IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 February 2024, accepted 7 March 2024, date of publication 19 March 2024, date of current version 25 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3378089

RESEARCH ARTICLE

A Hybrid Heuristic Harmony Search Algorithm for the Vehicle Routing Problem With Time Windows

YANG ZHANG[1] AND JIACHENG LI[2]
[1]Department of Computer Science, Kanagawa University, Yokohama 221-8686, Japan
[2]Department of Applied Systems and Mathematics, Kanagawa University, Yokohama 221-8686, Japan

Corresponding author: Yang Zhang (ft102212gh@kanagawa-u.ac.jp)

ABSTRACT In this paper, we present a hybrid heuristic method based on the harmonic search algorithm to solve the vehicle routing problem with time windows. The importance of efficient path planning in logistics and supply chain management is emphasized herein, particularly under complex constraints like time window and vehicle capacity limits. The hybrid heuristic harmony search algorithm combines the global search capability of the harmony search algorithm with the accuracy of the local search heuristic to efficiently explore and harness the solution space. Through rigorous testing on the Solomon dataset, the hybrid heuristic harmony search algorithm performed remarkably in the generation of competitive solutions while maintaining the computational efficiency of the solutions. The results showed that the algorithm achieved competitive solutions even under strict time window constraints. The convergence of the algorithm was examined, and its strong performance in handling complex instances was revealed. This study enhances the operational efficiency of an organization and provides perspectives and solutions for optimization strategies in the domains of logistics and supply chain management.

INDEX TERMS Harmony search, heuristic algorithm, solomon dataset, vehicle routing problem.

I. INTRODUCTION

In the field of modern logistics and supply chain management, route planning is extremely critical. Effective route planning is important in reducing costs and improving efficiency. This is because they directly affect the operational efficiency, cost control, and customer satisfaction of an organization [1]. The objective of route planning is to transport goods or services across locations in the most efficient way possible. This involves determining the shortest path while considering various constraints, such as traffic conditions [2], distribution time windows [3], and vehicle load limits [4]. In this context, the vehicle routing problem (VRP) is an important branch of route planning. The VRP involves finding an optimal set of routes under constrained conditions so that a vehicle can efficiently serve a set of customers and return to its starting point. Exploring the Vehicle Routing Problem (VRP) typically involves reducing the total distance or costs incurred, alongside accommodating various operational considerations like limitations on vehicle capacity and specified service time windows(TWs), and customer demand, which render the VRP complex and challenging. From the basic form, which ensures that a single vehicle visits all customer points along the shortest path and returns to the origin, to more complex variants, such as a VRP considering multiple vehicles [5], VRP with different types of customer demands [6], and VRP with dynamic road condition information [7], the VRP is diverse and highly adaptable. Thus, studying and solving the VRP is required to understand the basic principle of path planning and improve the efficiency and responsiveness of real logistics systems. With the advancement of technology, particularly in algorithm research, computing power, and data processing, the VRP and its related studies provide perspectives and solutions for modern logistics and supply chain management [8]. With the increasing diversification and sophistication of customer demands, the traditional VRP model can no longer fully satisfy the needs of complex real-world scenarios. The VRP with TW (VRPTW) has emerged to accurately simulate real-world distribution constraints [9].

The associate editor coordinating the review of this manuscript and approving it for publication was Abderrezak Rachedi.

In the VRPTW, in addition to factors such as the route and load of a vehicle, each customer has one or more TWs in which the vehicle must arrive and complete the service. This window can be fixed or flexible (i.e., with tolerance for early or late arrival). Vehicles arriving at the location of the customer outside the specified TW are not allowed, which may result in the service not being completed or requiring additional high costs to be processed. Thus, the VRPTW adds a time dimension to the basic VRP, resulting in a more complex problem. Fig. 1 shows a typical VRPTW, illustrating a network of multiple customer nodes, each with its geographic location and an assigned service window. As shown, three vehicles depart from a central depot, and each vehicle is assigned certain customer nodes to service. Importantly, the service of each customer is constrained by the geographic location and TWs. These windows specify the time frame in which the service can begin and must be completed, ensuring a timely service. The marker next to each customer node shows its TW information (Fig. 1), indicating that the vehicle must arrive and complete the service within that time frame.

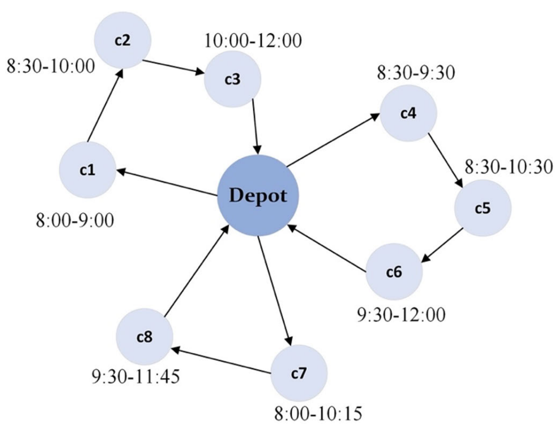

FIGURE 1. An example of a VRPTW.

In such a scenario, the goal of the VRPTW is to optimize vehicle routing to ensure that all customers are served within their time windows while minimizing the total distance and time traveled. This requires a complex algorithm that integrates TW constraints, vehicle capacity, and route efficiency. Considering that the VRP is a nondeterministic polynomial time (NP)-hard problem, its TW variant is considered to possess a high degree of computational complexity [10]. With the escalation in problem size, especially regarding the quantity of customers and the complexity of the Time Windows, there is a notable surge in the computational effort required to identify the optimal solution. Current solution methods fall into three main categories: exact, heuristic, and metaheuristic algorithms [11]. Exact algorithms are dedicated to finding the optimal solution to the problem and are typically employed to solve small VRPTW problems. However, as the problem's scale expands, the computational duration associated with these algorithms markedly escalates, thereby limiting their application to large-scale VRPTW problems [12], [13], [14]. Golden et al. [15] showed that exact algorithms are not effective for VRP problems with more than 50 clients. Heuristic algorithms can find suboptimal solutions in an acceptable time, although the optimal solution may not be guaranteed [16], [17]. Contrarily, meta-heuristic algorithms employ more general search strategies to optimize solutions by simulating natural processes or drawing on other intelligent behaviors. These algorithms comprise genetic algorithms (GAs) [18], particle swarm optimization [19], and simulated annealing algorithms [20]. They can find near-optimal solutions to a wide range of problems through a combination of iterative search and multiple heuristics. Qi and Sun proposed an improved algorithm based on an ant colony system (ACS), which demonstrated its effectiveness in dealing with the VRPTW by dual optimization on the primary objective (minimizing the number of vehicles) and the secondary objective (reducing the total cost of the trip) [21]. Zhang et al. proposed a hybrid algorithm combining the forbidden search and artificial swarm for a novel VRP with a TW and tray loading constraint algorithm optimized for complex requirements in the real logistics industry [22]. Bouchra et al. introduced a combined approach, integrating Genetic Algorithm (GA) with the variable neighborhood search technique, aimed at addressing the vehicle routing problem featuring a soft Time Window (STW) in both static (VRPSTW) and dynamic (D-VRPSTW) contexts [23]. Shen et al. proposed a hybrid algorithm that fuses an ACS and Brainstorm Optimization (BSO) algorithm as a hybrid swarm intelligence algorithm to solve the VRPTW. They obtained a competitive solution when experimenting on Solomon's benchmark instance [24]. Solving the VRPTW requires traditional cost and efficiency considerations, as well as time accuracy and customer satisfaction. Owing to this VRPTW characteristic, researchers have increasingly favored the use of heuristic algorithms, meta-heuristic algorithms, and a hybrid of both to find approximate or suboptimal solutions.

Harmony Search (HS) is a meta-heuristic algorithm that simulates the process of musical improvisation [25]. Since its emergence in 2001, it has demonstrated its effectiveness in solving optimization problems in several domains [26]. There are many applications in solving the VRPTW. For example, Yassen et al. devised a meta-heuristic Harmony Search Algorithm (meta-HSA) targeting this issue, aiming for an equilibrium between the exploratory capabilities of HSA and the refinement offered by the Local Search (LS) strategy, this balance was achieved through the adaptive selection of parameters for both HSA and LS, including the LS's neighborhood structure, thereby significantly enhancing the algorithm's performance [27]. Maleki et al. proposed a hybrid adaptive globally optimal HSA (HSGHSA) to solve the VRPTW. They enhanced the algorithmic capabilities by combining the improved HSGHSA with an adaptive mechanism to adjust the control parameters and employing six LS neighborhood structures [28]. These studies focused on how to harness the global search capability of HSAs to explore diverse solutions. This effectiveness stems from

HSA's primary strength - its formidable capability for global search. This attribute enables it to proficiently navigate the solution space, thereby facilitating the discovery of solutions that are close to the optimal. Compared with other meta-heuristic algorithms, the lesser parameter-tuning requirement of the HSA makes it easier to operate and implement during the optimization search process. Thus, the HSA is efficient and flexible in handling complex problems, such as the VRPTW. However, although the HSA is effective in global search, it may suffer the challenge of slow convergence in certain cases, particularly when dealing with large-scale problems. In addition, the HSA may fall into local optima while maintaining its stochastic nature, thereby affecting the quality of the final solution. In practice, an important challenge for HSAs is how to balance their stochastic and deterministic elements to ensure that premature convergence is avoided and the search direction remains stable. Thus, this study examined how to improve and sound the search algorithm, i.e., to improve the convergence speed of the HSA while avoiding falling into local optima as much as possible. In addition, we investigated how to balance the stochastic and deterministic elements by design to enhance the performance of the algorithm in solving VRPTWs.

Researchers have analyzed the effects of various heuristics and meta-heuristics on the VRP [29], [30]. They concluded that no single heuristic or meta-heuristic could outperform others in all cases. Hybrid algorithms can enhance the advantages of two or more methods by combining or competing approaches to produce better solutions. This study focused on the VRPTW, specifically the Solomon dataset. The Solomon dataset is a standard test set that is widely used in VRPTW research, containing VRPTW instances of different sizes and characteristics. It is an ideal testbed for algorithm performance evaluation. The vehicle must arrive within the TW specified by the customer, not early nor late. If early, it incurs a waiting time, and if late, it is denied service. To effectively solve this problem under strict time constraints, we propose a Hybrid Heuristic HSA (HHHSA). This algorithm is based on the HS framework, which introduces inter-route and intra-route improvement strategies. The present paper reports computational results on the Solomon dataset, comparing its performance with the best known solutions (BKS) and solutions obtained using other meta-heuristics. The results showed that the HHHSA could find competitive solutions in an acceptable time. Furthermore, we examined the convergence of the HHHSA, corroborating its remarkable performance in handling complex instances under strict TW constraints. This study is important for understanding and solving the VRPTW and provides perspectives and effective solutions for path planning in logistics and supply chain management.

The rest of the paper is structured as follows: Chapter II details the mathematical model and constraints of the vehicle path problem and its TW variant. Chapter III reviews the principles of the HSA and its application to the path optimization problem. Chapter IV demonstrates the improved HHHSA proposed herein, detailing its improvement points and expected advantages. Chapter V analyzes and compares the performance of the proposed algorithm with existing algorithms on standard test sets. Finally, Chapter VI provides the conclusions.

II. PROBLEM DEFINITION AND MODELING

The mixed integer model of the VRPTW can be described as $V = \{0, 1, 2, \ldots, n\}$, where 0 denotes the center, and $V_c = \{1, 2, \ldots, n\}$ denotes the set of customers. K is the set of vehicles, and the onboard capacities are all Q. The distance and time costs for a vehicle to travel from customers i to j are d_{ij} and t_{ij}, respectively. The demand of customer i is q_i, the required service time is s_i, and the service time window is $[e_i, l_i]$, with e_i being the earliest and l_i being the latest allowable starts of the service, respectively. Vehicle k arrives at customer i at AT_i^k, and the waiting time is WT_i^k. If vehicle k travels from customercustomer i to customer j and successively serves two customers one after another, then $x_{ij}^k = 1$, otherwise $x_{ij}^k = 0$. The mathematical model of VRPTW was constructed as follows:

$$\min Z = \sum_{i \in V} \sum_{j \in V/\{i\}} \sum_{k \in K} x_{ij}^k \cdot d_{ij}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{i \in V/\{j\}} x_{ij}^k = 1 \quad (\forall j \in V_c) \tag{1}$$

$$\sum_{k \in K} \sum_{j \in V/\{i\}} x_{ij}^k = 1 \quad (\forall i \in V_c) \tag{2}$$

$$\sum_{i \in V_c} x_{i0}^k = \sum_{j \in V_c} x_{0j}^k = 1 \quad (\forall k \in K) \tag{3}$$

$$\sum_{j \in V_c/\{i\}} x_{ij}^k = \sum_{j \in V_c/\{i\}} x_{ji}^k \le 1, \forall i \in V_c, \forall k \in K \tag{4}$$

$$\sum_{i \in V} \sum_{j \in V/\{i\}} x_{ij}^k \cdot q_i \le Q, \forall k \in K \tag{5}$$

$$\sum_{i \in S} \sum_{j \in S/\{i\}} x_{ij}^k \le |S| - 1, \forall S \subseteq V_c, |S| \ge 2, \forall k \in K \tag{6}$$

$$e_i \le AT_i^k + WT_i^k \le l_i, \forall k \in K, \forall i \in V \tag{7}$$

$$WT_i^k = \max\left\{0, \left(e_i - AT_i^k\right)\right\}, \forall i \in V_c, \forall k \in K \tag{8}$$

$$AT_i^k + WT_i^k + s_i + t_{ij} = AT_j^k, \forall i, j \in V_c, i \ne j, \forall k \in K \tag{9}$$

$$x_{ij}^k = \begin{cases} 1, & \text{if vehicle } k \text{ travels from customer } i \text{ to customer } j \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where Z is the total distance traveled by all the vehicles. In the constraints, (1) and (2) show only one vehicle arriving at each customer and only one vehicle departing from each customer, respectively. This implies that only one vehicle served each customer. Equation (3) ensures that each vehicle k departs from and returns to the depot once. This implies that each vehicle began its route from the warehouse, served a series of customers, and returned to the warehouse. Equation (4) shows that each vehicle k departed from one customer only to another while each customer i was being served and that each customer i was served only once by a vehicle k. This constraint helps to prevent situations where each vehicle serves the same customer multiple times or a customer is served by multiple vehicles. Equation (5) shows that the sum of the demands of the customer visited by each vehicle did not exceed the onboard capacity of the vehicle. Equation (6) shows that there are no two customers for repeated visits and was used to eliminate sub-loops. Equation (7) shows that if a vehicle exists to service a customer, the start time of the customer is within its TW, i.e., the TW constraint. Equation (8) denotes the waiting time for the vehicle to arrive at the customer it is to serve. Equation (9) denotes the arrival time relationship constraint of a vehicle for two customers served consecutively.

III. BASIC HARMONY SEARCH ALGORITHM

The HSA, inspired by the process of music composition, is an optimization method based on intelligent search. Geem et al. proposed this algorithm in 2001, aiming to solve the optimization issues by simulating the process of discovering a harmonic melody in a music performance [25]. Just as a musician refines his melody through continuous practice, the optimization algorithm improves the evaluated value of the objective function through an iterative process. HSAs generate better harmonies by selecting or adjusting harmonies in an already existing harmony library (solution set). After repeated iterations, the optimal solution to the problem, i.e., the best harmony, is found. Compared with traditional GAs, HSAs integrate the genetic characteristics of the population and focus on the range of the values of individuals, to better inherit the good characteristics and avoid falling into the local optimum.

The flow of the HSA is as follows:

A. SETTING UP THE PROBLEM AND ALGORITHM PARAMETERS

We can summarize the global optimization problem in this way:

$f(x)$ was minimized subject to

$$x(i) \in X_i, \quad i = 1, 2, \ldots, N \tag{11}$$

where $f(x)$ represents the objective function, x denotes the array of decision variables $x(i)$, N is the total count of decision variables, and X_i specifies the range of feasible values for each decision variable. Furthermore, the algorithm's initial phase requires appropriate configuration of parameters.

B. SETTING UP THE HARMONY MEMORY

A harmony memory (HM), characterized by a Harmony Memory Size (HMS), was established in accordance with the solution space, in the following manner.

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \ldots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \ldots & x_{N-1}^2 & x_N^2 \\ . & . & \ldots & . & . \\ . & . & \ldots & . & . \\ . & . & \ldots & . & . \\ x_1^{HMS-1} & x_2^{HMS-1} & \ldots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \ldots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix} \tag{12}$$

Continuous problems were randomly generated according to the following formula:

$$x_i^j = LB(i) + (UB(i) - LB(i)) * r \tag{13}$$

where $i = 1, 2, \ldots, N, j = 1, 2, \ldots HMS$, and r is a random number between 0 and 1.

For discrete optimization problems, the following formula was followed:

$$x_i^j \in X_i = \{x_i(1), x_i(2) \ldots, x_i(K)\} \tag{14}$$

where $x_i(K)$ is an alternative discrete value in the i-th dimension.

C. CREATING A NEW HARMONY

The formation of a new harmony vector involves three distinct rules: (a) memory consideration, (b) pitch adjustment, and (c) random selection [31]. Initially, a random number, r_1, within the range [0, 1], is generated and compared with the pre-set Harmony Memory Consideration Rate (HMCR). If $r_1 < HMCR$, a variable from the pre-established HM, known as the memory consideration, is chosen randomly. If not, it is sourced through random selection, meaning it's generated randomly within the search boundaries. Subsequently, the harmony variable is updated. In case of an update via memory consideration, an adjustment is necessary. Here, another variable r_2 within the [0, 1] range is randomly produced. If $r_2 < PAR$, the variable undergoes a modification based on the preset Bandwidth (BW), resulting in a newly adjusted variable, referred to as the pitch adjustment. The rule for pitch adjustment is outlined as follows:

$$x_i^{new} = x_i^{new} \pm r * BW \tag{15}$$

where r is a random number between 0 and 1.

D. UPDATE THE HARMONY MEMORY

Should the newly generated harmony yield a more favorable outcome in terms of the objective function compared to the least effective solution in the previously initialized HM, it will replace the least effective harmony within the HM.

E. DETERMINING TERMINATION

The process involved assessing whether the current count of iterations had reached the pre-established maximum number of iterations, T_{max}. If this threshold had not been met, the procedure encompassed in Steps (3) and (4) would be reiterated until the maximum iteration count was attained.

According to the vehicle path planning problem and the characteristics of the HSA, a harmony represents a customer or vehicle code. We designed the basic framework of the HSA for solving the VRPTW as follows:

Step 1. Initialize the HM.

Step 2. Generate the i-th solution component of the new harmony according to the three rules of new solution generation.

Step 3. Determine if the generated i-th solution component is legal (if repeated with the previously generated $i-1$ solution components). If not, execute Step 2, otherwise, $i = i+1$.

Step 4. Determine if a set of completed harmony has been generated. If so, execute Step 5; otherwise, execute Step 2.

Step 5. Calculate the objective function value of the new harmony. Ascertain if the harmony satisfies constraints, such as the maximum driving distance and maximum vehicle capacity limit. If not, add the corresponding penalty.

Step 6. Assess whether the newly formulated harmony outperforms the least effective harmony in the HM. In cases where it does, proceed to substitute the least effective harmony with this new one.

Step 7. Determine if the termination condition is satisfied. If so, stop; otherwise, execute Step 2.

IV. PROPOSED METHOD

The specific algorithm design process was as follows:

A. ENCODING

Here, sequential coding was employed to solve the vehicle path planning problem. Sequential encoding is intuitive and prevents repetition. For a given path planning problem with n customers and m vehicles, we used $n+m-1$ consecutive natural numbers to represent the encoding. Here, the first n bits represent the customer, and the last $m-1$ bits represent the vehicle identification. For example, for a path planning problem with 10 customers and 3 vehicles, we used the following encoding: 1, 2, 3, 11, 4, 5, 6, 7, 12, 8, 9, 10. Here, 11 and 12 represent the vehicle identification code, indicating that the route of the first vehicle is 0, 1, 2, 3, 0, and the second vehicle serves customers 0, 4, 5, 6, 7, 0. In turn, the third vehicle serves customers 0, 8, 9, 10, 0. This encoding method intuitively represents the path-planning situation of the vehicle.

B. GENERATION OF SOLUTIONS

The basic harmony algorithm for new solution generation includes three keys: random generation, harmony retention probability, and harmony perturbation probability.

If the current solution to be generated is the i-th solution component x_i of the new solution $X = (x_1, x_2, \ldots x_i \ldots x_N)$, a random number r_1 is first generated. if $r_1 < HMCR$, then x_i is randomly selected from column i of the HM. If the randomly selected position is h, then $x_i = HM(h, i)$. If $r_1 > HMCR$, x_i takes random values from its value interval. For the VRP comprising n customers and m vehicles using sequential coding, the value space of x_i is $(1, 2, \ldots, N, \ldots, N+M-1)$. For $r_1 < HMCR$, x_i was randomly selected from the i-th column of the HM, and random number r_2 was generated. If $r_2 < PAR$, then one of the previous solution vectors was selected to swap with the i-th bit solution vector.

C. HANDLING STRATEGIES FOR ILLEGAL ENCODINGS

As this study adopted the natural number coding, considering the natural number coding characteristics and model constraints, the code must be guaranteed to be unique and non-repeating. However, in the process of generating the new solution of the harmonic algorithm, duplicate harmonies will inevitably be generated, such as coding 1, 11, 2, 3, 2, 5, 6, 7, 3, 8, 9, 10. For example, the aforementioned code has duplicates of 2 and 3, and such code is illegal. To ensure that the encoding of the new solution x_i is legal, the following rejection strategy was used in the algorithm design and processing.

Let us assume that the new solution vector to be generated is $X' = \left(x'_1, x'_2, \ldots x'_i \ldots x'_N\right), T = \emptyset$, and x_i is the i-th solution component generated according to the new solution generation rule of harmony. The rejection process is shown in Fig. 2.

Step 1. Determine if $i > N$; if yes, stop generating solutions. Otherwise, execute Step 2.

Step 2. Determine if $i = 1$; if yes, $x'_i = x_i$, $T = T \cup \left\{x'_i\right\}$, and $i = i+1$. Otherwise, execute Step 3.

Step 3. Determine if $x_i \in T$; if yes, generate the nearest node x_i to x'_{i-1} that satisfies $x'_i \nexists T$, and perform $T = T \cup \left\{x'_i\right\}$, $i = i+1$. If $x_i \in T$ does not hold, then $x'_i = x_i$, $T = T \cup \left\{x'_i\right\}$, and $i = i+1$. Execute Step 1.

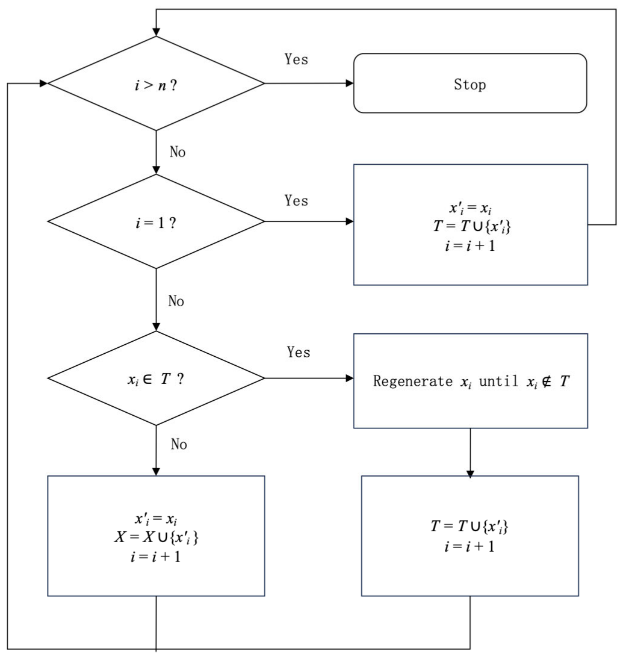

FIGURE 2. Rejection process of an illegal encoding.

D. PLUG-IN HEURISTIC PATH CONSTRUCTION

The basic idea of the plug-in heuristic method is to select customer nodes to construct a path following certain rules until a feasible solution is formed. For the present VRPTW, we first decoded a feasible path based on the constraints of vehicle loading. Second, considering that this study attempted to solve the VRPTW, the decoded feasible path may not meet the TW. To adjust the generated path, we first selected a seed customer in the path to form the initial path of a single customer point and ensured that the remaining path in the customer not added to the path was inserted into the formed path. Thereafter, the customer not added to the path was selected and inserted into the formed path. The selected path had already completed the judgment of the feasible path. Thus, the new customer point was only placed in the most suitable position according to the TW to ensure that the TW was strictly enforced from beginning to end.

If there were 10 existing customer points, coded as 1, 2, 3, 11, 4, 5, 6, 7, 12, 8, 9, 10, according to the aforementioned decoding principle, the first path was 0-1-2-3-0. For this path, we first randomly selected a node (e.g., node 2) and randomly selected the remaining unselected customer points on this path such as 1. We adjusted the left TW according to 1 and 2, keeping the left TW of the previous customer point smaller than that of the latter. Assuming that it was now adjusted to 2,1, we selected customer point 3, following the idea of the plug-in heuristic method. At this time, 3 was inserted into the position, according to the size of the left TW inserted in the appropriate position. This was to ensure that the left TW of all the customer points for the smallest to the largest were in order.

E. LOCAL NEIGHBORHOOD SEARCH

To balance the vehicle tasks and further improve the quality of the solution, this paper proposes the following perturbation strategies, which are mainly the deletion and reinsertion operators. The deletion operator proposed in this paper is the correlation deletion, and the reinsertion operator is the feasible optimal reinsertion. They were implemented as follows:

1) CORRELATION DELETION OPERATOR

First, the number of customer nodes, TR, to be deleted was determined. The number of regular deletions was 10%–20% of the total number of all the customer points. Second, customer i was randomly selected, and the TR customers with higher correlation with customer i from the corresponding path were removed. The correlation between clients i and j was defined as

$$R_{ij} = \frac{1}{c + V}$$

where $c = \dfrac{d}{d_{max}}$, d_{max} is the distance between this node (customer i) and the farthest node feasible

$$V = \begin{cases} 1, & \text{Customer } i \text{ and customer } j \\ & \quad \text{are not on the same route} \\ 0, & \text{Customer } i \text{ and customer } j \text{ are on the same route} \end{cases}$$

2) FEASIBLE OPTIMAL REINSERTION OPERATOR

The process of feasible optimal insertion was as follows: let us consider the insertion of customer j. First, we calculated all the feasible insertion locations and their path distance increments after insertion into the paths of the current solution R that satisfies the TW constraints and the vehicle capacity constraints. Thereafter, we inserted customer j into the location with the smallest path distance increment. If there was more than one location satisfying the insertion condition, the one with a smaller path distance length was preferred. If there was no location satisfying the insertion condition, a new path $r_t = \{0, j, 0\}$ was added and incorporated into the current solution, R.

F. IMPROVED HYBRID HEURISTIC HARMONY SEARCH ALGORITHM

We set the number of new solutions that had been generated as newger. iter was the current number of iteration generations of the loop, and IT was the set maximum number of iteration generations. The new solution was X, and the set holding the new solutions was NEW. Thus, the update process of the new solution set was $NEW = [NEW, X]$, and n was the number of dimensions of the solution, i.e., the number of variables. The improved HHHSA process was as follows:

Step 1: Initialize the HM and parameters.

Step 2: Determine if iter exceeded IT. If yes, output the optimal solution; otherwise, execute Step 3.

Step 3: Determine if newger exceeded the set maximum number of new solutions to be generated, HMS. If so, update the HM, and $iter = iter + 1$, then execute Step 2. Otherwise, execute Step 4.

Step 4: Determine if the number of generated solution components, i, reaches n. If yes, a completely new solution X was generated. Update the new solution set NEW, i.e., $NEW = [NEW, X]$, $newger = newger + 1$, then execute Step 3; otherwise, execute Step 5.

Step 5: Generate a random number, r_1. If $r_1 < HMCR$, the i-th solution component of the new solution X was randomly selected in the i-th column of the HM. Execute Step 6; otherwise, the i-th solution component of the new solution X was randomly selected in its own value space, then execute Step 8.

Step 6: Generate a random number, r_2. If $r_2 < PAR$, pitch adjustment was applied to a randomly selected number from column i of the HM; otherwise, it remained unchanged.

Step 7: Perturbation of the new harmony.

Step 8: $i = i + 1$, and execute Step 4.

The flowchart of the algorithm is shown in Fig. 3.

V. EXPERIMENTS AND DISCUSSION

This study selected Solomon's Vehicle Path Planning problem dataset as the benchmark problem instance. In assessing the effectiveness of the proposed Hybrid Heuristic Harmony Search Algorithm (HHHSA) in addressing the VRPTW, an initial comparison was made with the Best Known

TABLE 1. Categorization of solomon's VRPTW datasets.

Dataset	Number of instances	Number of customers	Number of vehicle	Vehicle Capacity	Distribution Type	Width of time window
C1	9	100	25	200	Cluster	Small
C2	8	100	25	700	Cluster	Large
R1	12	100	25	200	Random	Small
R2	11	100	25	1000	Random	Large
RC1	8	100	25	200	Random/ Cluster	Small
RC2	8	100	25	1000	Random/ Cluster	Large

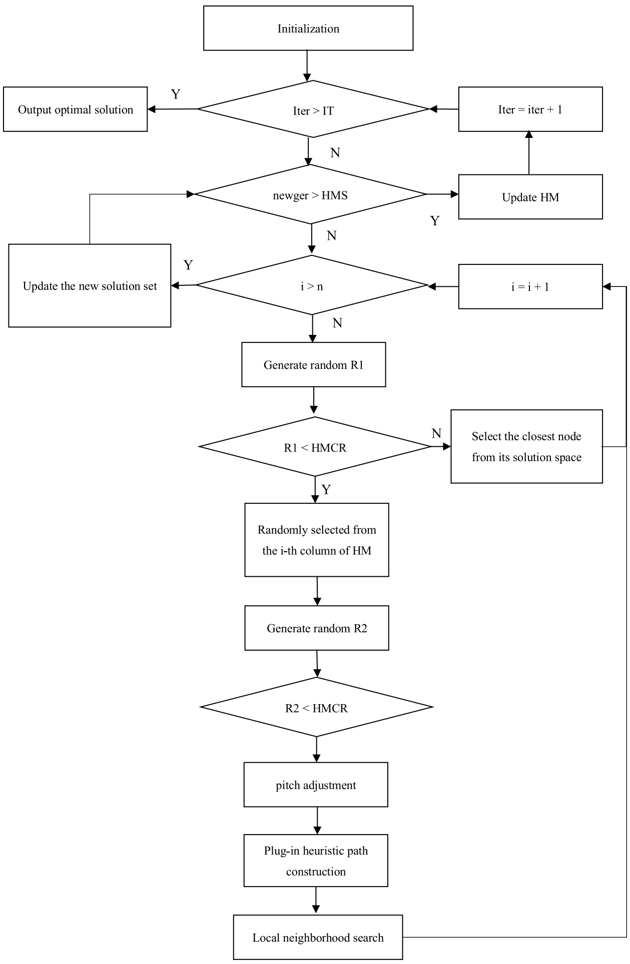

FIGURE 3. Flowchart of the HHHSA.

Solutions (BKS), as well as the standard HSA and its prominent variants. Subsequently, we evaluated the average performance against other meta-heuristic algorithms and scrutinized the convergence efficiency of our proposed algorithm.

A. BENCHMARK PROBLEM INSTANCES

The Solomon dataset contains 56 problem instances (each with different characteristics and complexities). The benchmark instances were classified into six categories (C1, C2, R1, R2, RC1, and RC2), which can be employed to comprehensively evaluate the performance of algorithms.

TABLE 2. Performance of different HMS values.

	Upper HMS					
Instances	5	10	20	50	80	100
C102	829.12	835.99	875.99	904.46	891.22	971.35
C206	588.49	588.69	589.12	599.16	607.68	598.57
R103	1242.95	1261.92	1260.92	1288.66	1292.1	1300.19
R201	1187.22	1229.92	1219.64	1249.43	1220.32	1238.88
RC101	1697.88	1738.85	1765.55	1817.67	1908.24	1849.53
RC201	1302.81	1329.35	1342.11	1341.14	1359.38	1355.98
Average	1141.41	1164.12	1175.56	1200.09	1213.16	1219.08

TABLE 3. Performance of different PAR values.

Instances	0.1	0.2	0.3
C102	829.12	831.98	836.1
C206	588.49	589.12	591.06
R103	1242.95	1290.1	1331.12
R201	1187.22	1190.12	1220.1
RC101	1697.88	1721.01	1760.66
RC201	1302.81	1320.98	1329.98
Average	1141.41	1157.22	1178.17

These problem instances cover different customer distributions, capacity limits, and TW constraints. Class C was the clustered distribution of customer points, class R was a random distribution of customer points, and class RC was a mix of random and clustered distributions. Each class was divided into two subclasses according to the vehicle capacity and the length of the TW. Classes C1, R1, and RC1 had a vehicle capacity of 200, and the TWs of their algorithms were tighter than those of C2, R2, and RC2. The vehicle capacities of classes C2, R2, and RC2 were 700, 1000, and 1000, respectively. The problem sets R1, C1, and RC1 featured brief scheduling cycles, accommodating a limited number of customers (around 5-10) per route. In contrast, sets R2, C2, and RC2 encompassed extended scheduling intervals, enabling service to a larger group of customers (exceeding 30) using the same vehicle. Each instance comprised 100 customer nodes and a single depot node, all distributed across a 100 $\times$ 100 Euclidean grid, where travel times were equivalent to the respective distances. Table 1 presents the specific classification information.

B. EXPERIMENTAL SETUP

The proposed algorithm was implemented and tested using Matlab_R2023b, and all the experiments were conducted using a MacBook (2.3 GHz 8-core Intel Core i9 processor, 16 GB 2667 MHz DDR4 RAM). The HHHS parameters were as follows. The HMCR parameter was established at 0.9, adhering to the guidelines suggested in a previous study [27]. Following the outcomes of initial testing (refer to Table 2), the HMS was determined to be 5. Similarly, drawing from preliminary results shown in Table 3, the PAR was set to 0.1. The upper limit for iterations was established at 100.

C. RESULT ANALYSIS

To verify the effectiveness of the proposed algorithm in dealing with the VRPTW, three experimental analyses were performed, solving each arithmetic case 10 times and counting their performances. The current findings were juxtaposed with the reported Best Known Solutions (BKS), alongside comparisons with outcomes from the standard Harmony Search Algorithm (HSA) and its renowned variations, along with the average performances of other metaheuristic algorithms. Finally, a case study was implemented to examine the convergence of the HHHSA.

1) COMPARISON WITH THE BEST KNOWN SOLUTIONS

We compared the reported BKS [24], and the references of each BKS corresponding to the solution method are labeled in Table 4. "N*" and "N*" denote the number of vehicles and the total distance traveled in the current BKS for that instance, respectively. "BN" and "BD" denote the total number of vehicles and total distance traveled in the optimal results of the 10 solutions of the proposed algorithm, respectively. To examine the gap between the proposed method and BKS, we calculated the GAP using $\mathrm{GAP} = (\mathrm{BD} - \mathrm{D}^*)/\mathrm{D}^*$.

We determined the route with the shortest distance, and the proposed algorithm achieved good results compared with those of the BKS using the Solomon dataset. For C1 and C2, the HHHSA found all the BKS, except C104 and C204. For R and RC, the proposed algorithm approached the BKS in many cases, demonstrating its effectiveness in dealing with the VRPTW in different environments. The average gaps for R1, R2, RC1, and RC2 were 3.27%, 1.61%, 3.55%, and 1.71%, respectively. Although gaps existed in certain cases, most of them were relatively small, suggesting that the algorithm was robust in various scenarios. These gaps highlight potential areas for improvement but validate the ability of the algorithm to effectively solve the complex constraints of the VRPTW.

2) COMPARISON OF STANDARD HARMONY SEARCH ALGORITHMS AND WELL-KNOWN VARIANTS

The solution outcomes of our proposed algorithm were evaluated against those of the standard Harmony Search Algorithm (HSA) and its recognized variants (Tables 5, 6, and 7 for the Best, Avr, and Std., respectively), and the winning solutions are marked in bold.

The results showed that the HHHSA demonstrated significant advantages in several instances. In the best result comparison, the HHHSA met or exceeded the existing HSA and its variants in most cases. In the average result and standard deviation comparisons, the HHHSA exhibited high stability and superiority, confirming its effectiveness and robustness in solving the VRPTW. In specific instances, such as C102, C206, and C208, the standard deviation of the HHHSA was significantly lower than that of other algorithms, suggesting that its solution was less volatile and more stable. Noteworthily, in specific problem instances, the advantage of the HHHSA was not evident. This may have been due to the special nature of the problem instances or a mismatch between the search strategy of the HHHSA and the characteristics of the problem instances, which is a direction of our future research.

3) COMPARISON WITH OTHER METAHEURISTIC ALGORITHMS

Table 8 lists the results of the application of the HHHSA, the well-known Tabu search (TS), TS-CP, GA, hybrid GA (HGA), AC-TC, and Meta-HSA algorithms to Solomon's benchmark dataset. TS is a probabilistic TS proposed in a previous study [36]. TS-CP represents a straightforward framework that integrates Tabu Search (TS) with constraint programming methodologies [48]. GA and HGA were proposed in previous studies [17], [49]. AC-TS is a reported hybrid AC with TS [50]. Meta-HSA is one of the best reported HSA variants [27].

The average results (Avr.) in Table 8 shows that the average performance of the HHHSA was consistent and stable, indicating that its performance was less volatile across runs and provided a reliable solution for the solver. The HHHSA significantly outperformed other algorithms when solving R2 and RC2. In the R1, C1, and C2 instances, the HHHSA performed similarly to the best results in Table 8. Although it did not completely outperform all the other algorithms, it still showed its competitiveness and adaptability. In terms of runtime, HHHSA performed extremely well. Its runtime was relatively low compared with those of other algorithms, such as the GA and HGA. This is important for practical applications where good quality solutions are required in a reasonable time.

Our goal was not to defeat all the existing solutions but to propose a solution that minimizes the need to manually adjust and that can deliver good results in solving the VRPTW. The experimental results confirmed its usefulness.

4) CASE STUDY ON CONVERGENCE ANALYSIS

We selected the simpler C102 and the more complex R204, RC201 with a strict TW as representatives of the case studies. Fig. 4 shows the convergence change curve for the C102 instance. As the number of iterations increased, the objective value continually decreased, indicating that the HHHSA effectively optimized the route to reduce the total distance. The colored lines represent the average value of each iteration, and the red line indicates the best value for each iteration. The convergence results were evident around generation 13 and finally converged to the optimal solution 828.94 at generation 75. Fig. 5 shows the route result, and Table 9 presents the detailed route information.

TABLE 4. Results for solomon's 56 instances with 100 customers.

Instances	BKS		HHHSA		GAP	Instances	BKS		HHHSA		GAP
	N*	D*	N	BD			N*	D*	N	BD	
C101	10	828.94 [32]	10	828.94	0.00%	R112	10	953.63 [32]	10	997.19	4.57%
C102	10	828.94 [32]	10	828.94	0.00%	R201	8	1147.8 [33]	7	1162.2	1.25%
C103	10	828.06 [32]	10	828.06	0.00%	R202	8	1034.35 [32]	5	1044.89	1.02%
C104	10	824.78 [32]	10	846.78	2.67%	R203	6	874.87 [32]	6	882.29	0.85%
C105	10	828.94 [32]	10	828.94	0.00%	R204	5	735.8 [33]	4	737.23	0.19%
C106	10	828.94 [32]	10	828.94	0.00%	R205	5	954.16 [33]	5	975.77	2.26%
C107	10	828.94 [32]	10	828.94	0.00%	R206	5	879.89 [32]	6	905.91	2.96%
C108	10	828.94 [32]	10	828.94	0.00%	R207	4	799.86 [34]	5	818.53	2.33%
C109	10	828.94 [32]	10	828.94	0.00%	R208	4	705.45 [35]	4	718.54	1.86%
C201	3	591.56 [32]	3	591.56	0.00%	R209	5	859.39 [32]	5	866.4	0.82%
C202	3	591.56 [32]	3	591.56	0.00%	R210	5	910.7 [35]	7	924.87	1.56%
C203	3	591.17 [32]	3	591.17	0.00%	R211	4	755.96 [34]	4	776.02	2.65%
C204	3	590.60 [32]	3	596.55	1.01%	RC101	15	1623.58 [36]	16	1672.9	3.04%
C205	3	588.88 [32]	3	588.88	0.00%	RC102	14	1461.23 [32]	14	1505.9	3.06%
C206	3	588.49 [32]	3	588.49	0.00%	RC103	11	1261.67 [37]	12	1290.3	2.27%
C207	3	588.29 [32]	3	588.29	0.00%	RC104	10	1135.48 [38]	10	1153.8	1.61%
C208	3	588.32 [32]	3	588.32	0.00%	RC105	16	1518.58 [32]	15	1564.8	3.04%
R101	20	1642.88 [32]	19	1656.5	0.83%	RC106	13	1371.69 [39]	14	1417.25	3.32%
R102	18	1472.62 [40]	18	1477.7	0.34%	RC107	12	1212.83 [34]	12	1280.28	5.56%
R103	14	1213.62 [40]	15	1225.84	1.01%	RC108	11	1117.53 [40]	11	1190.15	6.50%
R104	11	976.61 [32]	11	1017.5	4.19%	RC201	9	1265.56 [32]	7	1283.82	1.44%
R105	15	1360.78 [32]	15	1416.7	4.11%	RC202	8	1095.64 [32]	7	1106.85	1.02%
R106	13	1240.47 [34]	13	1269.3	2.32%	RC203	5	928.51 [34]	5	951.06	2.43%
R107	11	1073.34 [34]	12	1112.68	3.67%	RC204	4	786.38 [33]	4	801.43	1.91%
R108	10	947.55 [34]	10	984.91	3.94%	RC205	7	1157.55 [33]	7	1169.22	1.01%
R109	13	1151.84 [34]	13	1203.7	4.50%	RC206	7	1054.61 [32]	6	1082.66	2.66%
R110	12	1072.41 [32]	12	1139.3	5.87%	RC207	6	966.08 [32]	6	984.93	1.95%
R111	12	1053.50 [32]	11	1094.3	3.87%	RC208	4	779.31 [34]	5	788.9	1.23%

TABLE 5. Best performance of the HHHSA in relation to HSA and its variants.

Instances	Best									
	HAS [25]	IHSA [41]	GHS [42]	SGHSA [43]	DHSA [44]	MHSA [45]	SHSA [46]	ITHSA [47]	Meta-HAS [27]	HHHSA
R101	1704.11	1692.5	2389.48	2025.19	1872.75	1745.09	1690.39	1722.76	**1642.88**	1656.5
R103	1387.59	1412.28	2125.59	1646.74	1489.3	1476.33	1394.02	1399	1232.96	**1225.84**
R201	1858.5	1824.19	2106.68	1846.44	1740.26	1907.6	1419.44	1810.93	1202.96	**1162.2**
C102	1228.2	1176.51	2708.38	1616.49	1314.26	1337.5	1079.36	1128.79	828.94	**828.94**
C109	1362.78	1418.72	2894.76	1690.83	1500.72	1574.62	1099.79	1365.04	832.29	**828.94**
C206	1516.25	1505.35	2109.02	1677.9	1702.53	1706.81	817.02	1314.86	595.37	**588.49**
C208	1414.39	1297.89	1805.01	1751.98	1696.58	1609.91	769.57	1241.73	594.7	**588.32**
RC101	1734.57	1703.34	2537.67	2097.18	1870.16	1794.66	1713.62	1746.66	**1639.73**	1672.9
RC201	2106.93	2084.07	2497.04	2227.3	2079.33	2189.64	1689.17	2074.96	1345.16	**1283.82**

TABLE 6. Mean performance of the HHHSA in relation to HSA and its variants.

Instances	Avr									
	HSA [25]	IHSA [41]	GHS [42]	SGHSA [43]	DHSA [44]	MHSA [45]	SHSA [46]	ITHSA [47]	Meta-HSA [27]	HHHSA
R101	1762.02	1767	2534.3	2245.04	1956.56	1836.07	1741.15	1771.57	**1644.01**	1659.15
R103	1497.12	1507.72	2366.45	1846.03	1613.44	1586.46	1447.27	1485.82	1255.23	**1242.95**
R201	1990.66	1983.83	2294.26	2150.17	2024.42	2099.55	1521.87	1991.73	1232.95	**1187.22**
C102	1456.81	1402.57	2920.21	2051.46	1623.65	1614.25	1197.21	1334.46	831.67	**829.12**
C109	1615.31	1659.52	3079.97	2127.45	1841.43	1833.45	1225.69	1603.77	**847.45**	872.99
C206	1802.84	1758.84	2308.17	2226.03	2051.09	1968.11	888.88	2050.26	632.87	**588.49**
C208	1679.24	1666.59	2161.45	2091.27	2021.68	1938.39	862.95	2042.63	638.68	**588.32**
RC101	1814.3	1811.43	2742	2325.94	1998.28	1893.67	1800.66	1830.43	**1663.5**	1697.88
RC201	2237.13	2260.14	2701.32	2448.81	2362.23	2412.18	1772.53	2282.6	1385.5	**1302.81**

Figs. 6 and 7 show the convergence curve and routing results of R204, respectively. The convergence graph shows significant convergence near generation 6 and finally converges to the best solution 737.23 at generation 64, with only 0.19% Gap with BKS. Table 10 shows the routing information of the final solution.

TABLE 7. Standard deviation outcomes for the HHHSA in comparison to HSA and its variants.

Instances	Std									
	HAS [25]	IHSA [41]	GHS [42]	SGHSA [43]	DHSA [44]	MHSA [45]	SHSA [46]	ITHSA [47]	Meta-HAS [27]	HHHSA
R101	29.13	37.05	85.81	104.94	57.73	43.84	21.65	31.61	1.7	9.45
R103	39.37	50.76	93.58	99.78	57.72	44.7	23.83	34.75	15.2	**12.1**
R201	75.73	96.69	85.78	133.94	140.67	95.18	37.16	130.52	22.5	**18.65**
C102	104	117.08	129.48	273.28	147.43	147.89	55.87	121.16	3.65	**0.37**
C109	133.59	111.6	111.29	248.93	151.67	142.58	64.8	120.06	9.99	25.07
C206	171.22	148.17	124.97	265.53	199.4	213.28	43.71	426.17	20.86	**0**
C208	130.67	153.07	130.52	242.35	171.32	166.3	40.16	501.94	22.26	**0**
RC101	48.28	42.16	101.4	135.17	92.51	68.19	37.9	50.74	20.7	**12.16**
RC201	89.22	90.68	111.01	158.03	110	146.35	38.89	105.77	24.16	**11.16**

TABLE 8. Comparison of the HHHSA and different heuristics.

Instances	TS	TS-CP	GA	HGA	AC-TC	Meta-HSA	HHHSA	GAP
R1	1209.35	1214.86	1333	1220	1241.24	**1207.76**	1217.85	0.84%
R2	980.27	930.18	1124	985.69	961.11	977.19	**906.61**	**0.00%**
C1	**828.38**	829.77	872	851.05	843.55	838.47	830.82	0.29%
C2	**589.86**	604.84	641	620.12	611.12	605.41	590.60	0.13%
RC1	1389.22	1385.12	1547	**1366.62**	1419.14	1381.96	1386.39	1.45%
RC2	1117.44	1099.96	1343	1108.5	1119.24	1099.12	**1032.04**	**0.00%**
Avr.	1019.09	1013.98	1143.33	1025.33	1032.57	1018.32	**1003.87**	**0.00%**
Running time	–	160.8	709	1340	404.83	314.86	301.12	–

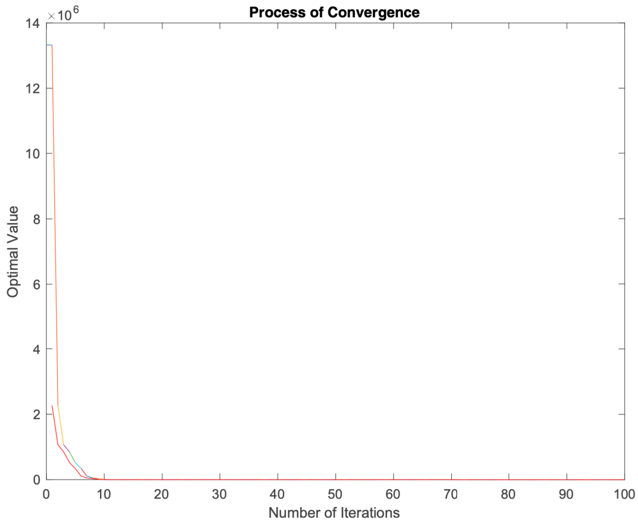

FIGURE 4. Convergence curves of C102.

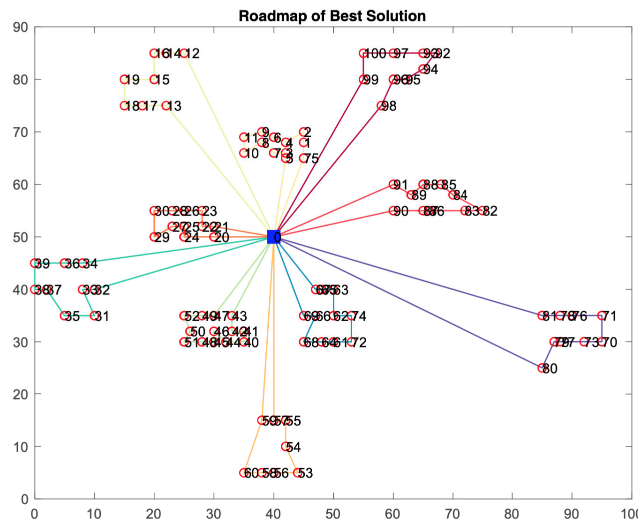

FIGURE 5. Route results of C102.

TABLE 9. Route information of C102.

Vehicle	Solution
1	0→81→78→76→71→70→73→77→79→80→0
2	0→32→33→31→35→37→38→39→36→34→0
3	0→57→55→54→53→56→58→60→59→0
4	0→43→42→41→40→44→46→45→48→51→50→52→49→47→0
5	0→90→87→86→83→82→84→85→88→89→91→0
6	0→20→24→25→27→29→30→28→26→23→22→21→0
7	0→98→96→95→94→92→93→97→100→99→0
8	0→67→65→63→62→74→72→61→64→68→66→69→0
9	0→5→3→7→8→10→11→9→6→4→2→1→75→0
10	0→13→17→18→19→15→16→14→12→0

TABLE 10. Route information of R204.

Vehicle	Solution
1	0→52→31→70→30→32→90→63→10→62→88→7→82→48→19→11 →64→49→36→47→46→8→45→17→84→85→98→37→100→13→0
2	0→6→94→95→92→42→43→15→57→41→22→75 →56→23→67→39→4→72→74→73→21→40→58→0
3	0→89→18→83→60→5→61→16→86→38→14→44 →91→93→99→96→59→97→87→2→53→0
4	0→27→69→1→50→76→3→79→33→81→9→51→20→66→65→71→35 →34→78→29→24→55→25→54→80→68→77→12→26→28→0

Similarly, Figs. 8 and 9 show the convergence curve and route result of RC201, respectively, demonstrating the effectiveness of the HHHSA in the optimization process. The algorithm significantly converged around generation 9 and finally converged at the current optimal solution of 1283.82 at generation 40. Table 11 shows the routing information of RC201 with evidence of the strong performance of HHHSA in handling complex instances under strict TW constraints.

As a new hybrid optimization algorithm, the HHHSA achieved good results in dealing with the VRPTW in different

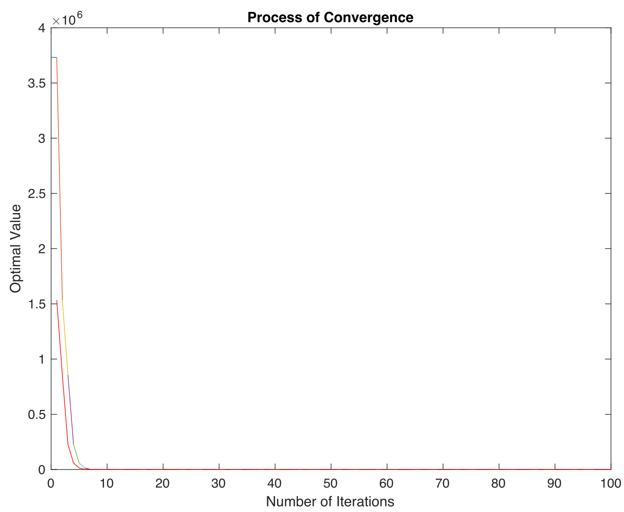

FIGURE 6. Convergence curves for R204.

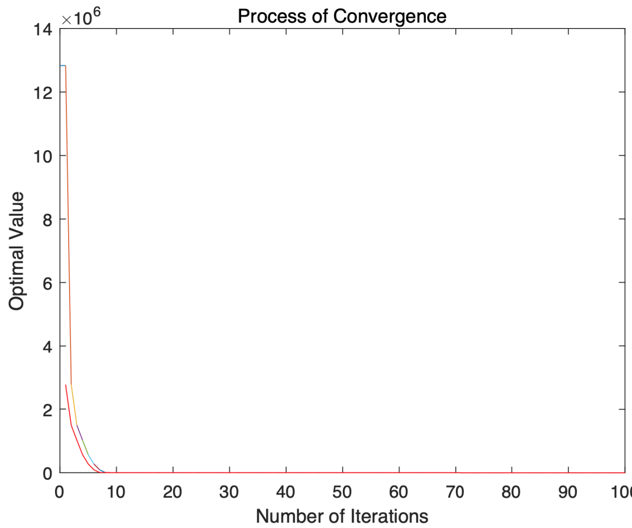

FIGURE 8. Convergence curves for RC201.

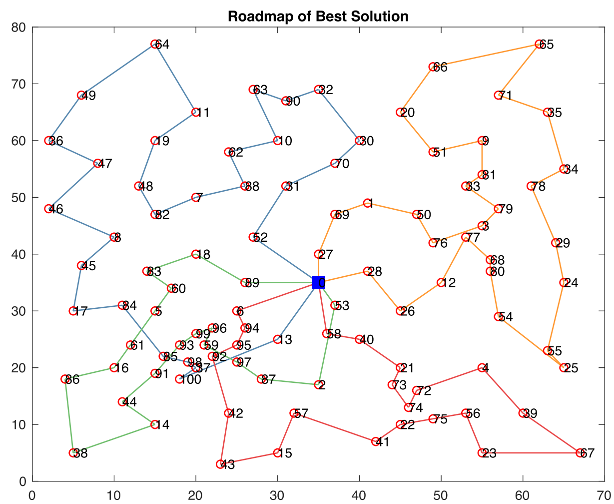

FIGURE 7. Route results for R204.

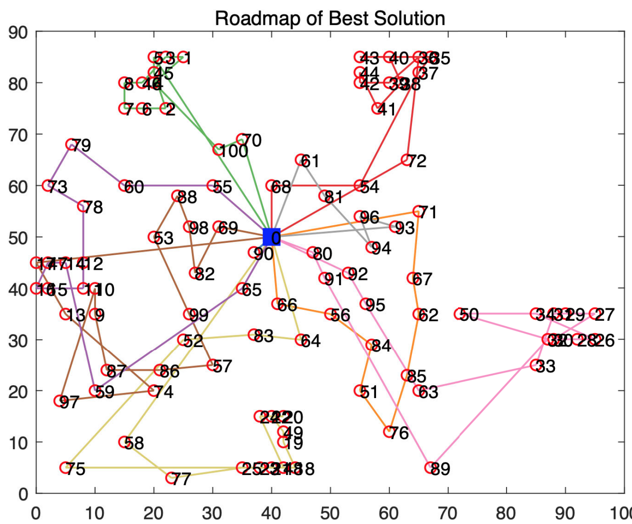

FIGURE 9. Route results for RC201.

TABLE 11. Route information of RC201.

Vehicle	Solution
1	0→72→36→39→42→44→41→38→40→43→35→37→54→68→0
2	0→90→0
3	0→5→45→2→6→7→8→46→3→1→4→100→70→0
4	0→71→67→62→85→76→51→84→56→66→0
5	0→64→83→52→75→23→21→18→19→49→22→20→24→48→25→77→58→0
6	0→69→82→98→88→53→99→57→86→87→9→10→97→74→13→17→0
7	0→92→95→63→33→31→29→27→30→28→26→34→50→32→89→91→80→0
8	0→61→81→94→96→93→0
9	0→65→59→14→47→16→15→11→12→78→73→79→60→55→0

scenarios. It achieved optimal or near-optimal results in several instances, particularly when dealing with complex and large-scale instances. Simultaneously, the stability and efficient runtime of the HHHSA confirmed its ability to solve real VRPTWs. There is still room for further optimization of the performance of the algorithm for specific types of problem instances. Future research can focus on the tuning of algorithmic parameters and the improvement of the search strategy to enhance the applicability and solution quality of the algorithm.

VI. CONCLUSION

We proposed the HHHSA to provide an efficient solution to the VRPTW. The VRPTW is a combinatorial optimization problem with complex constraints, where the core objective is to minimize the total distance traveled while satisfying a customer-specific service TW. The HHHSA was designed to overcome these challenges by combining robust HSAs and efficient heuristic techniques to improve the efficiency and quality of path planning.

Here, the HHHSA performance was thoroughly tested against the widely recognized Solomon benchmark dataset. The results showed that the HHHSA had a clear advantage in generating high-quality solutions and outperformed many traditional methods in terms of computational efficiency. The HHHSA demonstrated excellent performance and stability when dealing with large-scale and complex VRPTW instances. Its success was attributed to its ability to efficiently combine global and local search strategies, fully exploring the search space. The strategy encoding, solution generation, and optimization processes of the algorithm were carefully designed to ensure that constraints were effectively handled during the search process while optimizing the primary objective of the total distance traveled. Furthermore, the robustness and versatility demonstrated by the HHHSA across different instance types of the Solomon benchmark confirmed that it could handle different types of customer distributions, vehicle capacities, and TW constraints. Thus, the HHHSA is a powerful tool for solving VRPTWs in real logistics and supply chain problems.

The remarkable results of the HHHSA in solving VRPTW problems provide perspectives and methods for optimization in logistics and supply chain management. Furthermore, the study findings significantly contribute to the advancement of operations research and intelligent systems. Future research can explore the application of the HHHSA in other VRP variants and how it can be integrated with real-time data for more efficient and adaptive dynamic path planning.

REFERENCES

[1] A. Rushton, P. Croucher, and P. Baker, *The Handbook of Logistics and Distribution Management: Understanding the Supply Chain*. London, U.K.: Kogan Page Publishers, 2022.

[2] G. Kim, Y. S. Ong, T. Cheong, and P. S. Tan, "Solving the dynamic vehicle routing problem under traffic congestion," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2367–2380, Aug. 2016.

[3] C.-I. Hsu, S.-F. Hung, and H.-C. Li, "Vehicle routing problem with time-windows for perishable food delivery," *J. Food Eng.*, vol. 80, no. 2, pp. 465–475, May 2007.

[4] L. Junqueira, J. F. Oliveira, M. A. Carravilla, and R. Morabito, "An optimization model for the vehicle routing problem with practical three-dimensional loading constraints," *Int. Trans. Oper. Res.*, vol. 20, no. 5, pp. 645–666, Sep. 2013.

[5] C.-G. Lee, M. A. Epelman, C. C. White, and Y. A. Bozer, "A shortest path approach to the multiple-vehicle routing problem with split pick-ups," *Transp. Res. B, Methodol.*, vol. 40, no. 4, pp. 265–284, May 2006.

[6] M. A. Figliozzi, "Planning approximations to the average length of vehicle routing problems with varying customer demands and routing constraints," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2089, no. 1, pp. 1–8, Jan. 2008.

[7] G. Yu and Y. Yang, "Dynamic routing with real-time traffic information," *Oper. Res.*, vol. 19, no. 4, pp. 1033–1058, Dec. 2019.

[8] E. Demir, K. Huckle, A. Syntetos, A. Lahy, and M. Wilson, "Vehicle routing problem: Past and future," in *Contemporary Operations and Logistics: Achieving Excellence in Turbulent Times*. Berlin, Germany: Springer, 2019, pp. 97–117.

[9] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, Apr. 1987.

[10] M. Kim and J. Park, "Learning collaborative policies to solve NP-hard routing problems," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 10418–10430.

[11] N. A. El-Sherbeny, "Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods," *J. King Saud Univ.-Sci.*, vol. 22, no. 3, pp. 123–131, Jul. 2010.

[12] R. Baldacci, E. Bartolini, A. Mingozzi, and R. Roberti, "An exact solution framework for a broad class of vehicle routing problems," *Comput. Manage. Sci.*, vol. 7, no. 3, pp. 229–268, Jul. 2010.

[13] A. G. Qureshi, E. Taniguchi, and T. Yamada, "An analysis of exact VRPTW solutions on ITS data-based logistics instances," *Int. J. Intell. Transp. Syst. Res.*, vol. 10, no. 1, pp. 34–46, Jan. 2012.

[14] A. G. Qureshi, E. Taniguchi, and T. Yamada, "An exact solution approach for vehicle routing and scheduling problems with soft time windows," *Transp. Res. E, Logistics Transp. Rev.*, vol. 45, no. 6, pp. 960–977, Nov. 2009.

[15] B. L. Golden, S. Raghavan, and E. A. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*, vol. 43. New York, NY, USA: Springer, 2008.

[16] N. Balakrishnan, "Simple heuristics for the vehicle routeing problem with soft time windows," *J. Oper. Res. Soc.*, vol. 44, no. 3, pp. 279–287, Mar. 1993.

[17] K. C. Tan, L. H. Lee, Q. L. Zhu, and K. Ou, "Heuristic methods for vehicle routing problem with time windows," *Artif. Intell. Eng.*, vol. 15, no. 3, pp. 281–295, Jul. 2001.

[18] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992.

[19] Y. Shi, "Particle swarm optimization," *IEEE Connection*, vol. 2, no. 1, pp. 8–13, Feb. 2004.

[20] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statist. Sci.*, vol. 8, no. 1, pp. 10–15, 1993.

[21] C. Qi and Y. Sun, "An improved ant colony algorithm for VRPTW," in *Proc. Int. Conf. Comput. Sci. Softw. Eng.*, 2008, pp. 455–458.

[22] D. Zhang, S. Cai, F. Ye, Y.-W. Si, and T. T. Nguyen, "A hybrid algorithm for a vehicle routing problem with realistic constraints," *Inf. Sci.*, vols. 394–395, pp. 167–182, Jul. 2017.

[23] B. Bouchra, D. Btissam, and C. Mohammad, "A hybrid genetic algorithm for the static and dynamic vehicle routing problem with soft time windows," in *Proc. 3rd Int. Conf. Logistics Oper. Manage. (GOL)*, May 2016, pp. 1–9.

[24] 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*, vol. 8, pp. 93882–93893, 2020.

[25] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, Feb. 2001.

[26] D. Manjarres, I. Landa-Torres, S. Gil-Lopez, J. D. Ser, M. N. Bilbao, S. Salcedo-Sanz, and Z. W. Geem, "A survey on applications of the harmony search algorithm," *Eng. Appl. Artif. Intell.*, vol. 26, no. 8, pp. 1818–1831, 2013.

[27] E. Taha Yassen, M. Ayob, M. Z. Ahmad Nazri, and N. R. Sabar, "Meta-harmony search algorithm for the vehicle routing problem with time windows," *Inf. Sci.*, vol. 325, pp. 140–158, Dec. 2015.

[28] F. Maleki, M. Yousefikhoshbakht, and A. Rahati, "A hybrid self-adaptive global best harmony search algorithm for the vehicle routing problem with time windows," *Broad Res. Artif. Intell. Neurosci.*, vol. 8, no. 4, pp. 65–84, 2017.

[29] G. Laporte, "Classical and modern heuristics for the vehicle routing problem," *Int. Trans. Oper. Res.*, vol. 7, nos. 4–5, pp. 285–300, Sep. 2000.

[30] R. Martí, M. Gallego, A. Duarte, and E. G. Pardo, "Heuristics and metaheuristics for the maximum diversity problem," *J. Heuristics*, vol. 19, no. 4, pp. 591–615, Aug. 2013.

[31] Y. Zhang, J. Li, and L. Li, "An improved clustering-based harmony search algorithm (IC-HS)," in *Proc. SAI Intell. Syst. Conf.*, vol. 2. Cham, Switzerland: Springer, 2021, pp. 115–124.

[32] S. Jung and B. R. Moon, "A hybrid genetic algorithm for the vehicle routing problem with time windows," in *Proc. GECCO*, 2002, pp. 1309–1316.

[33] H. C. B. de Oliveira and G. C. Vasconcelos, "A hybrid search method for the vehicle routing problem with time windows," *Ann. Oper. Res.*, vol. 180, no. 1, pp. 125–144, Nov. 2010.

[34] J. Berger, M. Salois, and R. Begin, "A hybrid genetic algorithm for the vehicle routing problem with time windows," in *Proc. Conf. Can. Soc. Comput. Stud. Intell.*, Vancouver, BC, Canada. Cham, Switzerland: Springer, 1998, pp. 114–127.

[35] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "Time-window relaxations in vehicle routing heuristics," *J. Heuristics*, vol. 21, no. 3, pp. 329–358, Jun. 2015.

[36] Y. Rochat and É. D. Taillard, "Probabilistic diversification and intensification in local search for vehicle routing," *J. Heuristics*, vol. 1, no. 1, pp. 147–167, Sep. 1995.

[37] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," in *Proc. Int. Conf. Princ. Pract. Constraint Program.* Cham, Switzerland: Springer, 1998, pp. 417–431.

[38] J.-F. Cordeau, G. Laporte, and A. Mercier, "A unified Tabu search heuristic for vehicle routing problems with time windows," *J. Oper. Res. Soc.*, vol. 52, no. 8, pp. 928–936, Aug. 2001.

[39] K. C. Tan, Y. H. Chew, and L. H. Lee, "A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows," *Comput. Optim. Appl.*, vol. 34, no. 1, pp. 115–151, May 2006.

[40] G. B. Alvarenga, G. R. Mateus, and G. de Tomi, "A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 34, no. 6, pp. 1561–1584, Jun. 2007.

[41] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Appl. Math. Comput.*, vol. 188, no. 2, pp. 1567–1579, May 2007.

[42] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Appl. Math. Comput.*, vol. 198, no. 2, pp. 643–656, May 2008.

[43] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, "A self-adaptive global best harmony search algorithm for continuous optimization problems," *Appl. Math. Comput.*, vol. 216, no. 3, pp. 830–848, Apr. 2010.

[44] Q.-K. Pan, P. N. Suganthan, J. J. Liang, and M. F. Tasgetiren, "A local-best harmony search algorithm with dynamic sub-harmony memories for lot-streaming flow shop scheduling problem," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3252–3259, Apr. 2011.

[45] Y. M. Cheng, L. Li, T. Lansivaara, S. C. Chi, and Y. J. Sun, "An improved harmony search minimization algorithm using different slip surface generation methods for slope stability analysis," *Eng. Optim.*, vol. 40, no. 2, pp. 95–115, Feb. 2008.

[46] C.-M. Wang and Y.-F. Huang, "Self-adaptive harmony search algorithm for optimization," *Expert Syst. Appl.*, vol. 37, no. 4, pp. 2826–2837, Apr. 2010.

[47] P. Yadav, R. Kumar, S. K. Panda, and C. S. Chang, "An intelligent tuned harmony search algorithm for optimisation," *Inf. Sci.*, vol. 196, pp. 47–72, Aug. 2012.

[48] B. De Backer and V. Furnon, "Meta-heuristics in constraint programming experiments with Tabu search on the vehicle routing problem," in *Proc. 2nd Int. Conf. Metaheuristics*, Sophia Antipolis, France, 1997.

[49] K. C. Tan, L. H. Lee, and K. Ou, "Artificial intelligence heuristics in solving vehicle routing problems with time window constraints," *Eng. Appl. Artif. Intell.*, vol. 14, no. 6, pp. 825–837, Dec. 2001.

[50] B. Yu, Z. Z. Yang, and B. Z. Yao, "A hybrid algorithm for vehicle routing problem with time windows," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 435–441, 2011.

YANG ZHANG was born in Sheyang, Yancheng, Jiangsu, China, in 1990. She received the bachelor's degree in management from Yancheng Teachers University, China, in 2012, and the master's and Ph.D. degrees in computer engineering from Hosei University, Japan, in 2020 and 2023, respectively.

In April 2023, she joined Kanagawa University, Japan, as a Teaching Assistant. Her main research interests include heuristic algorithms, optimization problems, and mathematical computation.

Ms. Zhang was a recipient of several prestigious awards and honors, including the Scholarship for the 100th Anniversary of Hosei University and the Scholarship from the Ministry of Science and Culture of Japan.

JIACHENG LI was born in Jiangsu, China, in 1990. He received the bachelor's degree in computer engineering from Jiangsu University, China, in 2012, and the Ph.D. degree in computer engineering from Hosei University, Japan, in 2020. He is currently an Assistant Professor with the Department of Applied Systems and Mathematics, Kanagawa University, Japan. His research interests include fast algorithms, parallel algorithms, genetic algorithms, swarm intelligence algorithms, optimization theory, and vehicle routing problems. He has published more than 30 papers in these areas, including in refereed journals, conference proceedings, and book chapters. His awards and honors include the Hosei University Centennial Commemorative Scholarship and Japanese Ministry of Science and Culture Scholarship. He has been involved in more than ten conferences and workshops, serving as the program/general/organizing chair.