An Iterated Local Search Algorithm for the Heterogeneous Fixed Multi-compartment Vehicle Routing Problem

Hengrui Ma, Yan-e Hou, and Haizhe Xu

***Abstract*—This paper addresses the multi-compartment vehicle routing problem with heterogeneous fleets with maximum number limitations. To solve this problem, we propose an iterated local search algorithm integrating with multiple strategies to find a solution with minimum total cost. The proposed algorithm starts by generating an initial solution using a cheapest insertion heuristic. Then, a load adjustment strategy is applied to transfer customers from larger vehicle to smaller one. Four neighborhood operators such as relocation, 2-opt, exchange and or-opt, are applied sequentially to refine the solution, allowing for vehicle type adjustments during the local search process. Additionally, a perturbation mechanism based on destroy-and-repair principle is introduced to explore a wider search space. Meanwhile, an acceptance strategy based on route load variance is designed to permit the acceptance of sub-optimal solutions. Finally, the best global solution undergoes further optimization through a vehicle assignment strategy to improve vehicle allocation. Experimental results on benchmark instances demonstrate that the proposed algorithm outperforms existing methods.**

***Index Terms*—Heterogeneous fleet, iterated local search, multi-compartment vehicle, routing problem, metaheuristic.**

I. Introduction

THE diversification of goods transported in logistics supply chain has raised the demand for transportation vehicles. When transporting oil, medicine, fresh food and other items, the different requirements of the transportation environment for each item may lead to the fact that it cannot be distributed in the same vehicle, which will increase the cost of transportation. Multi-compartment vehicle is a kind of transport vehicle with multiple compartments to transport different types of goods at the same time. It can reduce the risk caused by mixed loading of different goods and improve the flexibility and safety of transportation.

The Multi-compartment Vehicle Routing Problem (MCVRP) is a variant of the Vehicle Routing Problem (VRP). Different from the traditional VRP, each vehicle in MCVRP has multiple independent compartments, and can distribute multiple types of items at the same time, so as to achieve the goal of efficient and safe transportation. The study of MCVRP was carried out relatively early, the first work of it [1] introduced the multi-compartment vehicle problem attribute in the transportation of oil products. In the study of MCVRP, some scholars [2], [3], [4], [5] have accomplished research on traditional MCVRP, where multi-compartment vehicles are used to solve the Capacitated Vehicle Routing Problem. As research progressed, some typical attributes of classic VRP[6], such as multiple depots [7], stochastic demand [8], and time windows [9], [10], have been gradually introduced into the study of MCVRP. The latest literature review on MCVRP can be referred to reference [11].

The Heterogeneous Multi-compartment Vehicle Routing Problem (HMCVRP) is a new variant of MCVRP, and it uses a variety of multi-compartment vehicles with different capacities and costs. Similar to the traditional heterogeneous vehicle routing problem, HMCVRP is also divided into mixed fleet Multi-compartment Vehicle Routing Problem (MSFMCVRP) and heterogeneous fixed Multi-compartment Vehicle Routing Problem (HFMCVRP) problems. The difference between them is that the number of vehicles for each type is limited in HFMCVRP. These problems are both widely applied in real application.

For the MSFMCVRP, Cornillier et al. [12] proposed an exact algorithm based on the branch-and-bound method to solve the problem of refueling station replenishment. The experimental results demonstrated that the algorithm effectively found optimal solutions for small and medium-sized problems, and the computation times within an acceptable range. Abdelaziz et al. [13] used a variable neighborhood search heuristic to find the best solutions, and they employed nine different vehicle types with varying numbers of compartments and capacities. For the HFMCVRP, Wang et al. [14] improved the Tabu Search (TS) algorithm by combining reactive and guiding mechanisms to propose a hybrid guiding-reactive TS algorithm to solve the heterogeneous multi-compartment VRP, achieving a balance between search depth and search breadth during the solution process. Avella et al. [15] conducted experiments using three types of vehicles, totaling six vehicles. In the experiments, they applied both branch-and-price and heuristic algorithms based on savings algorithm to solve the problem. The transportation costs obtained by the two algorithms decreased by 22%-25% and 12%-15%, respectively. Sethanan and Pitakaso [16] investigated how to optimize the transportation scheduling of fresh milk using a differential evolution algorithm. They proposed several improved metaheuristic algorithms based on the differential evolution algorithm, and they verified that the proposed algorithm performed excellently in opti-

Manuscript received November 14, 2024; revised March 14, 2025. This work was supported by National Natural Science Foundation of China (42471459) and Key Scientific and Technological Project of Henan Province of China (242102210080).

Hengrui Ma is a graduate student of Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng, 475004, China. (e-mail:a854303716@163.com)

Yan-e Hou is a professor of College of Computer and Information Engineering, Henan University, Kaifeng, 475004, China. (Corresponding author, e-mail: houyane@henu.edu.cn)

Haizhe Xu is a graduate student of College of Computer and Information Engineering, Henan University, Kaifeng, 475004, China. (e-mail:xuhaizhe@henu.edu.cn)

mizing transportation routes, reducing transportation costs, and improving efficiency. Urli and Kilby [17] conducted a study to provide theoretical support for fleet configuration by addressing the multi-compartment split-delivery vehicle routing problem, and then designed a hybrid model for the issued problem.

These research on the HMCVRP has promoted the development of this problem, but there are still some shortcomings in algorithmic performance. HMCVRP is a combinatorial optimization problem with high difficulty and is closely related to practical applications, thus it is still a challenge to explore efficient solution algorithms for HMCVRP.

Iterated Local Search (ILS) algorithm is a neighborhood-based metaheuristic, which can not only avoid falling into local optimal solution through perturbation, but also has high solving efficiency. ILS has remarkable performance in a variety of combinatorial optimization problems [18], and has potential application in solving HMCVRP problem.

Therefore, this paper studies HMCVRP that the number of each vehicle type is limited, named as HFMCVRP. The optimization objective of the address problem is to minimize the total cost. Then, an improved iterated local search combined multiple strategies, denoted as MS_ILS, is designed to solve this problem. In the framework of ILS algorithm, three strategies including Load Adjustment Strategy(LAS), Load Variance(LV) acceptance strategy, and Vehicle Model Optimizing Strategy(VOS) are designed and mixed with ILS to improve the quality of the solution. The designed algorithm has been experimentally verified and analyzed on benchmark case sets, confirming its effectiveness.

The remainder of paper is structured as follows. Section II is the description of problem model. The designed algorithm is given in Section III. Section IV is the experimental analysis,and the effectiveness of the improvement proposed in this paper is verified by comparative experiments. Finally, the conclusion is shown in Section V.

II. PROBLEM STATEMENT

A. Problem Definition

In this paper, the HFMCVRP problem can be described as follows. There are various types of multi-compartment vehicles in the distribution center named depot. Each vehicle needs to provide transportation services to some customers. Each compartment of vehicle has the maximum load capacity. The number of vehicles of each type is limited, that is to say, the number of certain vehicle type cannot exceed the maximum number limitation. Each customer has multiple goods to deliver, and the certain type of goods must be placed in the certain compartment of vehicle. Additionally, it is necessary to reasonably arrange vehicles to minimize the routing cost under the premise of meeting customer demands, constraints, and vehicle scheduling.

In order to solve this problem, this paper makes the following assumptions.

(1) The depot is unique, and it has a certain number of multi-compartment vehicles and products.

(2) The locations of the depot and all customers are known and fixed, all customers are interconnected. All vehicles need to start from the depot and return to the depot after completing the distribution task.

(3) All vehicles are multi-compartment vehicles with multiple independent compartments, and vehicles of the same type share identical parameters. The total number of each type of vehicle is limited.

(4) Each customer has multiple types goods to deliver. Each customer can only be serviced by a single vehicle, that is, any customer can only be satisfied by one vehicle.

(5) Each vehicle is responsible for one single delivery, and the products carried by the vehicle when departing from the depot cannot exceed the capacity of that vehicle type.

B. Model Description

Table I presents the parameters required for defining the model. Based on this, a mathematical model for the problem is established with the objective of minimizing the routing cost, as shown below.

Minimize

$$\sum_{i\in V}\sum_{j\in V}\sum_{m\in M} x_{ijm} * d_{ij} * c_m \tag{1}$$

Subject to:

$$\sum_{m\in M}\sum_{i\in C} x_{ijm} = \sum_{m\in M}\sum_{j\in C} x_{ijm} = 1, \forall i, j \in C \tag{2}$$

$$\sum_{i\in C} x_{0im} = \sum_{j\in C} x_{j0m} = 1, \forall m \in M \tag{3}$$

$$d_{ip} \le L_{ipm} \le Q_{pm}, \forall i \in V, p \in P, m \in M \tag{4}$$

$$\sum_{i\in V} X_{i0m} \le N_m, \forall m \in M \tag{5}$$

$$(L_{ipm} + d_{jp}) x_{ijm} \le Q_{pm}, \forall i \in V, p \in P, m \in M \tag{6}$$

$$X_{ijm} = \{0, 1\}, i \ne j, \forall i, j \in V, m \in M \tag{7}$$

Equation (1) is the objective function of the model, which represents the minimization of routing cost. Constraint (2) means that each customer can only be served by one vehicle. Constraint (3) ensures the vehicle start from depot and return to depot. Constraint (4) ensures that the quantity of products delivered by all vehicles doesn't exceed the maximum load capacity of compartment. Constraint (5) indicates that the number of vehicles used for each vehicle type cannot exceed the limit for that vehicle type. Constraint (6) describes the accumulation of each product in a vehicle. Constraint (7) indicates the decision variable.

III. ALGORITHM DESIGN

A. Algorithm Framework

The ILS algorithm consists of four parts: initial solution construction, perturbation, local search, and acceptance criteria. By iteratively applying perturbation, local search, and acceptance criteria to find the global best solution. Based on the ILS framework, this paper designs multiple strategies, including load adjustment, acceptance criteria, and vehicle model optimization, to solve the HFMCVRP effectively.

The algorithm in this paper is denoted MS_ILS, and its pseudocode is shown in Algorithm 1. In Algorithm 1, Step 2 calls the initial solution generation algorithm to obtain the initial solution. In step 3, the initial solution is improved

TABLE I
DEFINITION OF RELEVANT PARAMETERS

Parameter	Description
V	Set of nodes $V = \{0, 1, \ldots, n\}$, where node 0 represents the depot, and nodes 1 to n represent customers.
E	Set of edges $E = \{(i, j); i, j \in V\}$, representing all edges between node i and node j.
C	Set of customers $C = \{1, 2, ..., n\}$.
M	Set of fleet types $M = \{1, 2, 3, ..., m\}$.
P	Types of goods to be served, $P = \{1, 2, 3, ..., p\}$.
d_{ip}	Customer i has a known demand that is the demand of product p.
Q_{pm}	The capacity of corresponding compartment for vehicle m that the product p must be put into.
d_{ij}	The distance between nodes i and j, $d_{ij} = d_{ji}$; when $i = j$, d_{ij}=0.
c_m	Unit transportation cost for vehicle type m.
N_m	Maximum number of vehicle type m.
L_{ipm}	The total quantity of product p by vehicle m after visiting node i.
X_{ijm}	Decision variable,where $X_{ijk} = 1$ if vehicle m travels from node i to node j; otherwise, $X_{ijm} = 0$.

Algorithm 1 MS_ILS

Input: Iteration count $iter$, neighborhood size nb, perturbation destruction factor p, search strategy $rules$
Output: Global best solution S^*

1: S^*=Null; t=0;
2: S_0=GetInitSolution(); // Construct the initial solution
3: S_1=ApplyLAS(S_0); // Apply the LAS to improve S_0
4: S_c=LocalSearch(S_1,nb,$rules$);
5: S^*=S_c;
6: **while** $t <= iter$ **do**
7: S_b = Pertrub(S_c,p);
8: S_d=ApplyLAS(S_b);
9: S_n = LocalSearch(S_d,nb,$rules$);
10: **if** AcceptLV(S_c,S_n) **then**
11: S_c =S_n;
12: **end if**
13: **if** $S_c < S^*$ **then**
14: S^* = S_c;
15: **end if**
16: t++;
17: **end while**
18: S^*= ApplyVOS(S^*);
19: **return** S^*

by applying the LAS to ensure that a high quality initial solution is obtained. Steps 6 to 18 are the main loop of the algorithm, within which the perturbation method, LAS, local search and LV acceptance strategy are invoked in turn, and the LV is responsible for determining whether the new solution should be accepted. If the new solution meets the acceptance rules, it's accepted, then update the solution. Step 19 is to perform a post-optimization process on the global best solution obtained after the loop, and the solution is improved by the VOS designed in this paper. Finally, the algorithm outputs the global best solution.

B. Initial Solution Generation

For HFMCVRP, the maximum quantity constraint for each vehicle type makes the generation of the initial solution quite challenging. In order to solve this problem, we designed an initial solution construction method based on the cheapest insertion algorithm, which allows customers with large demand to be inserted into the route first, and then the others. To ensure the quality of the initial solution, the algorithm will look for the position with the cheapest insertion cost and then insert the customer into it. The steps of this initial solution construction are described as follows:

(1) Based on the vehicle set, construct M initial routes in the form of 0-0 (where M is the number of all vehicles, 0 represents the depot), and then assign appropriate vehicles to these routes.

(2) From the set C, select $\lceil M/2 \rceil$ large demand customers and insert them into the routes, obtained $\lceil M/2 \rceil$ routes in the form of 0-i-0. Next, randomly select M-$\lceil M/2 \rceil$ customers from C and also insert them into the routes, then remove these customers from C. Finally, this process generates M routes in the form of 0-i-0.

(3) Arrange the remaining customer in set C in descending order based on demand. Sequentially select customer j and traverse all positions in the routes to find the position that satisfies the vehicle capacity constraints and has the cheapest insertion cost, then insert customer j into the position and remove j from set C. If no suitable insertion position can be found, jump back to step (1) and continue execution.

(4) Repeat step (3) until the set C is empty.

C. Load Adjustment Strategy

Since the capacity and unit distance cost of each vehicle type are different, the load on each route may vary significantly. To address this,this paper designs a LAS to increase the load on smaller vehicles, thereby freeing up space on larger vehicles, which enhances the diversity and flexibility of local search.

Algorithm 2 presents the process of the LAS. The execution steps are as follows:

(1) Obtain the set of all vehicle types used in the current solution $M = \{1, 2, \ldots, m\}$. Traverse from the largest vehicle type m to type 2 in descending order, covering all non-minimal types. Set the current vehicle type $r = m$.

(2) When traversing vehicle type r, obtain the set of all routes using vehicle type $F_r = \{R_1, R_2, \ldots, R_i\}$, where $r \in M$ and $r \neq 1$.

(3) For each route R_i in F_r, traverse the nodes in route R_i and try to insert the nodes into the routes using vehicle type t ($t < r$). It ensures that the insertion doesn't violate the capacity constraint and doesn't increase the cost.

(4) After all routes in F_r have been traversed, $r = r$-1.

Algorithm 2 Load Adjustment Strategy

Input: Initial solution $initial_{sol}$
Output: Optimal solution $optimal_{sol}$

typenum=Get-typenum($initial_{sol}$); //Obtain the number of vehicle types
tournum=Get-tournum($initial_{sol}$); //Obtain the number of routes
for (i= typenum;i>1;i- -) **do**
  $Route_i$=Get-route($initial_{sol}$);
  for (j=1;j<= tournum;j++) **do**
    $Node_i$=Get-node($Route_i$);
    $Nodenum_i$=Get-nodenum($Route_i$);
    for (c=1;c<= $Nodenum_i$;c++) **do**
      $Route_j$=Get-route($initial_{sol}$);
      $Nodenum_j$=Get-nodenum($Route_j$);
      if $Route_j$=insert($Node_i$) **then**
        for (k=1;k<= $Nodenum_j$;k++) **do**
          $Route_j$=insert($Node_i$); //select the position in the route with the minimum cost to insert node.
        end for
      else
        $Route_j$=$Route_{j+1}$;
      end if
      $Node_i$= $Node_{i+1}$;
    end for
    $Route_i$ = $Route_{i+1}$;
  end for
end for
$optimal_{sol}$=update($initial_{sol}$);
return $optimal_{sol}$;

(5) If $r > 1$, return to step (2) and continue; otherwise, exit the strategy.

D. Local Search

In the local search process, this paper sequentially applies four neighborhood operators: relocate, exchange, 2-Opt, and Or-Opt to find the best solution within the neighborhood. None of these operators involve depot, and their functionalities are described as follows:

(1) Relocate: Select a node on the route, remove it from its original position, and then insert it into a position on the same or another route.

(2) Exchange: Choose two nodes from the same route or different routes, swap their positions, and then insert them back into their corresponding routes.

(3) 2-Opt: Within the same route, select two non-intersecting edges to break. After breaking, reverse the segment of nodes between these edges and then reconnect the route segments. Fig.1 illustrates the execution of 2-Opt on a route, where 0 represents the depot and the letters represent customers.

(4) Or-Opt: Select a segment of a specified length from the route and move it to another position within the same route or to a different route. Fig.2 illustrates the execution of Or-Opt.

During the local search process, when performing intra-path operations such as relocate, exchange, and Or-Opt, a

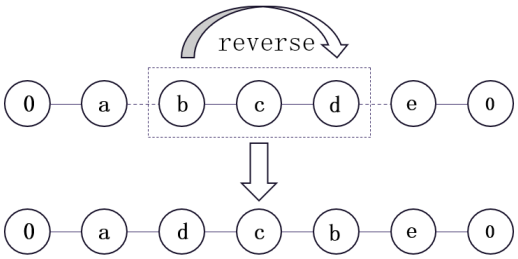

Fig. 1. Example Of 2-Opt Operator

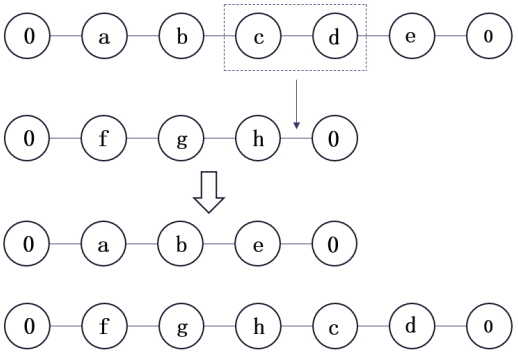

Fig. 2. Example Of Or-Opt Operator

vehicle type exchange operation is conducted. The underlying idea is that if the load capacities of the two routes don't exceed each other's capacity constraints and exchanging the vehicle types results in a cheaper route cost, then the vehicle types corresponding to the two route are swapped. The purpose of this step is to further reduce the route cost and facilitate the subsequent enhancement of the solution quality.

E. Perturbation

When solving vehicle routing problems,the results often get trapped in local optima, which affects the global search capability of the algorithm. By introducing the perturbation method, this issue can be effectively avoided. The perturbation method will actively disrupts the structure of the solution, forcing the algorithm to jump out of the local optimum and explore new spaces, which enhances the diversity of solutions and increases the likelihood of finding the global optimal solution.

In the selection of perturbation method, this paper uses the fixed random perturbation method. During each perturbation method, the algorithm randomly selects a node from the route, then removes several nodes within its neighborhood range, and temporarily stores these nodes in an array. Subsequently, the removed nodes are randomly inserted into the route. Finally, if the new obtained solution, is better than the perturbed solution, the new solution will replace the perturb solution. This process could be repeated many times, and them the best perturbation solution can be found.

Through the perturbation method described above, local optima can be effectively avoided, the diversity of the solution is enhanced.

F. Load Variance Acceptance Strategy

To enhance the diversity of the solution, the algorithm allows the acceptance of worse solutions. When a better

solution with a cheaper cost than the current solution is found, this solution is accepted. If the cost is higher than the current solution, an acceptance strategy called Load Variance based on the variance of route load rates is used, which tends to accept more stable solutions. Its definition is shown in Equation (8).

$$f(x) = \sum_{i=1}^{n} \frac{(p_i - P)^2}{n} \tag{8}$$

In (8), n represents the number of routes in solution x, P_i represents the load rate of vehicle on route R_i, which is calculated as the ratio of the load on the vehicle along route R_i to the maximum capacity of the vehicle type, P denotes the average load rate across all routes. Thus, the acceptance strategy for the current solution can be defined as shown in Equation (9).

$$S_c = \begin{cases} S_n, F(S_n) < F(S_c) \\ S_n, F(S_n) \geq F(S_c) \quad and \quad f(S_n) < f(S_c) \\ S_c, F(S_n) \geq F(S_c) \quad and \quad f(S_n) \geq f(S_c) \end{cases} \tag{9}$$

where S_c represents the current solution, S_n represents the new solution found through the local search, and $F(x)$ represents the objective function. In this strategy, the new solution is accepted if its $F(x)$ decreases. If the $F(x)$ remains the same or worsens but $f(x)$ decreases, the new solution is also accepted.

G. Vehicle model Optimizing Strategy

During the solution process of the multi-vehicle routing problems, unreasonable vehicle type allocation may occur. If the vehicle types are allocated simply based on traversing routes, it may result in a situation that the longer routes being assigned to larger vehicle types, which doesn't guarantee that the final solution is optimal under the current conditions. To avoid this, this paper employs a Vehicle model Optimizing Strategy to optimize the final solution, ensuring that the vehicle type allocation is better.

The process of this strategy is as follows:

1) Obtain the set of vehicle types used in the solution M = {1, 2,..., m}, and traverse from vehicle type 1 to m in order. Set the current vehicle type i = 1;

2) Find the set of routes corresponding to vehicle type i as f_i = {R_1, R_2,..., R_i}, and then obtain the set of routes with loading capacities not exceeding the maximum capacity of vehicle type i as F_I = {R_1, R_2,..., R_I}. If i is less than I, sort the routes R_1 to R_I in descending order based on their lengths, and then assign vehicle type i to these routes in the order.

3) i++ ; If $i \leq m$, jump back to (2) to continue; otherwise, exit the strategy.

This method ensures that the vehicle assignment across all routes in the final solution is optimized. If i is equal to I, this allocation will not affect the cost. If i is less than I, it indicates that there are routes in the set F_I = {R_1, R_2,..., R_I} that are assigned a larger vehicle type than i. Since vehicle allocation is done in descending order of route length, vehicle type i will be allocated to the longer routes, while the remaining shorter routes will enter the next iteration, waiting to be assigned to larger vehicle types than i. Therefore, it prevents situations where shorter routes are assigned to smaller vehicles and longer routes to larger vehicles.

Algorithm 3 Vehicle Model Optimizing Strategy

Input: Current solution $current_{sol}$
Output: Optimal solution $optimal_{sol}$

```
S1=Null,S2=Null;
typenum=Get-typenum(current_sol); //Obtain the number of vehicle types
tournum=Get-tournum(current_sol); //Obtain the number of routes
for (i=1;i≤ typenum;i++) do
    for (j=1;j≤ tournum;j++) do
        S1=Get-route(current_sol); //Iterate through all routes using vehicle type i
        S2=Descending(S1); //Sort all routes in descending order
        current_sol=Allocation(S2); //Assign vehicle type i to the routes in this order
    end for
end for
optimal_sol=current_sol
return optimal_sol
```

The process of the Vehicle Model Optimizing Strategy is shown in Algorithm 3.

IV. EXPERIMENTAL ANALYSIS

The proposed algorithm was implemented by C# and run on the PC with following specifications: Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz,16.0 GB RAM, and a Windows 11 64-bit operating system. The parameters setting are as follows. The number of iterations is set to 100, the neighborhood size is 30, the number of perturbations is 30, the perturbation parameter is set to 0.2, and every instance was executed independently by the MS_ILS algorithm 10 times. The MS_ILS algorithm was used to solve the benchmark dataset TC-8 come from reference [14], which is a modified from the international standard instance T-8 of the heterogeneous fixed fleet vehicle routing problem.

In the following table, AVG represents the average of the best values obtained over 10 runs. GAP is the percentage improvement of the best solution obtained by the corresponding algorithm over that one found by the first algorithm, NUM represents the number of customers in this case. BEST represents the best result obtained by the algorithm, and T represents the running time of the algorithm. All experimental environments were kept consistent.

A. Performance of Load Adjustment Strategy

This paper adopts a LAS aimed at redistributing the loading structure among routes, providing more capacity for high-load vehicle types as much as possible and increasing the adjustment space for subsequent local operators. To further verify the effectiveness of this strategy, a comparison is conducted using the inclusion or exclusion of the LAS as a variable.

The detailed data of best solutions is shown in Table II. Fig.3 gives the comparison results of the average solutions obtained by ILS and ILS combined with LAS strategy.

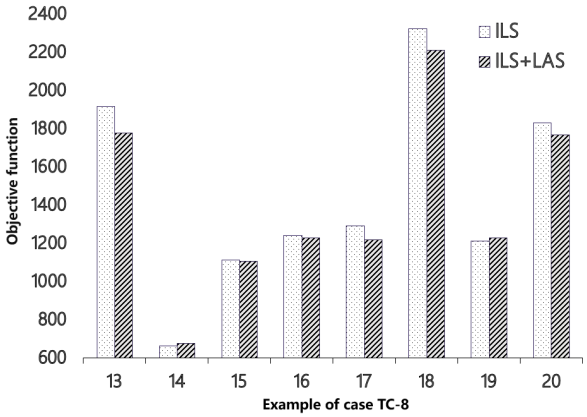

Fig. 3. Comparison Average solutions of ILS and ILS+LAS

TABLE II
COMPARISON RESULTS OF ILS AND ILS+LAS

TC-8	NUM	ILS	ILS+LAS	
		BEST	BEST	GAP/%
13	50	1750.35	1693.45	3.25
14	50	642.41	645.01	-0.40
15	50	1042.90	1050.97	-0.77
16	50	1207.65	1181.82	2.14
17	75	1210.32	1154.68	4.60
18	75	2131.60	2057.45	3.48
19	100	1168.61	1192.56	-2.05
20	100	1705.43	1689.42	0.94
AVG		1357.41	1333.17	1.40

From Table II, we can see that compared to the basic ILS, the ILS+LAS algorithm achieves an average cost improvement of 1.40%. Therefore, it can be concluded that the solution quality is significantly improved by introducing this strategy, verifying its effectiveness.

B. Performance of Load Variance Strategy

To test the performance of the load variance acceptance strategy on the TC-8 benchmark set, this paper conducted experiments with and without the LV acceptance strategy for comparison. The results are shown below.

The results in Fig.4 show that by introducing the LV acceptance strategy, the costs of average solutions with the LV acceptance strategy are superior to those without it in multiple cases. As shown in Table III, the best improvement of the ILS+LV algorithm compared to the basic ILS algorithm is 1.60% , and the maximum improvement reaches to 4.66%. Therefore, it is concluded that the introduction of the LV acceptance strategy significantly improves the solution quality, verifying the effectiveness of this strategy.

C. Performance of Vehicle Model Optimizing Strategy

As mentioned in the previous sections, the VOS Strategy can effectively prevent unreasonable vehicle assignments. Therefore, to evaluate the performance of Vehicle model Optimizing Strategy, comparative experiments were conducted with and without VOS, and the results are shown below.

From Fig.5, it is evident that in six instances, the average solutions obtained using the VOS outperform those without it. Specific data in Table IV shows that the average cost reduction of all the instances is 1.78%. The improvement in Case 13 is the most significant and the cost decreases

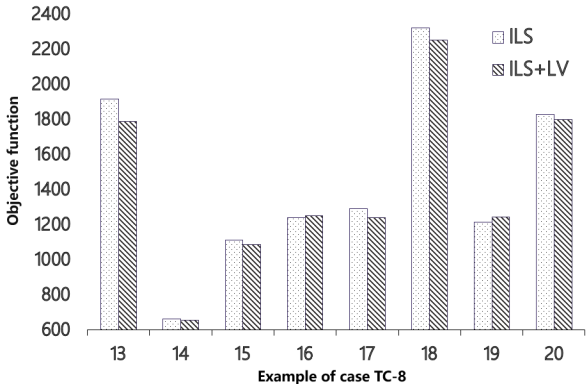

Fig. 4. Comparison Average Solutions of ILS and ILS+LV

TABLE III
COMPARISON RESULTS OF ILS AND ILS+LV

TC-8	NUM	ILS	ILS+LV	
		BEST	BEST	GAP/%
13	50	1750.35	1668.78	4.66
14	50	642.41	627.16	2.37
15	50	1042.90	1039.88	0.29
16	50	1207.65	1197.25	0.86
17	75	1210.32	1160.99	4.08
18	75	2131.60	2072.92	2.75
19	100	1168.61	1211.30	-3.65
20	100	1705.43	1681.01	1.43
AVG		1357.41	1332.41	1.60

by 6.50% compared with basic ILS. This confirms that the application of the VOS significantly enhances the solution quality.

D. Comparison of Hybrid Strategies

In the previous experiments, we have already verified that the improvement achieved by each single strategy is effective. In order to verify whether the combination of these strategies is effective, we conducted four comparative experiments. The specific data is shown in Table V.

From Table V, we can see that the best results are obtained using all three strategies. Compared with basic ILS, our proposed algorithm decrease the cost by 5.24% on average. Among of all the hybrid strategies, the improvement percentage value of ILS+LV+VOS is smallest just only 1.76%, but it is better than the ILS+LV mentioned above. The improvement of ILS+LAS+LV is the most significant, with an increase of 2.83%. When two or three strategies are used, the improvement percentage value increases, which indicates the combination of these strategies leads to better results.

E. Comparison with Existing Methods

In order to verify the overall effect of MS_ILS algorithm, we compared the proposed algorithm with the existing methods [14], including TS, TS-R, TS-G, RGTS. TS-R is the Reactive Tabu Search algorithm, TS-G represents the Guided Tabu Search algorithm, and RGTS is the Guided Reactive Tabu Search algorithm. The results are shown in table VI and table VII. The best result is marked in bold.

As reported in table VI, it can be seen that the MS_ILS algorithm outperforms other algorithms in half of the cases, and obtains the same number of optimal solutions as RGTS.

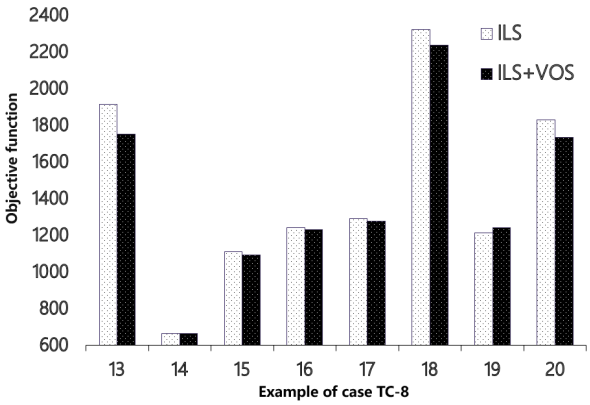

Fig. 5. Comparison Average Solutions of ILS and ILS+VOS

TABLE IV
COMPARISON RESULT OF ILS AND ILS+VOS

TC-8	NUM	ILS	ILS+VOS	
		BEST	BEST	GAP/%
13	50	1750.35	1636.53	6.50
14	50	642.41	632.01	1.62
15	50	1042.90	1053.85	-1.05
16	50	1207.65	1185.98	1.79
17	75	1210.32	1181.00	2.42
18	75	2131.60	2074.62	2.67
19	100	1168.61	1197.34	-2.46
20	100	1705.43	1659.15	2.71
AVG		1357.41	1327.56	1.78

Meanwhile, the average cost obtained by the MS_ILS algorithm is 1276.49, which is better than the RGTS algorithm.

Table VII mainly reflects the improvements of the algorithms. BKS represents the best result among all algorithms. From the table VII, it can be observed that the MS_ILS algorithm has found four best solutions, matching the count of RGTS. The average deviation value of the MS_ILS algorithm is 0.59%, outperforming RGTS's 1.04% in terms of algorithm improvement. Thus, it can be concluded that the MS_ILS algorithm not only achieved the same number of best solutions as the RGTS algorithm, but also has the best average results among all algorithms, which verifies the effectiveness of the improvement.

F. Convergence Analysis Of the Proposed Algorithm

To evaluate the performance of the proposed algorithm in solving the HFMCVRP, we analyze the algorithm's behavior after multiple iterations to determine whether it converges to the optimal solution after a sufficient number of iterations, ensuring its stability and effectiveness. In the experiment, we selected three different-sized cases as representatives: case 16 with 50 customers, case 17 with 75 customers, and case 19 with 100 customers. The convergence curves are shown in Fig.6, where the vertical axis represents the cost and the horizontal axis represents the number of iterations.

Through the convergence analysis of the algorithm, we have some findings. First, when the iteration number reaches 40, the algorithm converges quickly. Second, when the iteration number reaches 80, the algorithm still experiences some fluctuations. Finally, when the iteration number reaches 100, the algorithm tends to stabilize and approach the optimal cost. The results indicate that as the number of iterations increases, the algorithm gradually stabilizes, and the cost approaches the optimal solution. This demonstrates that the algorithm can effectively converge after multiple iterations, validating its good convergence property and proving its effectiveness in solving the HFMCVRP.

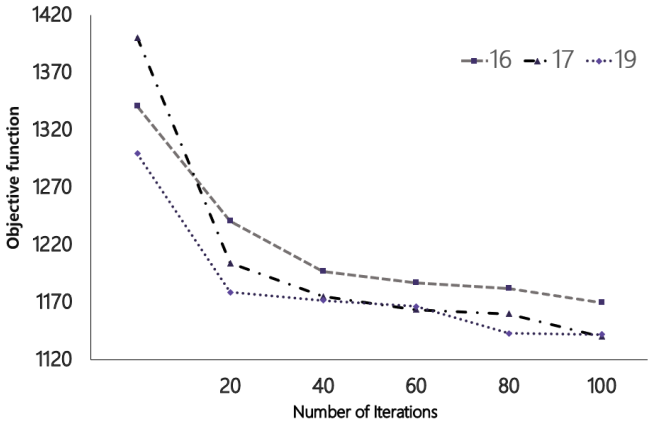

Fig. 6. Convergence Curves Of Different Problem Scale Instances

V. Conclusion

This paper mainly focuses on the HFMCVRP, which has different capacity and cost vehicles and each vehicle type of vehicle has limited number. Then, we proposed an improved iterated local search algorithm to solve it. These improvements include one method for generating initial solutions and three strategies such as the LAS strategy to provide adjustment flexibility, the LV strategy to ensure the diversification of solutions, and the VOS strategy to ensure optimal vehicle allocation. The proposed algorithm was verified on benchmark instances, and it outperforms existing methods for the same problem.

Further, we carried out some experiments to evaluate the effectiveness of designed strategies. There are some findings made from the experimental results. First, the results of using any single strategy improvement method are better than the basic ILS algorithm. When the basic ILS algorithm is combined with LAS, LV or VOS, the improvement percentage values are 1.40%, 1.60% and 1.78% respectively. Second, when the combination of two or three strategies is applied in the proposed algorithm, the better solutions can be found. The combination of all three strategies yielded the best results, the improvement percentage value reaches 5.24%. These results prove that our designed strategies are effective. In summary, the proposed algorithm proposed in this paper can effectively solve the HFMCVRP.

In future, more effective algorithm could be proposed owing to the complexity of the addressed problem. Meanwhile, more problem attributes such as multiple depots, electric vehicles, and split delivery can be introduced into the research of problem.

References

[1] N. Christofides, A. Mingozzi, and P. Toth, "The vehicle routing problem," *Traveling Salesman Problem*, vol. 10, pp. 315-338, 1979.
[2] M. Reed, A. Yiannakou, and R. Evering, "An ant colony algorithm for the multi-compartment vehicle routing problem," *Applied Soft Computing*, vol. 15, pp. 169-176, 2014.
[3] M. Abdulkader, Y. Cajpal, and T. Elmekkawy, "Hybridized ant colony algorithm for the multi compartment vehicle routing problem," *Applied Soft Computing*, vol. 37, pp. 196-203, 2015.

TABLE V
HYBRID STRATEGY IMPROVEMENT VERIFICATION

Strategy	Module				AVG	GAP/%	T/s
	ILS	LAS	LV	VOS			
ILS	√				1357.41	0.00	7.47
ILS+LAS+LV	√	√	√		1311.50	2.83	12.04
ILS+LAS+VOS	√	√		√	1322.98	2.36	11.04
ILS+LV+VOS	√		√	√	1328.12	1.76	10.66
ILS+LAS+LV+VOS(MS_ILS)	√	√	√	√	1276.49	5.24	10.10

TABLE VI
COMPARISON RESULTS OF DIFFERENT ALGORITHMS

TC-8	TS		TS-R		TS-G		RGTS		MS_ILS		
	BEST	T/s	BEST	T/s	BEST	T/s	BEST	T/s	BEST	AVG	T/s
13	1588.97	81	1686.08	45	1588.97	99	**1560.97**	155	1578.71	1736.98	4.66
14	730.07	103	652.13	206	718.90	73	625.06	148	**616.69**	649.53	3.08
15	1076.15	118	1077.11	84	1076.14	95	**1025.76**	168	1039.72	1090.69	7.11
16	1225.80	82	1203.39	68	1173.31	82	1168.25	80	**1159.17**	1223.41	4.96
17	1169.44	304	1156.89	475	1169.44	241	**1114.63**	725	1133.69	1191.94	12.35
18	2036.12	329	2013.31	421	1972.22	612	**1939.85**	908	1949.13	2199.41	14.33
19	1197.53	1978	1212.17	790	1200.56	1095	1186.70	2001	**1140.46**	1240.89	18.51
20	1680.11	589	1666.95	964	1651.32	1116	1628.49	1345	**1594.32**	1700.45	15.79
AVG	1338.02	448	1333.50	381	1318.86	426	1281.21	691	**1276.49**	1379.16	10.10

TABLE VII
COMPARISON DEVIATION VALUES COMPARED WITH BEST SOLUTION

TC-8	BKS	TS	TS-R	TS-G	RGTS	MS_ILS
13	1560.97	1.79	8.01	1.79	**0.00**	1.14
14	616.69	18.39	5.75	16.57	1.36	**0.00**
15	1025.76	4.91	5.01	4.91	**0.00**	1.36
16	1159.17	5.75	3.81	1.22	0.78	**0.00**
17	1114.63	4.92	3.79	4.92	**0.00**	1.71
18	1939.85	4.96	3.79	1.67	**0.00**	0.48
19	1140.46	5.00	6.29	5.27	4.05	**0.00**
20	1594.32	5.38	4.56	3.58	2.14	**0.00**
AVG	1268.98	6.39	5.13	4.99	1.04	**0.59**

[4] P. V. Silvestrin, and M. Ritt, "An iterated tabu search for the multi-compartment vehicle routing problem," *Computers & Operations Research*, vol. 81, pp. 192-202, 2017.

[5] H. Yahyaoui, I. Kaabachi, S. Krichen, and A. Dekdouk , "Two metaheuristic approaches for solving the multi-compartment vehicle routing problem," *Computers & Operations Research*, vol. 1, pp. 1-24, 2018.

[6] Y. Luo, X. F. Zhang, and J. K. Wu, "An Improved Hybrid Particle Swarm Optimization Path Planning Algorithm Based on Particle Reactivation," *IAENG International Journal of Computer Science*, vol. 51, no. 10, pp. 1534-1545, 2024.

[7] F. Cornillier, F. Boctor, and J. Renaud , "Heuristics for the multi-depot petrol station replenishment problem with time Windows," *European Journal of Operational Research*, vol. 220, no. 2, pp. 361-369, 2012.

[8] J. C. Coodson, "A priori policy evaluation and cyclic-order-based simulated annealing for the multi-compartment vehicle routing problem with stochastic demands," *European Journal of Operational Research*, vol. 241, no. 2, pp. 361-369, 2015.

[9] H. Kaabi, "Hybrid metaheuristic to solve the selective multi-compartment vehicle routing problem with time windows," in *Proceedings of the Second International Afro-European Conference for Industrial Advancement*, vol. 427, pp. 185-194, 2016.

[10] H. Kaabi, and K. Jabeur, "Hybrid algorithm for solving the multi-compartment vehicle routing problem with time windows and profit," in *12th International Conference on Informatics in Control, Automation and Robotics*, vol. 1, pp. 324-329, 2015.

[11] L. J. Sun, Y. X. Zhou, Y. Teng, and X. P. Hu, "Multi-compartment vehicle routing problem: Status and perspectives," *Systems Engineering-Theory & Practice*, vol. 41, no. 6, pp. 1535-1546, 2021.

[12] F. Cornillier, F. F. Boctor, G. Laporte, and J. Renaud , "An exact algorithm for the petrol station replenishment problem," *Journal of the Operational Research Society*, vol. 59, no. 5, pp. 607-615, 2008.

[13] F. Ben Abdelaziz, C. Roucairol, and C. Bacha, "Deliveries of liquid fuels to SNDP gas stations using vehicles with multiple compartments," in *2022 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 478-483, 2002.

[14] Q. Wang, Q. Ji, and C. Chiu, "Optimal routing for heterogeneous fixed fleets of multi-compartment vehicles," *Mathematical Problems in Engineering*, vol. 2014, pp. 1-11, 2015.

[15] P. Avella, M. Boccia, and A. Sforza, "Solving a fuel delivery problem by heuristic and exact approaches," *European Journal of Operational Research*, vol. 152, no. 1, pp. 170-179, 2004.

[16] K. Sethanan, and R. Pitakaso, "Differential evolution algorithms for scheduling raw milk transportation," *Computers and Electronics in Agriculture*, vol. 121, pp. 245-259, 2016.

[17] T. Urli, and P. Kilby, "Constraint-based fleet design optimisation for multi-compartment split-delivery rich vehicle Routing," in *International Conference on Principles and Practice of Constraint Programming*, vol. 10416, pp. 414-430, 2017.

[18] R. Elshaer and H. Awad, "A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants," *Computers & Industrial Engineering*, vol. 140, 106242, 2020.