Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 7, No. 4, October 2025, pp: 991-1011 e-ISSN: 2656-8632

RESEARCH ARTICLE OPEN ACCESS

Optimizing Medical Logistics Networks: A Hybrid Bat-ALNS Approach for Multi-Depot VRPTW and Simultaneous Pickup-Delivery

Taha Anass[1], Elatar Said[2], El Bazzi Mohamed Salim[3], Ait Ider Abdelouahed[1], and Lotfi Najdi[1]

[1] Laboratory of System Computer Engineering, Mathematics and Applications, Ibn Zohr university, Agadir, Morocco.

[2] Laboratory of Research in Optimization, Emerging Systems, Networks, and Imaging, Chouaib Doukkali university, Faculty of Science, El Jadida, Morocco.

[3] Laboratory of Image and Pattern Recognition, Intelligent and Communicating Systems, Ibn Zohr university, Agadir, Morocco.

Corresponding author: Taha Anass (e-mail: an.taha@uiz.ac.ma), **Author(s) Email**: Elatar Said (e-mail: elatar.said@gmail.com, El Bazzi Mohamed Salim (e-mail: m.elbazzi@uiz.ac.ma), Ait Ider Abdelouahed (e-mail: a.aitider@uiz.ac.ma), Lotfi Najdi (e-mail: l.najdi@uiz.ac.ma)

Abstract **This paper tackles the multi-depot heterogeneous-fleet vehicle-routing problem with time windows and simultaneous pickup and delivery (MDHF-VRPTW-SPD), a variant that mirrors he growing complexity of modern healthcare logistics. The primary purpose of this study is to model this complex routing problem as a mixed-integer linear program and to develop and validate a novel hybrid metaheuristic, B-ALNS, capable of delivering robust, high-quality solutions. The proposed B-ALNS combines a discrete Bat Algorithm with Adaptive Large Neighborhood Search, where the bat component supplies frequency-guided diversification, while ALNS adaptively selects destroy and repair operators and exploits elite memory for focused intensification. Extensive experiments were conducted on twenty new benchmark instances (ranging from 48 to 288 customers), derived from Cordeau's data and enriched with pickups and a four-class fleet. Results show that B-ALNS attains a mean cost 1.15 % lower than a standalone discrete BA and 2.78 % lower than a simple LNS, achieving the best average cost on 17/20 instances and the global best solution in 85% of test instances. Statistical tests further confirm the superiority of the hybrid B-ALNS, a Friedman test and Wilcoxon signed-rank comparisons give p-value of 0.0013 versus BA and p-value of 0.0002 versus LNS, respectively. Although B-ALNS trades speed for quality (182.65 seconds average runtime versus 54.04 seconds for BA and 11.61 seconds for LNS), it produces markedly more robust solutions, with the lowest cost standard deviation and consistently balanced routes. These results demonstrate that the hybrid B-ALNS delivers statistically significant, high-quality solutions within tactical planning times, offering a practical decision-support tool for secure, cold-chain-compliant healthcare logistics.**

Keywords **Pharmaceutical Distribution, *Medical Logistics Networks,* Vehicle Routing Problem, Hybrid Metaheuristic, Bat Algorithm, Adaptive Large Neighborhood Search.**

I. Introduction

Efficient distribution products from depots to customers is a vital concern in modern logistics, impacting sectors such as pharmaceutical distribution, home healthcare, and other time-sensitive medical services. In these contexts, timely delivery of critical medications, optimized routing, and effective management of returns are crucial for reducing operational costs and improving service quality. This challenge is frequently modelled as the Vehicle Routing Problem (VRP), where a set of geographically dispersed customers must be served by a fleet of vehicles subject to numerous constraints. Since its inception by Dantzig and Ramser in 1959 [1], the VRP has evolved into numerous variants, such as the Capacitated VRP [2], VRP with Time Windows (VRPTW) [3], Heterogeneous Fleet VRP (HFVRP) [4], Pickup and Delivery VRP (PDVRP) [5], and Multi-Depot

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

VRP (MDVRP) [6], each designed to address specific practical considerations.

This paper focuses on a specialized variant, the Multi-Depot Heterogeneous Fleet VRP with Time Windows and Simultaneous Pickup and Delivery (MDHF VRPTW SPD). This formulation closely aligns with pharmaceutical distribution requirements, where medications must be delivered to local pharmacies while concurrently collecting patient lab samples, expired drugs, or biohazardous waste, all under strict time windows (TW). The problem setup involves multiple depots, each housing different types of vehicles with distinct capacity and cost structures. The main objective is to minimize travel costs and reduce the number of vehicles used, while respecting vehicle capacity constraints, time windows, and depot-specific vehicle availability.

To address the complexity nature of the MDHF-VRPTW-SPD problem, we propose a novel hybrid approach (B-ALNS), that integrates the Bat Algorithm (BA) [7] with Adaptive Large Neighborhood Search (ALNS) [8]. The BA, inspired by the echolocation behavior of bats, effectively balances the search between exploring the entire solution space and exploiting local areas, which helps prevent early convergence to suboptimal solutions. In parallel, ALNS iteratively removes and reinserts parts of the solution to explore the search space thoroughly, a process that is especially effective for large-scale, real-world vehicle routing problems.

This combined strategy ensures a balanced approach between global exploration and local refinement, leading to high-quality routing solutions while making efficient use of computational resources within the MDHF-VRPTW-SPD context. By applying the proposed hybrid B-ALNS method to 20 test instances, we observed that it consistently delivers high-quality routing solutions. The method not only minimizes travel costs and reduces the number of vehicles required but also efficiently manages computational resources, even as problem complexity increases. These promising results highlight the robustness and practical applicability of our approach for addressing the challenging MDHF-VRPTW-SPD in real-world pharmaceutical distribution scenarios.

The remainder of this paper is structured as follows. Section IIpresents a comprehensive review of the existing literature, emphasizing recent developments and identifying gaps in the research on complex vehicle routing problems. Section III provides a formal description of MDHF-VRPTW-SPD, presenting its mathematical formulation along with the key constraints involved. Section IVintroduces the proposed hybrid B-ALNS approach, detailing its algorithmic framework and operational mechanics. Section V reports the experimental results gathered from 20 benchmark instances, showcasing the effectiveness and efficiency of the methodology. Finally, section VI wraps up the paper with concluding remarks and suggests potential directions for future research.

II. Literature survey

The MDHF-VRPTW-SPD problem considered in this study shares characteristics with several well-known VRP variants, especially the pickup and delivery VRP with time windows (PD-VRPTW) [9] and the multi-depot VRP with time windows (MD-VRPTW) [6] [45]. In recent years, these variants have attracted significant research interest, leading to the development of various extensions. These include optimizing Simultaneous pickup and delivery VRP for personnel transportation in COVID-19 [10] and formulating multi-depot VRPs with simultaneous pickup and delivery within the context of the physical internet [11], all aiming to better reflect the complexities of real-world distribution scenarios. Applications of these rich VRP variants are found in diverse industries, ranging from last-mile e-commerce logistics [12], milk collection in dairy supply chains [13], and Vehicle routing problem in cold-Chain logistics [14]. number of vehicles used, while respecting vehicle capacity constraints, time windows, and depot-specific vehicle availability.

Recent advancements in vehicle routing have refined both modeling and solution strategies to address the complex challenges of modern healthcare logistics. For instance, [15] optimizes medical-waste collection by combining time-window limits with stochastic travel-time reliability considerations in public-health operations. Similarly, [16] extends the classical VRP to a multi-depot, multimodal home-health-care setting, where routing and scheduling decisions must respect both vehicle–mode compatibility and depot assignment. Addressing direct patient service, [17] formulates a VRP with simultaneous delivery and pickup, time windows, and vehicle-capacity limits to ensure that drug deliveries and bio-sample returns are coordinated within clinically acceptable windows references. To guarantee safe disposal pathways, [18] introduces multi-compartment vehicles, enforcing the segregation of hazardous versus non-hazardous waste throughout the route references. Finally, [19] tackles emergency medical logistics with a heterogeneous fleet, coupling pickup-and-delivery tasks and strict time windows.

To tackle the complexity of problems such as the MDHF-VRPTW-SPD, researchers have developed various methods, including exact algorithms, heuristics, and metaheuristics. Exact algorithms, such as Branch-and-Bound [20], Branch-and-Price [21], and dynamic programming [22], can find optimal solutions but are often impractical for large-scale instances due

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

to the NP-hard nature of these problems. As problem size increases, computation time grows exponentially, limiting the applicability of exact methods to smaller problems. Laporte [23] reviews several exact algorithms and classical heuristics developed to solve the VRP. Among these heuristics, the Clarke-Wright Savings Algorithm [24] and the sweep algorithm [25] are widely studied, though often tailored to particular problem scenarios, which may limit their broader applicability.

Considering these limitations, metaheuristics provide effective solutions for addressing complex VRP variants. Population-based methods, including genetic algorithm [26] and differential evolution [27], as well as swarm intelligence techniques such as ant colony optimization [28], particle swarm optimization [29], firefly algorithm [30], and bat algorithm, effectively navigate extensive solution spaces. Local search techniques, including simulated annealing [31] and tabu search [32], improve solutions by escaping local optima. Large neighborhood search (LNS) [33] enhances the solution quality through an iterative process of destruction and repair operations. Hybrid metaheuristics [34] integrate global and local search strategies, resulting in enhanced performance.

ALNS has demonstrated considerable efficacy across many VRP types. Ropke and Pisinger [8] initially demonstrated its efficacy on the PDPTW by adaptively selecting destroy and repair operators, resulting performance enhancements across more than half of the benchmarks. Naccache et al. [35] improved ALNS for the multi-pickup and delivery problem with time windows by incorporating specialized operators to address precedence constraints. Sacramento et al. [36] employed ALNS for truck-drone routing, effectively exploring the solution space. While Mofid-Nakhaee and Barzinpour [37] hybridized ALNS with a whale optimization algorithm to address a multi-compartment waste collecting issue, showcasing effective hybridization. Chen et al. [38] expanded ALNS to synchronize vans and delivery robots in VRPTW, emphasizing its flexibility in supporting emerging logistics technologies.

Bat-inspired metaheuristics have progressively evolved to address diverse VRP challenges. Yang [7] first introduced the Bat Algorithm (BA), outperforming classical methods on continuous benchmarks. Taha et al. [2] later adapted BA for the Capacitated VRP (ABA), achieving high precision on standard CVRP instances. Osaba et al. [39] developed a discrete, hamming distance–based BA for medical goods distribution with waste collection, validated via rigorous statistical tests. To handle time-window constraints, Osaba et al. [40] proposed an evolutionary discrete BA with random reinsertion, unifying route minimization and diversification. Finally, Taha et al. [41] hybridized BA with Large Neighborhood Search to solve the VRPTW, demonstrating competitive convergence on Solomon's benchmarks.

Despite considerable advances, most VRP models still address only isolated features, time windows, heterogeneous fleets, or pickup and delivery, rather than their combination. The MDHF VRPTW SPD variant bridges this gap by unifying multiple depots, diverse vehicle types, strict time windows, and simultaneous pickup and delivery into a single framework. This holistic approach mirrors real-world pharmaceutical distribution, where logisticians must coordinate varied fleets across several depots to manage both deliveries and returns within tight schedules. On the other hand, to address the proposed problem, we have developed a hybridization of a discrete version of the BA and the ALNS. To the best of our knowledge, no prior work has fully applied the hybrid B-ALNS to such a complex VRP, underscoring the novel contribution of our study.

III. Problem formulation

The application of the MDHF-VRP-TWSPD is commonly observed in the pharmaceutical and healthcare logistics, where last-mile parcel services deliver customers' orders and, during the same scheduled time window, simultaneously collect return packages. Our objective is to minimize the total transport cost, defined as the sum of fixed dispatch charges for each vehicle used and variable expenses proportionate to the distance traveled along the routes. A feasible solution to the MDHF-VRPTW-SPD must satisfy a strict set of operational constraints.

Fundamentally, each customer must be visited exactly once by a single vehicle, ensuring that neither delivery nor pickup tasks are split across multiple visits or vehicles. Each vehicle must depart from and return to its originating depot. Throughout its tour, a vehicle's onboard load must remain within its designated capacity, accounting for the decrease in load after deliveries and the increase after pickups. The simultaneous nature of the problem dictates that both delivery and pickup operations at a customer location occur during the same stop. Critically, all services must comply with customer time windows; vehicles may wait if they arrive early, but service must be completed before the customer's closing time. Operationally, the number of vehicles of each type dispatched from a depot cannot exceed its available fleet inventory. Finally, to ensure driver and vehicle compliance, the total duration of any route, including all driving, waiting, and service times, must not exceed the maximum duty time allowed for that vehicle type, and all vehicle types are treated as heterogeneous with distinct capacity and cost attributes.

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

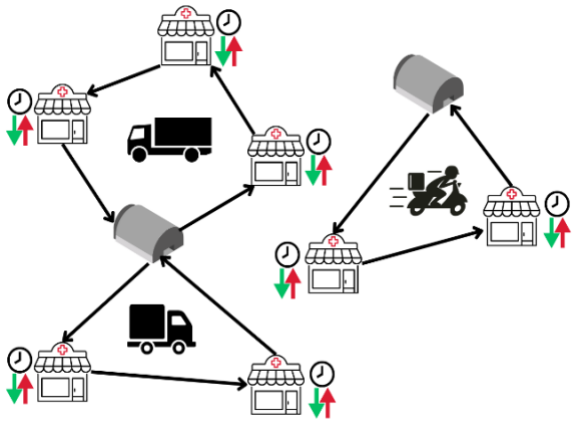

Fig. 1. The Visual representation of the MDHF-VRP-TWSPD in the pharmaceutical distribution.

Fig.1 represents a problem instance with 8 customers (e.g. local pharmacies). Each customer requires a medical delivery (green arrows) and has a simultaneous collection task for waste or expired medicine (red arrows). A heterogeneous fleet of three specialized vehicle types, dispatched from two different depots, serves these customers. Finally, the clocks refer to the time window that vehicles must respect when performing service.

To formulate the MDHF-VRP-TWSPD as a mixed-integer linear program, we first define its core components. The problem is defined on a directed graph consisting of a set of customers C indexed by i (or j when used as a destination index) and a set of depots D, indexed by d. The complete node set is therefore $V = D \cup C$, while contains $A \subseteq V \times V$ all directed arcs (i,j) with $i \neq j$. A heterogeneous fleet is represented by the set of vehicle types K, indexed by k; each depot $d \in D$ disposes of F_{dk} vehicles of type k. We write $N = |C|$ for the number of customers.

Each vehicle type k offers a capacity Q_k and incurs a fixed dispatch cost f_k together with a distance-proportional cost c_k. Every customer i requests a delivery quantity a_i and a pick-up quantity b_i, service at that location can begin no earlier than e_i and must finish by l_i, occupying s_i service time units. Travelling along arc (i,j) consumes τ_{ij} units of time and covers a distance d_{ij}. Each vehicle type is further subject to a maximum route duration T_k^{max}. To linearize load, time-propagation and return-duration constraints we define three Big-M constants: $M_k^w = Q_k$ for load conservation, $M_{ij,k}^t = l_j - e_i - s_i - \tau_{ij}$ for time propagation on every arc $(i,j) \in A$, and $M_{id,k}^r = T_k^{max} - s_i - \tau_{id}$ for enforcing the route-duration limit on return arcs (i,d).

Binary decision variables $x_{ij}^{kd} \in \{0,1\}$ equal 1 when a type-k vehicle is based at the depot d traverses arc (i,j). Two continuous auxiliary variables track route state: $w_i^{dk} \geq 0$ records the vehicle load immediately after servicing node i, while $t_i^{dk} \geq 0$ stores the service-completion time at that node. Finally, classical Miller–Tucker–Zemlin (MTZ) indices $u_i \in \{1, \dots, N\}$ eliminate sub-tours by imposing an ordering on the customer set.

The objective function Eq. (1) optimizes transport cost by summing the fixed dispatch cost for each vehicle and the distance-proportional cost on each traveled arc. Constraint Eq. (2) [1] ensures single service by forcing each vehicle to visit each client exactly once, whereas Eq. (3) [1] conserves flow by requiring vehicles to enter and leave customers. Constraint Eq. (4) [1] limits the number of vehicles of each type that can depart from (and return to) a depot to its available inventory, combining fleet availability and depot consistency. Constraints Eq. (5) and Eq. (6) [9] subtract deliveries and add pickups to distribute vehicle loads over the network while ensuring vehicle capacity limits are respected.

$$Min \sum_{d\in D}\sum_{k\in K}\left[f_k \sum_{j:(d,j)\in A} x_{dj}^{dk} + \sum_{(i,j)\in A} c_k d_{ij} x_{ij}^{dk}\right] \quad (1)$$

The problem is subject to the following constraints:

$$\sum_{d\in D}\sum_{k\in K}\sum_{i:(i,j)\in A} x_{ij}^{dk} = 1, \quad \forall j \in C \quad (2)$$

$$\sum_{j:(i,j)\in A} x_{ij}^{dk} = \sum_{h:(h,i)\in A} x_{hi}^{dk}, \quad \forall i \in C, d, k \quad (3)$$

$$\sum_{j\in C} x_{dj}^{dk} = \sum_{i\in C} x_{id}^{dk} \leq F_{dk}, \quad \forall d, k \quad (4)$$

$$w_j^{dk} \geq w_i^{dk} - a_j + b_j - M_k^w(1 - x_{ij}^{dk}), \quad \forall (i,j) \in A, d, k \quad (5)$$

$$0 \leq w_i^{dk} \leq Q_k, \forall i \in V, d, k \quad (6)$$

$$t_j^{dk} \geq t_i^{dk} + s_i + \tau_{ij} - M_{ij,k}^t\left(1 - x_{ij}^{dk}\right), \quad \forall (i,j) \in A, d, k \quad (7)$$

$$e_i \leq t_i^{dk} \leq l_i, \quad \forall i \in C, d, k \quad (8)$$

$$t_i^{dk} + s_i + \tau_{id} \leq T_k^{max} + M_{id,k}^r\left(1 - x_{id}^{dk}\right), \quad \forall i \in C, d, k \quad (9)$$

$$u_i - u_j + N\sum_{d\in D}\sum_{k\in K} x_{ij}^{dk} \leq N - 1, \quad \forall i \neq j \in C \quad (10)$$

$$1 \leq u_i \leq N, \quad \forall i \in C \quad (11)$$

$$w_d^{dk} = 0, \quad t_d^{dk} = 0, \quad \forall d \in D, k \in K \quad (12)$$

$$\begin{aligned} &x_{ij}^{kd} \in \{0,1\}, \\ &w_i^{dk} \geq 0, \\ &t_i^{dk} \geq 0, \\ &u_i \in \mathbb{Z} \cap [1, N] \end{aligned} \qquad \forall i,j \in C, d, k \quad (13)$$

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

Service completion, travel, and waiting durations are carried forward by constraints Eq. (7) and Eq. (8) [3], ensuring each service occurs within its prescribed time window. Constraint Eq. (9) [3] limits route duration, so no vehicle exceeds its duty time. Finally, the MTZ subtour-elimination constraints Eq. (10) and Eq. (11) [46] organize customers' visits, preventing customer-only loops and ensuring that all routes are connected to a depot. Constraint Eq. (12) [6] fixes each route to its origin by setting the load and time at every depot to zero, providing a valid starting point for subsequent propagation. The variable domains Eq. (13) are as follows: each routing decision x_{ij}^{kd} is binary; the load w_i^{dk} and time t_i^{dk} variables are continuous and nonnegative; and each ordering index u_i is an integer between 1 and N.

IV. Hybrid B-ALNS

A. Original Bat Algorithm

The original Bat Algorithm (BA), introduced by Yang in 2010 [7], simulates the echolocation behavior of microbats for global optimization. It integrates three key mechanisms, frequency tuning, loudness control, and pulse-emission adaptation, while employing simple yet effective update rules for pulse frequency, velocity, and position, complemented by a local random walk for intensified search. The BA operates under three idealized rules:

1. Each bat uses echolocation, emitting acoustic pulses and listening for echoes, to explore the search space and avoid obstacles.
2. Each bat $\boldsymbol{i}$ flies randomly with velocity $\boldsymbol{v_i}$ at position $\boldsymbol{x_i}$, emitting pulses at frequency $\boldsymbol{f_i} \in [\boldsymbol{f_{min}}, \boldsymbol{f_{max}}]$, wavelength $\boldsymbol{\lambda}$, and loudness $\boldsymbol{A_i}$. Bats adjust the frequency $\boldsymbol{f_i}$ and their pulse-emission rate $\boldsymbol{r_i} \in [\boldsymbol{0}, \boldsymbol{1}]$ adaptively based on how close they are to promising solutions.
3. The loudness $\boldsymbol{A_i}$ decays from an initial value $\boldsymbol{A_0}$ toward a minimum $\boldsymbol{A_{min}}$ as bats converge on prey (i.e., optimal solutions).

At each iteration t, the pulse frequency of the bat i is updated via Eq. (14) [7]:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (14)$$

where f_i is the frequency of the bat i, f_{min} and f_{max} are the minimum and maximum frequencies, respectively, and β is a random vector drawn from a uniform distribution $U(0,1)$. Each bat in the population moves through the search space by updating its velocity v_i and position x_i using the Eq(15) and Eq. (16) [7]:

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \quad (15)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (16)$$

Algorithm 1: Original Bat Algorithm

Input: objective function $f(x)$, population size n, termination criterion
Output: best solution x_*
1. Initialize the bat population x_i, $(i = 1, \ldots, n)$
2. **for** each bat x_i in the population **do**
3. Initialize the velocity v_i, the pulse rate r_i and the loudness A_i
4. Define the frequency f_i at position x_i
5. **end for**
6. Evaluate $f(x_i)$ for all bats and set $x_* \leftarrow best\ x_i$
7. **while** termination criterion not reached **do**
8. **for** each bat x_i in the population **do**
9. Generate new solutions by adjusting frequency, and updating velocity and position [Eq. (14) to Eq. (16) [7]]
10. **if** $rand() > r_i$ **then**
11. Select one solution among the current best solutions
12. Generate a local solution around the selected solution [Eq. (17) [7]]
13. **end if**
14. Generate a new solution by flying randomly
15. **if** $rand() < A_i$ and $f(x_i) < f(x_*)$ **then**
16. accept the new solutions;
17. increase r_i and reduce A_i [Eq. (18) and Eq. (19) [7]]
18. **End if**
19. **end for**
20. rank the bats and find the current best solution x_*
21. **end while**
22. Return x_*

with x_* denoting the global best solution found so far. To enhance local search, a random walk around the current best is performed using Eq. (17) [7]:

$$x_{new} = x_{old} + \varepsilon A^t \quad (17)$$

where x_{new} and x_{old} are the new and old generated solutions respectively, ε is a random number in $[-1, 1]$ and A^t is the average loudness of the swarm at time step t.

$$A_i^{t+1} = \alpha A_i^t \quad (18)$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \quad (19)$$

where $\alpha \in (0,1)$ is a constant that controls loudness decay and $\gamma > 0$ is a constant that governs the increase of r_i. Upon locating better solutions, bats decrease their loudness and increase pulse emission rates using Eq. (18) and Eq. (19) [7]. As $t \to \infty$, $A_i^t \to 0$ and $r_i^t \to r_i^0$, ensuring convergence. As Yang indicates in his paper [7], the parameters γ and α are typically set to values close to 1 for simplicity, for example, many studies in the literature choose $\gamma = \alpha = 0.98$.

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

This combination of global moves guided by echolocation and local random walks endows the BA with both exploration and exploitation capabilities, making it a robust choice for continuous optimization tasks. The overall procedure is summarized in Algorithm 1, which describes the pseudocode of the original BA.

B. Adaptive Large Neighborhood Search

The ALNS heuristic, first proposed by Ropke and Pisinger [8], explores vast solution spaces by iteratively "destroying" and "repairing" candidate solutions. In each iteration, a removal operator (RO) selectively removes a subset of elements from the current solution, creating a partial solution. An insertion operator (IO) then rebuilds it by reinserting the removed elements in the best possible positions. By dynamically selecting among multiple ROs and IOs based on their past performance, ALNS balances diversification and intensification to identify high-quality solutions efficiently.

The pseudocode presented in Algorithm 2 begins by setting the best solution s_{best} to the initial solution S (line 1). The main loop (lines 2-15) repeats until the termination criterion is reached. In each iteration, a temporary copy S' of the current solution S is made (line 3). The algorithm then determines the number m of customers to remove by drawing uniformly between $\min\{0.1N, 30\}$ and $min\{0.4N, 60\}$, where N is the total customer count (line 4). Next, one removal operator rem and one insertion operator ins are chosen from their respective sets of removal and insertion operators (RO and IO) via a roulette-wheel selection that favors higher-scoring operators, where scores increase every time an operator improves a solution (lines 5-6). Applying rem to S' removes m customers from S', and ins then reinserts them in feasible positions (lines 7-8). If the repaired solution S' outperforms the global best solution S_{best}, it updates it (lines 9–11). To avoid getting trapped in local optima, the acceptance of a new solution is governed by a Simulated Annealing (SA) criterion. An improved solution S' is always accepted, while a worse solution may be accepted with a probability $e^{\frac{-(f(S')-f(S))}{T}}$ (line 12) [8]. Here, the temperature $T > 0$ decays exponentially at each step and $f(S)$ represent the cost of a solution S (lines 13-14).

In our work, we employ four distinct removal (destroy) operators and a single insertion (repair) operator, whose choice is controlled by adaptive weights. The first operator, Random Removal, selects a subset of customers uniformly at random and removes them from the current solution. This operator promotes broad exploration by disrupting diverse parts of the route.

Algorithm 2: Adaptive Large Neighborhood Search

Input: termination criterion, Initial solution S
Input: Set of destroy operators RO and repair operators IO
1. $S_{best} \leftarrow S$
2. **while** termination criterion not reached **do**
3. $S' \leftarrow S$
4. $m \leftarrow$ random number of customers to remove
5. $rem \leftarrow$ select a random operator from RO
6. $ins \leftarrow$ select a random operator from IO
7. remove m customer from S' using rem
8. insert removed customers into S' using ins
9. **if** $f(S') > f(S_{best})$ **then**
10. $S_{best} \leftarrow S'$
11. **end if**
12. **if** $accept(S', S)$ **then**
13. $S \leftarrow S'$
14. **end if**
15. **end while**

A more structured approach is Cluster-Based Removal, which partitions the customer set into spatially or structurally coherent groups, using k-means clustering or minimum spanning trees, to form clusters $C_1, \dots, C_K$. At each iteration, one or more of these clusters is selected randomly, and all customers within the chosen cluster are removed from the solution.

The third operator, Shaw Removal, which was proposed by Shaw in 1997 [42], employs a relatedness measure described by Eq.(20) [42]:

$$rel_{sh}(i,j) = \lambda d_{ij} + \mu|T_i - T_j| + \nu|q_i - q_j| + \xi(1 - \big||K_i| - |K_j|\big|) \quad (20)$$

Where d_{ij} is the distance between customers i and j, T_i their service start times, q_i their demands, and $|K_i|$ the number of feasible vehicles, with weights $\lambda, \mu, \nu, \xi > 0$. Starting from a random seed customer, the operator iteratively removes its $n-1$ most related neighbors, refining routes by reassembling closely connected customers.

Finally, Worst Removal focuses the search on poorly placed customers by targeting those whose removal yields te most significant immediate cost benefit. For each customer i in the current solution S, we calculate $\Delta_i = f(S) - f(S \setminus \{i\})$ [8], where Δ_i represent the cost benefit of removing customer i, $f(S)$ denotes the objective function value of the current solution S and $f(S \setminus \{i\})$is the objective function value of the solution without customer i. Then, customers are ranked by descending cap delta sub i. and probabilistically remove those that most degrade solution quality, thus focusing repair on the routes needing greatest improvement.

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

For the repair phase, we use a single, powerful insertion operator: the k-regret insertion heuristic [8]. For each removed customer i, it calculates the cost difference between inserting it in its best possible position c_{i1} versus its k-th best position c_{ik}. The regret value is calculated as $\Delta f_i = \sum_{k=2}^{K} c_{ik} - c_{i1}$ [8]. At each step, the customer with the highest regret value is inserted into its best position, balancing immediate cost reduction with future insertion difficulty. In our implementation, K is set to a value between 2 and 5, depending on the number of un-routed customers..

The choice of removal operator is governed by a roulette-wheel mechanism [8], where each operator i has a weight w_i reflecting its past performance. At each iteration, any operator that yields an improved solution is rewarded by increasing its weight by σ, thereby boosting its future selection probability. The selection probability is recalculated as $p_i = w_i / \sum_{j=1}^{m} w_j$, where m is the total number of removal operators. This reinforcement mechanism ensures that more effective operators are chosen more frequently, dynamically steering the search toward the most promising destroy–repair strategies.

C. The proposed Hybrid B-ALNS

In this section, we present the hybrid Bat-Adaptive Large Neighborhood Search (B-ALNS) developed explicitly for the MDHF-VRPTW-SPD. The integration of BA and ALNS follows a master-slave architecture where the Bat Algorithm acts as a high-level controller, guiding the overall search strategy. At the same time, the Adaptive Large Neighborhood Search functions as a powerful engine for generating new candidate solutions through targeted destroy-and-repair operations. This synergistic relationship allows the hybrid B-ALNS to balance global exploration with local intensification.

Because the original Bat Algorithm was designed for continuous optimization, we adapted it to handle the combinatorial nature of the MDHF-VRPTW-SPD. In this approach, each bat corresponds to a feasible routing solution, and bats iteratively move toward the best-known solution to minimize the total routing cost. We retain the fundamental BA parameters (pulse rate r_i, loudness A_i, frequency f_i, and velocity v_i), with adapted interpretations suitable for discrete problems. The pulse rate and loudness maintain their original roles: as bats approach optimal solutions, their loudness decreases while their pulse emission rates increase. To reflect this behavior, the pulse rate update remains unchanged Eq. (19) [7], while we defined the loudness to follow an exponential cooling schedule expressed as Eq. (21) [2]:

$$A_i^{t+1} = A_i^0 \left[\exp\left(-log2\right) \frac{t}{\alpha t_{max}}\right] \quad (21)$$

Where α is the loudness decay factor, a constant typically set to a value close to 1, A_i^0 is the initial loudness, t is the current iteration and t_{max} is the maximum allowed iterations.

Additionally, each bat selects a discrete frequency f_i uniformly within a range $[f_{min}, f_{max}] \subseteq \{1, \dots, n\}$, where n is the number of customers. The frequency f_i determines the portion of the solution that will be destroyed by a removal operator, in order to be repaired later by an insertion operator: a higher value corresponds to larger perturbations (moving toward distant solutions), while lower frequencies facilitate local improvements through minor adjustments. This discrete frequency adaptation allows our hybrid B-ALNS to effectively balance broad exploration with local refinement in solving the MDHF-VRPTW-SPD.

Finally, we adapted the definition of the bat velocity v_i to suit the discrete nature of the problem. In the original BA, velocity depends on the distance between the current position of the bat i at iteration $t-1$ and the best-known position of the swarm. Since solutions to the MDHF-VRPTW-SPD are represented as sequences of routes, we measure this distance using the edge-difference distance, which counts the number of arcs (connections between customers or depots) differ between two solutions.

Thus, we redefine the velocity v_i^t of bat i at iteration t as a random number drawn uniformly from the interval between 2 and the calculated edge-difference distance, as shown in Eq. (22) [2]:

$$v_i^t = Rand(2, EdgeDifference(x_i^{t-1}, x_*)) \quad (22)$$

This formulation ensures the bats' movements correspond meaningfully to their closeness to the best solution.

In the original Bat Algorithm, bats update their positions using continuous Eq. (16) [7], which cannot be directly applied to the combinatorial structure of the MDHF-VRPTW-SPD. Therefore, in our hybrid B-ALNS, bat movement is simulated using the remove-and-insert (R&I) paradigm from Large Neighborhood Search, formulated as in Eq. (23):

$$x_i^t = R\&I(x_i^{t-1}, v_i^t, f_i) \quad (23)$$

Specifically, each bat applies the R&I operators iteratively v_i^t times, each time removing f_i customers using one of the four previously described removal operators and then reinserting them through the k-regret heuristic. After evaluating these candidate solutions, the bat selects the best-performing solution as its new position. This evolutionary approach effectively replaces the directional movements of bats through the search space.

In our proposed Hybrid B-ALNS, all operational constraints defined in the mathematical model are

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

Algorithm 3: Hybrid B-ALNS

Input: objective function $f(x)$, population size n, $iter_{max}$, bat memory list size M, frequency bounds f_{min}/f_{max}, parameters α, γ, removal set $RO = \{rem_1 \dots rem_4\}$, insertion operator INS
Output: global solution x_*
1. initialize population $X = \{x_1, \dots, x_n\}$ via k-regret insertion heuristic
2. **for** each bat i in population **do**
3. evaluate fitness $f(x_i)$; initialize pulse rate r_i^0 and loudness A_i^0
4. initialize the weights of the removal and insertion operators to 1
5. initialize each bat's memory with its initial solution x_i
6. **end for**
7. initialize the global solution x_* as the best initial bat
8. **for** $t = 1 \dots iter_{max}$, **do**
9. **for** each bat i in population **do**
10. adjust the frequency f_i via Eq. (14) [7]
11. update the velocity v_i via Eq. (22) [2]
// Diversification step
12. **for** $k = 1 \dots v_i$ **do**
13. choose removal and insertion operator via Roulette-Wheel mechanism
14. generate new solutions using Eq. (23)
15. **end for**
16. $x_{div} \leftarrow$ best generated solution
17. **if** $f(x_{div}) < f(x_i)$ **then**
18. increase the weights of the removal and insertion operators by σ
19. **end if**
// Bat memory update
20. update bat memory list with x_{div}
// Intensification step
21. **if** $rand() > r_i$ **then**
22. select a random solution x' from the bat's memory list
23. apply a local $R\&I$ move on x' Eq. (23)
24. **if** $f(x') < f(x_i)$ **then** $x_i \leftarrow x'$ **end if**
25. **else**
26. $x_i \leftarrow x_{div}$
27. **end if**
// Acceptance and update
28. **if** $rand() < A_i$ and $f(x_i) < f(x_*)$ **then** $x_* \leftarrow x_i$ **end if**
29. update r_i Eq. (19) [7] and A_i Eq. (21) [2]
30. **end for**
31. **end for**
32. return x_*

treated as hard constraints that cannot be violated. This is enforced directly within the solution generation process rather than through penalty terms in the objective function. Specifically, the destroy and repair operators, particularly the k-regret insertion heuristic, are designed to be feasibility-preserving. During the repair phase, a customer can only be inserted into a route if the move does not violate any constraints, including vehicle capacity, customer time windows, and maximum route duration. Any potential move that would result in an infeasible solution is immediately discarded from consideration.

The detailed workflow of the Hybrid B-ALNS is outlined in Algorithm 3. The process begins by seeding a population of n feasible solutions using the k-regret insertion heuristic, where $k = 2$ (line 1). Each bat's fitness, parameters, and memory are initialized (lines 2 to 6), and the global solution x_* is initialized to the best-performing solution among the initial population (line 7). The main iterative loop (line 8) then proceeds as follows for each bat i : First, the BA component acts as a high-level controller by adjusting its frequency f_i using Eq. (14) (line 10) and then calculates a velocity v_i using Eq. (22) (line 11) to determine the search intensity.

This control is then passed to the ALNS engine for the diversification step (lines 12 to 19), where it executes a loop v_i times to generate a new candidate solution using Eq. (23) each iteration (line 14). The specific destroy-and-repair operators for each move are selected via the adaptive roulette-wheel mechanism, which favors operators that have recently produced high-quality solutions and rewards any operator that yields an improved solution (lines 17 to 19). The best-found candidate from this phase, x_{div}, is added to the bat's elite memory list (line 20), pruning the worst if it exceeds the size of M. This memory serves two purposes: it preserves high-quality solutions from being lost and it provides a pool of elite candidates for the intensification step.

Next, a probabilistic choice determines the final candidate solution for this iteration. With a probability r_i (the bat's pulse rate), the algorithm enters the intensification phase (line 21 to line 27). It selects a random solution x' from the bat's elite memory, and applies Eq. (23) with $v_i = 1$ and $f_i = f_{min}$ to perform a single, focused local move to generate a locally-refined solution x' (line 23). A critical check occurs: this refined solution x' is only adopted if it is superior to the actual solution x_i (line 24). This ensures that intensification is only pursued if it yields a tangible improvement. Otherwise, if the intensification condition is not met (with probability $1 - r_i$), the algorithm forgoes this extra local search. It directly updates x_i by the best solution, x_{div}, from the diversification phase (line 26).

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

Finally, the decision to accept x_i as the global solution x_* is governed entirely by the Bat Algorithm's core acceptance mechanism (line 28). which considers both solution quality and the current loudness A_i, as defined in Eq. (21) [2]. This mechanism allows the algorithm to occasionally accept worse solutions, particularly in the early search stages, which is crucial for escaping local optima. The bat's loudness and pulse rate are then updated using Eq. (19) and Eq. (21) [2] for the next iteration (line 29), ensuring a dynamic balance between exploration and exploitation. This entire process repeats for all bats in the population until the termination criterion, $iter_{max}$, is reached (line 31). At that point, the algorithm concludes and returns the global best solution x_* found across all bats and all iterations (line 32).

This integrated process, where BA provides high-level strategic direction (how many moves to make, whether to intensify or intensify the search) and ALNS provides the tactical execution (performing the moves themselves), ensures a robust and well-coordinated search process, and remains robust across the rich constraints of the MDHF-VRPTW-SPD.

V. Computational experiments

In this section, we describe our computational experiments. First, we describe the characteristics of the MDHF-VRPTW-SPD problem test instance, followed by detailed and extensive computational experiments.

A. The benchmark proposed for the MDHF-VRPTW-SPD:

To the best of our knowledge, no public benchmark simultaneously covers multi-depot, heterogeneous fleet, hard time windows, and simultaneous pickup-and-delivery; existing data sets omit at least one of these dimensions. In this sense, and adhering to the instance-generation guidelines in [43], we extended the multi-depot time-window data of Cordeau et al. [44] to produce a deterministic suite of 20 MDHF-VRPTW-SPD instances spanning 48 to 288 customers served by four to six depots. All customer coordinates, number of depots and their limits D, and time windows $[e_i, l_i]$ were kept intact, while 3 deterministic steps enriched the data, to reflect real-world pharmaceutical-distribution operations:

Demand categories: each customer i is first assigned to one of three classes as in Eq. (24) to a uniform draw $u_i \in [0,1]$:

$$cat_i = \begin{cases} A : u_i < 0.7 \\ B : 0.7 \le u_i < 0.9 \\ C : u_i \ge 0.9 \end{cases} \quad (24)$$

Where category A is delivery and low return, B is delivery and significant return and C is return only.

The original Cordeau delivery demand D_i is retained for Categories A and B, while Category C receives $D_i = 0$. Pickup quantities P_i are then generated by Eq. (25):

$$P_i = \begin{cases} \lceil r_A D_i \rceil : category\ A \\ \lceil r_B D_i \rceil : category\ B \\ rand[1, \min\{3, \lceil 0.15\bar{D} \rceil\}] : category\ C \end{cases} \quad (25)$$

Where $r_A \sim U[0.05,0.20]$, $r_B \sim U[0.20,0.50]$ and $\bar{D}$ is the average delivery demand in the instance. This produces a realistic portfolio in which roughly 70 % of service points receive medications and hand back small quantities of waste or expired drugs, 20 % return a substantial share, and 10 % are pure pickup stops, giving an overall pickup volume of about 25 % of total deliveries, consistent with reported pharmaceutical reverse-logistics volumes.

Service time: To reflect simultaneous handling, the original service duration d_i from Cordeau et al. instances [44] was divided into delivery service time d_i^{del} and pickup service time d_i^{pick} using Eq. (26) and Eq. (27):

$$d_i^{del} = base_i + (d_i + 2base_i)\frac{D_i}{T_i} \quad (26)$$

$$d_i^{pick} = base_i + (d_i - 2base_i)\frac{P_i}{T_i} \quad (27)$$

Where $base_i = min(0.2d_i, 1)$, the total demand $T_i = D_i + P_i$, $D_i > 0$ and $P_i > 0$. Pickup-only stops keep the full original d_i, and if $e_i + d_i^{del} + d_i^{pick} > l_i$, the latest time l_i is shifted just enough to accommodate the total service time and no other time windows are altered.

Four-class heterogeneous fleet: Each depot now offers four vehicle types $k \in \{1,2,3,4\}$ with capacities with capacities Q_k, fixed dispatch costs F_k, and distance-proportional costs α_k, defined in Eq. (28) to Eq. (30).

$$Q_k = \{0.25, 0.50, 0.80, 1.20\}Q^* \quad (28)$$

$$F_k = 60 + 60(k-1) \quad (29)$$

$$\alpha_k = 0.45 + 0.15(k-1) \quad (30)$$

where Q^* is the single-type capacity in the original Cordeau et al. [44] data files. The vehicle capacities were defined to ensure that for any given customer i, the total demand T_i does not exceed the capacity of the smallest available vehicle.

B. Parameter settings and calibration:

To ensure methodological transparency and reproducibility, all parameters for the hybrid B-ALNS, the discrete BA, and the simple LNS are explicitly defined in Table 1. The selection of these parameters

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

Table 1. Parametrization of Hybrid B-ALNS, Discrete BA and LNS.

Parameters	Hybrid B-ALNS	Discrete BA	LNS
Population size	100	100	1
Initialization heuristic	k-regret insertion ($k = 2$)	Best insertion	Best insertion
Maximum iterations	500	500	5000
Update parameters α & β	0.98, 0.98	0.98, 0.98	N/A
Initial loudness A_i^0	$U[0.7,1]$	$U[0.7,1]$	N/A
Initial pulse rate r_i^0	$U[0.05,0.3]$	$U[0.05,0.3]$	N/A
Frequency $[f_{min}, f_{max}]$/destruction rate	$U[0.075,0.4]$	$U[0.075,0.4]$	$U[0.075,0.4]$
Bat memory M	10	N/A	N/A
Successor operator	k-regret insertion ($k = 4$) Random, Cluster, Shaw & Worst ruin	2-opt & 3-opt	Best insertion Random & Radial Ruin
Operator Selector	roulette-wheel ($\sigma = 0.2$)	random	random
Acceptance Criterion	BA Loudness-based (Eq. 20)	BA Loudness-based (Eq. 18)	Simulated Annealing

was guided by a combination of systematic calibration for the most sensitive parameters and standard values from the literature for baseline components.

A calibration procedure was performed to determine the key parameters for the hybrid B-ALNS, focusing on balancing solution quality against computational effort. We applied a two-stage factorial design on the on VRP02-n96 instance test with 96 customers. An initial broad search with a varying population size $N = \{40, 80, 120, 160\}$, maximum iterations $iter_{max} = \{200, 350, 500, 650\}$, and destruction fraction $f_{max} = \{0.05, 0.20, 0.35, 0.50\}$, reveals that solution quality gains saturate once the population exceeds 80 to 120 bats, with negligible gains observed beyond 500 iterations. It also showed that setting the maximum destruction rate f_{max} below 5 % provided too little perturbation, causing the search to stagnate in the local optima, while removing more than 40% resulted to inefficient, near-random search.

Following this, we conducted a finer search within the most promising intervals: population size ranging from 90 to 110, iteration limits from 450 to 550, and destruction rates from 0.07 to 0.40 in increments of 0.025. Multiple configurations were evaluated, and the final setting was selected based on the best cost-to-time performance ratio: a population of 100 bats, a 500 iterations limit, and a destruction rate drawn each iteration from the uniform distribution $U[0.075,0.4]$. The population size retains sufficient diversity to prevent premature convergence yet keeps the largest 288-customer instances within a three-minute computational time. The core BA parameters of our hybrid method, α, γ, initial loudness A_i^0 and the Initial pulse rate r_i^0 were set to values commonly used in the literature [7]. While this calibration identified a robust set of parameters for our benchmark suite, a full-scale sensitivity analysis across all parameters was considered beyond the scope of this study.

For a fair and rigorous comparison, the parameters for the benchmark algorithms were aligned with the calibrated Hybrid B-ALNS where appropriate. The Discrete BA mirrors the core settings derived from our calibration, using the same population size (100), maximum iteration (500), update parameters ($\alpha = \gamma = 0.98$), initial loudness, and initial pulse rate. This ensures that any performance difference is attributable to the algorithmic structure rather than parameter tuning. The single-solution LNS was allotted a comparable search effort of 5000 iterations, with its destruction fraction also drawn from the calibrated range $U[0.07, 0.4]$. The primary termination criterion for all algorithms was a fixed number of maximum iterations, as detailed in Table 1. To ensure a high-quality starting point, all methods were initialized using construction heuristics. The hybrid B-ALNS employs a more advanced k-regret insertion (with k=2) to seed its population, reflecting its sophisticated design. In contrast, the baseline Discrete BA and LNS were initialized with a standard best-insertion heuristic. The elite memory size (M=10) offers a sufficient pool of high-quality solutions for intensification without significant computational overhead, and the roulette-wheel reward (σ=0.2) is a standard value that

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

Table 2. Comparative performance of Hybrid B-ALNS, BA, and LNS across 20 benchmark instances.

Instance name	Hybrid B-ALNS			Discrete BA			LNS		
	Avg Cost	S. dev.	Avg Time (s)	Avg Cost	S. dev.	Avg Time (s)	Avg Cost	S. dev.	Avg Time (s)
VRP01-n48	**1225.22**	8.97	9	1266.42	17.98	2.89	1280.45	34.79	0.85
VRP02-n96	**2034.77**	19.34	34.61	2119.24	19.62	10.38	2153.95	38.23	2.48
VRP03-n144	**3087.72**	31.99	73.91	3162.01	29.69	20.68	3211.3	50.46	4.86
VRP04-n192	**4235.67**	39.58	154.6	4337.99	44.15	52.4	4401.1	66.72	14.09
VRP05-n240	**5488.39**	44.48	211.92	5535.7	45.35	43.97	5614.24	59.49	9.67
VRP06-n288	**6101.87**	54.24	263.85	6229.31	57.04	64.56	6269.31	86.29	13.87
VRP07-n72	**1637.49**	21.02	28.44	1714.33	18.74	8.94	1734.09	49.1	2.22
VRP08-n144	**3217.7**	39.67	104.5	3311.11	25.16	27.78	3389.31	49.3	6.23
VRP09-n216	**4541.33**	34.72	209.15	4652.73	53.4	56.85	4727.26	79.53	11.99
VRP10-n288	**6252.16**	57.48	461.67	6291.37	54.85	190.63	6361.02	89.36	35.92
VRP11-n48	**1097.34**	17.52	18.9	1131.44	16.06	5.8	1156.13	36.32	1.75
VRP12-n96	**2043.88**	24.3	68.94	2084.34	22.74	21.08	2122.45	32.79	4.83
VRP13-n144	**2950.89**	31.83	145.47	2993.42	45.48	42.83	3092.01	60.05	10.02
VRP14-n192	**4038.88**	39.3	259.42	4043.83	25.34	76.9	4127.68	72.84	15.17
VRP15-n240	5202.81	43.8	352.66	**5148.22**	38	87.01	5223.74	53.42	18.95
VRP16-n288	5803.97	38.69	376.07	**5773.41**	41.4	138.3	5891.61	96.68	30.35
VRP17-n72	**1490.27**	12.6	62.27	1530.36	18.34	22.65	1567.17	47.41	5.93
VRP18-n144	**3150.73**	15.58	125.45	3178.2	25.02	33.07	3229.77	60.16	7.09
VRP19-n216	**4326.77**	38.68	245.75	4335.83	44.7	65.67	4383.49	48	13.57
VRP20-n288	6275.22	78.55	446.51	**6232.53**	66.59	108.32	6380.61	88.04	22.44
Average	**3710.15**	34.62	182.65	3753.59	35.48	54.04	3815.83	59.95	11.61

effectively balances operator exploitation and exploration.

The Discrete BA relies on a classic 2-opt/3-opt local search as its successor operator, consistent with its original formulation [2]. Finally, the simple LNS was deliberately designed with a limited set of operators, Random and Radial Ruin with Best Insertion, to serve as a lightweight but effective baseline, highlighting the added value of the hybrid's adaptive, multi-operator framework.

C. Results

All numerical experiments were performed on a computer powered by an Intel Core i5 (12th-generation, 2.5 GHz base clock) and 16 GB of DDR4 RAM. The complete set of 20 MDHF-VRPTW-SPD benchmark instances was evaluated. Furthermore, each instance was subjected to 20 independent runs. As noted in the introduction, we conducted a comprehensive comparison between our proposed hybrid B-ALNS, the discrete Bat Algorithm (BA), and a straightforward Large Neighborhood Search (LNS). We selected discrete BA and simple LNS as benchmarks because they represent two well-established metaheuristic paradigms, swarm-based exploration and ruin-and-recreate intensification, that have proven effective across numerous routing applications. Demonstrating that B-ALNS consistently achieves lower routing costs than these standard approaches would underscore its strength and versatility as a metaheuristic for the complex MDHF-VRPTW-SPD.

Analyzing the solution quality (average cost) from Table 2, our proposed Hybrid B-ALNS consistently demonstrates superior performance. Its mean cost of 3710.15 is 1.15 % lower than that of the discrete BA (3753.59) and 2.78 % below LNS (3815.83). Delving into instance-specific performance, Hybrid B-ALNS obtained better average costs than Discrete BA in 85% of instances (17 out of 20) and outperformed LNS in 100%. These findings on average performance are further corroborated by examining the best solutions for each instance, as presented in Table 3. The hybrid B-ALNS obtained the best solution in 85% of the instances (17 out of 20), whereas Discrete BA tops the remaining three, and LNS never leads. The advantage of B-ALNS comes from pairing wide exploration, via frequency-controlled destroy-and-repair moves, with focused refinement driven by bat memory and adaptive

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

Table 3. Characteristics of the average solutions obtained by the best algorithm for each instance.

Instance name	Avg cost	Vehicles	Route balance CV	Load distribution quality	Algorithm
VRP01-n48	1225.22	9	0.2276	1.89	B-ALNS
VRP02-n96	2034.77	14	0.1390	4.22	B-ALNS
VRP03-n144	3087.72	20	0.1071	3.88	B-ALNS
VRP04-n192	4235.67	27	0.1569	5.10	B-ALNS
VRP05-n240	5488.39	30	0.1870	3.19	B-ALNS
VRP06-n288	6101.87	35	0.1649	3.00	B-ALNS
VRP07-n72	1637.49	13	0.1984	5.50	B-ALNS
VRP08-n144	3217.7	22	0.1468	5.08	B-ALNS
VRP09-n216	4541.33	27	0.1098	3.33	B-ALNS
VRP10-n288	6252.16	39	0.1604	6.48	B-ALNS
VRP11-n48	1097.34	6	0.1293	3.00	B-ALNS
VRP12-n96	2043.88	13	0.1739	3.40	B-ALNS
VRP13-n144	2950.89	17	0.1969	3.18	B-ALNS
VRP14-n192	4038.88	22	0.1792	6.96	B-ALNS
VRP15-n240	5148.22	23	0.1310	3.22	Discrete BA
VRP16-n288	5773.41	27	0.1397	3.51	Discrete BA
VRP17-n72	1490.27	11	0.1147	5.87	B-ALNS
VRP18-n144	3150.73	18	0.2238	3.27	B-ALNS
VRP19-n216	4326.77	23	0.1820	3.48	B-ALNS
VRP20-n288	6232.53	28	0.1786	5.58	Discrete BA

operator scores, letting it avoid the local optima that limit BA and LNS.

Beyond cost minimization, Table 3 also offers insights into the structural quality of these best solutions. The 'Route Balance CV' for solutions found by Hybrid B-ALNS consistently indicates well-balanced routes, with low CV values (e.g., ranging from 0.1071 on VRP03-n144 to 0.2276 on VRP01-n48). For the instances where Discrete BA found the best solution, route balance CV values were also low (e.g., 0.1310 on VRP15-n240). These low CVs suggest an even distribution of workload among vehicles. Furthermore, the Load Distribution Quality metric, representing the standard deviation of vehicle capacity utilization, is notably low for the best solutions. For Hybrid B-ALNS, these values (e.g., 1.89 for VRP01-n48 to 6.96 for VRP14-n192) and for Discrete BA (e.g., 3.22 for VRP15-n240 to 5.58 for VRP20-n288) fall well within the 0-10 percentage point range, signifying a very balanced fleet distribution.

To provide deeper insight into the algorithm's learning process and internal stability, and to supplement the average performance data in the tables, Fig. 2, Fig. 3, and Fig. 4 illustrate the convergence rate of the B-ALNS population for the best run on small, medium, and large-scale instances. Several key performance characteristics are evident across the different scales. First, the algorithm demonstrates rapid convergence. For the small-scale instance VRP01-n48 (Fig. 2), the best solution cost of 1203.97 was found in just 5.52 seconds, achieving a total improvement of 20.3%. The medium-scale VRP03-n144 (Fig. 3) shows a steep initial improvement within the first 120 iterations, ultimately finding its best solution of 3042.71 after 65.04 seconds of computation, securing a total cost reduction of 28.57%. This pattern of substantial early gains followed by continued refinement is most pronounced in the large-scale instance VRP06-n288 (Fig. 4), which achieved a 29.39% improvement, finding its best solution of 6031.4 after 253.3 seconds. This confirms that the algorithm efficiently explores the solution space, regardless of problem size. Second, the plots visually confirm the algorithm's strong population stability. For the small and medium instances (Fig. 2 and Fig. 3), the gap between the best (blue line), average (green line), and worst (red line) solutions in the population narrows dramatically. It becomes nearly indistinguishable after the initial convergence phase. This demonstrates that the entire population is effectively guided toward a very focused, high-quality region of the search space. While a small gap persists in the more complex large-scale instance (Fig. 4), the population still converges to a tight band, showcasing a high degree of stability even

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

under challenging conditions. This collective convergence is responsible for the low standard deviation of costs reported across multiple runs in Table 2.

Finally, the algorithm successfully avoids premature stagnation. Even after the initial rapid descent, minor, step-wise improvements are visible in the later iterations of all three plots. This is particularly noticeable in the large-scale instance (Fig. 4) between iterations 200 and 500, showcasing the effectiveness of the adaptive destroy-and-repair operators in continuously refining complex solutions long after simpler heuristics might have stalled.

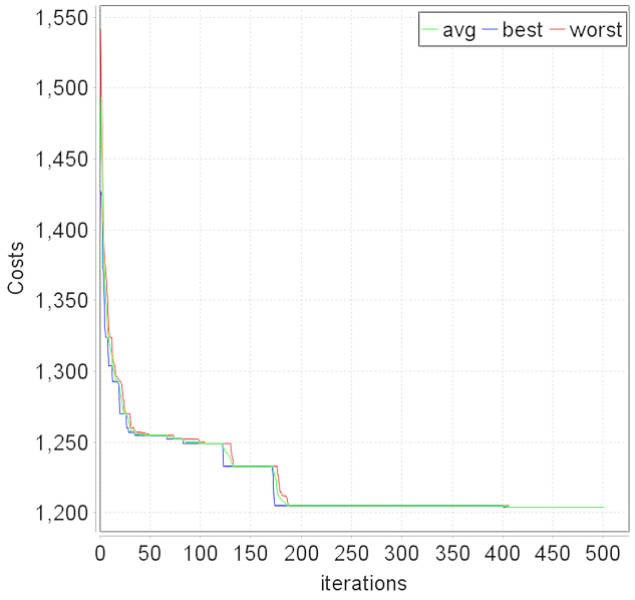

Fig. 2. Convergence rate of the Hybrid B-ALNS for the best run on VRP01-n48 (Small scale).

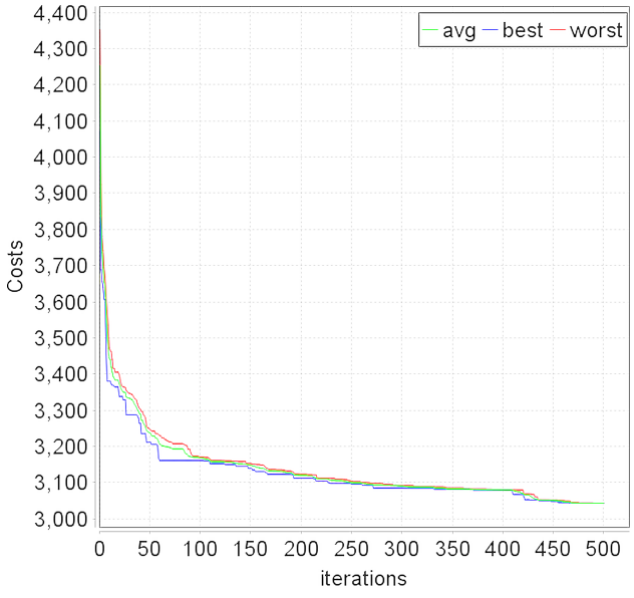

Fig. 3. Convergence rate of the Hybrid B-ALNS for the best run on VRP03-n144 (Medium scale).

In terms of computational effort (average time in seconds), the results in Table 2 highlight a classic trade-off between solution quality and runtime. The simple LNS is the quickest, averaging 11.61 seconds, followed by the Discrete BA at 54.04 seconds. The Hybrid B-ALNS, while delivering consistently superior solutions, required the most computation time, averaging 182.65 seconds per run. This increased effort is a direct consequence of its more sophisticated

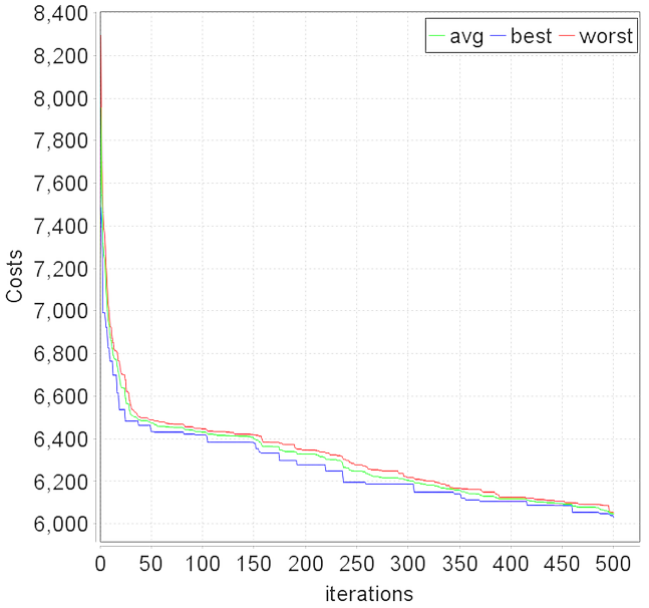

Fig. 4. Convergence rate of the Hybrid B-ALNS for the best run on VRP06-n288 (Large scale).

search mechanics, particularly the k-regret insertion heuristic. Regarding the algorithms' robustness and consistency, reflected by the standard deviation (S. dev.) of the solution costs in Table 2, Hybrid B-ALNS demonstrates the highest consistency, with the lowest value of 34.617. Discrete BA also shows good robustness, with a value of 35.4825. LNS is the quickest method, but its solution quality varies the most.

Because no public data set combines all four features of our full model (MDHF-VRPTW-SPD), we benchmarked the hybrid algorithm on the Cordeau MDVRPTW instances, which is the closest subset of the problem, to compare our approach with leading heuristics from the literature. Table 4 summarizes the best costs from ten independent B-ALNS runs alongside Firefly Algorithm (FA) [30], Tabu Search (TS)

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

Table 4. Best routing cost of 10 independent Hybrid B-ALNS runs, against FA, ACO and TS on MD-VRPTW instances.

Instance name	Hybrid B-ALNS			Firefly Algorithm [30]	Ant Colony Optimization [28]	Tabu Search [6]	ALNS [45]
	Best	S. dev.	Avg Time (s)	Best	Best	Best	Best
pr01	1074.12	1078.61	6.79	1083.98	1086.85	1083.98	1074.12
pr02	1734.61	1766.24	24.76	1763.02	1768.96	1763.07	1762.21
pr03	2400.41	2414.87	51.5	2408.43	2655.38	2408.42	2373.65
pr04	2786.08	2823.67	72.1	2958.21	3605.49	2958.23	2836.59
pr05	3010.95	3031.39	73.38	3134.04	3785.49	3134.04	2968.81
pr06	3547.91	3575.78	111.35	3904.16	4356.26	3904.07	3620.71
pr07	1415.64	1420.33	11.62	1423.33	1464.67	1423.35	1418.22
pr08	2085.97	2098.47	40.26	2150.22	2446.31	2150.22	2096.73
pr09	2741.86	2765.52	153.23	2831.95	3467.28	2833.80	2730.86
pr10	3489.99	3527.11	271.48	3714.77	4241.06	3717.22	3498.23

[6], Ant Colony Optimization (ACO) [28] and the stand-alone ALNS [45].

The hybrid achieves the lowest cost on every instance when compared with FA, TS, and ACO. Its advantage over ACO ranges from 1.2% on the easiest example to 22.7% on the most difficult. Regarding FA and TS, the gain reaches 9.1% on instance pr06. Compared to the ALNS, our hybrid B-ALNS is superior or equal in six cases, has ties in pr01, and slightly outperforms pr03, pr05, and pr09. On average, our approach reduces cost by 0.33% compared with stand-alone ALNS. The adaptive destroy-repair layer, especially the worst-removal and regret-insertion operators, refines routes that the single-solution ALNS sometimes leaves sub-optimal, while the frequency-guided bat population prevents the search stagnation observed in ACO for depot-rich networks. Solution times remain within seven to 272 seconds for problems with 288 customers, which is fast enough for daily tactical planning.

D. Statistical analysis of the results:

To formally validate that the performance differences observed in Table 2 are not due to random variation, we conducted a non-parametric statistical analysis. First, an omnibus Friedman test was applied to the best costs obtained by the three algorithms across the 20 benchmark instances. The resulting statistic yielded a Friedman statistic of 17.2 with a p-value of 0.0002. This p-value, being substantially less than the conventional significance level of 0.05, indicates statistically significant differences in the performance of at least one algorithm compared to the others.

To identify which specific pairs of algorithms differ, we first applied Nemenyi's post-hoc test, with the results shown in Table 5. The test identified a significant difference between the Hybrid B-ALNS and LNS ($p = 0.011$). Although the difference between Hybrid B-ALNS and Discrete BA (p = 0.2209) does not cross the 0.05 threshold, B-ALNS still shows the lowest mean cost in 17 of 20 cases.

For a more detailed and sensitive pairwise comparison, we conducted a two-tailed Wilcoxon signed-rank test, using Hybrid B-ALNS as the reference algorithm. Table 6 provides a comprehensive, instance-by-instance summary of this comparison. The "+/=/-" notation indicates instances where the reference algorithm (Hybrid B-ALNS) performed better, equal to, or worse than the competing algorithm. The "Avg. % Diff" column quantifies the average percentage improvement of B-ALNS over the other methods (positive when better, negative when worse). As the table shows, B-ALNS outperformed the Discrete BA on 15 out of 20 instances (with 3 ties) for an average cost reduction of 1.71%, and it outperformed LNS on 19 out of 20 cases (with 1 tie) for an average decrease of 3.46%. We aggregated the

Table 5. Nemenyi Post-hoc Test (p-values)

Algorithm	Hybrid B-ALNS	Discrete BA	LNS
Hybrid B-ALNS	-	0.2209	0.011
Discrete BA	0.2209	-	0.7907
LNS	0.011	0.7907	-

results using the R statistical computing environment to confirm that these observed trends are statistically significant. The test comparing Hybrid B-ALNS to Discrete BA yielded a highly significant result ($V = 28$, $Z = -2.875$ *and* $p = 0.0027$), with a 95% confidence

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

Table 6. pairwise Wilcoxon signed-rank analyses

Instance	Hybrid B-ALNS	Discrete BA	LNS
VRP01-n48	1.23E+03±8.97E+00	1.27E+03±1.80E+01+ (-3.4%)	1.28E+03±3.48E+01+ (-4.5%)
VRP02-n96	2.03E+03±1.93E+01	2.12E+03±1.96E+01+ (-4.2%)	2.15E+03±3.82E+01+ (-5.9%)
VRP03-n144	3.09E+03±3.20E+01	3.16E+03±2.97E+01+ (-2.4%)	3.21E+03±5.05E+01+ (-4.0%)
VRP04-n192	4.24E+03±3.96E+01	4.34E+03±4.42E+01+ (-2.4%)	4.40E+03±6.67E+01+ (-3.9%)
VRP05-n240	5.49E+03±4.45E+01	5.54E+03±4.53E+01+ (-0.9%)	5.61E+03±5.95E+01+ (-2.3%)
VRP06-n288	6.10E+03±5.42E+01	6.23E+03±5.70E+01+ (-2.1%)	6.27E+03±8.63E+01+ (-2.7%)
VRP07-n72	1.64E+03±2.10E+01	1.71E+03±1.87E+01+ (-4.7%)	1.73E+03±4.91E+01+ (-5.9%)
VRP08-n144	3.22E+03±3.97E+01	3.31E+03±2.52E+01+ (-2.9%)	3.39E+03±4.93E+01+ (-5.3%)
VRP09-n216	4.54E+03±3.47E+01	4.65E+03±5.34E+01+ (-2.5%)	4.73E+03±7.95E+01+ (-4.1%)
VRP10-n288	6.25E+03±5.75E+01	6.29E+03±5.48E+01+ (-0.6%)	6.36E+03±8.94E+01+ (-1.7%)
VRP11-n48	1.10E+03±1.75E+01	1.13E+03±1.61E+01+ (-3.1%)	1.16E+03±3.63E+01+ (-5.4%)
VRP12-n96	2.04E+03±2.43E+01	2.08E+03±2.27E+01+ (-2.0%)	2.12E+03±3.28E+01+ (-3.8%)
VRP13-n144	2.95E+03±3.18E+01	2.99E+03±4.55E+01+ (-1.4%)	3.09E+03±6.00E+01+ (-4.8%)
VRP14-n192	4.04E+03±3.93E+01	4.04E+03±2.53E+01= (-0.1%)	4.13E+03±7.28E+01+ (-2.2%)
VRP15-n240	5.20E+03±4.38E+01	5.15E+03±3.80E+01- (+1.0%)	5.22E+03±5.34E+01= (-0.4%)
VRP16-n288	5.80E+03±3.87E+01	5.77E+03±4.14E+01- (+0.5%)	5.89E+03±9.67E+01+ (-1.5%)
VRP17-n72	1.49E+03±1.26E+01	1.53E+03±1.83E+01+ (-2.7%)	1.57E+03±4.74E+01+ (-5.2%)
VRP18-n144	3.15E+03±1.56E+01	3.18E+03±2.50E+01+ (-0.9%)	3.23E+03±6.02E+01+ (-2.5%)
VRP19-n216	4.33E+03±3.87E+01	4.34E+03±4.47E+01= (-0.2%)	4.38E+03±4.80E+01+ (-1.3%)
VRP20-n288	6.28E+03±7.85E+01	6.23E+03±6.66E+01= (+0.7%)	6.38E+03±8.80E+01+ (-1.7%)
+/=/−		15/3/2	19/1/0
Avg. % Diff		-1.71%	-3.46%

interval for the median cost difference of $[-68.21, -22.52]$.

This formally proves the superiority of the hybrid approach. The comparison with LNS was even more statistically decisive $(V = 0,\ Z = -3.920\ and\ p < 0.0001)$, with a 95% confidence interval of $[-126.54, -82.85]$. These aggregated statistics provide the formal evidence to support the instance-level results in Table 6, confirming that the Hybrid B-ALNS is significantly more effective than both baseline algorithms.

E. Discussion

The superior cost-minimization achieved by the hybrid B-ALNS stems from how its adaptive removal and insertion phase complements the frequency-guided exploration of the Bat algorithm core. During search, the roulette-wheel mechanism continually re-weights the four removal operators according to their recent contribution to improvement, allowing the algorithm to pivot toward the neighborhood structure most likely to unlock further savings. Optimization logs show that worst-removal dominates, succeeding on 77.9 % of the calls that led to an accepted solution; by ejecting the highest-cost customers, it systematically dismantles expensive route segments that the discrete BA's 2-opt/3-opt moves cannot access. Random removal (29.28 %) provides the diversification needed to escape medium-sized basins, while Shaw removal (16.49 %) and the more selective cluster-based heuristic (6.33 %) target geographically or temporally cohesive subsets, refining route balance once the solution is near a local optimum. Crucially, every removal phase is paired with regret-k insertion, which evaluates multiple alternative positions for each reinserted node and chooses the placement that would cause the greatest future penalty if postponed. This explains why the hybrid costs 1.15% and 2.78 % greater cost reduction compared tothe other methods.

Analyzing the results in Table 2 across instance sizes reveals a key strength of the hybrid approach. While the runtimes for all methods scale with problem size, the solution quality gap between B-ALNS and the baseline methods widens on larger instances. For example, the average cost advantage of B-ALNS over LNS nearly triples from approximately 55 units on the VRP01-n48 instance to over 167 units on the VRP06-n288 instance. A similar, though less pronounced, trend is observed against the Discrete BA, where the cost gap increases from 41 units to nearly 128 units. This trend suggests that the sophisticated search mechanics of B-ALNS provide a compounding

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

advantage as the solution space expands and better scalability for complex, large-scale problems.

A critical analysis also reveals scenarios where B-ALNS was not superior. The Discrete BA found better solutions on three large-scale instances (VRP15, VRP16, and VRP20) (Table 3). This suggests that for certain large-scale instances, the powerful but disruptive destroy-and-repair operators of ALNS may occasionally break apart a delicate, near-optimal structure. In these cases, the more conservative 2-opt/3-opt refinement of the BA proves more effective for final optimization.

To contextualize the results presented in Table 4, we benchmarked the performance of all five algorithms against the Best Known Solutions (BKS) for the Cordeau et al. instances, as reported in the NEO research group's public repository (neo.lcc.uma.es). Table 7 summarizes this analysis, showing the average percentage deviation (gap) from the BKS and the number of new BKS found by each method. A negative gap indicates an improvement over the previously published BKS. Our proposed B-ALNS algorithm not only achieved the best overall performance with an average improvement of 3.46% over the BKS, but it also discovered better alternative solutions for all 10 benchmark instances. This performance surpasses that of the standalone ALNS [45], which also proved highly effective but did not improve upon one instance. In contrast, other methods like Tabu Search [6] and Ant Colony Optimization [28] were, on average, unable to match the quality of the BKS.

The trade-off between solution quality and computational effort is well-justified within the practical context of healthcare logistics. The VRP is typically a tactical planning problem, where routes are determined hours or even a day in advance, not in real-time. In such a planning environment, an optimization runtime of a few minutes is a negligible investment when weighed against the benefits of a superior solution. The resulting 1% to 3% reduction in operational costs, compounded daily, may lead to substantial annual savings. Furthermore, the higher consistency and improved balance of the routes generated by B-ALNS contribute to more reliable service, which is a critical priority in medical logistics.

While a formal complexity analysis of the hybrid B-ALNS is intricate, its computational cost is primarily driven by the most intensive phase: the k-regret insertion heuristic within the ALNS engine. The complexity of a single remove and insertion operation is influenced by the number of customers to be removed m and the total number of customers N. For each of the m removed customers, the k-regret heuristic must evaluate potential insertion positions across all routes, leading to significant cost calculations. Consequently, the algorithm's runtime scales polynomially with the problem size N. This is empirically confirmed in Table 2, where the average runtime increases from 9 seconds for 48 customers to over 7 minutes for the largest 288-customer instances. The main parameters influencing this scalability are the population size n, the maximum number of iterations $iter_{max}$, and the destruction fraction f_{min}/f_{max}, which determines m. No specific optimization techniques to reduce runtime, such as maintaining a memory of previously evaluated move costs, were employed in this study, leaving this as a key avenue for future work to enhance performance for large-scale or real-time applications.

The findings of this research offer several important implications. From a practical standpoint, the validated B-ALNS algorithm provides a powerful decision-support tool for pharmaceutical distributors. The demonstrated ability to consistently reduce routing costs can lead to substantial annual operational savings, while the generation of well-balanced and robust routes enhances service reliability, a critical priority in healthcare logistics. By efficiently integrating simultaneous pickups and deliveries, the model also supports regulatory compliance and improves the efficiency of reverse logistics chains. Theoretically, this study confirms that hybridizing a population-based metaheuristic (BA) with a powerful local search framework (ALNS) is a highly effective strategy for rich, multi-constraint VRPs, offering a valuable template for future algorithm designs. Furthermore, developing a new suite of 20 benchmark instances for the MDHF-VRPTW-SPD provides a dataset for the research community, enabling rigorous comparison of future

Table 7. Average Gap from BKS for Hybrid B-ALNS and State-of-the-Art Heuristics on Cordeau et al. Instances.

Author	Methods	Avg. Gap from BKS (%)	Improved BKS
R. Yesodha et. al. [30]	Firefly Algorithm	-0.01%	3/10
J. Gao et. al. [28]	Ant Colony Optimization	0.00%	0/10
M. Polacek et. Al. [6]	Tabu Search	11.82%	0/10
S. Wang et. al. [45]	ALNS	-3.14%	9/10
Proposed method	Hybrid B-ALNS	-3.46%	10/10

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

algorithms targeting complex healthcare logistics challenges.

Despite these contributions, the study has several limitations that should guide future work. First, the MILP and meta-heuristic assume deterministic travel times, service durations, and demand volumes. stochastic traffic delays, variable loading times, and emergency order spikes common in healthcare logistics are ignored and may erode solution robustness. Second, the fleet typology is restricted to four generic vehicle classes and omits driver-hour and rest-break regulations. Third, all computational experiments rely on synthetic instances adapted from Cordeau's benchmark; they lack the geocoding noise, asymmetric street networks, and recurrent time-window violations observed in real dispatch scenarios, limiting the realism. Finally, the current Java prototype runs serially. It therefore cannot exploit GPU acceleration or distributed parallelism required to shorten optimization times to the sub-minute level demanded by large-scale, real-time operations that directly impact patient safety.

VI. Conclusion

The primary purpose of this study was to address the Multi-Depot Heterogeneous Fleet Vehicle Routing Problem with Time Windows and Simultaneous Pickup and Delivery (MDHF-VRPTW-SPD), a challenge central to modern healthcare logistics. We aimed to model this complex problem formally and develop and validate a novel hybrid metaheuristic, B-ALNS, capable of producing high-quality, robust solutions.

We developed a discrete Bat Algorithm hybridized with an Adaptive Large Neighborhood Search to achieve this. Computational experiments on a new suite of 20 benchmark instances demonstrated the clear superiority of the proposed B-ALNS. On average, the algorithm achieved a cost reduction of 1.15% compared to a standalone discrete Bat Algorithm and 2.78% compared to a simple LNS. Wilcoxon signed-rank tests confirmed These performance gains were statistically significant ($p < 0.003$ vs. BA; $p < 0.0001$ vs. LNS). Furthermore, the B-ALNS found the best-known solution in 85% of the test cases and consistently produced routes with superior balance and lower cost variance, all within computational times suitable for tactical planning. Building on the promising results of this study, future research will advance along two primary tracks.

Building on these promising results of this study, future research will advance along two primary tracks. The first track will focus on algorithmic enhancement and scalability. To address the main computational bottleneck, we will investigate parallelizing the k-regret insertion phase and implementing caching mechanisms for move evaluations. This could significantly reduce runtimes for large-scale applications. The second track involves increasing model realism by extending the current deterministic model to incorporate stochastic elements, such as variable travel times and uncertain demands, as well as more complex operational constraints like multi-compartment vehicles. These advancements will bring the model closer to the dynamic, high-stakes environment of daily healthcare logistics operations.

Acknowledgment

The authors are grateful for the insightful feedback and constructive suggestions provided by the editor and the reviewers, which have greatly enhanced the clarity and rigor of this manuscript. We also extend our thanks to our colleagues at the ISIMA Laboratory for their continued support.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The test instances developed for MDHF-VRPTW-SPD are available under request to the corresponding author of this paper.

Author Contribution

All authors contributed equally to the conception of the research problem and the design of the study. Taha Anass led the mathematical formulation of the MDHF-VRPTW-SPD. B.C. and D.E. developed and coded the hybrid Bat–ALNS metaheuristic and executed the computational experiments. All authors of this manuscript wrote and then revised previous versions of the manuscript. All the authors read and approved the final manuscript.

Declarations

Ethical Approval

This research paper contains no studies with human participants or animals performed by any authors. Institutional ethical approval was not required as a computational study based on publicly available and generated benchmark data.

Consent for Publication Participants.

Consent for publication was given by all participants

Competing Interests

The authors declare that they have no competing interests.

References

[1] G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," Manage Sci, vol. 6, no. 1, pp. 80–91, 1959.

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

[2] A. Taha, M. Hachimi, and A. Moudden, "Adapted bat algorithm for capacitated vehicle routing problem," International Review on Computers and Software, vol. 10, no. 6, pp. 610–619, 2015.

[3] B. Kallehauge, J. Larsen, O. B. G. Madsen, and M. M. Solomon, "Vehicle Routing Problem with Time Windows," Column Generation, pp. 67–98, 2005.

[4] J. Jiang, K. M. Ng, K. L. Poh, and K. M. Teo, "Vehicle routing problem with a heterogeneous fleet and time windows," Expert Syst Appl, vol. 41, no. 8, pp. 3748–3760, Jun. 2014.

[5] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems," Journal fur Betriebswirtschaft, vol. 58, no. 1, pp. 21–51, 2008.

[6] M. Polacek, R. F. Hartl, K. Doerner, and M. Reimann, "A Variable Neighborhood Search for the Multi Depot Vehicle Routing Problem with Time Windows," Journal of Heuristics, vol. 10, no. 6, pp. 613–627, Dec. 2004.

[7] X. Yang, "A New Metaheuristic Bat-Inspired Algorithm," Nicso 2010, pp. 65–74, 2010.

[8] D. Pisinger and S. Ropke, "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows," Transportation Science, vol. 40, no. August, pp. 455–472, 2006.

[9] J. Wang, Y. Zhou, Y. Wang, J. Zhang, C. L. P. Chen, and Z. Zheng, "Multiobjective Vehicle Routing Problems with Simultaneous Delivery and Pickup and Time Windows: Formulation, Instances, and Algorithms," IEEE Trans Cybern, vol. 46, no. 3, pp. 582–594, 2016.

[10] E. Köse, A. Kokmazer, D. Danı¸sment Vural, G. Gül, G. Glu, and P. P. Pınar¸savlı, "Simultaneous pickup and delivery model suggestion for personnel transportation in COVID-19 pandemic conditions," vol. 33, no. 4, 2023.

[11] A. Kantasa-Ard, T. Chargui, A. Bekrar, A. Aitelcadi, and Y. Sallez, "Dynamic sustainable multiple-depot vehicle routing problem with simultaneous pickup and delivery in the context of the physical internet," Journal of International Logistics and Trade, vol. 21, no. 3, pp. 110–134, 2023.

[12] V. F. Yu, H. Susanto, Y. H. Yeh, S. W. Lin, and Y. T. Huang, "The Vehicle Routing Problem with Simultaneous Pickup and Delivery and Parcel Lockers," Mathematics 2022, Vol. 10, Page 920, vol. 10, no. 6, p. 920, Mar. 2022.

[13] J. Ruiz-Meza, A. Torregroza Espinoza, D. Mejía-Ayala, and G. Mendoza Ortega, "Vehicle routing problem with simultaneous pickup and delivery for milk collection in Galeras, Sucre", Sept 2020.

[14] G. Liu, J. Hu, Y. Yang, S. Xia, and M. K. Lim, "Vehicle routing problem in cold Chain logistics: A joint distribution model with carbon trading mechanisms," Resour Conserv Recycl, vol. 156, p. 104715, May 2020.

[15] X. Xu et al., "Optimization of reliable vehicle routing problem for medical waste collection with time windows in stochastic transportation networks," Engineering Optimization, Aug. 2024.

[16] F. Ghiasvand Ghiasi, M. Yazdani, B. Vahdani, and A. Kazemi, "Multi-depot home health care routing and scheduling problem with multimodal transportation: Mathematical model and solution methods," Scientia Iranica, vol. 31, no. 11, pp. 825–846, Jun. 2024.

[17] R. Liu, X. Xie, V. Augusto, and C. Rodriguez, "Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care," Eur J Oper Res, vol. 230, no. 3, pp. 475–486, Nov. 2013.

[18] N. Ouertani, H. Ben-Romdhane, I. Nouaouri, H. Allaoui, and S. Krichen, "A multi-compartment VRP model for the health care waste transportation problem," J Comput Sci, vol. 72, p. 102104, Sep. 2023.

[19] Z. Lin, X. Xu, E. Demir, and G. Laporte, "Optimizing task assignment and routing operations with a heterogeneous fleet of unmanned aerial vehicles for emergency healthcare services," Comput Oper Res, vol. 174, p. 106890, Feb. 2025.

[20] C. P. Tomazella and M. S. Nagano, "A comprehensive review of Branch-and-Bound algorithms: Guidelines and directions for further research on the flowshop scheduling problem," Expert Syst Appl, vol. 158, p. 113556, Nov. 2020.

[21] A. Bettinelli, A. Ceselli, and G. Righini, "A branch-and-price algorithm for the multi-depot heterogeneous-fleet pickup and delivery problem with soft time windows," Math Program Comput, vol. 6, no. 2, pp. 171–197, 2014.

[22] E. A. G. de Souza, M. S. Nagano, and G. A. Rolim, "Dynamic Programming algorithms and their applications in machine scheduling: A review," Expert Syst Appl, vol. 190, p. 116180, Mar. 2022.

[23] D. Management, H. E. C. Montréal, Q. Ht, and G. Laporte, "Fifty Years of Vehicle Routing," vol. 43, no. 4, pp. 408–416, 2009.

[24] B. Herdianto and Komarudin, "Guided Clarke and Wright Algorithm to Solve Large Scale of

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

Capacitated Vehicle Routing Problem," 2021 IEEE 8th International Conference on Industrial Engineering and Applications, ICIEA 2021, pp. 449–453, Apr. 2021.

[25] C. Hertrich, P. Hungerländer, and C. Truden, "Sweep Algorithms for the Capacitated Vehicle Routing Problem with Structured Time Windows," pp. 127–133, 2019.

[26] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," Multimed Tools Appl, vol. 80, no. 5, pp. 8091–8126, Feb. 2021.

[27] A. Singh and S. Kumar, "Differential evolution: An overview," Advances in Intelligent Systems and Computing, vol. 436, pp. 209–217, 2016.

[28] J. Gao, F. Gu, P. Hu, Y. Xie, and B. Yao, "Automobile chain maintenance parts delivery problem using an improved ant colony algorithm," Advances in Mechanical Engineering, vol. 8, no. 9, pp. 1–13, Sep. 2016.

[29] T. J. Ai and V. Kachitvichyanukul, "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery," Comput Oper Res, vol. 36, no. 5, pp. 1693–1702, 2009.

[30] R. Yesodha and T. Amudha, "A bio-inspired approach: Firefly algorithm for Multi-Depot Vehicle Routing Problem with Time Windows," Comput Commun, vol. 190, pp. 48–56, Jun. 2022.

[31] W. C. Chiang and R. A. Russell, "Simulated annealing metaheuristics for the vehicle routing problem with time windows," Ann Oper Res, vol. 63, no. 1, pp. 3–27, 1996.

[32] E. Taillard, P. Badeau, M. Gendreau, F. Geurtin, and J. Y. Potvin, "A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows," Transportation Science, vol. 31, no. December 2016, pp. 170–186, 1997.

[33] D. Pisinger and S. Ropke, "Large neighborhood search," Handbook of Metaheuristics, vol. 146, pp. 1–22, 2010.

[34] C. Blum, "Hybrid metaheuristics in combinatorial optimization: A survey," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7505 LNCS, no. 6, pp. 1–10, 2012.

[35] S. Naccache, J. F. Côté, and L. C. Coelho, "The multi-pickup and delivery problem with time windows," Eur J Oper Res, vol. 269, no. 1, pp. 353–362, Aug. 2018.

[36] D. Sacramento, D. Pisinger, and S. Ropke, "An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones," Transp Res Part C Emerg Technol, vol. 102, pp. 289–315, May 2019.

[37] E. Mofid-Nakhaee and F. Barzinpour, "A multi-compartment capacitated arc routing problem with intermediate facilities for solid waste collection using hybrid adaptive large neighborhood search and whale algorithm," Waste Manag Res, vol. 37, no. 1, pp. 38–47, Jan. 2019.

[38] C. Chen, E. Demir, and Y. Huang, "An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots," Eur J Oper Res, vol. 294, no. 3, pp. 1164–1180, Nov. 2021.

[39] E. Osaba, X. S. Yang, I. Fister, J. Del Ser, P. Lopez-Garcia, and A. J. Vazquez-Pardavila, "A Discrete and Improved Bat Algorithm for solving a medical goods distribution problem with pharmacological waste collection," Swarm Evol Comput, vol. 44, pp. 273–286, Feb. 2019.

[40] E. Osaba, R. Carballedo, X. S. Yang, I. Fister, P. Lopez-Garcia, and J. Del Ser, "On Efficiently Solving the Vehicle Routing Problem with Time Windows Using the Bat Algorithm with Random Reinsertion Operators," Studies in Computational Intelligence, vol. 744, pp. 69–89, 2018.

[41] A. Taha, M. Hachimi, and A. Moudden, "A discrete Bat Algorithm for the vehicle routing problem with time windows," 2017 International Colloquium on Logistics and Supply Chain Management: Competitiveness and Innovation in Automobile and Aeronautics Industries, LOGISTIQUA 2017, pp. 65–70, Jun. 2017.

[42] P. Shaw, "A new local search algorithm providing high quality solutions to vehicle routing problems," APES Group, Dept of Computer Science, University of …, pp. 1–12, 1997.

[43] E. Osaba, R. Carballedo, F. Diaz, E. Onieva, A. D. Masegosa, and A. Perallos, "Good practice proposal for the implementation, presentation, and comparison of metaheuristics for solving routing problems," Neurocomputing, vol. 271, pp. 2–8, Jan. 2018.

[44] J. F. Cordeau, G. Laporte, and A. Mercier, "A unified tabu search heuristic for vehicle routing problems with time windows," Journal of the Operational Research Society, vol. 52, no. 8, pp. 928–936, 2001.

[45] S. Wang, W. Sun, and M. Huang, "An adaptive large neighborhood search for the multi-depot dynamic vehicle routing problem with time

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

windows," Comput Ind Eng, vol. 191, p. 110122, May 2024.

[46] C. E. Miller, R. A. Zemlin, and A. W. Tucker, "Integer Programming Formulation of Traveling Salesman Problems," Journal of the ACM (JACM), vol. 7, no. 4, pp. 326–329, Oct. 1960.

.

Author Biography

Dr. Anass TAHA, Ing., PhD, is an Associate Professor in the Department of Mathematics and Informatics at the Polydisciplinary Faculty of Taroudant (FPT), part of Université Ibn Zohr (UIZ) in Agadir, Morocco. He is an active member of the ISIMA Laboratory (Computer Systems Engineering, Mathematics and Applications). His primary research interests lie at the intersection of operational research, metaheuristic optimization, and data science. He focuses on developing advanced computational methods to solve complex combinatorial optimization problems, particularly within supply chain management and industrial scheduling. Dr. Taha's work is centered on the design and implementation of novel swarm intelligence and hybrid algorithms, with specific expertise in vehicle routing problems and job-shop scheduling. He can be contacted via email at an.taha@uiz.ac.ma.

Mr. Elatar Said is a PhD candidate in Operations Research and Computer Science at Chouaib Doukkali University, Faculty of Science, in El Jadida, Morocco. He holds a engineering degree in industrial engineering, which he earned from the National School of Applied Sciences in Safi in 2013. His current doctoral research is at the intersection of supply chain management and advanced optimization techniques. Mr. Said specializes in applying metaheuristics, with a particular emphasis on swarm intelligence algorithms, to address combinatorial optimization problems. His work also explores the integration of the Internet of Things (IoT) to develop dynamic decision-making models, aiming to enhance the efficiency and responsiveness of modern logistical systems. He can be contacted via email at elatar.said@gmail.com.

Dr. Mohamed Salim El Bazzi, PhD, is an Associate Professor in the Department of Mathematics and Informatics at the Polydisciplinary Faculty of Taroudant (FPT), part of Université Ibn Zohr (UIZ) in Agadir, Morocco. He is an active member of the IRF-SIC Laboratory (Image and Pattern Recognition, Intelligent and Communicating Systems). His primary research interests lie at the intersection of machine learning, text mining, and Natural Language Processing (NLP). His specific research focuses on developing robust algorithms for text indexing, automated classification, document clustering, opinion mining, and sophisticated feature extraction from unstructured data. Dr. El Bazzi works to advance the theoretical foundations of these fields while also building practical applications that transform raw text into structured, actionable knowledge for various domains. He can be contacted via email at m.elbazzi@uiz.ac.ma.

Dr. Ait Ider Abdelouahed, PhD, is an Associate Professor of Computer Science at the Polydisciplinary Faculty of Taroudant, Ibnou Zohr University, Morocco. He earned his Ph.D. in Computer Science in 2018 from the Faculty of Sciences and Technologies at Sultan Moulay Slimane University, Beni-Mellal. As a key member of the ISIMA Laboratory, his work is situated at the forefront of modern computational intelligence. He specializes in applying Artificial Intelligence, Machine Learning, and particularly Deep Learning techniques to solve complex challenges in Data Science and Computer Vision. His research is pivotal in developing sophisticated models for tasks such as automated image analysis in the medical field and pattern recognition. He can be contacted via email at a.aitider@uiz.ac.ma.

Dr. Lotfi Najdi, Ing., PhD, is a Moroccan computer science and artificial intelligence scholar currently serving as Associate Professor in the Department of Mathematics and Informatics at the Polydisciplinary Faculty of Taroudant (FPT), Université Ibn Zohr (UIZ), Agadir, Morocco. He is affiliated with the ISIMA Laboratory (Computer Systems Engineering, Mathematics and Applications), where he conducts research at the intersection of computational intelligence and educational analytics. Dr. Najdi's research interests encompass educational data mining, time-series forecasting, and decision-support systems, particularly on leveraging artificial intelligence and advanced

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054

analytics to enhance higher education management and institutional effectiveness. His work contributes to the growing field of learning analytics and its applications in academic administration and student success initiatives. He can be contacted via email at l.najdi@uiz.ac.ma.

Manuscript received 10 June 2025; Revised 5 August 2025; Accepted 22 August 2025; Available online 5 September 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1054