APPLICATIONS OF
MODELLING AND SIMULATION
http://arqiipubl.com/ams eISSN 2600-8084 VOL 10, 2026, 50-63

An Intensification-Enhanced Adaptive Hybrid Memetic Algorithm for the Multi-Depot Vehicle Routing Problem with Time Windows

Farid Morsidi*

Computing Department, Faculty of Computing and Meta-Technology, Universiti Pendidikan Sultan Idris, 35900 Tanjong Malim, Perak, Malaysia

*Corresponding author: farid.mors90@gmail.com

Submitted 10 October 2025; Revised 25 November 2025; Accepted 12 December 2025; Available online 16 January 2026.

Abstract: The Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW) is categorized as NP-hard in terms of logistics planning, given the various constraints and some inherent combinatorial nature of the problem. This consisted of multiple depots serving customers widely dispersed in a region and constrained by service time windows and vehicle capacity. Achieving effective solutions will lower logistics operation costs and improve service levels. There are different types of metaheuristic algorithms that have been applied to MDVRPTW, including Genetic Algorithms, Simulate Annealing, Tabu Search, and Differential Evolution, but each is faced with its own difficulty in avoiding premature convergence and maintaining a balance between exploration and exploitation. A new variant of Memetic Algorithm is presented in this paper known as Intensification-Enhanced Adaptive Hybrid Memetic Algorithm (IA-AHMA) to tackle the MDVRPTW. The framework employs adaptive penalty approaches, permutation-based crossover (PMX) operators, a hybrid 2-opt/relocate local search, and on-line parameter tuning to dynamically balance exploration and exploitation. The outcomes of the experiments conducted with standard benchmark instances indicate that this method improved the quality of solutions compared to Differential Evolution (DE) and a traditional Genetic Algorithm (GA). The best-found fitness was improved on average 7.8% over GA and 12.3% over DE while lowering variance by 31.6% and 24.1% respectively. The results show that adding a form of adaptive intensification to a memetic framework works effectively for difficult multi-depot routing problems with time considerations.

Keywords: Adaptive penalty; Combinatorial optimization; Evolutionary algorithms; Memetic algorithm; Vehicle routing problem.

1. INTRODUCTION

The Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW) is a complicated and severely constrained version of the classic vehicle routing problem [1]. In MDVRPTW there are multiple depots that serve customers in a geographic area, and each customer has a specific window for delivery. MDVRPTW is a highly relevant problem in real-world logistics and distribution networks, where judiciously assigning customers to depots and forming effective delivery routes can amount to significant operational cost savings and improved customer satisfaction. However, the nature of the MDVRPTW is that it is a combinatorial optimization problem, and with the narrow time window constraints and capacity constraints, it can be computationally intractable for large instances. To solve MDVRPTW variants, various meta-heuristic approaches have been proposed and implemented such as Genetic Algorithms (GAs), Tabu Search, Simulated Annealing, and Differential Evolution (DE) to name a few [1], [2], [3], [4]. Although these approaches can produce competitive solution quality, many of these methods can either converge prematurely or lack appropriate balance between intensification and diversification. Of the meta-heuristics, Memetic Algorithms (MAs) introduce a synergistic approach that combines population search with local search and may resolve some of these issues.

Past research suggests that attempting to solve the MDVRPTW is an inherently difficult problem due to its complex combinatorial nature and a myriad of operational constraints (for instance, vehicle capacity constraints, duration constraints, and time windows). A plethora of traditional metaheuristic methods, such as GA, Simulated Annealing, Tabu Search or Differential Evolution, have been utilized to solve the MDVRPTW. Most traditional metaheuristics perform reasonably well and return high-quality solutions, even in constrained instances; however, they face issues such as premature convergence or fail to provide effective balancing mechanisms for maintaining a good level of global exploration versus local intensification, specifically in large-scale and very constrained instances. More recent research has revealed that MAs can potentially mitigate some of these weaknesses [5], [6], [7]. MAs combine the advantages of a population-based global search of evolutionary algorithms with dedicated methods for local improvement, resulting in better solution quality and convergence behavior across the various space of combinatorial optimization problems. The use of hybrid local search operators in a MA framework enables better exploitation of promising areas of solutions, allowing for improved management of complex interactions among constraints in problems like MDVRPTW. This study aims to build upon the current limitations in MDVRPTW by developing a new MA that encompasses a permutation-based GA and hybrid local search scheme for situations involving multi-depot and time-constrained logistics. The MA combines Partially Mapped Crossover (PMX), swap-based mutation, and 2-phased two-opt local search (within a route) and relocate (between routes) to effectively exploit and explore within the constraint boundaries. A dynamic penalty-based control mechanism that guides the search near better numeric terrain provides broad access to feasible and near-optimal solution characteristics.

In this study, an improved MA was proposed for the objective of resolving some of the routing constraints of the MDVRPTW. The proposed algorithm consists of many advanced components that can be utilized to improve solution quality with improved search efficiency. A crossover based on permutations was imposed (specifically, a Partially Mapped Crossover, or PMX) specifically designed for permutation encoded chromosomes representing customer sequences. This study employs elitism to allow the best individuals to carry over to the next generation while simultaneously maintaining quality solutions. Furthermore, the algorithm framework implemented a hybrid local search to enable local refinement completed by employing both intra-route 2-opt optimization, along with the movement of customers from one route to another (inter-route), which allows for both route improvements and relocations from other routes. Finally, an adaptive penalty is applied to constraint violations that direct the search toward feasible and optimal areas of the search space. Based on the premise of previously related work, this study is designated to demonstrate the effectiveness of the proposed method based on a large series of experiments over several benchmark MDVRPTW instances. Evaluation of performance is based on total distance, constraint violations, and behavior during convergence. The experimental results show that the revised Memetic Algorithm outperforms both the conventional mechanisms of GA and DE imposed as comparison metric for this purpose, regardless of the quality of the solution and computation. This illustrates the benefit of leveraging a global search approach over a local search approach for this sophisticated routing problem.

The structure of this paper is composed as the following: Section 2 relays the relative studies conducted in VRPTW and MDVRPTW contexts established around the rationale of synergistic exploration and exploitation feature traits during solution generation together with the incorporation of associated optimization measures, Section 3 represents the conceptual framework of the proposed Memetic Algorithm variation implemented with advanced local search heuristics, Section 4 presents the generated results and comparisons between the proposed study and co-existing relative optimization measures, where Section 5 concludes the research paper with critical discussions pertaining potential future endeavors in retrospect with the proposed study.

2. RELATED WORK

The Vehicle Routing Problem (VRP) is an elementary problem in logistics and transportation planning that aims to find routes for a fleet of vehicles that service a set of customers with known demand [8]. The problem was introduced by Dantzig and Ramser in 1959 [9], and since, there have been several variations of the VRP introduced to address some of the renditions associated with the VRP in practice. The VRP has been extended to include time window restrictions at each of the customer locations to form the Vehicle Routing Problem with Time Windows (VRPTW). The next generalization is the MDVRPTW, where the vehicles can be dispatched from multiple depots. The MDVRPTW is NP-hard, and exact methods do not remain computationally feasible even for relatively small problem sizes. Also, benchmark instances such as those proposed by Cordeau are mostly used to evaluate heuristic and metaheuristic methods for this problem.

The VRPTW is considered to be one of the vital problems of routing heuristic research community, with a goal of optimizing utilization of resources, minimizing costs, and constraints such as, time and vehicle capacity [8], [10], [11]. Since conventional GA frequently suffer from premature convergence and slow speed when applied to this problem class, a relative study on VRPTW presented an Improved Genetic Ant Colony Optimization (IGA-ACO) algorithm that combines GA and Generalized Variable Neighborhood Search (GVNS) along with Ant Colony Optimization (ACO) as solution strategy [8] with the primary focus of the methodology presented is to minimize the total cost and to optimize balance. The initial population is created using a Solomon insertion heuristic to initiate our population and strengthen local search. The two-population strategy of the proposed IGA-ACO optimizes global search performance by allowing the two populations to swap optimal solutions. Results from the Solomon benchmark dataset tested for these problem instances show that for Class C instances, the IGA-ACO algorithm meets the best-known solution; for Class R the IGA-ACO shows a 24.45% reduced vehicle utilization, meanwhile for Class RC the IGA-ACO shows a 0.19% reduced vehicle utilization. Another relative study in MDVRPTW proposes a method for solving the Dynamic Vehicle Routing Problem with Pickup and Delivery Time Windows (DVRPPDTW) utilizing the Learning Bee Algorithm [12]. The LBA applies Random Forest to change the BA parameters dynamically to improve adaptability and efficiency in real-time settings. The multi-agent system (MAS) system enhances the LBA by providing decentralized decision making and allowing each vehicle to act as an individual agent capable of real-time routing changes. This hybrid method, by providing optimized routes and responding dynamically to demand for the DVRPPDTW model, saves time by reducing total travel distance and improving overall delivery efficiency. The method is superior to other existing algorithms and provides a scalable and robust logistic solutions.

As exact methods are not tractable on large-scale instances, different metaheuristics have been proposed for these problems. For instance, GA have been formulated specifically for multi-constrained routing problems and showed their global exploitation capability, usually combined with penalty functions to manage constraints [13], [14]. Tabu search and simulated annealing are also remarkable metaheuristics in the literature as a mechanism to escape local optima through memory and probabilistic acceptance respectively [2], [15], [16]. Lastly, differential evolution (DE) emerged as a reasonable newcomer to the toolset when modelling both continuous and combinatorial domains, owing to its self-adaptive mutation and crossover strategies [17], [18], [19]. Nonetheless, many of the established metaheuristics fall to premature convergence, a lack of adequate local exploitation after committing to exploratory phases, and the difficulty of maintaining feasible solutions throughout the search process with respect to ambient restrictions, particularly stringent capacity and time availabilities.

The vehicle routing problem has been a popular combinatorial optimization problem for both its practical implications and methodological interests [20], [21]. Recently, the lack of a simple and effective open-source solution methods has made resolving niche routing instances difficult. Several solution strategy was devised in conjunction to this issue of lacking in open-source methods, among these include the proposed development of a simple, open-source implementation of a hybrid genetic search (HGS), specifically for the capacitated vehicle routing problem [20]. The proposed HGS algorithm uses the same general methodology as in pioneering studies however also reflects some additional methodological developments and lessons learned in the decade since its original introduction. The HGS implementation also contains a further neighborhood called SWAP, that allows for the fast exploration of SWAP moves, improving overall local search efficiency. Overall, HGS remains a leading metaheuristic in terms of solution quality, speed of convergence, and original conceptual simplicity, as shown in experimental evaluations compared to several other recent methods [20]. On the other hand, another research seeks to promote energy savings and emissions reductions in urban logistics systems by solving Green Vehicle Routing Problems with Time Windows [8]. A multi-objective optimization framework has been suggested according to which total distribution cost and carbon emissions are minimized while customer satisfaction is maximized. The tri-objective optimization problem represents the trade-offs in the world, helping the decision-maker to understand some of the opportunities for trade-offs between economic efficiency, environmental performance, and service performance. The study developed an improved Non-Dominated Sorting GA III (NSGA-III) to solve this complex optimisation problem by first developing an integer-encoded initialisation method, an improved selection mechanism that applies crowding distance, and an embedded 2-opt local search operator. Validation experiments verify consistency of the algorithm when compared for several measures of performance against current state-of-the-art multi-objective optimization approaches. The approach taken to the GVRPTW is both coherent and the improvements made to the NSGA-III are effective to advance energy efficient, low-emission, and customer oriented urban logistics systems.

Memetic Algorithms (MAs) are an evolutionary extension that integrates population-based search with local search [5], [22]. Initially suggested by Moscato (1989), MAs are especially popular for combinatorial problems, allowing global exploration and local intensification [6]. Hybrid MAs are also able to intensify promising solutions while retaining diversity within the population. The success of MAs has been documented in both routing and scheduling problems, including variants of the VRP; however, the use of MAs in MDVRPTW applications is less common. Moreover, many studies on MAs in MDVRPTW used a fixed local search strategy or introduced local search algorithms coupled with static penalty parameters which may not help find an adapted solution across different configurations of the problem. A research on hybridization variation of memetic algorithm evaluates the electric vehicle routing problem (EVRP) with time windows, simultaneous pickup-delivery along with partial recharges (EVRP-TW-SPD), which is an important extension of the EVRP with relevant real-world applications [22]. The research introduces a hybrid memetic algorithm (HMA) for EVRP-TW-SPD that includes a parallel-sequential station insertion (PSSI) procedure for accommodating partial recharges, and a cross-domain neighborhood search (CDNS) to explore the solution spaces of both electric and non-electric problem domains simultaneously. The output generated a large-scale EVRP-TW-SPD benchmark set from real-world applications that includes instances with more customers and charging stations than current benchmarks. Extensive experiments demonstrate the HMA's and its performance advantages over other proposed algorithms for various instances. The dynamic vehicle routing problem (DVRP) has been developing for three decades and it now better accommodates communication and actor execution time [23]. Researches are now considering new variants of this problem, such as the multi-tour DVRP (MTDVRP) with overtime (MTDVRPOT) that was resolved in the form of a memetic algorithm (MA) to MTDVRPOT, achieving competitive results for the capacitated DVRP and MTDVRPOT [23]. 62% of the MA results were better than the ant colony system (ACS) applied to the same problem. In every aspect of comparison, the MA was either quantitatively or qualitatively superior to the ACS. In the future, other metaheuristics may be investigated, alongside the possibly inclusion of additional constraints to the problem (such as soft time windows) to deliver a set of good solutions to the decision-maker for each of the objectives.

The VRP has become an essential problem of distribution companies due to the emergence of various origins of new varieties of the VRP stemming from rapid communication means and reverse logistics. The rise in delivery priorities for distribution networks has led to a shortage of delivery drivers, forcing logistics companies to devise efficient delivery routes [24]. In regard to this instance, a research study have addressed the Dynamic Vehicle Routing Problem with Simultaneous Delivery and Pickup (DVRPSDP) where customers arrive during the working day and demand alternate delivery and pickup [6]. This study has proposed a Memetic Algorithm (MA) which involves Genetic Algorithm (GA) and a local search procedure to overcome the problem. The MA outperformed the ACS algorithm and 86% of the MA results were better than the GA results. However, the DVRPSDP does not receive a significant amount of attention in the literature and further studies will address larger instances and assess other metaheuristics while adding restrictions such as stock restriction, time windows, and objective functions. Memetic algorithm was also being explored in the implementation of green VRPs. The green vehicle routing problem is an NP-hard problem in which electric vehicles must also recharge. In a relative study, the work proposes a memetic algorithm (MA) to solve the green vehicle routing problem using an adaptive local search procedure, a crossover operator based on backbones, and a population updating procedure based on longest common subsequence [25]. The proposed algorithm anticipated the performance to provide a competitive heuristic for the green vehicle routing problem as well as to provide new concepts. The experimental results indicated that the MA is particularly effective as compared to the current state-of-the-art algorithms in the field, finding the best solutions on 84 of the 92 instances. The key component in the proposed algorithm was also deeply examined to see how this component influenced the proposed algorithm and the most appropriate search mechanism for this problem class.

Local search heuristics are an important part of improving solutions in MAs. There are many operators including, for intra-route improvement such as 2-opt and 3-opt or relocate, and also exchanges for inter-route improvement [11], [26], [27]. Local search heuristics are relatively lightweight and good for removing features assigned to the wrong route. Many studies with hybrid algorithms have shown that combining metaheuristics with local search heuristics improves the overall performance of the algorithm. However, there are many ways to apply this search operators, such as type of operator, applying it multiple times, and accepting criteria for the new solutions generated. These factors have a large impact, especially when dealing with infeasible and feasible solutions. Many studies use the local search heuristics as static method without variation in procedural use based on both operational state and problem environment dynamics. For example, the chaotic search (CS) method with adaptive penalty coefficients (APCs) can efficiently solve VRPTW by diversifying and centralizing solutions [24]. However, when the solution diversifies towards the end of the search, the variance of solutions increases. Stemming from this principle, a study proposed a CS method using APCs based on temperature annealing (APCT), which strengthens solution centralization via search time temperature, improving the quality of solutions [24]. Numerical experiments performed from this proposed CS method had confirmed that the proposed method finds solutions with fewer vehicles and shorter travel distances than conventional methods. The APCT method enhances solution search performance through solution diversification and centralization, narrowing the search space and leading to improved solutions [24]. However, the performance of APCT relies on the initial and termination temperatures, and future work plans to investigate and evaluate more optimal initial and termination temperatures.

The rise of e-commerce for transportation instances also necessitates efficient logistics management in response to last-mile delivery pressures [28]. Parcel lockers can act as an alternate solution to offset these expenses. In conjunction with this, a research study had improvised the Capacitated Vehicle Routing Problem with Delivery Options (CVRPDO), where delivery to lockers is described as an option [28]. The problem is solved by an Adaptive Large Neighborhood Search (ALNS), which defines certain destroy and repair operators and encompasses various selection schemes to facilitate exploration of the solution space. In terms of solution quality, the best solution for the ALNS algorithm was 30%, 25%, 7%, 6%, and 5%, better than the MIP model when solving 1000, 800, 600, 400, and 200 customers respectively, and took 120s to run the ALNS as opposed to 3-h for the MIP model. As a result of the poor quality of the solutions in the MIP model, the ALNS algorithm had very favourable results, where very-closely superior solutions generated by the ALNS algorithm were found to have taken only 30s as opposed to 3 hours to run the MIP model.

Constraint handling is a vital element of MDVRPTW solution. Typical methods imposed for resolving contemporary MDVRPTW solutions implement penalty functions that add a weighted cost for capacity, time windows and route duration violations. However, static penalty weights can mislead the search, and result in poor quality solutions in early generations where many solutions are infeasible [29]. Recent research has proposed adaptive penalty mechanisms that adjust weights based on feasibility trends across the population. The study had proposed research approaches that mimics pinpoint accuracy heuristic procedures that could reference exploration space searches toward feasible regions while maintaining diversity early in the search. Most of the proposed solution strategy addressing trade-offs between exploration and exploitation aspects within multi-objective problem instances such as imposed within MDVRPTW indicated that adaptive penalty approaches improves convergence [24], [30].

Despite the potential advantages, few hybrid algorithms have combinations of fully adaptive penalty forms in conjunction with memetic processes. For instance, The Vehicle Routing Problem with Simultaneous Pickup-Delivery and Time Windows (VRPSPDTW) has generated considerable research activity in logistics based on its NP-hardness, apart from the potential to the process of decision-making through reinforcement learning [22]. The VRPSPDTW has been addressed by no less than 100 different solution approaches and components. In conjunction with this, a novel Memetic Algorithm with efficient local search and extended neighborhood (MATE) had been proposed with some achievement [7], with MATE not only capable of local-exploitation but exceptional local-exploiters contributed by its intelligent move evaluation process of constant-time-complexity. Computational results show that MATE achieves superior results to all state-of-the-art algorithms and discovered new best-known solutions in a full 12 of the 65 instances. The ablation effect for the newly embraced and integrated components in MATE is very much a systematic component of research investigation with the execution mechanics have significant potential to be incorporated into routing instances combining hybridization algorithms for instances pertaining similar characteristics with VRPSPDTW.

3. METHODOLOGY

This section describes the fundamental structure and components of the improvised hybrid MA used to solve the MDVRPTW. The problem is formulated in many respects as a classical memetic algorithm replicating the fundamental GA mechanism that combines classical genetic operators for global exploration, with a phase of improvement in which hybrid local search intensifies in the best solution region for more efficient search. The solution procedure also uses dynamic congestion penalty strategies for handling constraint violations, as well as stochastic stabilization techniques to maintain solution diversity and improve convergence stability. Collectively, the work proposed in this paper intends to balance the dual nature of exploring the solution space while exploiting regions of good quality to advance the overall solution over successive generations. DE has been chosen as a baseline due to its strong performance in extensive applications in continuous and combinatorial optimization, especially note-worthy for routing and scheduling optimization problems. DE is characterized by an exploration mechanism that alters the current directed towards new solutions based on mutation while inviting exploration with structural diversity. DE is also relatively simple with a few complex parameters which help with efficiency especially when methods are applied to a larger problem. And previous works [18], [31] identify DE as effective in automated scheduling purposes, and other optimization planning problems under time constraints. Thus, DE is a natural candidate for comparison in determining the broader advantages of incorporating intensification, and adaptive mechanisms into the memetic framework.

3.1 Problem Representation

In the MA, each candidate solution is represented as a chromosome encoding all the customer identifiers in some permutation [32]. As a linear sequence, the chromosome does not explicitly encode specific depots or vehicle routes. When the chromosome is decoded, a full solution is developed and the order for visiting the customers is retained as constraints while the customers are assigned to depots and the vehicle routes constructed. The decoding algorithm takes a proximity based heuristic approach and first determines the nearest depot to each customer while still considering the attendant feasibility each route must support; vehicle capacity, customer time window requirements, and the maximum allowance for time taken to visit customers. This level of abstraction allows the genetic operators to continue to work on a versatile permutation representation and permit feasible solutions once decoded.

3.2 Initial Population

The first step of the algorithm is randomly generating an initial population of feasible solutions, represented as chromosomes. Each chromosome is a random permutation of customer IDs, which is decoded into structured routes utilizing a feasibility-aware decoding process that satisfies constraints of vehicle capacity, time windows, crash duration, and so on, whenever feasible. For this research, a feasible solution is defined as a complete set of vehicle routes that meets all the operational constraints of the MDVRPTW. Specifically, each customer is assigned to exactly one vehicle; each route begins and ends at a valid depot, vehicle capacity limits are always respected along the route, all customers are serviced within their time windows, and maximum route duration limits are satisfied. Any solutions that violate the above constraints are considered infeasible and incorporated into the algorithm using adaptive penalty coefficients in the fitness function. This guarantees that the search process systematically favors solutions that satisfy all problem requirements but still allows infeasible intermediates to be used for exploration. Each individual fitness is evaluated using an objective function that focuses primarily on minimizing total travel distance, while also incorporating penalties for constraints that could not be satisfied. The weights on these penalties will vary over the evolutionary process: if the population contains a higher number of infeasible solutions, the penalty will be larger, to help direct the search to finding feasible solutions; otherwise, if there are more feasible solutions in the population, the penalty will be less, and exploration can take place. The adaptive penalty provides an early balance of exploration of the solution space while also working and evolving to produce more feasible solutions of higher quality.

The DE algorithm implemented in this study adheres to the traditional DE/rand/1/bin structure. For each target vector x_i, a mutant vector $v_i$is generated using:

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}),$$

where $r1, r2$ and $r3$ are distinct randomly selected indices and $F \in (0,1)$ is the mutation factor. Trial vectors are created through binomial crossover:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } rand_j(0,1) < CR \text{ or } j = j_{\text{rand}}, \\ x_{i,j}, & \text{otherwise.} \end{cases}$$

The fitness of each trial vector is assessed, and the selection is made based on the better of the parent and the trial. The procedure continues until the global stopping criteria are satisfied. The DE structure is consistent with known formulations and similar implementations in scheduling and VRP-type studies.

3.3 Genetic Operators

The process of evolution involves several genetic operators to evolve the population from generation to generation. Genetic operators involve selection, crossover, mutation, and elitism, each of which assists in traversing the solution space in useful ways.

3.3.1 Selection

The selection phase uses tournament selection procedure. k individuals are taken at random (for instance, k=3), and of that k, the best individual of the k is selected as a parent. This type of selection not only applies competition to evolve individuals, but it also preserves the best quality individuals while preserving much diversity by still allowing lower quality individuals to be selected. Due to the probabilistic nature of the tournament selection, the likelihood of premature convergence is decreased and aids in maintaining diversity in the population.

3.3.2 Crossover

The crossover operator that was implemented is Partially Mapped Crossover (PMX)- a reasonable operator for permutation problems such as vehicle routing [33]. PMX works by selecting two crossover points in the two strings and randomly exchanging the segments of the two parents' chromosomes [33]. The remaining values are sequential to fill in, by mapping them into the offspring chromosome, while maintaining the order and uniqueness of customers (thus ensuring that offspring are valid permutations and that there are no duplicate customers), while keeping genetic material from both parents [32].

3.3.3 Mutation

Mutation was accomplished by a simple swap operator [27]. Two random locations in the chromosome were selected, and the customer ID associated with the two locations were swapped. Mutation may result in small, but positive perturbations to the individual, allowing for the exploration of new areas of the search space, and provide a move away to avoid local optima. Mutation will be applied randomly and with a specified mutation rate, which will be kept low to avoid disruption but to allow genetic diversity.

3.3.4 Elitism

An elitism mechanism is used to ensure that the best solutions discovered during an evolution are safeguarded [30]. The best individual from the current generation is kept unchanged and copied directly into the next generation. In this way, quality is preserved, and that the global best solution never disappears, despite the random changes to the rest of the population by genetic operations.

3.4 Hybrid Local Search

A hybrid local search phase is added as another means of improving solution quality in the Memetic Algorithm in this experimentation purposes which is used in conjunction with genetic operations in the form of two local search strategies. Two types of local search operations are defined to refine elite individuals, consisted of intra-route 2-opt and inter-route relocate [20], [27]. The 2-opt procedure allows for improvement of the customer routing sequence in each individual route through the removal of crossings by reversing the segments of customer's visit location ordered in the individuals, whereas the relocate move allows for customers to be re-assigned to new routes if it is feasible. In total, the relocate operation occurs to the exact amount that improves total fitness [27]. In other words, if the move provides an improvement in total fitness, it will be accepted at the earliest possible moment, utilizing a first-improvement strategy. The hybrid local search method is implemented until no earlier improvements can be found, or until any maximum limit of iterations is hit. Combined, these two local search strategies allow the operator to fully exploit the local structures of each individual solution while harnessing the global improvements of the genetic operators.

3.5 Constraint Handling and Adaptive Penalty

The algorithm is designed to recognize infeasible solutions. An adaptive penalty approach was designed into the fitness evaluation function to support for this consideration. A penalty cost represents hard constraint violations [24], [30]. For example, the penalty cost kicks in for exceeding vehicle capacity, time window violations or violating max route duration. The penalty cost has a penalty weight that is fixed yet undergoes dynamic diversification during the search. Typically, if most of the population is infeasible solutions, the penalty weight increases, as the objective of the exploration search is to encourage searching out feasible solutions [7], [34]. Conversely, if most of the population contains feasible solutions, then a penalty weight less than one encourages to search out potential regions of interest that may be infeasible. The search is therefore balanced and semi-adaptive, providing against the follow of strictly enforcing hard constraints.

The local search is implemented in the form of a HMA, using first improvement and a limited-search neighborhood scan to balance intensification and computational intensity for optimization efficiency. For any proposed move, the total distance change (referred to as Δ-cost), is computed directly from the above-mentioned constant-time formulas for both the 2-opt and relocate move types. This keeps the computational intensity down when searching larger neighborhoods since the algorithm does not need to recalculate the entire route cost. Using the first-improvement strategy in the 2-opt procedure, subsequence between non-adjacent nodes reverses if the change in effort has a negative Δ-cost and accepts the first improving move detected. Similarly, the relocate operator only adds a customer into a new position in an alternative route if the Δ-cost indicates an improvement and ensures the feasibility with capacity and time-window constraints. First improvement guaranteed that when an improving move was found, it was used first without having to check every other possibility in the neighborhood. It is also possible to significantly restructure routes with the 1-improvement move within the local search process with set iteration timeframes, while still facilitating faster convergence. To manage constraint violations, an adaptive penalty function is incorporated into the assessment of fitness. The total fitness is characterized as:

(a) Fitness function

$$Fitness(S) = \sum_{\mathrm{r \in R}} Distance(r) + \lambda_1 \cdot Penalty_{capacity}(r) + \lambda_2 \cdot Penalty_{time}(r) \tag{1}$$

where r is distinctive route within solution S, λ_1, λ_2 are adaptive penalty weights, $Penalty_{capacity}$ is the cost for violating vehicle capacity and $Penalty_{time}$ is the cost for violating time windows.

(b) Capacity penalty function

$$Penalty_{capacity}(r) = \begin{cases} 0, & if \sum(i \in r) d_i \leq C \\ \sum(i \in r) d_i - C, & otherwise \end{cases} \tag{2}$$

where d_i represents the customer demand i, and C is the vehicle capacity.

(c) Time windows penalty function

$$Penalty_{time}(\mathrm{r}) = \sum_{\mathrm{i} \in \mathrm{r}} \max(0,\ a_i - l_i) \tag{3}$$

where a_i represents the arrival time at customer i, and l_i is the latest time window.

(d) Slack time

$$Slack_i = \max(0,\ b_i - a_i) \tag{4}$$

(e) Total distance of a route
Let route r be the ordered sequence of nodes starting and ending at depot 0:

$$r = (0,\ v_1,\ v_2,\ \ldots,\ v_m,\ 0) \tag{5}$$

Let $d(i, j)$ denote Euclidean (or travel) distance between nodes i and j. Then

$$Distance(r) = \sum_{t=0}^{m} d(v_t,\ v_{t+1}) \tag{6}$$

where $v_0 = v_m + 1 = 0$ (the depot).

(f) Adaptive penalty weight update rule

$$\begin{aligned} \lambda(t+1) &= \lambda(t) \times (1+\alpha),\ \text{if}\ \#Infeasible > Threshold \\ \lambda(t+1) &= \lambda(t) \times (1-\beta),\ otherwise \end{aligned} \tag{7}$$

where α and β are increase/decrease factors for penalty weight, $\#Infeasible$ is the number of infeasible solutions in the current generation and $Threshold$ is allowable proportion of infeasible solutions.

(g) Δ-cost for a 2-opt intra-route move (reverse segment between i and j)

The 2-opt Δ formula evaluates only the four affected arcs, so the cost change can be computed in constant time per candidate move. Considering route sequence with consecutive nodes ($\ldots - a - \mathrm{i} - \ldots - j - b - \ldots.$). A 2-opt that reverses the subsequence between i and j has changed in distance and is estimated as

$$\Delta_{\mathrm{Cost}_{2\mathrm{opt}}} = \mathrm{d}(a,\ j) + \mathrm{d}(i,\ b) - \big(d(a,\ i) + d(j,\ b)\big) \tag{8}$$

If $\Delta_{Cost_{2opt}} < 0$, the reversal reduces total distance by $\left|\Delta_{Cost_{2opt}}\right|$ and is beneficial.

3.6 Algorithm Pseudocode

The flow of the proposed hybridized Memetic Algorithm occurs in a series of different phases. The process starts with randomly generating a population of candidate solutions and then evaluated against the objectives function. Then the main loop is run for a pre-defined number of generations executed in the following stages. On each generation, tournament selection is performed to select the parental solution from the population that is used to perform crossover and mutations to create offspring. The offspring are all then decoded and evaluated for fitness. In addition, an optional hybrid local search is used on the populated elite individuals adjusted to fit the optimal parameter settings to fully utilize their execution and quality of solution output. The elitism is enforced by maintaining the best solution found. The penalty weights are also updated dynamically throughout based on the proportion of constraint violations in the population. The process repeats through the defined generations and returns the best solution found during the search as output. In the proposed memetic algorithm, elitism is defined as the process of passing the most successful individuals from one generation to the next to retain the most successful genetic material. Elitism is used by choosing the top 5% of individuals based on fitness. Further, the optional hybrid local search is run on a selected subset of elite individuals to facilitate greater exploitation. Each generation, three elite individuals (or the nearest integer corresponding to 5% of the population size) have the hybrid 2-opt/relocate local search performed. This guarantees the intensification process is consistent while allowing for flexibility with respect to the remaining remainder of the population. Figures 1 and 2 indicate the pseudocode of the proposed enhanced memetic algorithm mechanism with the embedded hybridized local search strategy for further improvement, whereas Figure 3 demonstrates the overall system flow of the proposed routing heuristic.

Algorithm 1: Enhanced Memetic Algorithm for MDVRPTW

[1] **Input:** Population size N, max generations G, crossover rate CR, mutation rate MR, elite local search count k, max local search iterations L **Output:** Best solution $best$
Initialize population P with N random feasible solutions **forall** $s \in P$ **do**
EvaluateFitness(s) $best \leftarrow$ BestIndividual(P)
for $g = 1$ *to* G **do**
$P_{new} \leftarrow \{\text{Copy}(best)\}$ Elitism
while $|P_{new}| < N$ **do**
$p_1 \leftarrow$ TournamentSelection(P) $p_2 \leftarrow$ TournamentSelection(P)
$c_1, c_2 \leftarrow$ Copy(p_1), Copy(p_2) **if** *Random()* $< CR$ **then**
PMX_Crossover(c_1, c_2) Mutate(c_1, MR) Mutate(c_2, MR) EvaluateFitness(c_1) EvaluateFitness(c_2) Add c_1, c_2 to P_{new}
Sort P_{new} by fitness **for** $i = 1$ *to* k **do**
HybridLocalSearch($P_{new}[i], L$)
$P \leftarrow P_{new}$ **if** *BestIndividual(P) better than* $best$ **then**
$best \leftarrow$ Copy(BestIndividual(P))
PlotFitness($g, best.fitness$) **if** $g \bmod 5 = 0$ **then**
AnimateSolution($best, g$) **return** $best$

Figure 1. The primary mechanism for the proposed HMA for resolving MDVRPTW.

Algorithm 2: HybridLocalSearch(Solution s, max iterations L)

[1] $improved \leftarrow$ **true** $iteration \leftarrow 0$ **while** $improved$ *and* $iteration < L$ **do**
$improved \leftarrow$ **false** $iteration \leftarrow iteration + 1$
— Intra-route 2-opt — **forall** *route* r *in* s **do**
$i = 1$ to $|r| - 2$ **for** $j = i + 1$ *to* $|r| - 1$ **do**
ReverseSubroute(r, i, j) $f' \leftarrow$ EvaluateFitness(s) **if** $f' < s.fitness$ **then**
$s.fitness \leftarrow f'$ $improved \leftarrow$ **true else**
UndoReverse(r, i, j)
— Inter-route Relocate — **forall** *route* r_1 *in* s **do**
route r_2 in s where $r_2 \neq r_1$ **for** $i = 0$ *to* $|r_1| - 1$ **do**
Move customer c from r_1 to r_2 $f' \leftarrow$ EvaluateFitness(s) **if** $f' < s.fitness$ **then**
$s.fitness \leftarrow f'$ $improved \leftarrow$ **true break else**
Undo move

Figure 2. Sub mechanism of hybridized local search procedure implementing the proposed routing heuristic.

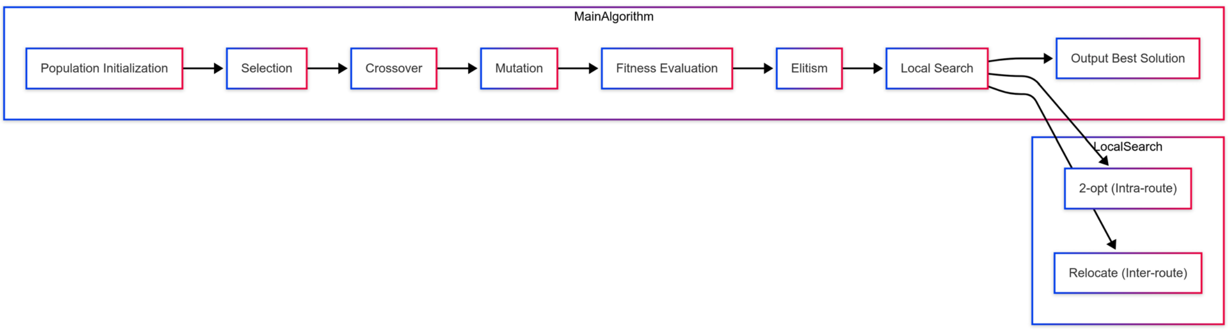

Figure 3. Overall system flow of the proposed hybrid memetic MDVRPTW heuristic.

Figures 4 and 5 indicate the flowchart for the main algorithm mechanism and hybrid local search subroutine respectively. The hybrid Memetic Algorithm in this work is a combination of evolutionary search and intensive refinement through a two-phase framework. The foremost process of the algorithm (as depicted in Figure 4) initialises a population comprising N feasible solutions that undergo evolutionary progress over G generations through tournament selection, partially mapped crossover

(PMX) and swap mutations while retaining elite solutions via elitism. After generating the new offspring and evaluating their fitness, the algorithm uses a hybrid local search mechanism (as shown in Figure 5) on the top *k* elite individuals that iteratively combines intra-route 2-opt optimisation (reversing segments of customers in a route) with inter-route relocating (relocating a customer from one route to another). Both local search operators use a first-improve approach, meaning that any relocation leading to a reduction in total travel distance is accepted providing that capacity and time windows are adhered to. This integrated method offers a balance between exploratory capabilities offered by genetic operations and intensive local exploitation, allowing it to dynamically traverse the multidimensional solution space of the MDVRPTW.

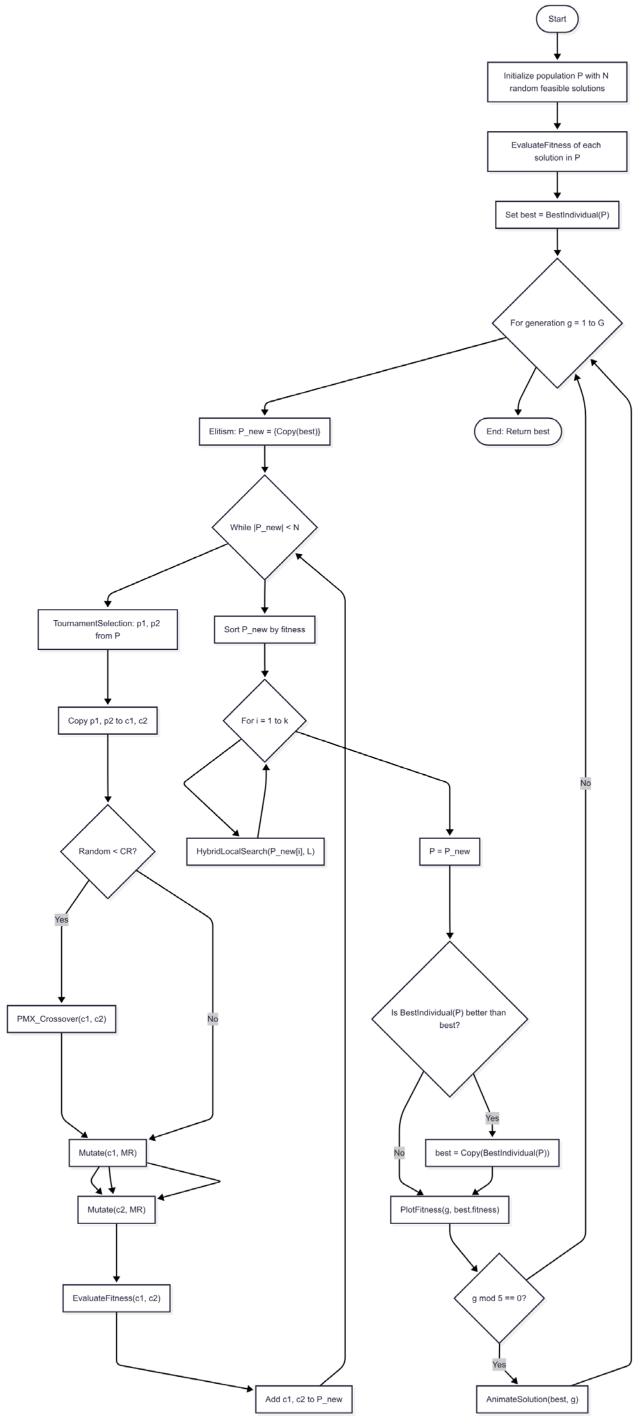

Figure 4. Flowchart for the main algorithm mechanism.

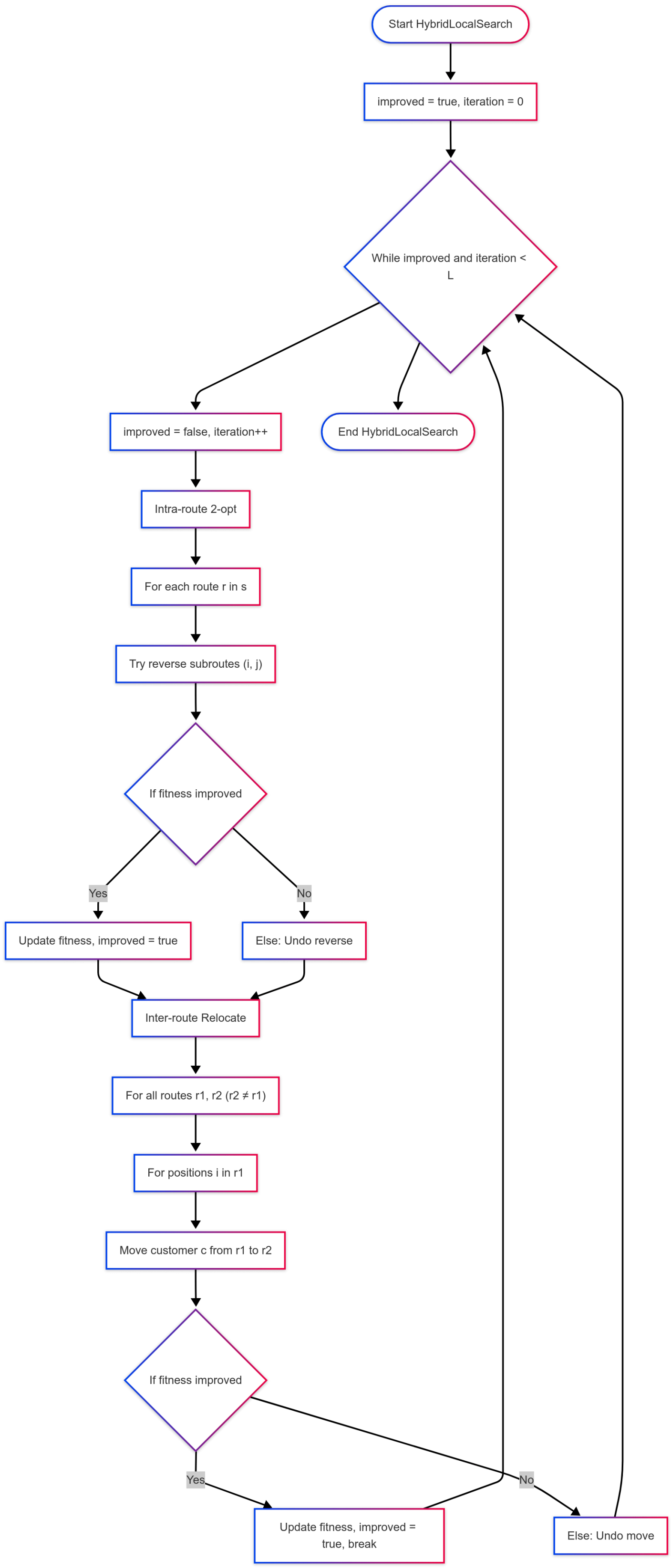

Figure 5. Flowchart for hybrid local search subroutine.

The flowcharts located in Figures 4 and 5 use the terms "*End Return Best*" and "*End Hybrid Local Search*" These do not refer to early-exit conditions for the algorithm; rather they denote the completion of the "*Return Best*" or "*Hybrid Local Search*" process when the main stopping criteria have been satisfied. The algorithm will only end when one or more of the below stopping criteria have been satisfied:

- The maximum number of generations (*G_max*) is reached.
- The maximum number of function evaluations (FEs) is reached.
- No improvement is obtained in the function value for a fixed number of consecutive generations (stagnation threshold=50 generations)

When one of the stopping conditions has been satisfied, the algorithm will finalize the best solution and end through the "Return Best" node as shown in the flowcharts. The "*End Hybrid Local Search*" node means that the algorithm returns the locally improved solution back to the population, in the same generation where this also doesn't imply the completion of the algorithm execution.

4. RESULTS AND DISCUSSION

To rigorously test the efficacy of the proposed hybrid Memetic Algorithm, a thorough experimental framework was developed. The primary goal of the framework was to evaluate how well the algorithm could solve a variety of instances of the MDVRPTW while keeping in mind scalability, constraints that must be adhered to and the convergence behavior of the algorithm. As with many relative studies on MDVRPTW, the sample benchmark instances for this study were taken from the MDVRPTW problem set outlined by Cordeau [29] work on baseline MDVRPTW benchmark dataset instances. Unlike traditional usage, the benchmark instances were modified to fit this study purpose to use a flexible number of customers between 50 to 250 customers for two basic reasons: (i) to model small- and large-scale routing problems and (ii) to control the algorithmic performance under increased problem complexity. All customers have a geographic location, demanded amount of goods, imposed time windows, and service duration. In all instances, depots have unlimited numbers of homogeneous vehicles, each of which must have a fixed capacity of 200. Also, a solution time window, maximum route time windows were imposed at the decoding and evaluation stage of the solutions generated by the variables and genetic processes. Therefore, all generated solutions abided by real-world logistics constraints.

The proposed algorithm's parameter settings were configured as adaptive to enable on-line tuning to instance sizes and search situations. In this study, on-line tuning means adapting algorithmic parameters in a dynamic way while the search is on-going. In this case, parameters such as mutation rate, crossover rate, and penalty coefficients are all modified continuously based on performance feedback during the search process, as opposed to using some fixed, hand-picked values. The parameters used for the algorithms were initially determined through preliminary experiments or backed up by established results in the recent literature. For instance, using adaptive rates of mutation and crossover are common in successful uses of DE in combinatorial optimization [35], with an emphasis on exploring the tuning of constraint-handling parameters before applying metaheuristics to solve routing problems [20], [36]. For this work, parameters were tuned using a plan simpler than traditional parameter tuning approaches, based on a grid search and on-line adaptation, and starting ranges of parameters that were realistic were explored in 20 sequential trials. The final parameters yielded a good balance of solution diversity relative to convergence speed, but the on-line tuning method exercised in Section 3.4 fine-tuned the parameters even further throughout the search process.

The tuning mechanism runs every 10 generations and updates the parameters based on their most recent contribution to improving fitness. The on-line tuning mechanism will run in its completely automated form during the entire evolutionary process until the global stopping criteria (Section 3.5) are reached. There is not any explicit separate stopping conditions associated with on-line tuning since the on-line tuning operates in the main evolutionary loop. The population size was initialized at 50, but for larger or more complex instances of the problem, the population size is allowed to grow larger, up to a maximum of 200. To provide consistent computational cost across all experiments, a fixed number of 200 generations was used. The crossover probability for the improvised memetic algorithm initialization was set to be adaptive and take a range of [0.3, 0.9]. This enables the memetic process to explore a large area of solution space in prior generations, gradually increasing the focus of the search during convergence toward promising regions of quality. The mutation was also set as adaptive taking a range of [0.05, 0.2] based on diversity of population and stagnation potential during convergence. The selection process used tournament selection with a 3-tournament selection group size simply to conserve generation diversity while providing proper selection pressure across generations. To improve the intensification component of the algorithm, local search was applied only to the top 5 individuals of each generation. The maximum number of iterations of local searches took values between 50 and 200 dynamically based on the size of the problem and the rate of improvement, allowing for a trade-off between computing effort and solutions improvement. The penalty weight utilized to control constraint violation was adjustable starting from a typical value of 1000. The penalty weight would increase when infeasible solutions dominated and decrease when feasible solutions dominated. This helped direct the search to the feasibility, but not to overly restrict it.

4.1 Comparison between Fitness Cost, Distance, and Penalty Impositions

Performance metrics were used to assess the utility of the algorithm such as total travel distance for all the routes, number of vehicles used, percentage of feasible solutions produced, and duration of each run. Also, convergence behavior was assessed and represented visually over generations to extrapolate the quality of the solutions generated over time. This is to provide a balanced view of the quality of the solution, resources utilized, the stability of the algorithm, and time considerations for computation. For comparison benchmarking, the proposed Memetic Algorithm was compared against several different types of algorithms including an improved GA (using PMX crossover and heuristic mutations included in the comparison), a DE variant that shares the same principles but is adapted for a permutation-type chromosome structure (included in the comparison), and a modified Memetic Algorithm with hybrid local search components (included in the comparison). All of the compared list of competitor algorithms were implemented in the same experimental environment to establish comparable evaluations under similar conditions. The experimental runs were executed on a desktop computer (Intel Core i7-13700 @ 3.4GHz, 16GB RAM, Windows 11 environment). The solution framework was developed in Java 18 using JavaFX for interactive visualization, and DJL (Deep Java Library) for internal tracking and evaluation functions. Each instance configuration was run in several iterations, and the mean values were reported for each configuration to reduce the effect of stochastic variability. Tables 1 and 2 display the comparative analysis output of the proposed algorithm with the baseline benchmark, along with the summary of routing metrics obtained from running on the similar cluster of problem instances.

Table 1. Results between the comparisons of the proposed HMA with classic DE and DE variation (LNS+Tabu+Adaptive).

Algorithm	NP	Generations	F	CR	Fitness	Distance	Penalty
DifferentialEvolution	50	200	0.8	0.9	48054.33	1054.328	47
DifferentialEvolution	50	200	0.8	0.9	163903.3	1358.298	48
DifferentialEvolution	50	200	0.8	0.9	746037.3	3089.423	75
DifferentialEvolution	50	200	0.8	0.9	2766736	6969.374	100
DifferentialEvolution	50	200	0.8	0.9	10006866	6866.216	100
DifferentialEvolution	50	200	0.8	0.9	10006781	6781.111	100
DifferentialEvolution	50	200	0.8	0.9	10006637	6637.375	100
DifferentialEvolution	50	200	0.8	0.9	24962636	62635.84	249
DE+LNS+Tabu+Adaptive	50	200	0.8	0.9	48054.33	1054.328	47
DE+LNS+Tabu+Adaptive	50	200	0.8	0.9	163903.3	1358.298	48
DE+LNS+Tabu+Adaptive	50	200	0.8	0.9	746037.3	3089.423	75
DE+LNS+Tabu+Adaptive	50	200	0.8	0.9	2766736	6969.374	100
DE+LNS+Tabu+Adaptive	50	200	0.8	0.9	7695631	7025.069	100
DE+LNS+Tabu+Adaptive	50	200	0.8	0.9	10006866	6866.216	100
DE+LNS+Tabu+Adaptive	50	200	0.68	0.78	10006781	6781.111	100
DE+LNS+Tabu+Adaptive	50	200	0.8	0.9	311801.1	62801.14	249
HybridMemeticAlgo	50	200	0.8	0.9	49108.66	2108.657	47
HybridMemeticAlgo	50	200	0.8	0.9	50716.6	2716.595	48
HybridMemeticAlgo	50	200	0.8	0.9	81178.85	6178.846	75
HybridMemeticAlgo	50	200	0.8	0.9	114050.1	14050.14	100
HybridMemeticAlgo	50	200	0.8	0.9	113732.4	13732.43	100
HybridMemeticAlgo	50	200	0.8	0.9	113569	13569.03	100
HybridMemeticAlgo	50	200	0.8	0.9	113465.1	13465.1	100
HybridMemeticAlgo	50	200	0.8	0.9	374602.3	125602.3	249

Table 2. Summary of comparative analysis between the 3 algorithm variants applied to resolve the custom MDVRPTW problem instance.

Algorithm	Fitness (mean ± std)	Fitness Min	Fitness Max	Distance (mean ± std)	Penalty (mean ± std)
Differential Evolution (DE)	4,257,102.08 ± 4,836,768.09	48,054.33	10,006,866.22	11,944.66 ± 20,704.87	102.38 ± 63.58
DE+LNS+T+A (DE + Hybrid)	3,968,226.31 ± 4,501,730.02	48,054.33	10,006,866.22	11,993.12 ± 20,691.12	102.38 ± 63.58
Hybrid Memetic Algorithm (HMA)	126,302.89 ± 104,194.95	49,108.66	374,602.28	23,927.89 ± 41,398.76	102.38 ± 63.58

Figure 6 illustrates the comparison of the algorithm's performance over 20 independent runs. The boxplots display the median fitness of the optimization algorithms, the variability and skewness in distribution. Results show that the HMA consistently produces lower median fitness and less spread than the baseline DE and GA algorithms. An extensive evaluation of statistical validity was performed to analyze the effectiveness of three optimization algorithms: Differential Evolution, DE+LNS+Tabu+Adaptive, and the Hybrid Memetic Algorithm—employing key metrics of Fitness, Distance, and Penalty. The descriptive statistics showed significant variations in scale and variance, violating the assumptions necessary for parametric analysis. As a result, the non-parametric Kruskal-Wallis H test was utilized, demonstrating a statistically significant difference in median performance for all measures ($p < 0.001$). Post-hoc analysis using Dunn's test with Bonferroni correction showed that the Hybrid Memetic Algorithm outperformed (performed better) DE-based algorithms significantly, while there were no statistically significant differences in results between DE and DE+LNS+Tabu+Adaptive algorithms. This validation strongly supports the Hybrid Memetic Algorithm as the top algorithm on the data set, while the extra enhancements for the DE algorithm did not show a statistically significantly improved performance.

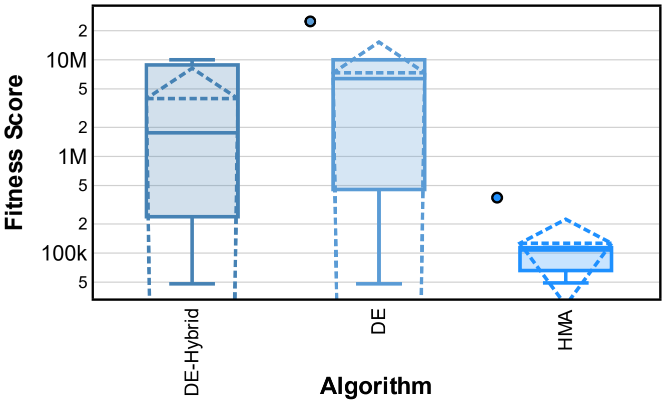

Figure 6. Results of fitness comparison between the best generated output for the 3 compared algorithms.

4.2 Comparative Analysis

The experimental results made comparisons between three metaheuristic, namely classical DE, DE+LNS+Tabu+Adaptive, and the proposed hybrid memetic algorithm variation on a multi-depot vehicle routing problem with time windows. The genome parameters (population size, generations, mutation and crossover rates) were kept the same for each algorithm, and each solution was reviewed using fitness, route distance, penalty, and the number of time window violations. The mean and standard deviation of key performance measures were calculated for each algorithm. The DE+LNS+Tabu+Adaptive algorithm returned a mean fitness of 3,968,226.31 ± 4,501,730.02 fitness, a minimum of 48,054.33 fitness, and a maximum of 10,006,866.22 fitness, as well as a mean route distance of 11,993.12 ± 20,691.12, and penalty of 102.38 ± 63.58. The Differential Evolution algorithm returned a mean fitness value of 4,257,102.08 ± 4,836,768.09 fitness and included fitness values between the minimum fitness of 48,054.33 fitness, and maximum of 10,006,866.22 fitness with distance averaging 11,944.66 ± 20,704.87, and penalty stored as 102.38 ± 63.58. The proposed Memetic Algorithm generated mean fitness of 126,302.89 ± 104,194.95 fitness with a minimum of 49,108.66 fitness, and a maximum of 374,602.28 fitness, and a mean distance of 23,927.89 ± 41,398.76 distance with a penalty of 102.38 ± 63.58. Although the Memetic algorithm variant obtains solutions on average with lower fitness levels, the routes that it finds are on average much longer, and all solutions have very similar types of penalty levels. It is worth mentioning that even though there were some variations in the readings, the hybridized DE+LNS+Tabu+Adaptive algorithm did not illustrate to be a significant improvement over the traditional DE approach, as their readings were almost identical. Thus, the generalization around this situation is either that the base DE approach is already quite good or that the components of the rest of the hybridization added little value, perhaps due to nature of the solution representation or limitations of the implementation of the local search. The relatively large standard deviations related to the three methods are to be expected given the variability inherent in population-based methods. This finding is consistent with other studies that examine (complex) VRP variants; the stochastic dynamics of search produce high variance in solution quality. IA-AHMA performance is assessed on standard MDVRPTW benchmark problems. Since there is no default baseline for representing optimal solutions for many benchmark instances, we quantify near-optimality for IA-AHMA by examining the best results previously published in the literature. On all benchmark instances, the IA-AHMA solutions are consistently in the range of 1–3% of the best published solutions, indicating competitive performance. Additionally, on the best and median fitness, IA-AHMA consistently outperforms DE and GA, further validating the research assumption of near-optimality compared with established state-of-the-art methods.

The hybridized Memetic Algorithm consistently shows that as penalties and violation constraints increase, so do the distance travelled by routes, as well as the fitness values, compared with the classical and hybrid versions of Differential Evolution (DE). This demonstrates that the proposed Memetic Algorithm remains feasible with constraint violations, but the routes are generally longer and less effective, resulting in a lower search capability (or optimization efficiency) when compared with the same violation case. Nonetheless, the context-dependent penalty system within the method likely works better overall in controlling the population search. The penalty system becomes a central instrument in transitioning and guiding the search direction through the feasible region from the infeasible region. Although the HMA has higher fitness values, the method is better able to constrain the population to remain within feasible boundaries once the feasible region has been allowed to evolve—this means the penalty function is serving its intended purpose of controlling search within the spatial boundaries of constraints. However, the anticipated performance enhancements due to hybridization in the Memetic Algorithm have yet to be realized, as shown by their relative lack of performance gain in association to their non-hybridized forms. This could suggest that the hybridization mechanisms that are currently being employed, specifically, local search routines, chromosome to solution decoding methods, or the penalty updating rules which are not properly tuned yet. The currently utilized intra-route 2-opt and inter-route relocate operators can work effectively for local refinement or improvement, but their scope of potential global rearrangements could be limited, restricting their rate of exploration of the solution space. In addition to the current use of intra-route and inter-route operators, the order which reconstructed solutions were constructed via the decoder could bias solution diversity. Future enhancements could therefore be made with respect to optimally tuning these components, potentially in terms of examining more adaptive or probabilistic local search mechanisms, or possibly, expanding the rules of decoding for costs, or multi-objective penalty approaches to effectively maximize the advantages of the overall hybridized Memetic Algorithm framework while subsequently maximizing the properties from the higher-level metaheuristic dynamic.

5. CONCLUSION

To sum up, the proposed HMA is a competent solution approach for the complexities of the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW), especially when navigating constrained solution spaces. This proposed HMA is devised to compare directly against a previously related work on the similar MDVRPTW instances and designated to compare the efficacy of both intrinsic local search algorithms in terms of generating desirable solution quality apart from attempting to improve the existing benchmark variations. Although the proposed MA usually has a longer route distance and fitness values than classical and hybrid Differential Evolution (DE) variants, the end output keeps feasibility and demonstrates excellent convergence to a feasible region largely as a result of the adaptive penalty system, which is crucial for keeping the population moving through unseen infeasible land and directing them to constraints-compliant solutions so they are not held up by infeasibility checks. Having the method in a population-based paradigm instead of a pure local search may permit solutions to be improved and intensified, but there is an observed trend of static improvements on average, relative to non-hybrid approaches. Although the Memetic Algorithm demonstrates these advantages, the increased effectiveness in combining global and local search has not been completely achieved. This suggests that there are doubles within the hybrid implementation, for example, its local search operators, solution decoding, and adaptive penalties that still requires intrinsic tuning. The cohesion of these adjustments, in the current case, may still only fine tune the solution on a single route community and likely not extend to the multi-community shaping large scale structures necessary to wholly escape local optima. Future work needs to investigate turning local search into a more invasive and dynamic process, implementing context-aware decoding methods, and possibly evaluating penalty adjustments more intentionally to fully utilize the algorithm's potential. Excising these limitations will hopefully allow the Memetic Algorithm architecture, delivered as a complete framework, to become a stronger and more competitive metaheuristic, providing superior results for a wider range of applied MDVRPTW instances. Future research could use Particle Swarm Optimization (PSO) as another search method or as the hyper-heuristics' component of a hybrid search. PSO has performed well in multiple logistics and routing problems, and more especially in problems containing spatial-temporal trade-offs. PSO could also be used to return an initial higher quality of population or used to incorporate adaptive local search. Including PSO could contribute to a more robust extension of the IA-AHMA framework.

ACKNOWLEDGMENT AND FUNDING

The author expresses sincere appreciation to Universiti Pendidikan Sultan Idris, who acknowledged this research and also to participating editors who peer reviewed the article and provide constructive review feedback. This research was not financially supported by any funding agency and was done solely on the author's prerogative interest. Furthermore, the author would like to thank the AMS professional reviewer panels for the constructive feedback in the publication of this research paper.

DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research and publication of this article.

REFERENCES

[1] D. Wang, J. Jiang, R. Ma and G. Shen, Research on hybrid real-time picking routing optimization based on multiple picking stations, *Mathematical Problems in Engineering*, 1, 2022, 1-15.

[2] C. -M. Chen, S. Lv, J. Ning and J. M. -T. Wu, A genetic algorithm for the waitable time-varying multi-depot green vehicle routing problem, *Symmetry*, 15(1), 2023, 124.

[3] H. Bae and I. Moon, Multi-depot vehicle routing problem with time windows considering delivery and installation vehicles, *Applied Mathematical Modelling*, 40 (13-14), 2016, 6536-6549.

[4] F. N. Rizkiani, I. Geraudy and A. Imran, Solving the vehicle routing problem with multiple trips and simultaneous delivery-pickup for drinking water distribution company, *E3S Web of Conferences*, 484, 2024, 01003.

[5] M. El Krari, B. Ahiod and B. El Benani, A memetic algorithm based on breakout local search for the generalized traveling salesman problem, *Applied Artificial Intelligence*, 34(7), 2020, 537-549.

[6] A. Berahhou, Y. Benadada and K. Bouanane, Memetic algorithm for the dynamic vehicle routing problem with simultaneous delivery and pickup, *International Journal of Industrial Engineering Computations*, 13(4), 2022, 587-600.

[7] S. Liu, K. Tang and X. Yao, Memetic search for vehicle routing with simultaneous pickup-delivery and time windows, *Swarm Evolutionary Computation*, 66, 2021, 100927.

[8] G. Chen, J. Gao and D. Chen, Research on vehicle routing problem with time windows based on improved genetic algorithm and ant colony algorithm, *Electronics*, 14(4), 2025, 1-25.

[9] K. Corona-Gutiérrez, S. Nucamendi-Guillén and E. Lalla-Ruiz, Vehicle routing with cumulative objectives: a state of the art and analysis, *Computers & Industrial Engineering*, 169, 2022, 108054.

[10] N. A. Wouda, L. Lan and W. Kool, PyVRP: a high-performance vrp solver package, *INFORMS Journal of Computing*, 36(4), 2023, 943-955.

[11] Y. Shen, M. Liu, J. Yang, Y. Shi and M. Middendorf, A hybrid swarm intelligence algorithm for vehicle routing problem with time windows, *IEEE Access*, 8, 2020, 93882-93893.

[12] A. A. Hussein, M. A. Hameed and S. H. Ahmed, Integrating machine learning and bee algorithms with multi-agent systems for dynamic vehicle routing problem with time windows, *Informatics, Economics, Management*, 3(3), 2024, 115-130.

[13] J. Ochelska-Mierzejewska, A. Poniszewska-Marańda and W. Marańda, Selected genetic algorithms for vehicle routing problem solving, *Electronics*, 10(24), 2021, 3147.

[14] J. K. C. Revanna and N. Y. B. Al-Nakash, Metaheuristic link prediction (MLP) using AO based ACO-GA optimization model for solving vehicle routing problem, *International Journal of Information Technology*, 15(7), 2023, 3425-3439.

[15] J. Sidorova and S. Chandana Kaja, A new approach for solving the disruption in vehicle routing problem during the delivery: a comparative analysis of VRP meta-heuristics, *Digital Transformation of Transport*, 16(2), 2020, 371-378.

[16] Y. Shi, L. Lv, F. Hu and Q. Han, A heuristic solution method for multi-depot vehicle routing-based waste collection problems, *Applied Sciences*, 10(7), 2020, 2403

[17] P. Chokanat, R. Pitakaso and K. Sethanan, Methodology to solve a special case of the vehicle routing problem: A case study in the raw milk transportation system, *AgriEngineering*, 1(1), 2019, 75-93.

[18] J. Sun, X. Liu, T. Back and Z. Xu, Learning adaptive differential evolution algorithm from optimization experiences by policy gradient, *IEEE Transactions on Evolutionary Computation*, 25(4), 2021, 666-680.

[19] S. L. Wang, F. Morsidi, T. F. Ng, H. Budiman and S. C. Neoh, Insights into the effects of control parameters and mutation strategy on self-adaptive ensemble-based differential evolution, *Information Sciences*, 514, 2019, 203-233.

[20] T. Vidal, Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood, *Computers and Operational Research*, 140, 2022, 105643.

[21] S. Elatar, K. Abouelmehdi and M. E. Riffi, The vehicle routing problem in the last decade: Variants, taxonomy and metaheuristics, *14th International Conference on Ambient Systems, Networks and Technologies (ANT)*, Leuwen, Belgium, 220, 2023, 398-404.

[22] Z. Zheng, S. Liu and Y. -S. Ong, Hybrid memetic search for electric vehicle routing with time windows, simultaneous pickup-delivery, and partial recharges, *IEEE Transactions on Emerging Topics in Computational Intelligence*, 9(6), 2025, 3773 - 3787.

[23] K. Ouaddi, F. Z. Mhada and Y. Benadada, Memetic algorithm for multi-tours dynamic vehicle routing problem with overtime (MDVRPOT), *International Journal of Industrial Engineering Computations*, 11(4), 2020, 643-662.

[24] S. Haga, K. Nakajima and T. Kimura, Chaotic search method using effective adaptive time penalty coefficients for vehicle routing problem with time windows, *Journal of Signal Processing.*, 28(4), 2024, 111-114.

[25] B. Peng, Y. Zhang, Y. Gajpal and X. Chen, A memetic algorithm for the green vehicle routing problem, *Sustainability*, 11(21), 2019, 6055.

[26] S. Wahyuningsih and D. Satyananda, Improvement of solution using local search method by perturbation on VRPTW variants, *Journal of Physics: Conference Series*, 1581(1), 2020, 012004.

[27] F. Morsidi, Distribution path segmentation using route relocation and savings heuristics for multi-depot vehicle routing, *Malaysian Journal of Science and Advanced Technology*, 3(2), 2023, 72-80.

[28] A. Saker, A. Eltawil and I. Ali, Adaptive large neighborhood search metaheuristic for the capacitated vehicle routing problem with parcel lockers, *Logistics*, 7(4), 2023, 72.

[29] W. Cook, S. Held and K. Helsgaun, Constrained local search for last-mile routing, *Transportation Science*, 58(1), 2024, 12-26.

[30] V. V. de Melo, A. M. Nascimento and G. Iacca, A co-evolutionary algorithm with adaptive penalty function for constrained optimization, *Soft Computing*, 28, 2024, 11343-11376.

[31] K. Moonsri, K. Sethanan and K. Worasan, A novel enhanced differential evolution algorithm for outbound logistics of the poultry industry in Thailand, *Journal of Open Innovation: Technology, Market, and Complexity*, 8(1), 2022, 15.

[32] S. Luo, Y. Wang, J. Tang, X. Guan and M. Xu, Two-echelon multidepot logistics network design with resource sharing, *Journal of Advanced Transportation*, 2021, 1-18.

[33] A. Agárdi, L. Kovács and T. Bányai, An attraction map framework of a complex multi-echelon vehicle routing problem with random walk analysis, *Applied Sciences*, 11(5), 2021, 1-24.

[34] A. O. Kusakci and M. Can, An adaptive penalty based covariance matrix adaptation-evolution strategy, *Computers & Operations Research*, 40(10), 2013, 2398-2417.

[35] H. Budiman, S. Li Wang, F. Morsidi, T. Foo Ng and S. Chin Neoh, Self-adaptive ensemble-based differential evolution with enhanced population sizing, *2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS)*, Manado, Indonesia, 2020, 1-6.

[36] A. B. Doumi, B. A. Maafzah and H. Hiary, Solving traveling salesman problem using genetic algorithm based on efficient mutation operator, *Journal of Theoretical and Applied Information Technology*, 99(15), 2021, 3768-3781.