A Selection Hyper-heuristic for the Multi-compartment Vehicle Routing Problem Considering Carbon Emission

Yan-e Hou, Lanxue Dang, Hengrui Ma, and Chunyang Zhang

***Abstract*—This paper presents a selection hyper-heuristic algorithm to solve the multi-compartment vehicle routing problem whose optimization objective includes vehicle fixed cost, total distance cost and carbon emissions cost. A multi-armed bandit method is used as high-level selection strategy to adaptively select low-level heuristics based on their cumulative rewards during the optimization process. To ensure effective exploration of the solution space, a ruin-and-recreated low-level heuristic has been developed, incorporating six commonly-used neighborhood operators. For the candidate solution obtained by low-level heuristic, a native acceptance rule is implemented to allow accepting inferior solutions by a certain probability, thereby maintaining the diversification of solutions. Experimental results on benchmark instances reveal that the proposed algorithm can effectively solve the problem addressed in this paper and its standard problem. When compared with existing state-of-the-art approaches, the proposed algorithm demonstrates superior performance and stability.**

***Index Terms*—multi-compartment vehicles; vehicle routing problem; hyper-heuristic; carbon emission; multi-armed bandit;**

I. INTRODUCTION

WITH the rapid development of society economic, global warming has become a real problem, which poses a threat to the survival of human beings and animals on earth [1]. It is generally believed that greenhouse gases are the main cause of global warming. In order to combat with the global warming, it is crucial to reducing the greenhouse gases emissions especially carbon dioxide. Many countries have developed relevant policies to achieve low carbon or carbon neutral targets. According to the International Energy Agency data statistics, carbon dioxide emissions from transportation sector are to blame for about 23% of the worlds total carbon emission in 2021. For transport companies, reducing transport costs and carbon emission has become an issue that cannot be ignored in their development process. Therefore, planning the routes of transport vehicles and taking into account both economic and carbon costs are essential for transport logistic companies.

Multi-compartment Vehicle Routing Problem (MCVRP) is a new variant of Vehicle Routing Problem (VRP)[2,3]. Different from traditional VRP, MCVRP uses the vehicles with multiple compartments to transport multiple types of products together. MCVRP has been received much attention in recent years [4], especially in the fields of petrol distribution [5, 6], waste collection [7], agricultural food transportation [8], and cold chain logistics [9], etc. Due to the higher complexity of MCVRP, some heuristics methods, such as ant colony optimization [10, 11], genetic algorithm [6], artificial bee colony [5], variable neighborhood search [5, 6] and iterated local search [12], have been developed. These algorithms have inspired the research of MCVRP, but they still need to be improved. Taking into account the great application value of MCVRP, it is worth investigating it further. It is a great challenge to develop an efficient algorithm for MCVRP especially for the complex real application.

Hyper-heuristic is a novel general-purpose heuristic algorithm that operates on the space of heuristics instead of solution space [13, 14]. A high-level strategy intelligently manages a set of pre-designed and problem-dependent low-level heuristics (LLHs), selecting or generating LLHs to solve cross-domain problems or different variants within the same domain. Hyper-heuristic has been successfully developed for some VRP variants [15-18]. To the best of my knowledge, there is currently no hyper-heuristic for MCVRP. The successful experience in VRP filed using hyper-heuristic algorithms encourages us to develop one for MCVRP.

In this paper, we aim to develop a hyper-heuristic for MCVRP considering carbon emission. The key contributions of this study is described. Firstly, we propose a selection hyper-heuristic, namely HHMAB, to tackle MCVRP whose objectives include fixed cost, operation cost and carbon emission costs. Secondly, we adopt a multi-armed bandit method with upper bound as a high-level selection strategy to select appropriate low-level heuristic by historical outcomes. For each solution obtained by the chosen low-level heuristic, we apply a naive acceptance rule to facilitate a broader acceptance of potentially worse solutions, as a high-level acceptance strategy. Thirdly, we have designed six regular neighborhood structures and an operator based on the ruin-and-recreate principle, in order to improve the solution. Finally, we conducted several experiments to assess the efficacy of the suggested algorithm. Additionally, we examined the functionalities of the devised high-level selection strategy and move acceptance strategy, and analyzed the influences of ruin-and-recreate low-level heuristic and its

Manuscript received May 6, 2024; revised September 8, 2024. This work was supported by the Key Research Project of Higher Education Institutions of Henan Province (23A520014) and the Key Scientific and Technological Project of Henan Province(242102210080).

Yan-e Hou is a professor of Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng, 475004, China(e-mail:houyane@henu.edu.cn)

Lanxue Dang is a professor of Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng, 475004, China(Corresponding author,e-mail:danglx@vip.henu.edu.cn)

Hengrui Ma is a graduate student of School of Computer and Information Engineering, Henan University, Kaifeng, 475004, China(e-mail:10475221353@henu.edu.cn)

Chunyang Zhang is a graduate student of School of Computer and Information Engineering, Henan University, Kaifeng, 475004, China(e-mail:cyzh@henu.edu.cn)

different destruction strength.

The remainder of this paper is described as follows. Section II summaries the related work about MCVRP and hyper-heuristic. Section III gives the description and mathematic model of the issued problem. The proposed algorithm is introduced in Section IV. Section V provides the experimental results and correlative analysis. Finally, we conclude and give the research directions in future.

II. Related Work

A. Multi-compartment vehicle routing problem

The MCVRP was early applied in petroleum products supplying, and the first study about it was provided by Brown and Graves [19]. After that, the MCVRP-related researches have been developed vigorously, owing to increasing availability of multi-compartments vehicles in different application fields. The general MCVRP only considers the capacity constraint, which differs from Capacitated Vehicle Routing Problem (CVRP) simply in the used vehicles. On this basis, many MCVRP variants are derived by adding path length limits [5-6, 11], time windows [20], multiple depots [21], flexible compartment sizes and other attributes. The relevant reviews of MCVRP can be found in [4, 22-23].

For basic MCVRP, exact methods based on branch and bound were developed to solve small scale problem [24, 25]. Muyldermans and Pang [7] presented a guided local search metaheuristic for the MCVRP and revealed the benefits of co-collection delivery by multi-compartment vehicles. Reed et al. [10] demonstrated the efficiency of an ant colony system for CVRP and MCVRP, which combined with 2-opt local search and k-means clustering method. In addition, they also gave a kind of generation strategy of MCVRP benchmark instances. Guo et al. [26] proposed an improved ant colony optimization algorithm combined with two types of variable neighborhood descent methods to solve 14 instances.

Fallahi et al. [27] introduced maximum route length constraints into their MCVRP model, and then proposed a memetic algorithm (MA) and a tabu search to solve the distribution of cattle food. The MA combined with path relinking and they both integrated with a local search procedure. The experimental results revealed their good performance. Abdulkader et al. [11] proposed an ant colony algorithm (ACS) by hybridizing several local search procedures to solve 28 new generated benchmark instances. The experimental results show their algorithm was superior to the method provided by [10]. Silvestrin and Ritt [28] dealt with a MCVRP problem that a customer may be visited multiple times by different vehicles and proposed an iterated tabu search (ITS) algorithm to solve single-visit and multi-visit MCVRP. Kaabachi et al. [5] presented a hybrid self-adaptive variable neighborhood search and artificial bee colonies approach to solve the petrol replenishment problem. These two algorithms were firstly tested on a set of randomly-generated small-sized MCVRP and benchmark instances proposed by [11]. The results still demonstrate that the proposed algorithms outperformed the existing methods developed by [10-11, 28]. Yahyaoui et al. [6] developed an adaptive variable neighborhood search and a genetic algorithm based on the partially matched crossover to solve the same problem defined in [5]. Hou et al. [12] presented a hybrid iterated local search for MCVRP, which is combined with a large neighborhood search as perturbation method and a simulated annealing-based acceptance rules. The results prove that the proposed method outperforms existing state-of-the-art six MCVRP algorithms. Recently, Guo et al. [29] addressed the MCVRP problem considering carbon emissions, whose optimization objective is the total transport cost rather than the total distance. They designed a three-dimensional ant colony optimization (TDACO) approach to solve this problem. Extensive experiment results show that TDACO can perform well on the issued problem and standard MCVRP.

B. Hyper-heuristic for specific VRP

Hyper-heuristic utilizes high-level strategy to select or generate heuristics, which can effectively solve cross-domain problem. In VRP fields, there are some successful specifically designed hyper-heuristics for basic VRP and variants. Marshall et al.[30] designed six selection methods and eight acceptance criteria to construct forty-eight combinations to compare their performance over randomly generated instances of CVRP. Garrido and Castro [15] presented a hill-climbing based hyper-heuristic to solve CVRP by employing some constructive-perturbative pairs of low-level heuristics to construct and improve partial solutions. Garrido and Riff [16] proposed an evolutionary-based hyper-heuristic approach for solving dynamic vehicle routing problem, which includes constructive, perturbative and noise heuristics three types of low-level heuristics in their collaborative framework. Tarhini et al. [31] proposed an evolutionary Cuckoo Search-based hyper-heuristic for the Vehicle Routing with Prioritized Customers (VRPC) and compared it with the modified Clarke Wright algorithm. The results indicate the solution selected by the proposed hyper-heuristic outperformed the modified Clarke Wright algorithm.

Recently, there emerge some hyper-heuristics which use machine learning methods as high-level strategy to evaluate and select low-level heuristics. Sabar et al. [17] proposed an effective hyper-heuristic for a large-scale vehicle routing problem with time windows (VRPTW), which uses column generation to construct an initial solution and then employs a multi-armed bandit selection approach to select low-level heuristics. Qin et al. [18] develops a hyper-heuristic based on policy-based reinforcement learning for heterogeneous vehicle routing problem (HVRP), which aims to minimize the maximum routing time of vehicles. Hou et al. [32] presented a two-stage selection hyper-heuristic for CVRP, which takes a set partitioning procedure as post-optimization technology. In the selection hyper-heuristic framework, a multi-armed bandit method is used to select low-level heuristics and the routes of improved solutions found at each iteration are recorded. Then, a set partitioning model was constructed according to the recorded routes, and was solved by CPLEX 12.6. The results on 82 CVRP instances reveal that the two-stage method is superior to existing CVRP approaches.

From this knowable, the excellent performance of the hyper-heuristic methods for VRP and its variants indicates that the hyper-heuristic algorithm has great application potential in solving MCVRP.

III. Problem Description and formulation

A. Problem Description

The multi-compartment vehicle routing problem can be described as follows. Suppose that there is a depot denoted as 0 and n customers in a region. There are some homogeneous multi-compartment vehicles located at the depot. Each customer has m different types of products to be served. Each vehicle has m compartments whose capacities may be different to serve different products. The product m must be located in its dedicated compartment, which just picks up product m. The vehicle starts from the depot, visits some customers, and then returns to the depot. Each customer is visited only once by only one vehicle. At any time, the total load of different products must not exceed the capacity of the compartment. Additionally, the total travel distance that is traveled by every vehicle cannot exceed the predefined maximum route length. The optimization objective of the addressed problem in this paper is to minimize the total cost including vehicle fixed cost, total distance and carbon missions cost.

TABLE I gives the definitions of parameters and decision variable for the addressed problem in this paper.

B. Mathematical Model

First of all, we give the definition of the optimization objective. The objective consists of fixed cost, total operation cost and carbon emission cost.

(1) Fixed cost

The fixed cost is related to the number of vehicle, which includes vehicles purchase or rent cost, driver wages and other cost. It is defined in Eq. (1), w_1 is the cost coefficient per vehicle.

$$F_1 = w_1 \sum_{k\in K}\sum_{j\in V} x_{0jk} \tag{1}$$

(2) Operation cost

The operation cost is correlation with total travel distance and it is calculated by Eq. (2), where w_2 is cost of per unit distance.

$$F_2 = w_2 \sum_{k\in K}\sum_{i\in V}\sum_{j\in V} d_{ij}x_{ijk} \tag{2}$$

(3) Carbon emission cost

Carbon emission cost mainly is related to fuel consumption. Like Guo et al.[29], we also adopt the comprehensive modal emission model(CEME) [33,34] to accurately calculate the cost of carbon emission. The fuel consumption in liter between node i and node j is calculated by Eq. (3), which includes engine, vehicle weight and vehicle driving three components. c_1, c_2 and c_3 are the coefficients of each component. So, the total carbon emission cost is calculated by Eq. (4), where w_3 is the cost of per liter.

$$FC_{ij} = c_1 d_{ij}/v_{ij} + c_2 d_{ij}(W + l_{ij}) + c_3 d_{ij}{v_{ij}}^2 \tag{3}$$

$$F_3 = w_3 \sum_{k\in K}\sum_{i\in V}\sum_{j\in V} FC_{ij}x_{ijk} \tag{4}$$

According to the values of parameters of CEME proposed by Guo et al. [29], W is set to 6350 kg, v_{ij} is set to constant 35 km/h. The values of c_1, c_2 and c_3 are calculated to be 3.66, $8.4 * 10^{-6}$ and $1.09 * 10^{-5}$ respectively. It should be noted that the unit of each variables has been uniformed. The fuel consumption is gotten by Eq.(3) in liter. The value of three weight coefficients w_1, w_2 and w_3 are set to 300, 5, and 12.6 respectively.

Then, based on the above analysis, the mathematical formulation of the issued MCVRP is defined based on [12], and its description is as follows.

Minimize:

$$Z = F_1 + F_2 + F_3 \tag{5}$$

Subject to:

$$\sum_{k\in K}\sum_{i\in C} x_{ijk} = 1, \forall j \in C \tag{6}$$

$$\sum_{k\in K}\sum_{j\in C} x_{ijk} = 1, \forall i \in C \tag{7}$$

$$\sum_{i\in C} x_{0ik} = \sum_{j\in C} x_{j0k} = 1, \forall k \in K \tag{8}$$

$$d_{im} \le Q_{im}^k \le Q_m, \forall i \in V, k \in K, m \in M \tag{9}$$

$$(Q_{im}^k + q_{jm})x_{ijk} \le Q_{jm}^k, \forall i \in V, k \in K, m \in M \tag{10}$$

$$\sum_{i\in V}\sum_{j\in V} d_{ij}x_{ijk} \le L_{max}, \forall k \in K \tag{11}$$

$$x_{ijk} \in \{0,1\}, \forall i \in V, j \in V, i \ne j, k \in K \tag{12}$$

Eq. (5) defines the composition optimization objective. Constraints (6) and (7) ensure that every customer must be served only once by one vehicle. Constraint (8) indicates that every vehicle must start from the depot and end at the depot. Constraint (9) ensures that the total quantity of each product must not exceed the capacity of the compartment at any time. Constraint (10) represents the accumulation of each product in a vehicle. Constraint (11) states the total travel distance of each vehicle cannot exceed the maximum route length. Constraint (12) defines the value of decision variable x_{ijk}.

IV. Proposed Algorithm

A. Description of HHMAB

The proposed algorithm is a single-solution based selection hyper-heuristic. It starts with a single solution obtained by sweep algorithm, and then iteratively executes the optimization process until it meets with termination condition. During each iteration, a high-level selection strategy chooses one low-level heuristic, which is then applied to the current solution in order to generate a new candidate solution. For the new obtained solution, the move acceptance rule is employed to decide to accept or refuse it. The pseudocode of HHMAB is described in Algorithm 1.

B. High-level Selection Strategy

1) Multi-armed Bandit Selection Method: The primary role of high-level selection strategy is to select one suitable low-level heuristic from a set of pre-defined low-level heuristics during the optimization process. It is important to design the evaluation and selection mechanism of low-level heuristic so as to improve the solution quality of the selection hyper-heuristic. The commonly used selection approaches, such as random, greedy[35], probability matching(PM)[36] and adaptive pursuit strategy (AP)[37] have been applied in selection hyper-heuristics. To some extent, the selection

TABLE I
SYMBOLS DEFINITION OF PARAMETERS

Parameters	Description
V	sets of depot and customers,$V = \{0, 1, 2, ...n\} = \{0\} \bigcup C$.
C	sets of customers, $C = \{1, 2, ...n\}$.
K	sets of multi-compartment vehicles, $K = \{1, 2, 3, ...k\}$.
M	sets of types of products, $M = \{1, 2, 3, ...m\}$.
d_{im}	total demand of product of type m of customer i.
Q_{im}	capacity of compartment that product of type m is placed.
Q_{im}^k	total quantity of product of type m by vehicle k after visiting node i.
d_{ij}	distance cost between node i and node j.
v_{ij}	speed of vehicle between node i and node j. v_{ij} is a constant.
l_{ij}	total load of products when the vehicle travels between node i and node j.
W	vehicle weight. W is a constant.
L_{max}	maximum route length of each path.
x_{ijk}	x_{ijk} is a decision variable. if the vehicle k directly visits node j after visiting node i, $x_{ijk} = 1$;otherwise $x_{ijk} = 0$.

Algorithm 1 HHMAB

Input: maximum iteration number $maxiter$, neighborhood list size nb, scaling factor C, the destruction strength list $plist$ and maximum trail number $iter$

Output: the best solution S^*

Get the list of low-level heuristic $hList$ and initialize its related parameters values;
Generate an initial solution S_c by the modified sweep algorithm;
$S^* = S_c$;
iteration variable $t = 0$;
while $t < maxiter$ **do**
    if one or more LLHs that has not been used **then**
        $cLLH$ Randomly select one LLH from $hList$;
    else
        Select one LLH $cLLH$ that makes Eq.(13) have maximum value;
    Apply $cLLH$ to current solution S_c and transfer parameters nb,$plist$ and $iter$, then get a new solution S_n;
    Calculate the score and accumulative reward of the chosen LLH;
    if S_n meets the acceptance rule **then**
        $S_c = S_n$;
    if S_n is superior to S^* **then**
        $S^* = S_n$;
    t++;
return S^*;

method with excellent performance both take into account both exploitation and exploration.

Multi-armed bandit method (MAB), a specific reinforcement learning method, is an online selection mechanism that select appropriate arm to make the expected reward maximum. For the selection hype-heuristic, the selection of low-level heuristic(LLH) is similar with the arm selection in MAB problem. Inspired by the successful experience of MAB as high-level selection strategy[17,32,38], we use an MAB with upper confidence bound method as the selection mechanism, where each LLH can be considered as an arm in MAB problem. At the beginning of execution, the used number and the cumulative reward value of each pre-defined LLH are both set to 0. If there exists one or more low-level heuristics that have not been used, a random selection strategy is applied, which selects one LLH randomly from the set of low-level heuristics. Once the chosen LLH is executed, the evaluation score will be calculated. For the chosen LLH, the values of related parameters and its cumulative reward are also updated. When all the low-level heuristics have been used once, the LLH that has the largest value defined in Eq. (13), will be selected in the subsequent execution process.

$$q_{i(t)} + C \times \sqrt{\frac{2 \times ln \sum_{j=1}^{K} n_{j(t)}}{n_{i(t)}}} \tag{13}$$

In Eq.(13), $q_{i(t)}$ is the empirical reward, which indicates the empirical reward of ith low-level heuristic obtained from begin to the time t. $q_{i(t)}$ is calculated by the Eq.(14).

$$q_{i(t+1)} = \frac{n_{i(t-1)} \times q_{i(t)} + r_{i(t)}}{n_{i(t)}} \tag{14}$$

Where $r_{i(t)}$ the score is based on credit score assignment of the ith low-level heuristic at time t .

The second component in Eq. (13) is an upper confidence bound that is related to the number of used times $n_{i(t)}$. K is the number of low-level heuristics. The parameter C controls the balance between the exploitation and exploration.

2) Score Assignment Mechanism: The high-level strategy of hyper-heuristic is responsible for selecting appropriate LLH during optimization process. The performance evaluation of each LLH is very important and it decides which one is the best suitable to be executed next. The main function of the credit score assignment is to record how well each LLH is executed.

When one selected LLH is applied on the current solution, a new solution will be generated. The difference between the optimization objective values of current solution and new solution could reflect the performance of chosen LLH. But, the difference value is usually large at early stage and it will be small or not change with the execution of the search process. It is obvious that the raw change of optimization objective cannot be suitable to measure the performance of LLH in the overall search process. Thus, we take the improvement rate as the score of chosen LLH, as defined in

Eq. (15).

$$r = \frac{f(S_c) - f(S^{'})}{f(S_c)} \times 100 \tag{15}$$

Where $f(S)$ and $f(S^{'})$ are the objective values of current solution and the new solution.

C. Move Acceptance Strategy

The move acceptance strategy determines accept or reject the neighborhood solution that is found by the chosen LLH. To keep the diversification of solutions, we adopt naïve acceptance rule to decide the new solution whether to be accepted or rejected, defined in Eq. (16).

$$S_n = \left\{ \begin{array}{l} R, f(R) \le f(S) \\ R, f(R) > f(S), p \ge 0.5, p \in [0, 1] \end{array} \right\} \tag{16}$$

Where S and R are current solution and new solution respectively, and their objective values are denoted as $f(S)$ and $f(R)$; p is a random number between 0 and 1. If R is better or equals to S, R replaces S. Otherwise, R replaces S with 50% probability. The naïve acceptance rule allows accepting worse solutions with certain probability, which can enhance the solution's diversity and may lead the algorithm to find better solutions in the follow-up search.

D. Low-level Heuristics

The proposed algorithm designed a set of neighborhood operators as low-level heuristics, both of which are problem-specific operators. The pool of low-level heuristics used in this paper includes six regular local-search based neighborhood operator and one ruin-and-recreate local search method.

1) Local search-based Low-level heuristics: There are six local-search based operators are designed to generate a new neighborhood solution by modifying the current solution without violating any problem constraints. It is worth noted that each node or edge move excludes the depot in order to keep the solution feasible. The descriptions of them are given as follows.

(1) *One Point Move*. Randomly select one node and move it to another position in the same route or another different route.

(2) *Two Points Swap*. It consists of inter-route and intra-route operators. Two different customers are randomly selected from the same route or two different routes and then their positions are exchanged.

(3) *Point Edge Swap*. Randomly select one node and an edge in the same route or from two different routes, and then swap them.

(4) *2-opt*. In a route, two non-adjacent edges are chosen randomly and then the nodes between them are reversed to create a new route. It is a classical operator for travelling salesman problem.

(5) *Or-opt*. Select a sequence of nodes and shift them to another position in the same route or to another different route. The number of moved nodes is usually an integer within a range. In this paper, when this operator is executed, the range is set to [2, 4].

(6) *Cross*. It occurs between two different routes. Two edges are selected and then broken to get four route segments. Then, two new edges are added respectively to connect two segments from two different routes to form two new routes.

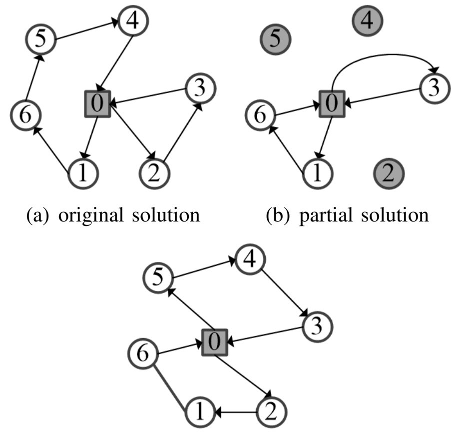


Fig. 1. example of the ruin-and-recreate operator

In addition, we employ two optimization techniques to improve the efficiency of these neighborhood operators due to the complexity of the researched problem. One is to use neighborhood list to limit the space of search to decrease computation time. For example, when a node i is selected by a neighborhood operator, the move operation just only happens between i and the nodes in its neighborhood list. The neighborhood list of i is constructed by the top t nearest nodes in distance, where t is the length of the neighborhood list. The other technique is to do pre-evaluation before executing any movement involving edges or nodes. If the movement results in violating capacity or maximum route length constraints or causing a bigger increase in the objective value, the movement will be discarded. In this way, the algorithm will not do useless search during the execution process.

2) Ruin-and-recreate Low-level heuristic: Taking into account the short-sighted shortcoming of local search based operators, we design a ruin-and-recreate low-level heuristic to explore relative bigger solution space. It is based on ruin-and-recreate principle [39], which has excellent performance in VRP fields by destroying and repairing solution.

Figure.1 shows the illustration of ruin-and-recreate operator. For an original solution shown in Figure.1 (a), customers 2, 4 and 5 are removed and an infeasible partial solution is obtained. Then, the removed customers are inserted into the partial solution to get a new solution. Thus, this operator can be regarded as a perturbation method to avoid falling into local optimality too early.

In this paper, we execute the destruction and repair procedures many times and take the best solution found in iteration process as the output solution. It is designed to exploit the benefits of destruction and reconstruction to search in bigger solution space. Algorithm 2 gives the description of the ruin-and-recreate LLH.

V. Computational Experiments

In this section, we will describe the parameters settings and conduct several experiments to prove the effectiveness of the proposed algorithm. First, we employed HHMAB to solve standard MCVRP problem and compare it with existing

Algorithm 2 ruin-and-recreate LLH

Input: current solution S_c, the destruction strength list $plist$ and maximum trail number $iter$

Output: the best solution S

Set $S = S_c$;
Set iteration variable $t = 0$;
Get minimize and maximize destruction factor from $plist$, and then assign to p_{min} and p_{max} respectively;
while $t < iter$ **do**
    $S_t = S_c$;
    Calculate the number of removed nodes by a number that is randomly selected between p_{min} and p_{max};
    Get a removed nodes list $rlist$ from S_t;
    Get a partial solution R by removing the nodes in $rlist$;
    while $rlist$ is not null **do**
        Select randomly a node i from $rlist$;
        Insert i to R by the cheapest insertion procedure and get a new partial solution $R^{'}$;
        Remove i from $rlist$;
        $R = R^{'}$;
    if R is superior to S **then**
        $S = R$;
    t++;
return S

eight state-of-the-art approaches. Second, we compared our proposed algorithm with TDACO [29] and HILS[12] on the addressed problem in this paper. Then, we carried out two experiments to verify the effectiveness of multi-armed bandit high-level selection strategy and native acceptance strategy used in our proposed algorithm. Finally, for ruin-and-recreate low-level heuristic, we tested its performance and the impacts of its different destruction strategies.

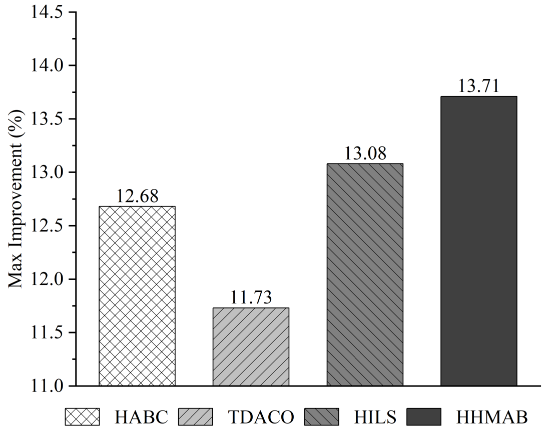
A. Experiments Setup and Parameters Settings

The proposed algorithm was programmed by C# in Visual Studio 2019, which is run on a personal computer with Intel i5-9500 3.0GHz CPU and 16GB RAM and running windows 10 64-bit operation systems. HHMAB was tested on 28 standard MCVRP instances designed by Abdulkader et al. [11]. Each instance is solved 10 times independently.

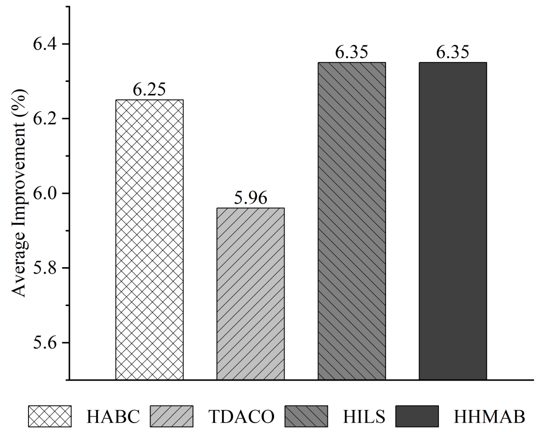
The parameters settings of HHMAB are described as following. The maximum iteration number is set to 200. The size of neighborhood list is set to 30. The scaling factor C in MAB is set to 4 according to the preliminary experiment. The destruction factor list includes 0.05 and 0.4 two numbers. The maximum attempt iteration number of ruin and recreate procedure is set to 30.

B. Results on Standard MCVRP

To validate the performance of the HHMAB algorithm, we use HHMAB to solve MCVRP problem whose optimization objective is the total distance. Subsequently, we compare HHMAB with existing state-of-the-art MCVRP methods, including ACS [10], HAC [11], HAVNS [5], HABC [5], AVNS [6], GAPMX [6], TDACO [29] and HILS [12]. The experimental results are shown in Table II.



(a) Maximum Improvement Percentage



(b) Average Improvement Percentage

Fig. 2. comparison of improvement percentage relative to ACS

As reported in Table II, HHMAB achieves the least total cost among of all the algorithms. Notably, both ACS and GAPMX fail to find any best solutions. TDACO just finds 1 best solution. On the other hand, HAC and HAVNS achieve 4 and 6 optimal solutions respectively while AVNS obtains 4 best solutions as well. Remarkably, HHMAB, HABC and HILS successfully hit 10 optimal solutions, demonstrating their good competition. It indicates that HHMAB possesses commendable capability in finding superior solutions compared to its counterparts on all instances.

Further, we take ACS as baseline method and compare HHMAB with HABC, TDACO and HILS algorithms, which are better than the remaining four comparison algorithms mentioned earlier. We calculate the improvement percentage achieved by each of these methods over ACS on every individual problem instance. The corresponding calculation results are illustrated in Fig.2.

Seen from Fig. 2, it can be observed that TDACO has least average and maximum improvement percentage values compared to others. HABC and HILS were specifically designed for MCVRP problem, and they exhibit greater competitiveness when compared against TDACO. However,

TABLE II
COMPARISON RESULTS OF HHMAB AND OTHER ALGORITHMS ON STANDARD MCVRP

Instance	ACS	HAC	HAVNS	HABC	AVNS	GAPMX	TDACO	HILS	HHMAB
vrpnc1a	569.564	550.70	550.70	550.42	550.70	551	550.70	550.70	550.70
vrpnc1b	569.118	551.94	506.18	502.83	506.18	551	548.77	548.77	548.77
vrpnc2a	957.525	890.68	881.84	890.68	890.68	901	876.76	875.13	870.89
vrpnc2b	954.856	918.96	918.96	914.00	918.96	914	888.92	880.31	884.36
vrpnc3a	964.132	874.07	878.83	875.24	880.84	894	870.80	865.23	865.23
vrpnc3b	959.327	895.26	898.64	895.39	900.63	914	867.51	865.82	865.21
vrpnc4a	1253.860	1126.12	1126.12	1126.12	1126.12	1243	1106.84	1089.91	1095.08
vrpnc4b	1254.510	1159.48	1159.48	1159.48	1159.48	1204	1126.79	1113.15	1110.97
vrpnc5a	1587.020	1444.29	1408.86	1385.71	1408.86	1470	1404.00	1380.69	1401.26
vrpnc5b	1640.590	1525.87	1515.25	1492.02	1515.25	1570	1467.39	1426.27	1415.69
vrpnc6a	573.274	557.49	557.49	557.49	557.49	560	557.49	557.49	557.49
vrpnc6b	573.378	559.37	503.82	503.94	505.56	563	555.43	555.43	555.43
vrpnc7a	997.007	928.24	928.24	928.24	928.24	980	930.16	927.62	932.43
vrpnv7b	969.337	932.67	932.67	932.67	932.67	960	933.76	930.66	932.27
vrpnc8a	963.381	882.96	888.50	885.50	890.20	880	880.75	876.67	876.55
vrpnc8b	976.212	884.85	887.35	885.14	889.30	884	880.89	874.82	874.82
vrpnc9a	1343.08	1228.88	1211.21	1221.00	1211.32	1221	1218.98	1212.66	1213.18
vrpnc9b	1346.63	1226.58	1221.26	1226.00	1220.58	1226	1215.30	1213.66	1214.27
vrpnc10a	1645.58	1511.65	1505.23	1504.68	1505.23	1680	1496.56	1493.77	1483.79
vrpnc10b	1659.94	1526.02	1517.65	1516.86	1517.65	1683	1508.04	1499.63	1501.39
vrpnc11a	1133.880	1110.45	1110.45	1110.45	1110.45	1130	1108.67	1115.97	1114.15
vrpnc11b	1247.490	1221.73	1221.73	1220.43	1221.73	1250	1223.87	1217.47	1210.36
vrpnc12a	911.861	912.64	901.22	901.15	901.36	913	907.76	902.59	905.79
vrpnc12b	970.833	950.79	935.62	923.25	936.25	960	954.43	950.79	950.79
vrpnc13a	1577.45	1556.46	1549.93	1549.82	1550.56	1570	1551.43	1563.66	1555.83
vrpnc13b	1572.11	1550.12	1538.67	1536.38	1540.37	1572	1555.44	1562.46	1557.83
vrpnc14a	914.857	911.35	911.35	911.35	911.35	913	912.59	911.98	911.98
vrpnc14b	970.933	965.84	965.84	965.84	965.84	973	967.80	966.30	966.66
Average	1109.205	1048.41	1040.47	1038.29	1041.21	1076.07	1038.14	1033.20	1032.94

in terms of overall performance, HHMAB has powerful ability to solve MCVRP problem. The results also mean that HHMAB can solve effectively standard MCVRP problem.

C. Results on MCVRP with Total Cost

In this section, we conducted a comparative analysis with state-of-the-art method for the MCVRP considering carbon emissions, namely TDACO [29]. Moreover, we also modify HILS [12] to suit the requirements of the addressed problem owing to its strong performance in solving MCVRP. The comparison results of three algorithms are shown in Table III.

The columns labeled Best, Mean and SD represent the best solution, average solution, and standard deviation respectively. The column T indicates the average computation time in seconds for each instance. For TDACO, execution time was determined as 1.2 times the number of customers [29]. Beyond that, bold formatting is used to highlight the optimal solutions for each instance within all tables for enhanced clarity.

As shown in Table III, it can be clearly seen that our proposed algorithm has the best performance. Compared with TDACO and HILS, HHMAB improves the best total cost on average by 1.61% and 0.19% respectively. HHMAB is better than TDACO on all remaining instances expect for two small instances. The maximum improvement percentage reaches to 5.15% on instance vrpnc5b. When compared with HILS, HHMAB obtains 22 best solutions, and the maximum improvement percentage is 2.10% on instance vrpnc12b. Although HILS gets 13 best solutions, HHMAB is better than it for the most instances especially for large-scale instances. Furthermore, HHMAB demonstrates remarkable stability among all algorithms based on average standard deviation values. Finally, the three algorithms exhibit average computation times of approximately 136, 99 and 94 seconds respectively. These findings indicate that our proposed algorithm HHMAB stands out as a high-efficiency method. To sum up, HHMAB is the most efficient and stable method of three comparison algorithms.

Further, we calculate the average best solution's objective value obtained by TDACO, HILS and HHMAB on different problem-scale instances, as shown in Table IV. Then, the average improvement percentage values of HHMAB compared with TDACO and HILS is denoted as g1 and g2 in Table IV. As shown in the table, three algorithms have almost equivalent performance on instances whose customer's number is between 50 and 120. Compared to TDACO and HILS, HHMAB has obvious improvement on large-scale instances. When the problem-scale of instances reaches to 150, HHMAB improves 2.38% and 0.16% respectively. When the number of customers is 199, the improvement values of HHMAB increase to 3.98% and 0.62%. It follows from the above that our proposed algorithm has a powerful ability to solve large-scale problem instances.

D. Analysis of Different High-level Section Strategies

To test the effectiveness of the multi-armed bandit high-level section strategy, we develop four variants of the pro-

TABLE III
COMPARISON RESULTS OF HHMAB AND OTHER ALGORITHMS ON MCVRP WITH TOTAL COST

Instance	N	TDACO			HILS				HHMAB			
		Best	Mean	SD	Best	Mean	SD	Time(s)	Best	Mean	SD	Time(s)
vrpnc1a	50	**5746**	5788	41	**5746**	5761	20	18	**5746**	5754	5	33
vrpnc1b	50	**5732**	5825	83	**5732**	5732	0	20	**5732**	5732	0	34
vrpnc2a	75	9668	9784	66	9584	9647	42	31	**9551**	9645	56	36
vrpnc2b	75	9776	9906	64	9600	9693	61	37	**9594**	9736	62	32
vrpnc3a	100	8990	9110	66	**8902**	8934	19	84	**8902**	8957	23	95
vrpnc3b	100	8999	9175	76	**8906**	8943	23	66	**8906**	8958	26	99
vrpnc4a	150	11984	12185	102	11708	11813	77	169	**11707**	11854	66	153
vrpnc4b	150	12198	12457	106	**11824**	11941	80	162	11826	11934	59	156
vrpnc5a	199	15738	16137	177	15282	15433	94	328	**15268**	15514	121	200
vrpnc5b	199	16515	16777	213	**15656**	16021	134	245	15664	16043	168	198
vrpnc6a	50	5869	5997	79	**5795**	5809	34	12	**5795**	5796	4	19
vrpnc6b	50	5841	5968	86	5808	5824	30	11	**5780**	5786	11	19
vrpnc7a	75	10346	10503	136	**10093**	10297	78	22	10266	10342	43	28
vrpnv7b	75	10410	10602	141	**10264**	10367	61	26	10265	10350	52	26
vrpnc8a	100	9076	9501	214	**8983**	9104	117	73	**8983**	9058	84	65
vrpnc8b	100	9092	9425	231	**8970**	9054	100	67	**8970**	9022	70	68
vrpnc9a	150	13446	13752	240	13196	13290	78	118	**13113**	13305	91	115
vrpnc9b	150	13379	13715	249	**13146**	13235	57	111	13149	13251	79	103
vrpnc10a	199	16833	17100	156	16229	16653	236	267	**16179**	16536	256	201
vrpnc10b	199	16916	17173	150	16605	16816	124	252	**16263**	16741	201	207
vrpnc11a	120	10481	10531	52	10429	10823	325	116	**10407**	10609	273	144
vrpnc11b	120	11194	11336	63	11110	11209	92	133	**11069**	11191	89	138
vrpnc12a	100	9735	9880	129	9489	9608	118	60	**9464**	9581	130	75
vrpnc12b	100	10191	10322	39	10115	10166	47	76	**9903**	10104	82	71
vrpnc13a	120	14417	14508	59	**14399**	14660	221	69	14423	14523	115	89
vrpnc13b	120	14406	14504	64	14440	14515	68	100	**14382**	14530	45	94
vrpnc14a	100	9844	9880	16	9837	9841	2	57	**9836**	9841	4	68
vrpnc14b	100	10241	10342	77	10228	10286	66	53	**10227**	10265	31	61
Average		10967	11149	113	10788	10910	86	99	**10763**	10891	80	94

TABLE IV
COMPARISON RESULTS OF THREE ALGORITHMS ON DIFFERENT PROBLEM-SCALE INSTANCES

Nodes	TDACO	HILS	HHMAB	g1(%)	g2(%)
50	5795	5770	5763	0.58	0.12
75	10050	9885	9919	1.30	-0.34
100	9521	9429	9399	1.28	0.32
120	12625	12595	12570	0.43	0.19
150	12752	12469	12449	2.38	0.16
199	16501	15943	15844	3.98	0.62

posed algorithm to compare their high-level section strategies.

We employ four commonly used selection methods in hyper-heuristic, namely random selection (SR), roulette wheel selection (RW), probability matching (PM) and adaptive pursuit strategy (AP), to evaluate the performance of our MAB selection strategy. The four variants, which are called HH_SR, HH_RW, HH_PM and HH_AP respectively, are based on HHMAB and differ only in terms of their selection methods. For HH_PM and HH_AP, the parameters setting of selection methods can be found in our previous study[32].

Table V summaries the comparative results of all five algorithms. Columns BestAvg, SDAvg and TimeAvg denote the average values of best solutions, standard deviations, and average computation times across all problem instances correspondingly. Column BestNum represents the count of best solutions obtained by each algorithm across all the problem instances.

As presented in Table V, it is evident that the HHMAB algorithm outperforms the other four variants. Among of all algorithms, HH_SR performs the poorest due to its randomness without considering individual performances. In contrast, the remaining three algorithms and HHMAB utilize rewards obtained by each LLH to adaptively adjust their selection probabilities. Obviously, HHMAB shows remarkable performance and finds 20 best solutions, which has higher success rate than others. In terms of computation time, HHMAB requires additional computation efforts to evaluate low-level heuristics performance. Even so, this slight increase in execution times does not undermine the effectiveness of HHMAB. Thus, employing a multi-armed bandit selection strategy proves effective and suitable for our proposed algorithm.

E. Analysis of Different Acceptance Strategies

In this section, we test the influence of different solution acceptance strategies on the proposed algorithm. Consequently, we construct two variants based on HHMAB using All Move(AM) acceptance and Only Improvement(OI) acceptance. The former always accepts any found solution regardless of its better or worse. The latter only accepts the solution better than current solution. These two variants are denoted as HHMAB_AM and HHMAB_OI respectively.

TABLE V
SUMMARY COMPARISON RESULTS OF FIVE ALGORITHMS

Methods	BestAvg	SDAvg	BestNum	TimeAvg(s)
HH_SR	10821	91	4	61
HH_RW	10818	97	8	70
HH_SR	10821	95	8	58
HH_RW	10818	85	6	59
HHMAB	10763	80	20	94

TABLE VI
COMPARISON OF THREE ALGORITHMS WITH DIFFERENT ACCEPTANCE STRATEGIES

Methods	BestAvg	SDAvg	BestNum	TimeAvg(s)
HH_AM	10792	84	12	93
HH_OI	10810	87	11	92
HHMAB	10763	80	21	94

TABLE VII
COMPARISON RESULTS OF FOUR ALGORITHMS

Methods	BestAvg	SDAvg	BestNum	TimeAvg(s)
HHMAB_NR	11160	206	0	4
HHMAB_F	10818	87	13	89
HHMAB_DR	10810	94	6	88
HHMAB	10763	80	22	94

According to the results of HHMAB and its two variants reported in Table VI, it reveals that HHMAB achieves the lowest average total cost, outperforming the comparisons variants. More specifically, HHMAB obtains 21 best solutions with a success rate of 0.75. In contrast, HHMAB_AM and HHMAB_OI find 12 and 11 best solutions respectively, resulting in a lower success rates. These findings demonstrate that HHMAB allow for occasional acceptance of worse ones through a probabilistic mechanism, maintaining algorithmic diversity. Notably, all algorithms have similar computational overheads. Overall, HHMAB using a native acceptance rule can consistently get better solutions in a reasonable execution time.

F. Analysis of Ruin-and-recreate LLH and its destruction Strategy

In this section, we analyze the performance ruin-and-recreate LLH and the impacts of its different destruction strategy. We construct three variants of HHMAB to carry out the experiments, namely HHMAB_NR, HHMAB_F and HHMAB_DR respectively. The first variant removes the ruin-and-recreate LLH. HHMAB_F uses a fixed destruction factor of 0.2. For HHMAB_DR, the destruction factor is randomly selected from the list {0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4}. Then, three variants are used to solve all the problems and the calculation results are shown in Table VII.

Seen from the Table VII, we have some findings. First, using ruin-and-recreate LLH in HHMAB algorithm can obtain better solutions. Among of fours algorithms, HHMAB_NR is the poorest because the neighborhood-based operators just only explore a relative smaller solution space, resulting in poor performance. When the ruin-and-recreate LLH is used in other algorithms, their performance has been significantly improved. Second, HHMAB achieves the best average solution among all algorithms. Compared with HHMAB_F and HHMAB_DR, HHMAB decreases total cost on average by 0.51% and 0.44% respectively. Besides, HHMAB finds 22 best solutions and performs better on large-scale instances demonstrating its superior optimization ability. Finally, the ruin-and-recreate LLH needs more computation times but does not be time-consuming.

VI. CONCLUSION

This paper deals with the multi-compartment vehicle routing problem with a focus on carbon emissions. The optimization objective comprises of vehicle cost, operation cost and carbon emissions cost. To tackle this problem effectively, we designed a selection hyper-heuristic called HHMAB, which employs multi-armed bandit algorithm as high-level selection strategy to choose low-level heuristics. The multi-armed bandit high-level selection can take advantage of the historical information of low-level heuristics to choose more appropriate ones during the execution of the hyper-heuristic. The set of low-level heuristics including six neighborhood search-based and one ruin-and-recreate operators were developed to explore the solution space. In addition, we adopt a probability acceptance rule to determine whether to accept or refuse newly obtained solutions.

We conduct several experiments to evaluate HHMAB. The experimental results display the superiority of HHMAB over the existing methods. For the standard MCVRP problem, HHMAB is more competitive than eight state-of-the-art MCVRP approaches. Moreover, we also make some discoveries from the experimental findings. Firstly, the multi-armed bandit high-level selection strategy outperforms these commonly used selection strategies, such as random selection, roulette wheel selection, probability matching and adaptive pursuit strategy, etc. Secondly, allowing accepting worse solutions can help keep solutions diversification, which may lead to find better solutions in the subsequent search process of the algorithm. Finally, the ruin-and-recreate low-level heuristic can explore a larger solution space to hit better solutions. We also find that the destruction strategy, with a destruction factor randomly selected from a range, can flexibly adjust the destruction strength, thereby searching better solution. These strategies mentioned above used in HHMAB makes it more effective than existing algorithms.

In future, we will extend the suggested algorithm for other MCVRP problems with additional attributes, such as time windows, heterogeneous vehicles, and multiple depots, etc. Additionally, other reinforcement learning methods may be considered as a high-level selection strategy to effectively select low-level heuristics.

REFERENCES

[1] Huang Y, Shen L and Liu H, "Grey relational analysis, principal component analysis and forecasting of carbon emissions based on long short-term memory in China, " *Journal of Cleaner Production*, vol. 29, pp. 415-423, 2019.

[2] Laporte G, " Fifty years of vehicle routing, " *Transportation Science*, vol. 43, pp. 408-416, 2009.

[3] Lahyani R, Khemakhem M and Semet F, " Rich vehicle routing problems: From a taxonomy to a definition, " *European Journal of Operational Research*, vol. 241, no. 1, pp. 1-14, 2015.

[4] Ostermeier M, Henke T, Hübner A and Wäscher G, " Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions, " *European Journal of Operational Research*, vol. 292, pp. 799-817, 2021.

[5] Kaabachi I, Yahyaoui H, Krichen S and Dekdouk A, " Measuring and evaluating hybrid metaheuristics for solving the multi-compartment vehicle routing problem," *Measurement*, vol. 141, pp. 407-419, 2019.

[6] Yahyaoui H, Kaabachi I, Krichen S and Dekdouk A, " Two metaheuristic approaches for solving the multi-compartment vehicle routing problem," *Operation Research*, vol. 20, pp. 2085-2108, 2020.

[7] Muyldermans L and Pang G, " A guided local search procedure for the multi-compartment capacitated arc routing problem," *Computers & Operations Research*, vol. 37, pp. 1662-1673, 2010.

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

[9] Chen L, Liu Y and Langevin A, "A multi-compartment vehicle routing problem in cold-chain distribution," *Computers & Operations Research*, vol. 111, pp. 58-66, 2019.

[10] Reed M, Yiannakou A and Evering R, " An ant colony algorithm for the multi-compartment vehicle routing problem," *Applied Soft Computing*, vol. 15, pp. 169-176, 2014.

[11] Abdulkader M M S, Gajpal Y and ElMekkawy T Y, " Hybridized ant colony algorithm for the Multi-Compartment Vehicle Routing Problem," *Applied Soft Computing*, vol. 37, pp. 196-203, 2015.

[12] Hou Y E, Wang C X, Wang C R and Fan G J, " A hybrid iterated local search algorithm for the multi-compartment vehicle routing problem," *Journal of Intelligent & Fuzzy Systems*, vol. 45, no. 1, pp. 257-268, 2023.

[13] Burke E K, Gendreau M, Hyde M, Kendall G and Ochoa G, "Hyper-heuristics: A survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695-1724, 2013.

[14] Drake J H, Kheiri A, Özcan E and Burke E K, " Recent advances in selection hyperheuristics," *European Journal of Operational Research*, vol. 285, no. 2, pp. 405-428, 2020.

[15] Garrido P and Castro C, " Stable solving of CVRPs using hyperheuristics, in " *Proceedings of the 11th Annual conference on Genetic and Evolutionary Computation*, pp. 255-262, 2009.

[16] Garrido P and Riff M C," DVRP: a hard dynamic combinatorial optimization problem tackled by an evolutionary hyper-heuristic," *Journal of Heuristics*, vol. 16, no. 3, pp. 795-834, 2010.

[17] Sabar N R, Zhang X J and Song A, " A math-hyper-heuristic approach for large-scale vehicle routing problems with time windows, in " *2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 830-837, 2015.

[18] Qin W, Zhuang Z, Huang Z and Huang H, " novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem," *Computers & Industrial Engineering*, vol. 156, 107252, 2021.

[19] Brown G G and Graves C W, " Real-time dispatch of petroleum tank trucks," *Management Science*, vol. 27, no. 1, pp. 19-32, 1981.

[20] Chen J and Shi J, " A multi-compartment vehicle routing problem with time windows for urban distribution-A comparison study on particle swarm optimization algorithms," *Computers & Industrial Engineering*, vol. 133, pp.95-106, 2019.

[21] Mahdi A and Nadia S, " Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search," *Omega*, vol. 76, pp. 85-99, 2018.

[22] Derigs U, Gottlieb J, Kalkoff J, Piesche M, Rothlauf F and Vogel U, " Vehicle routing with compartments: Applications, modelling and heuristics," *OR Spectrum*, vol. 33, no. 4, pp. 855-914, 2011.

[23] Coelho L C and Laporte G, " Classification, models and exact algorithms for multi-compartment delivery problem," *European Journal of Operational Research*, vol. 242, no. 3, pp. 854-864, 2015.

[24] Archetti C, Campbell A M, and Speranza M G, " Multicommodity vs. single-commodity routing," *Transportation Science*, vol. 50, no. 2, pp. 461-472, 2016.

[25] Mirzaei S, and Wohlk S, " Erratum to: A branch-and-price algorithm for two multi-compartment vehicle routing problem," *EURO Journal on Transportation and Logistics*, vol. 6, no. 2, pp. 185-218, 2017.

[26] Guo N, Qian B, Hu R, Jin H P and Xiang F H, " A hybrid Ant Colony Optimization Algorithm for Multi-compartment Vehicle Routing Problem," *Complexity*, 8839526, 2020.

[27] Fallahi A E, Prins C, and Wolfler C R, " A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem," *Computers & Operations Research*, vol. 35, no. 5, pp. 1725-1741, 2008.

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

[29] Guo N, Qian B, Na J, Hu R and Mao J L, " A three-dimensional ant colony optimization algorithm for multi-compartment vehicle routing problem considering carbon emissions," *Applied Soft Computing*, vol. 127, 109326, 2022.

[30] Marshall R J, Johnston M and Zhang M, " Hyper-heuristic operator selection and acceptance criteria, in " *European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 99-113, 2015.

[31] Tarhini A, Danach K and Harfouche A, " Swarm intelligence-based hyper-heuristic for the vehicle routing problem with prioritized customers," *Annals of Operations Research*, vol. 308, pp. 549-570, 2022.

[32] Hou Y, He W, Wang C and Ren X, " A Two-stage Selection Hyper-heuristic Algorithm for the Capacitated Vehicle Routing Problem," *IAENG International Journal of Applied Mathematics*, vol. 52, no. 4, pp. 979-988, 2022.

[33] Barth M, Scora G and Younglove T, " Modal emissions model for heavy-duty diesel vehicles," *Transportation Research*, vol. 1880, pp. 10-20, 2004.

[34] Demir E, Bektas T and Laport G, " A comparative analysis of several vehicle emission models for road freight transportation," *Transportation Research part D: Transport and Environment*, vol. 16,no. 5, pp. 347-357, 2011.

[35] Cowling P, Kendall G and Soubeiga E, " A hyperheuristic approach to scheduling a sales summit, in " *Proceedings of the International Conference on the Practice and Theory of Automated Timetabling*, pp. 176-190, 2000.

[36] Goldberg D E, " Probability matching, the magnitude of reinforcement, and classifier system bidding, " *Machine Learning*, vol. 5, no. 4, pp. 407C425, 1990.

[37] Thathachar M A L and Sastry P S, " A class of rapidly converging algorithms for learning automata, " *IEEE Transactions on Systems Man and Cybernetics*, vol. 15, pp. 168-175, 1985.

[38] Fialho I, Costa Da I, Schoenauer M and Sebag M, " Analyzing bandit-based adaptive operator selection mechanisms, " *Annals of Mathematics and Artificial Intelligence*, vol. 60, no. 1-2, pp. 25-64, 2010.

[39] Schrimpf G, Scheider J, Stamm-Wilbrandt H and Dueck G, " Record breaking optimization results using the ruin and recreate principle, " *Journal of Computational Physics*, vol. 159, no. 2, pp. 139-171, 2000.