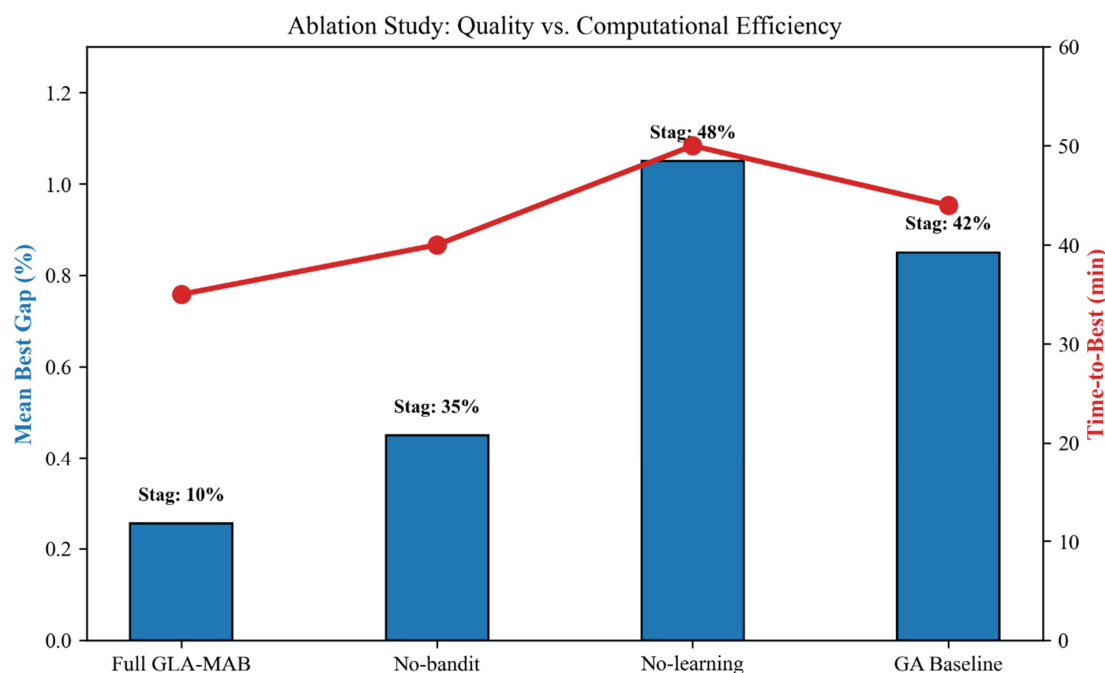


Figure 3. Ablation study trade-offs between solution quality and computational efficiency. The combo chart illustrates that the full GLA-MAB framework simultaneously minimizes the mean optimality gap and the time required to reach the best solution. Explicit stagnation rates are annotated above each variant, demonstrating the critical role of the MAB controller in preventing search arrest.

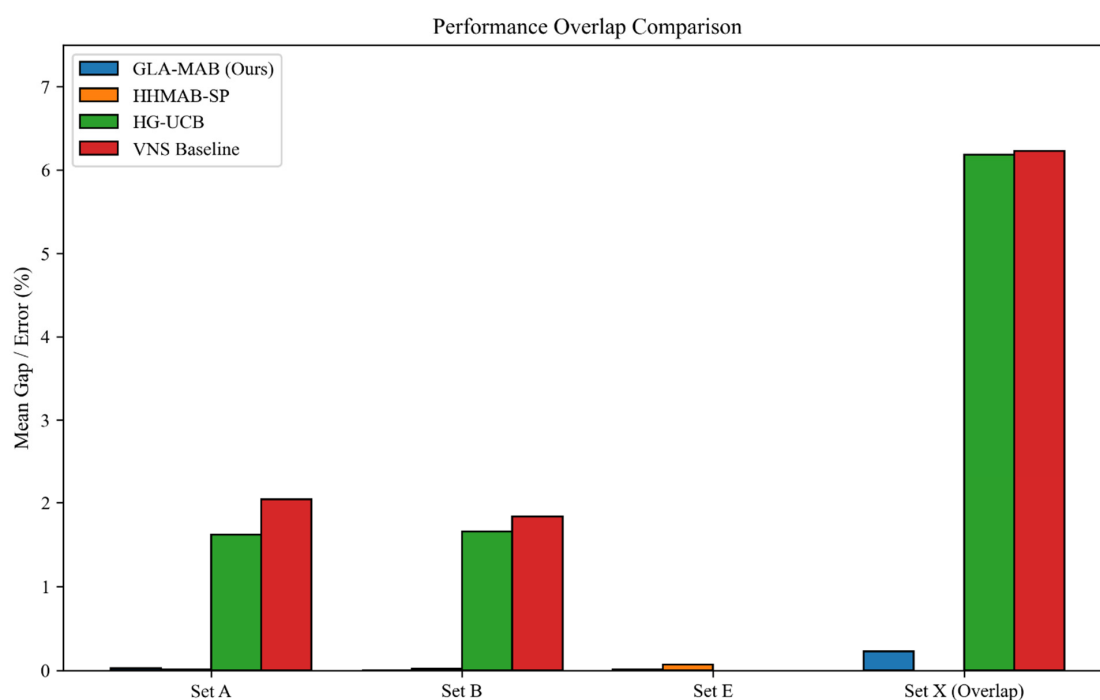


Figure 4. Shared-scale performance comparison between GLA-MAB and baseline state-of-the-art solvers. GLA-MAB achieves tighter gaps on classical sets (A, B, and E) compared to HHMAB-SP, and demonstrates significantly higher robustness on the massive, heterogeneous Set X when compared against HG-UCB and VNS baselines. Missing bars indicate instances where the respective baseline was not evaluated in its original literature.

4.4. Comparison with State-of-the-Art Baselines

To establish a foundational performance floor, the proposed framework was first evaluated against classical, unguided metaheuristics, represented in this study by the Variable Neighborhood Search (VNS) and the Genetic Algorithm (GA) baselines (detailed in Section 4.3). Traditional trajectory metaheuristics, encompassing methods such as Tabu Search and Simulated Annealing, and VNS, operate as unguided, point-to-point trajectory searches. While foundational, these algorithms rely heavily on localized, step-by-step neighborhood exploration, structurally predisposing them to become trapped in local optima when scaling to massive, heterogeneous networks. Conversely, traditional evolutionary methods, such as the baseline GA, avoid point-to-point traps but rely on stochastic, blind crossover mechanisms that frequently disrupt highly optimized routing edges.

As demonstrated in Figure 4 and Table 6, GLA-MAB establishes clear superiority over these traditional methods, which stall significantly on complex topologies (e.g., VNS yielding mean errors exceeding 6.1% on Set X). By leveraging explicit structural knowledge transfer via leader-learner dynamics, GLA-MAB bypasses both the local refinement traps of classical trajectory search and the stochastic inefficiencies of baseline evolutionary algorithms. Having validated this fundamental advantage, the remainder of our benchmarking focuses on a rigorous comparison against recent learning-based and MAB-controlled routing algorithms (HHMAB-SP [18] and HG-UCB [19]).

The GLA-MAB framework occupies a unique methodological gap compared to recent literature. While purely engineered metaheuristics, such as Hybrid Genetic Search (HGS) [1], achieve state-of-the-art absolute gaps, it is important to note that our methodological comparison relies on officially published benchmark results rather than experiments conducted under a unified computational environment. Because HGS utilizes a highly optimized implementation and differs in base hardware and evaluated time budgets, a direct one-to-one runtime comparison with our framework is fundamentally limited. Consequently, we do not claim that GLA-MAB dominates HGS in raw solution quality; HGS remains the definitive benchmark for absolute gap minimization. Instead, our objective is to demonstrate that GLA-MAB offers a distinct, training-free architectural alternative. It achieves competitive gaps on massive instances while providing dynamic, online adaptability and explicit structural knowledge transfer, which are capabilities not present in traditional phase-tuned solvers.

Conversely, while recent Deep Reinforcement Learning (DRL) routing solvers automate the search process, they demand prohibitive offline training costs and struggle with out-of-distribution scaling. GLA-MAB provides a highly competitive, training-free alternative. To demonstrate this, we benchmarked the framework against recent MAB-controlled routing algorithms: HHMAB-SP [18] and HG-UCB [19].

As illustrated in the shared-scale overlap analysis in Figure 4, GLA-MAB exhibits significant improvement in constraint management and solution quality. Against HHMAB-SP on Sets A, B, and E, the proposed framework achieves tighter mean gaps (reaching 0.000% on Set B). The contrast is starkest on the highly complex Set X overlap since GLA-MAB restricts its mean deviation to 0.226%, whereas the HG-UCB and baseline VNS frameworks report mean errors exceeding 6.1%. This confirms that integrating MAB control with explicit population learning yields a far more robust solver than integrating MAB with standard, unguided local search neighborhoods. Due to the performance observed on the Set Li topologies, detailed instance-level computational results are provided in Supplementary Table S6.

Finally, to guarantee a comprehensive evaluation, the framework was additionally contextualized against the absolute state-of-the-art in population-based routing: the Hybrid Genetic Search (HGS) [1]. Utilizing the officially published benchmark gaps for

HGS, a direct instance-level comparison on the complex Set X topologies reveals a distinct methodological trade-off. On small-to-medium topologies, GLA-MAB proves highly competitive, successfully matching the HGS optimal performance (0.000% gap) on instances such as X-n101-k25 and X-n115-k10. As instance dimensions scale toward 1000 nodes, HGS maintains a stricter performance ceiling due to its heavily engineered, implicit crossover mechanisms and static, tuned parameter schedules. In contrast, while yielding slightly wider gaps on massive scales, GLA-MAB achieves its results entirely through online MAB adaptation, bypassing the rigid calibration required by classical memetic solvers.

5. Conclusions

This paper introduced GLA-MAB, a training-free metaheuristic framework that successfully bridges the gap between computationally expensive offline deep-learning solvers and manually tuned classical metaheuristics for the Capacitated Vehicle Routing Problem (CVRP). By partitioning the search space into explicit leader-learner groups, the framework extracts and injects proven topological structures, transforming implicit evolutionary variation into directed knowledge transfer. Simultaneously, the integration of a hierarchical Multi-Armed Bandit (MAB) layer provides dynamic, online allocation of search operators and automated stagnation recovery without relying on fixed parameter schedules.

Comprehensive computational experiments across 117 standard CVRP instances demonstrated that GLA-MAB operates in a highly competitive regime. It achieved 100% match rates on classical sets (B and F) and maintained residual mean gaps under 0.03% on Sets A and E. Furthermore, the inclusion of a spatial decomposition wrapper proved highly effective, allowing the framework to scale predictably to massive, heterogeneous topologies of up to 1200 nodes (Sets Golden, Li, and X) while maintaining strict constraint feasibility. Explicit ablation studies confirmed that disabling either the MAB controller or the leader-guided learning mechanism results in severe performance degradation and high stagnation rates, validating the necessity of the bipartite architecture.

Moving forward, several promising avenues exist for extending the capabilities and operational scope of this framework. First, from a methodological perspective, the explicit knowledge transfer mechanisms within the group learning layer present strong opportunities for hybridization with exact methods. For instance, the high-quality, structurally diverse routes generated by the leader-learner dynamics could be actively archived to populate a matheuristic Set Partitioning framework, combining the rapid convergence of metaheuristics with the rigorous bounds of exact mathematical solvers. Additionally, future work will explore expanding the MAB controller to dynamically tune the spatial decomposition parameters online.

Second, while the experimental validation in this study focused on the standard CVRP to rigorously isolate the algorithmic contributions of the MAB and group learning layers, the architecture's practical application scope extends naturally to more complex, heavily constrained real-world routing variants. For example, in the Vehicle Routing Problem with Time Windows (VRPTW), the framework can be adapted by modifying the deterministic decoding policies to evaluate and penalize time-schedule violations, ensuring that the leader-to-learner structural injections maintain strict temporal feasibility. Similarly, the framework presents a significant structural advantage for dynamic and stochastic routing environments, because the hierarchical MAB controller is fundamentally designed to manage non-stationary reward landscapes. The GLA-MAB can seamlessly adapt to real-time operational disruptions (such as fluctuating customer demands or traffic delays) by continuously re-evaluating operator utilities and adjusting the search trajectory online based on immediate feedback.

Finally, because the GLA-MAB controller operates on a shared permutation-based core, the underlying learning mechanism itself is largely problem-agnostic, making it highly portable across a diverse range of combinatorial optimization applications. Our immediate future work involves transitioning this framework to multi-objective environments, specifically adapting it for the Employee Bus Routing Problem (EBRP). This transition does not require redesigning the MAB controller or its reward formulations to navigate Pareto fronts. However, it primarily involves replacing the standard capacity penalties with domain-specific route construction logic, such as enforcing routes that must sequentially visit designated bus stops before proceeding to workplaces, while optimizing for complex concurrent metrics like driver equity, passenger riding times, and overall operational fleet costs.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/asi9070143/s1>, Table S1: GLA-MAB results for set A; Table S2: GLA-MAB results for set B; Table S3: GLA-MAB results for set E; Table S4: GLA-MAB results for set F; Table S5: GLA-MAB results for set M; Table S6: GLA-MAB results for set Li; Table S7: GLA-MAB results for Golden; Table S8: GLA-MAB results for some instances of set X.

Author Contributions: Conceptualization, H.B. and K.O.; methodology, H.B. and K.O.; software, H.B.; validation, H.B., K.O. and R.E.; formal analysis, H.B. and A.G.; investigation, H.B.; resources, H.B. and R.E.; writing—original draft preparation, H.B.; writing—review and editing, H.B. and A.G.; visualization, K.O.; supervision, R.E. and A.G.; project administration, R.E. and A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The analyses and computational experiments reported in this study were conducted using the publicly available, open-access CRVLIP benchmark dataset. The data can be accessed through its public repository or open-access literature sources: <https://galgos.inf.puc-rio.br/cvrplib/en/instances> (accessed on 28 June 2026).

Acknowledgments: During the preparation of this work, the authors used language-support tools, including Grammarly (for Chrome Version 14.1308.0) and Paperpal (Extensive 2.0), to improve the language and readability of the manuscript. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vidal, T. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Comput. Oper. Res.* **2022**, *140*, 105643. [\[CrossRef\]](#)
2. Kwon, Y.D.; Choo, J.; Kim, B.; Yoon, I.; Gwon, Y.; Min, S. POMO: Policy optimization with multiple optima for reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21188–21198.
3. Zhao, W.; Zhang, Z.; Zhao, H.; Bian, X. A Q-Learning-assisted evolutionary optimization method for solving the capacitated vehicle routing problem. *Appl. Sci.* **2025**, *15*, 9332. [\[CrossRef\]](#)
4. Uchoa, E.; Pecin, D.; Pessoa, A.; Poggi, M.; Vidal, T.; Subramanian, A. New benchmark instances for the capacitated vehicle routing problem. *Eur. J. Oper. Res.* **2017**, *257*, 845–858. [\[CrossRef\]](#)
5. Bengio, Y.; Lodi, A.; Prouvost, A. Machine learning for combinatorial optimization: A methodological tour d’horizon. *Eur. J. Oper. Res.* **2021**, *290*, 405–421. [\[CrossRef\]](#)
6. Miao, W.; Li, L.; Wang, Z. A survey on deep reinforcement learning for traffic signal control. In Proceedings of the 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 1092–1097.
7. Kool, W.; Van Hoof, H.; Welling, M. Attention, learn to solve routing problems! *arXiv* **2018**, arXiv:1803.08475.
8. Cappart, Q.; Chélat, D.; Khalil, E.B.; Lodi, A.; Morris, C.; Veličković, P. Combinatorial optimization and reasoning with graph neural networks. *J. Mach. Learn. Res.* **2023**, *24*, 4348–4355.
9. Bi, J.; Cao, Z.; Chee, Y.M.; Chen, J.; Ma, Y.; Sun, Y.; Wang, J. Learning generalizable models for vehicle routing problems via knowledge distillation. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 31226–31238. [\[CrossRef\]](#)

10. Joshi, C.K.; Cappart, Q.; Rousseau, L.M.; Laurent, T. Learning the travelling salesperson problem requires rethinking generalization. *Constraints* **2022**, *27*, 70–98. [CrossRef]
11. Geisler, S.; Sommer, J.; Schuchardt, J.; Bojchevski, A.; Günnemann, S. Generalization of neural combinatorial solvers through the lens of adversarial robustness. *arXiv* **2021**, arXiv:2110.10942.
12. Wu, Y.; Song, W.; Cao, Z.; Zhang, J.; Lim, A. Learning improvement heuristics for solving routing problems. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 5057–5069.
13. Lu, H.; Zhang, X.; Yang, S. A learning-based iterative method for solving vehicle routing problems. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
14. Zhao, J.; Mao, M.; Zhao, X.; Zou, J. A hybrid of deep reinforcement learning and local search for the vehicle routing problems. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 7208–7218. [CrossRef]
15. Hottung, A.; Tanaka, S.; Tierney, K. Deep learning assisted heuristic tree search for the container pre-marshalling problem. *Comput. Oper. Res.* **2020**, *113*, 104781. [CrossRef]
16. Tian, Y.; Peng, S.; Zhang, X.; Rodemann, T.; Tan, K.C.; Jin, Y. A recommender system for metaheuristic algorithms for continuous optimization based on deep recurrent neural networks. *IEEE Trans. Artif. Intell.* **2020**, *1*, 5–18. [CrossRef]
17. Zhou, J.; Lai, X.; Chow, J.Y. Multi-armed bandit on-time arrival algorithms for sequential reliable route selection under uncertainty. *Transp. Res. Rec.* **2019**, *2673*, 673–682. [CrossRef]
18. Hou, Y.E.; He, W.; Wang, C.; Ren, X. A two-stage selection hyper-heuristic algorithm for the capacitated vehicle routing problem. *IAENG Int. J. Appl. Math.* **2022**, *52*, 979–988. Available online: http://www.iaeng.org/IJAM/issues_v52/issue_4/IJAM_52_4_26.pdf (accessed on 28 June 2026).
19. Kalatzantonakis, P.; Sifaleras, A.; Samaras, N. A reinforcement learning-variable neighborhood search method for the capacitated vehicle routing problem. *Expert Syst. Appl.* **2023**, *213*, 118812. [CrossRef]
20. Lagos, F.; Pereira, J. Multi-armed bandit-based hyper-heuristics for combinatorial optimization problems. *Eur. J. Oper. Res.* **2024**, *312*, 70–91. [CrossRef]
21. Sultana, N.; Chan, J.; Abbasi, B.; Sarwar, T.; Qin, A.K. Learning to guide local search optimisation for routing problems. *Oper. Res. Lett.* **2024**, *55*, 107136. [CrossRef]
22. Nagata, Y.; Kobayashi, S. A powerful genetic algorithm using edge assembly crossover for the traveling salesman problem. *Inf. J. Comput.* **2013**, *25*, 346–363. [CrossRef]
23. Vidal, T.; Crainic, T.G.; Gendreau, M.; Prins, C. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Comput. Oper. Res.* **2013**, *40*, 475–489. [CrossRef]
24. Queiroga, E.; Sadykov, R.; Uchoa, E. A POPMUSIC matheuristic for the capacitated vehicle routing problem. *Comput. Oper. Res.* **2021**, *136*, 105475. [CrossRef]
25. Karimi-Mamaghan, M.; Mohammadi, M.; Meyer, P.; Karimi-Mamaghan, A.M.; Talbi, E.G. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *Eur. J. Oper. Res.* **2022**, *296*, 393–422. [CrossRef]
26. Grinsztajn, L.; Oyallon, E.; Varoquaux, G. Why do tree-based models still outperform deep learning on typical tabular data? *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 507–520. [CrossRef]
27. Ayachi Amar, C.; Bouanane, K.; Aiadi, O. A reinforcement learning framework for scalable partitioning and optimization of large-scale capacitated vehicle routing problems. *Electronics* **2025**, *14*, 3879. [CrossRef]
28. Wang, L.; Zheng, J.; Xiong, Z.; He, K. Multi-armed bandit and backbone boost Lin-Kernighan-Helsgaun algorithm for the traveling salesman problems. *arXiv* **2025**, arXiv:2501.04072.
29. Kakkar, M.K.; Gupta, G.; Garg, N.; Singla, J. Solving the capacitated vehicle routing problem (CVRP) using clustering and meta-heuristic algorithm. In *Proceedings of the Congress on Control, Robotics, and Mechatronics, Warangal, India, 3–4 February 2024*; Springer: Berlin/Heidelberg, Germany, 2024.
30. Sultana, N.N.; Meisheri, H.; Baniwal, V.; Nath, S.; Ravindran, B.; Khadilkar, H. Reinforcement learning for multi-product multi-node inventory management in supply chains. *arXiv* **2020**, arXiv:2006.04037.
31. Wang, P.; Xue, B.; Liang, J.; Zhang, M. A survey on evolutionary multimodal optimization-driven machine learning. *IEEE Trans. Evol. Comput.* **2025**; early access. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.