Hindawi
Advances in Operations Research
Volume 2022, Article ID 1811689, 16 pages
https://doi.org/10.1155/2022/1811689

Research Article

Multiobjective Green Time-Dependent Location-Routing Problem and Algorithms

Hua-Xin Zhang[1] and Chun-Miao Zhang[2]

[1]*Department of Basic Education, Jiaxing Vocational and Technical College, Jiaxing 314036, China*
[2]*Department of Intelligent Manufacturing, Jiaxing Vocational and Technical College, Jiaxing 314036, China*

Correspondence should be addressed to Chun-Miao Zhang; 1121402110@zjut.edu.cn

Received 6 February 2022; Revised 18 March 2022; Accepted 25 April 2022; Published 21 May 2022

Academic Editor: Saurabh Pratap

To reduce the logistic cost and carbon emission and improve customer satisfaction, this study proposes a multiobjective green time-dependent location routing problem (MOGTDLRP) model in which the objectives are to minimize the distribution total cost, delivery time, and fuel consumption. This model will be solved by several hyperheuristic algorithms which include the high-level heuristics and the low-level heuristics. There are three acceptance criterions for the solution: improving and equal, all moves and accept all solutions, and dynamic acceptance criteria. Through the case, the performance of the algorithm and the influence of various factors on the solution are analyzed in this study. The experimental results show that the proposed model can effectively reduce logistic costs, carbon emissions, and vehicle travel time.

1. Introduction

Urban logistics plays an important role in the economic vitality of the cities. Green logistics is defined as planning and execution of logistics activities in a more environmentally friendly way by considering external factors such as waste, noise, energy usage, and greenhouse gas (GHG) emissions [1, 2]. At the same time, the urban freight transport is known to be responsible for congestion. The location-routing problem (LRP) is one of the most important combinational optimization problems in supply chain management and logistics system planning and conducts joint decision-making regarding the locations of arbitrary types of facilities and the routing of vehicles [3]. The LRP with environmental issues was called the green LRP (GLRP) which aims to reduce fuel consumption and vehicle exhaust emissions [4]. Koç [5] proposed a model which aims to minimize the total costs including the depot costs, vehicle and routing costs, and emissions costs. Yu [6] investigated the location-routing problem with time-dependent demands as an extension of the location-routing problem by considering the time-dependent demand characteristic. Alamatsaz [7] developed a mixed-integer programming model considering time windows for customers and drivers. There are many other researchers [8–11] who investigated the green LRP model with the objective of minimizing the total costs including depot costs, vehicle and routing costs, and emission costs. However, in addition to reducing costs, other factors such as travel time, fuel consumption, and customer satisfaction should be taken seriously. Therefore, multiobjective GLRP (MOGLRP) problems have gradually become a research hotspot since 2019. Toro [12] proposed a multiobjective model for the GLRP to minimize the operational cost and the environmental effect. Rabbani [13] introduced a new variant of the MOGLRP to minimize the total travel distance and the total costs including vehicle fixed cost and CO_2 emissions. Tang [14] established a biobjective model to reduce costs and carbon emissions from the perspective of a sustainable supply chain network.

For further analysis, the papers on GLRP in Table 1 are compared with five factors: (i) year, (ii) the number of objective functions, (iii) time window, (iv) vehicle speed, and (v) algorithm.

There are 11 papers about single GLRP and 10 papers about multiobjective GLRP in Table 1. In the above models, some do not consider the speed of the vehicle, and the rest

Table 1: The papers on GLRP.

References	Year	Objective	Time window	Speed	Algorithm	References	Year	Objective	Time window	Speed	Algorithm
[5]	2016	Single	√		Heuristic	[9]	2020	Single		√	Heuristic
[14]	2016	Multi			Heuristic	[17]	2020	Single			Heuristic
[10]	2017	Multi	√		Heuristic	[23]	2020	Multi	√		Heuristic
[12]	2017	Multi	√		Heuristic	[6]	2021	Single	√	√	Heuristic
[13]	2017	Single	√	√	Exact	[7]	2021	Multi	√		Heuristic
[15]	2018	Multi	√	√	Heuristic	[18]	2021	Multi			Heuristic
[16]	2018	Single	√	√	Heuristic	[19]	2021	Multi			Heuristic
[4]	2019	Single	√	√	Heuristic	[20]	2021	Multi	√		Heuristic
[24]	2019	Single	√	√	Heuristic	[21]	2021	Single	√		Heuristic
[11]	2020	Single			Heuristic	[23]	2021	Single			Heuristic
[8]	2020	Single	√	√	Heuristic						

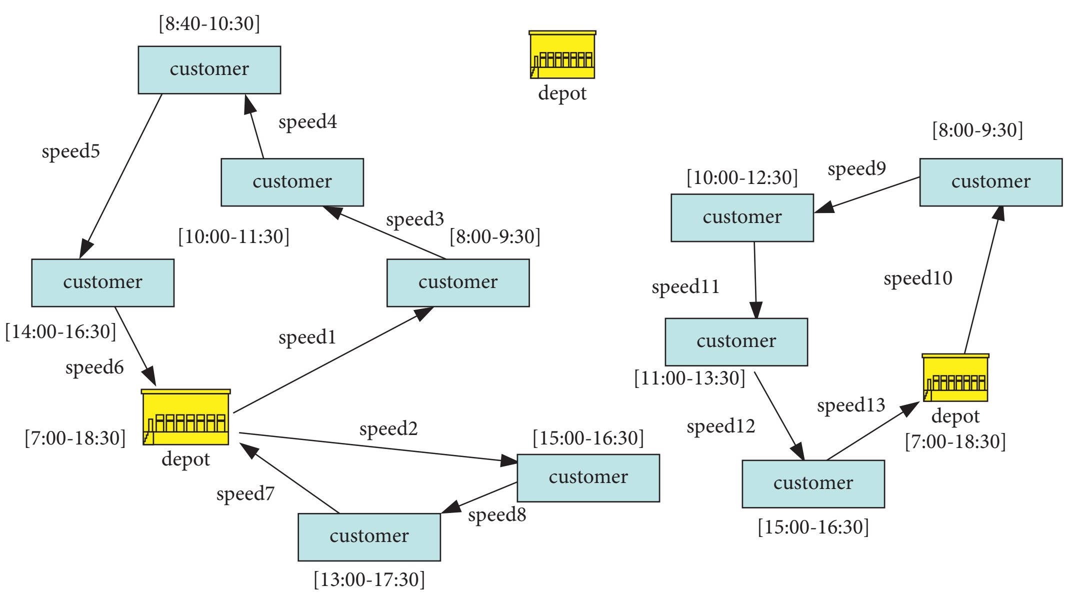

Figure 1: The diagram of the MOGTDLRP.

consider the speed to be a constant. However, speed is the critical parameter in calculating fuel consumption and emissions [25]. The fuel consumption is accepted which was affected by many factors in which the vehicle speed is the most critical one, and the vehicle emission is proportional to the fuel consumption [26]. In the real road network, the vehicle speed is the time-dependent function because of the traffic flow, weather, accidents, and other factors. The model based on time-dependent road network is more instructive to the real logistics distribution. So, a three-objective model named the multiobjective green time-dependent location routing problem (MOGTDLRP, as shown in Figure 1. Its goal is to reduce the total costs (including open depot costs, vehicle costs, and distribution costs), the vehicle fuel consumption, and the delivery time.

Since the LRP is a NP-hard problem, MOGTDLRP is also a NP-hard problem. In the MOGTDLRP, vehicles travel on different types of roads at different speeds at different times, and deliver goods to customers within the time window. Since exact algorithms can only solve small-scale optimization problems in a reasonable amount of time, metaheuristics are selected to solve the large-scale problems. These metaheuristics include genetic algorithm [27–29], the ant colony optimization [30], and A Tabu search [17]. Different from the metaheuristics, several hyperheuristics (HHs) are proposed in this study. The hyperheuristic system, attempting to find the right sequence of heuristics in a given situation rather than trying to solve a problem directly, was defined as a "heuristic selection heuristic" algorithm [32, 33]. In the HH, the domain barrier isolates the high-level heuristic (HLH), including the selection strategy of the operator and the receiving mechanism of the solution and the low-level heuristic (LLH) which contains a series of low-level operators, problem definitions, objective functions, and other information. The role of the selection strategy is to monitor the performance information of the LLH to select a good and suitable operator, and the receiving mechanism determines whether to replace the parent solution according to the quality of the child solution and controls the search direction of the algorithm and the convergence speed. In this study, we design HHs with heuristic algorithms (artificial bee colony algorithm (ABC), ant colony optimization (ACO), and Tabu search algorithm (TS)) as HLH to solve MOGTDLRP. And we design three acceptance criterions

Table 2: Parameters and values.

Symbols	Description	Symbols	Description
D	Set of the candidate depots	$V_h(t)$	Travel speed of the vehicle h
C	Set of the customers	K_{ijh}	Load of vehicle h leaving i and traveling to j
q_i	The demands of the customer i	ST_j	Service time of customer j
ST_j	Service time of customer j	AT_{jh}	The time when vehicle h arrives at node j
x_i, y_i	Coordinate of the node i	TT_{ij}	Travel time between nodes i and j
$[a_i, e_i]$	Time window of the node i	y_r	1 if depot r is open, and 0 otherwise
P_j	Capacity of the depot j		
FD_j	Cost of the depot j	zim	1 if customer i is served by depot m, and 0 otherwise
Q_h	Capacity of the vehicle h		
FV_h	Cost of the vehicle h	x_{ijh}	1 if customer i is served by depot m, and 0 otherwise
d_{ij}	Distance between the two nodes i and j		
F_{ijh}	Fuel consumption of the vehicle h from node i to node j		

(ACs) for the solution, Improving and Equal (IE), All Moves (AM) and Accept all solutions, and Dynamic Acceptance Criteria (DA) and a series of low-level operators that conform.

In summary, the main contributions and innovations of this study are as follows:

(1) The effects of multiple objectives, time-dependent network, and fuel consumption are considered, and the MOGTDLRP mathematical model is constructed

(2) Three HLHs and three ACs combine to form nine HH to analyze the performance of the algorithms

(3) We propose a set of benchmarks of the MOGTDLRP and analyze the simulation results

The construction of this study is as follows. Section 2 is about the MOGTDLRP model. Section 3 is an introduction to the HHs. Section 4 is the results and analysis. Section 5 is the conclusion and outlook.

2. The Model of MOGTDLRP

The model of MOGTDLRP with three optimization objectives will be introduced in this section. The MOGTDLRP can be defined on an asymmetric directed graph $G = (V, E)$, where V is composed of the set of the candidate depots D ($D = \{1, 2, \ldots, M\}$) and the set of customers C ($C = \{1, 2, \ldots, N\}$) and E is a set of edges ($E = \{(i, j)\colon i, j \in V, i \neq j\}/\{(i, j)\colon i, j \in D\}$). The demand q_i, service time ST_i, coordinate (x_i, y_i), and hard time window $[a_i, e_i]$ of customer iEC are known. The capacity P_j, rent cost FD_j, and coordinates (x_j, y_j) of the candidate depot j (jED) are known. Each vehicle h in the homogeneous fleet has the same capacity Q_h and rent cost FV_h. The symbol d_{ij} is the distance between the two nodes i and j ($i \neq j \in V$). The delivery time of a vehicle depends on the distance, where the speed changes according to the departure time and the arc being traversed. $V_h(t)$ is the travel speed of the vehicle h traversing different arcs. TT_{ij} is the travel time between nodes i and j. K_{ijh} is the load vehicle h leaving i and traveling to j. ST_j is the service time of customer j. AT_{jh} is the time when vehicle h arrives at node j. The decision variable y_r equals 1 if depot r is open and equals 0 otherwise. z_{im} equals 1 if customer i is served by depot m and equals 0 otherwise. x_{ijh} equals 1 if the vehicle h travels from i to j and equals 0 otherwise. F_{ijh}, which can be calculated by the CMEM model [8], is the fuel consumption of the vehicle h which travels from node i to node j. Table 2 gives the parameters used in the MOGTDLRP model.

Some assumptions need to be satisfied: (1) each customer must be served only once, (2) each vehicle must return to the original depot, (3) every vehicle shall not be overloaded, and (4) the load of each depot must not exceed its capacity.

Based on the above assumptions and definitions, the model can be represented as follows:

$$\min f_1 = \sum_{j\in D} FD_j y_j + \sum_{i\in V}\sum_{j\in V}\sum_{h\in H} FV_h x_{ijh} + \sum_{i\in V}\sum_{j\in V}\sum_{h\in H} F_{ijh} x_{ijh}, \quad (1)$$

$$\min f_2 = \sum_{i\in G}\sum_{j\in G}\sum_{k\in K}\sum_{m=1}^{M}\left(x^m_{ijk} TT^m_{ij} + \max\left(e_i - AT_i, 0\right) + ST_i\right), \quad (2)$$

$$\min f_3 = \sum_{i\in V}\sum_{j\in V}\sum_{h\in H} F_{ijh} x_{ijh}. \quad (3)$$

The following constraints must be satisfied:

$$\sum_{i\in V}\sum_{h\in H} x_{ijh} = 1, \quad \forall j \in C, \quad (4)$$

$$\sum_{h\in H}\sum_{i\in V} x_{ijh} = \sum_{h\in H}\sum_{i\in V} x_{jih}, \quad \forall j \in C, \quad (5)$$

$$\sum_{j\in D} z_{ij} = 1, \quad \forall i \in C, \quad (6)$$

$$x_{ijh} + \sum_{k\in H, k\neq h}\sum_{r\in V, r\neq j} x_{jrk} \leq 1, \quad i \in V, j \in V, i \neq j, h \in H, \quad (7)$$

$$\sum_{h\in H} x_{ijh} \leq z_{ij}, \quad \forall i \in C, j \in D, \quad (8)$$

$$\sum_{h\in H} x_{jih} \leq z_{ij}, \quad \forall i \in C, j \in D, \quad (9)$$

$$\sum_{h\in H} x_{ijh} + z_{ik} + \sum_{m\in D, m\neq k} z_{jm} \leq 2, \quad \forall i, j \in C, k \in D, i \neq j, \tag{10}$$

$$\sum_{i\in D}\sum_{j\in C} K_{ijh} = \sum_{i\in C}\sum_{j\in V} q_i x_{ijh}, \quad \forall h \in H, \tag{11}$$

$$\sum_{i\in C}\sum_{j\in D} K_{ijh} = 0, \quad \forall h \in H, \tag{12}$$

$$\sum_{i\in C} q_i z_{ik} \leq P_k y_k, \quad \forall k \in D, \tag{13}$$

$$\sum_{i\in V}\sum_{h\in H} \left(K_{ijh} - q_j\right) x_{ijh} = \sum_{i\in V}\sum_{h\in H} K_{jih} x_{jih}, \quad \forall j \in C, \tag{14}$$

$$K_{ijh} \leq Q_h x_{ijh}, \quad \forall i, j \in V, i \neq j, h \in H, \tag{15}$$

$$K_{ijh} \geq q_j x_{ijh}, \quad \forall i \in V, j \in C, h \in H, \tag{16}$$

$$AT_{jh} = x_{ijh} \cdot \left(\max\{e_i, AT_{ih}\} + ST_i + TT_{ijh}\right), \quad \forall i, j \in V, h \in H, \tag{17}$$

$$0 \leq AT_{jh} \leq l_j, \quad \forall j \in C, h \in H, \tag{18}$$

$$x_{ijh} \in \{0, 1\}, \quad \forall i, j \in V, h \in H, \tag{19}$$

$$y_j \in \{0, 1\}, \quad \forall j \in D, \tag{20}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i \in C, \forall j \in D. \tag{21}$$

The description and explanation are as follows. Equation (1) is the costs' objective. Equation (2) is travel time objective. Equation (3) is the third objective which is the total fuel consumption. Constraints (4) and (5) make sure that each customer is served exactly once. Constraints (6) and (7) impose that each customer is served by one depot and one vehicle, respectively. Constraints (8)–(10) forbid illegal routes that the corresponding vehicles do not return to the departure depots. Constraints (11) and (12) make sure that the demand of each customer is satisfied. Constraint (13) guarantees that the load of each selected depot must be less than its capacity. Constraint (14) is the dynamic equilibrium of the load of each vehicle after visiting customer j. Constraints (15) and (16) guarantee that overloading is not allowed. Constraints (17) and (18) are the bounds on the arrival time at each node: a vehicle must arrive at a customer before the closing time windows. Finally, the last three constraints are decision variables.

3. Algorithms

Metaheuristics are wide-ranging. However, choosing the best algorithm and configuring the parameters and operator to solve the problem is very difficult [34]. Numerical experiments have shown that it is impossible to develop a single metaheuristics algorithm that is always efficient for a diverse set of optimization problems. Hyperheuristic algorithm (HH) is a super heuristic algorithm, which provides a high-level heuristic (HLH) method. It can solve various combinatorial optimization problems by managing or manipulating a series of low-level heuristics (LLH). HHs can help researchers to reduce the inherent time and effort required to set up a new domain, as it is a difficult task for testers without deep prior domain-specific knowledge [35]. There are two main tasks in the design of HH: one is the high-level selection strategy and the other is the design of problem domain operator. Using the HH to solve the MOGTDGLRP problem can solve the selection problem of the local optimization operators. If iterating all operators, a lot of time will be wasted. The HH can decide whether to enable the operator through a reasonable selection mechanism according to the history of the operator, which not only saves time but also improves the quality of the solution. In the HLH, there are two different target strategies: operator selection strategy and reconciliation receiving mechanism. In addition, there is an information transmitter between the HLH and the LLH, which transmit information, such as the information of the selection operator, the judgment information of the receiving mechanism, the running time and frequency of the operators, and the number of consecutive unimproved times of the current solution. Therefore, the HLH plays a vital role in improving the performance of the HH. In this study, we have designed nine operators and three HLHs including Tabu search algorithm (TS), ant colony optimization (ACO), and artificial bee colony algorithm (ABC). Next, we will introduce HH according to the following steps: (1) coding method and initial population generation; (2) HLHs; (3) acceptance criteria; (4) the LLHs pool of operators, ξ.

3.1. Coding Method and Initial Population Generation. In the MOGTDLRP problem domain, a complete solution is a collection of all routes $R = \{r_1, r_2, \ldots, r_n\}$. Each route r_i has the same structure: the first and the end node is the opened depot, and the middle part is the customer number. For example, the route $\{D_2\text{-}C_1\text{-}C_3\text{-}C_{10}\text{-}D_2\}$ means that a certain vehicle departs from the depot D_2 to deliver goods to customers C_1, C_3, and C_{10} in order, and then, returns to D_2. The initial solution is critical to the quality of the final solution. In the traditional LRP, when generating an initial route, only the capacity constraints of the depots and vehicles need to be satisfied, and the optimization objective is the distribution distance. However, in the MOGTDLRP, it is also necessary to meet the time window of the depots and customers. In order to produce high-quality initial solutions, it is necessary to design more efficient algorithms. There are two classical algorithms for vehicle-routing problem (VRP) with time windows for reference: Solomon's Insertion algorithm [36] and IMPACT algorithm [37]. However, these two algorithms need to set a lot of parameters. Zhang [8], in her paper, designed one algorithm named the insertion method based on travel time (IMTT) to generate the initial solution. This method has good solution quality and does not need to set many parameters. Therefore, this study used

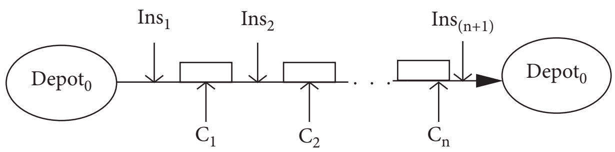

FIGURE 2: The generation process of the initial solution.

the IMTT to generate the initial solutions. Its steps are as follows:

- First step: select the opened depots according to the barycentric method
- Second step: randomly select a customer node to generate a new route
- Third step: calculate the total travel time of this route
- Fourth step: calculate the impact values of the remaining customers inserted into the route
- Fifth step: insert new customers into the route according to the impact values until one or more constraints are not satisfied

Figure 2 shows the generation process of the initial solution. $Depot_0$ is the selected depot; Ins_i represents the insertable position; C_i is the customer in the route.

3.2. High-Level Heuristic

3.2.1. The Artificial Bee Colony (ABC) Algorithm. The ABC includes three elements: foods, employed bees, and unemployed bees which have the onlooker bees and the scouts make up. The foods are the solutions in the MOGTDLRP. Employed bees are responsible for finding foods and carrying information about the foods. Onlooker bees follow the employed bees with a certain probability to select the foods. Scouts are responsible for searching for foods randomly to enhance the algorithm's ability to jump out of local optimal solutions. In the initialization phase of the ABC algorithm, SN solutions, that are the location of the food sources, are generated by IMTT. Then, these initial solutions are evaluated to obtain their fitness values.

In the ABCHH, the food sources are the bottom operators, and the optimization process is the process of the bee colony searching for foods. The high-level strategy idea of the ABC is as follows.

First, 20% of the bees in the colony are set as scout bees whose roles are to find excellent sources of food. Each scout bee calculates 20 times at each operator. The count table score (i) of the operator i will be added by 1 point when the fitness improves. The operators are sorted in the descending order according to their scores when excellent food has been found, all individuals become employed bees to enjoy the food, and individual is updated by the following equation:

$$X_i^{t+1} = X_i^t \cdot \left(p_i \cdot S_a + \text{rand}\left(S - S_a - S_{\text{best}}\right) + S_{\text{best}}\right),$$
$$i = 1, 2, \ldots, SN, a, b = 1, \ldots, |S|, a \neq b, \tag{22}$$
$$p_i = \frac{SC_a}{|S|} \text{ or } p_i = 1,$$

where i represents the i_{th} individual in the population, t is the number of iterations, S_a represents the ath operator in the operator set, SC_a is the score ranking of operator S_a, and $P_i{\cdot}S_a$ means that the S_a operator is selected with the probability of P_i. When the t generation solution X_{ti} is better than the t-1 generation solution, $p_i = 1$; otherwise, it is calculated according to equation. rand (S-S_a-S_{best}) means that an operator is randomly selected after the operator set S eliminates the S_a and S_{best} operators, and S_{best} is the operator with the highest score in the S set.

3.2.2. Related Definitions

Definition 1. Status minus.

For the problem of real number coding, let the state of bee i be x_{id}. According to equation (23), bee i randomly approaches the position x_{kd} of another bee for information transmission:

$$x_{i\,d}' = x_{i\,d} + \phi_{i\,d}\left(x_{i\,d} - x_{k\,d}\right), \quad i = 1, 2, \ldots, SN, d = 1, 2, \ldots, D, \tag{23}$$

which means an operation bit d is randomly selected, and the element corresponding to the d bit of x_{id} and x_{kd} forms a field $[a\ b]$, where a is a smaller value and b is a larger value. Then, λ (x_{id}-x_{kd}) is the reverse order of x_{id} from a to b.

For example, let $x_{i\,d} = [1\ 2\ 3\ 4\ 5\ 6]$, $x_{k\,d} = [3\ 5\ 2\ 1\ 6\ 4]$, and random number $\beta = 4$; then, x_{id} (4) = 4, x_{kd} (4) = 1, and $[a\ b] = [1\ 4]$. According to the definition, x_{id}–x_{kd} means that the 1–4 bits of x_{id} are arranged in the reverse order to produce a new solution $x_{i\,d}' = [4\ 3\ 2\ 1\ 5\ 6]$.

Definition 2. State number multiplication.

$C \times x_{id}$ represents the number multiplication of states that means taking the first.

$\lceil C \times D \rceil$ elements of x_{id}, and the symbol $\lceil\rceil$indicates rounding up.

For example, $x_{i\,d} = [1\ 2\ 3\ 4\ 5\ 6]$ and $C = 0.5$; then, $\lceil C \times D \rceil = 3$ and $x_{i\,d}' = [1\ 2\ 3]$.

Definition 3. Status sum.

Status sum are 2-opt switching operations.

For example, let $x_{i\,d} = [1\ 2\ 3\ 4\ 5\ 6]$ and XO = $[2\ 3\ 1\ 4]$. First, we convert XO to XO = ((2, 3) (1, 4)). Then, x_{id} + XO indicates that the second bit and the third bit of x_{id} exchange, $x_{i\,d1} = [\,1\;\;3\;\;2\;\;4\;\;5\;\;6\,]$, and then, the first bit and the fourth bit of x_{id} exchange are the result which is $x_{i\,d2} = [\,4\;\;3\;\;2\;\;1\;\;5\;\;6\,]$. If the XO element is odd, the last bit is omitted.

3.2.3. The Ant Colony Optimization (ACO) Algorithm. The high-level strategy is based on ACO. There are m ants distributed among the bottom operators. Each ant determines the next vertex according to the pheromone in each route, and the quality of the operator is evaluated by visibility function η_i:

$$\eta_j(t) = \gamma\eta_j(t-1) + \sum_k^m \frac{I_{kj}}{T_{kj}(t)}, \tag{24}$$

where I_{kj} (t) is the improved value (which can be negative) after ant k applies the bottom operator j in generation t, T_{kj} (t) is the CPU running time occupied by the calculation, and γ is a value between 0 and 1.

The equation of τ_{ij} (t), the pheromone on each arc, is as follows:

$$\tau_{ij}(t) = (1-\rho)\tau_{ij}(t-1) + \sum_k^m \frac{\#_{ij}(P_k(t))\cdot I(P_k(t))}{T(P_k(t))}, \tag{25}$$

where ρ $(0<\rho<1)$ is the evaporation coefficient of pheromone, $(1-\rho)$ represents the persistence coefficient of pheromone, $P_k(t)$ represents the order in which ant k selects the bottom operator, $\#_{ij}(P_k(t))$ represents the number of times the arc (i, j) appears in the route of ant k, $I(P_k(t))$ is the amount of improvement of ant k in the route, and $T(P_k(t))$ is the CPU time.

The ant colony selects operators based on pheromone and visibility; the equation is as follows:

$$V_{ij}(t) = \alpha\eta_j(t) + \beta\tau_{ij}(t). \tag{26}$$

There is a variable PV to improve the selection probability of some operators with poor performance:

$$\begin{cases} PV_{ij}(t) = \max\left\{V_{ij}(t), Q\sigma^{V_{ij}(t)}\right\}, \\ Q = \dfrac{\sum_{h\in H}\max\{0, Vih(t)+\varepsilon\}}{10n}, \end{cases} \tag{27}$$

where H is the set of bottom operators, $n = |H|$ is the number of the bottom operators, and ε and σ can ensure that the underlying heuristic operator with poor performance still has a small but nonzero probability of being chosen. If arc (i, j) is an illegal arc, $PV_{ij}(t)$ is 0. If the performance of all bottom operators is not good, $q = 0$, and set all PV values to 1 to ensure that all bottom operators are selected with equal probability. The probability that the arc (i, j) is selected can be expressed as

$$\text{probability}_{ijk}(t) = \frac{PV_{ij}(t)}{\sum_{h\in H} PV_{ih}(t)}. \tag{28}$$

3.2.4. Tabu Search (TS) Algorithm. The high-level strategy idea of the TS is as follows. Score evaluates the performance of operators. Then, select an operator with a larger score to improve the current solution in the iterative process. Each operator has the same initial score. When the operator improves the current solution, the score of this operator will add 1; otherwise, it will subtract 1. Detailed process is as follows.

First, an initial score $r_k = 0$ is set for each operator k. The score interval of each operator is $[r_{\min}, r_{\max}]$; $r_{\min}$ and $r_{\max}$ represent the lowest and highest scores, respectively. The operator k that is not in the Tabu list and has the highest score will be selected. If it can improve the objective function, the score of operator k will add 1. Otherwise, the score will subtract 1 and be put into the Tabu list with fixed Tabu length. Operators in the Tabu list are exempted according to the first in first-out principle.

3.3. Acceptance Criteria. The acceptance criterion is the process of whether the new solution is obtained after processing by the operator. If the new solution always unconditionally replaces the parent solution, the optimization time will be too long. However, the HH will fall into a local search if it just saves the improved solution and omits the inferior solution. Other acceptance criteria include the accepting improved solutions completely and accepting inferior solutions in a certain proportion.

In this study, three acceptance criteria are proposed:

(1) Improving and equal (IE): accept all improved solutions and reject nonimproved solutions.

(2) All moves (AM): accept all solutions.

(3) Dynamic acceptance criteria (DA): if the current optimal solution is not updated too many times, which means that the solution falls into local search, the acceptance probability of inferior solution should be increased to ensure global search. The acceptance probability is calculated as follows:

$$p = \frac{f_{t-1} - f_t}{\left(f_t + f_t^{\text{best}}\right)/2} + \frac{M_1}{M}, \tag{29}$$

where f_{t-1} is the parent solution, f_t is the child solution, f_t^{best} is the optimal solution of generation t, M is a constant, and M_1 is the number of times that f_t^{best} has not been updated.

3.4. Pool of LLH. This section introduces the nine LLH operators including four operators operating within one route, three operators operating between two routes, and two operators operating the selection of the depots:

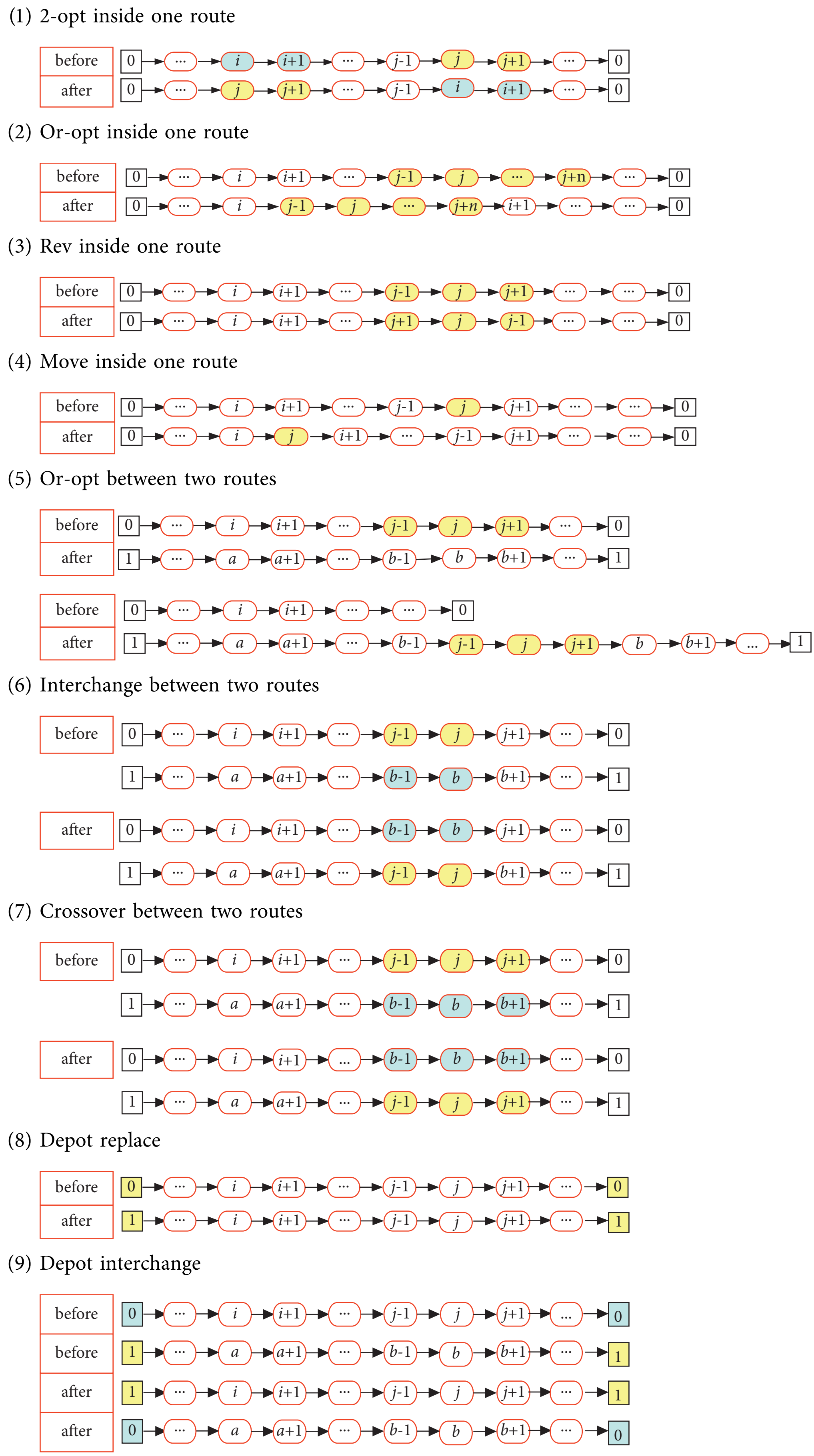
(1) 2-opt inside one route
before
after
(2) Or-opt inside one route
before
after
(3) Rev inside one route
before
after
(4) Move inside one route
before
after
(5) Or-opt between two routes
before
after
before
after
(6) Interchange between two routes
before
after
(7) Crossover between two routes
before
after
(8) Depot replace
before
after
(9) Depot interchange
before
before
after
after

Table 3: Pseudocode of the HH.

```
//Initialization
Set the parameters
Initialize Solution (Pop)
Current Solution = Pop (r)
Best Solution = Current Solution
Fitness (Best Solution) = Fitness (Current Solution)
  //Main loop
    while t < ITER do
    //High-level Selection Strategy
      operator = Select (ξ)
      //low-level heuristics
      [Child Solution] = Implement(Current Solution, operator)
      //High-level Acceptance Criterion
  if Fitness (Child Solution) <Fitness (Current Solution) then
      Current Solution = Child Solution
  else
      Current Solution = Accept or Not (Child Solution, Current Solution)
  end if
  //Save the global best solution
  if Fitness (Best Solution) >Fitness (Child Solution) then
      Best Solution = Child Solution
  end if
  Update related data
end while
```

3.5. Pseudocode of the HH. Table 3 lists the pseudocode of the HH, and Figure 3 shows the flowchart.

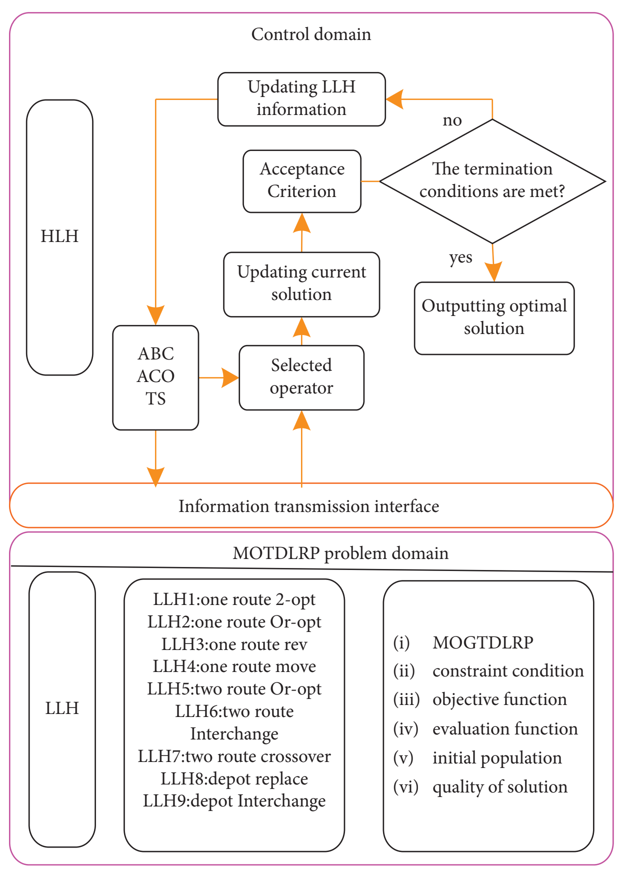

Figure 3: The flowchart of the HH.

4. Simulation Experiment and Analysis

All programs are coded in MATLAB R2018a and executed on a computer with an Intel (R) Core (TM) i5-5200U CPU @2.20GHZ, 4 GB of RAM, and Windows 7 operating system.

4.1. Benchmarks. The experimental data come from [8]. Table 4 shows the detailed depot data, and Table 5 includes the time windows of all depots. In the tables, "Ben" is benchmark's name, "Para" is the abbreviation of parameter, "cap" is the capacity, "coor" is the coordinate, "costs" is the fixed open costs, and "TW" notes the time window. Table 6 shows the data of the vehicles. The data about customers come from the Solomon benchmarks.

4.2. The Time-Dependent Speed Function. Ioannou [37] first proposed the vehicle travel time-dependent speed function in his study about vehicle-routing problem and calculated the travel time by the length of the road segment and the travel speed, that is, the travel speed is a time-dependent function. This time-dependent speed function can avoid jumps in travel time and ensure the FIFO characteristics of the time-varying network; that is, the vehicle that departs first arrives at the destination first. For the road network, it can be considered to conform to the FIFO characteristic without overtaking.

In this study, to simulate urban road conditions and rush hours, we divide the roads into five types and divide the total service time into four equal time periods. Table 7 shows the values.

The road type is obtained by equation (30), where *Grade* (i, j) is the type of road between nodes i and j and Mod (a, b) is the remainder of the number a divided by the number b:

Table 4: Data of the depots.

Ben	Para		Depot1	Depot2	Depot3	Depot4	Depot5	Depot6	Depot7	Depot8	Depot9	Depot10
C1	cap		990	800	900	850	840	970	1000	910	930	780
	coor	x	40	64	35	44	29	18	63	85	11	37
		y	50	13	79	57	40	82	93	8	63	17
	costs		40000	45000	42000	41000	48000	50000	38000	49000	47000	46000
C2	cap		950	800	1010	970	920	990	1030	930	870	890
	coor	x	40	8	91	35	20	51	29	60	98	96
		y	50	95	46	43	69	100	28	43	97	42
	costs		90000	100000	120000	95000	105000	97000	115000	112000	99000	117000
R1	cap	960	750	910	820	720	790	1000	800	790	890	
	coor	x	35	10	52	46	81	11	94	78	88	72
		y	35	54	56	60	24	59	40	77	66	49
	costs		20000	19000	22000	21000	18000	23000	24000	17000	25000	24000
R2	cap		1020	810	720	790	890	1070	740	700	1100	790
	coor	x	35	92	82	25	64	100	10	3	8	64
		y	35	77	17	82	17	87	72	51	99	60
	costs		85000	94000	94000	89000	100000	92000	97000	87000	99000	96000
RC1	cap		1050	900	1090	850	790	940	970	1180	900	1020
	coor	x	40	15	40	24	50	62	79	10	80	87
		y	50	52	3	93	76	60	100	95	11	75
	costs		18000	19000	17000	21000	26000	24000	23000	19000	24000	25000
RC2	cap		1300	1200	900	800	1080	780	1090	1240	900	1100
	coor	x	40	86	23	55	28	68	72	34	26	61
		y	50	37	94	100	92	52	19	61	88	44
	costs		86000	91000	87000	99000	96000	100000	85000	94000	93000	97000

Table 5: Time windows of the depots.

Ben	TW	Ben	TW
C1	[0.1236]	R2	[0.1000]
C2	[0.3390]	RC1	[0.240]
R1	[0.230]	RC2	[0.960]

Table 6: The data of the vehicles.

Ben	C1	C2	R1	R2	RC1	RC2
Capacity	200	700	200	1000	200	1000
Vehicle	800	2700	500	2500	450	2500

$$\mathrm{Mod}(a, b) + 1. \tag{30}$$

4.3. Parameter Settings. Population size is 100, the algorithm iteration number is 200, ant path length $LP = 11$, weight parameter $\alpha = (0.6,0.7,0.8)$, $\beta = (0.6,0.7,0.8)$, and $\gamma = (0.6,0.7,0.8)$, and enhancement coefficient $\varepsilon = 0.001$ and $\sigma = 1.001$. Tabu search algorithm parameter maximum score is rankingMax = 5, minimum score is rankingMin = 0, and Tabu length is tabuLength = 4.

4.4. Comparative Analysis of the Nine Combined HHs. Three high-level algorithms (ABC, ACO, and TS) and three acceptance criterion (AM, IE, and DA) can form nine HHs named ABC + AM, ABC + IE, ABC + DA, ACO + AM, ACO + IE, ACO + DA, TS + AM, TS + IE, and TS + DA These nine HHs solve the benchmarks of the MOGTDLRP.

Table 7: The time-dependent speed function.

Road type	S1	S2	S3	S4
1	1.80	2.20	1.60	2.40
2	1.60	2.40	1.80	2.20
3	1.40	2.60	1.00	3.00
4	1.20	2.80	1.40	2.60
5	1.00	3.00	1.20	2.80

The results can reflect the performance of the algorithm. The relevant data are as follows:

Objective function: f_1 minimum total cost.

Instances: C1 (C101–C109).

Comparison parameters: deviation percentage *RD* obtained as

$$R\ D = \frac{f_A - f_{best}}{f_{best}} \times 100, \tag{31}$$

where f_A is the best result obtained by algorithm A and f_{best} is the average of the results obtained by the nine HHs.

The flow of the nine HHs:

Step 1: generating the initial population by the IMTT algorithm and calculating the objective functions to get the initial father solutions S11, S21, and S31

Step 2: setting relevant algorithm parameters

Step 3: selecting the operator of the LLH according to HLH and calculating the objective functions to get the child solutions S12, S22, and S32

Table 8: Results of C101–C109 obtained by 9 HHs.

Number	HH	RD									Scores
		C101	C102	C103	C104	C105	C106	C107	C108	C109	
1	ABC + AM	1.11		1.20	1.00	1.50	0.98	1.74	1.15	1.20	1.43
30											
2	ABC + IE	1.05	1.10	1.04	0.96	0.78	1.12	0.67	0.00	1.09	48
3	ABC + DA	0.00	0.00	0.00	0.15	0.0	0.00	0.00	0.00	0.00	80
4	ACO + AM	1.95	1.50	1.12	1.43	0.97	0.89	1.20	1.52	1.05	27
5	ACO + IE	1.45	1.32	0.80	0.79	0.85	0.00	1.13	1.35	0.76	45
6	ACO + DA	0.95	0.75	0.58	0.00	0.18	0.00	0.99	1.13	0.00	69
7	TS + AM	1.13	1.81	1.28	0.78	0.35	0.93	1.21	1.45	0.79	34

Step 4: determining whether to accept the current child solutions according to the acceptance criteria

Step 5: updating related parameters

Step 6: if the algorithm termination condition is satisfied, the algorithm ends; otherwise, go to step 3

The results of C101–C109 obtained by nine HHs are in Table 8.

The maximum value of RD in Table 8 is 1.95, which indicates that the solutions obtained by these nine algorithms are not very different. We continue to analyze the performance of the algorithm with Boda algorithm. Each HH has 1 to 9 scores according to the RD value. The algorithm with the smallest RD gets 9 points, the algorithm with the second-smallest RD gets 8 points, and so on. Figure 4 shows the score of the nine HHs. The abscissas 1–9 in the figure correspond to 9 HHs. Figure 4 is a pie chart of the number of optimal solutions obtained by the 9 combination algorithms on C1 examples; that is, the number of times that *RD* is equal to 0.

Through the data of Table 8 and Figures 4 and 5, it is known that the ABC + DA combination algorithm scores the highest with 80 points, followed by ACO + DA with an equal score of 69, and the third place is TS + DA with a score of 67. The pie chart shows that the number of times ABC + DA algorithm obtains the optimal solution accounts for 47.06%; ACO + DA accounted for 17.65%; TS + DA accounted for 23.53%; ABC + IE and ACO + IE accounted for 5.88%, respectively, and other combinations do not get the optimal solution. Under ABC high-level strategy, the total score is 30 + 48 + 80 = 158; ACO senior strategy score is 27 + 45 + 69 = 141 points; TS high-level strategy score is 34 + 59 + 67 = 160 points. The score of the AM acceptance criteria is 30 + 27 + 34 + 91, IE is 48 + 45 + 59 = 152, and DA is 80 + 69 + 67 = 16 points. The scores of the three strategies have little difference, but the difference of the solution acceptance criteria is large. AE accepts all solutions, which is easy to cause too divergent search range and difficult to find the optimal solution. IE only accepts noninferior solutions, which is easy to fall into the trap of local optimization. DA accepts all noninferior solutions and conditionally selects inferior solutions. This method gets a higher score because of which it takes into account the advantages of AE and IE and avoid their shortcomings. In general, the ABC + DA is the best combined HH algorithm in the C1 case.

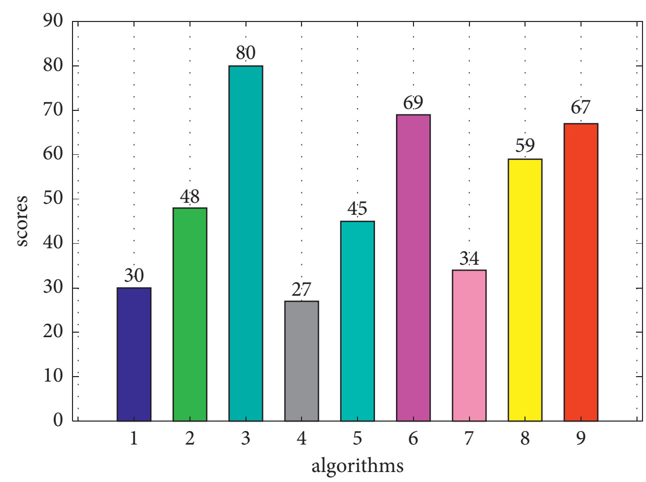

Figure 4: The bar chart of HHs scored on C1.

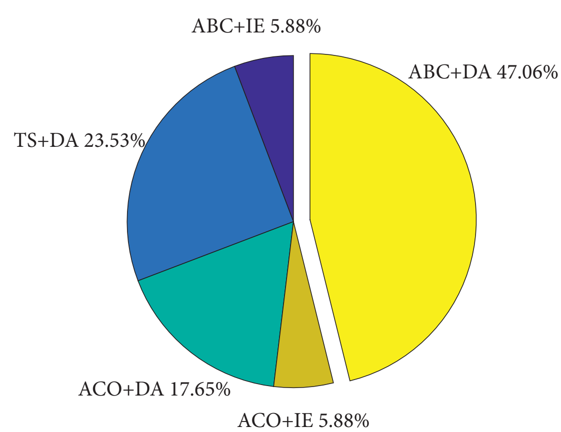

Figure 5: The pie chart of HHs scored on C1.

4.5. Analysis of the Single Objective Optimization Problem. MOGTDLTP is a multiobjective model, which includes three objectives: total cost, travel time, and fuel consumption. Different optimization objectives lead to different distribution routes. In this section, the difference of the solutions of the different single objective optimization problem is discussed.

Instances: RC2 (RC201–RC208).

Algorithm: ABC-DA

Table 9: The result of the three optimal objectives.

Benchmarks	OBJ	TC	RD_1	TI	RD_2	FC	RD_3
RC201	f_1	$1.8011e+05$	0	2458.9	5.29	1611.8	4.06
	f_2	$1.8925e+05$	5.07	2335.3	0	1751	13.05
	f_3	$1.8905e+05$	4.96	2428.2	3.98	1548.9	0
RC202	f_1	$1.7982e+05$	0	2748.7	12.09	1322.6	5.22
	f_2	$1.8629e+05$	3.6	2452.1	0	1788.8	42.30
	f_3	$1.9176e+05$	6.29	2730.7	11.36	1257	0
RC203	f_1	$1.7964e+05$	0	2066.1	6.35	1144.1	8.49
	f_2	$1.8868e+05$	5.03	1942.8	0	1181.5	12.01
	f_3	$1.8855e+05$	4.96	2025.2	4.24	1054.8	0
RC204	f_1	$1.7936e+05$	0	2648.8	41.45	859.11	8.08
	f_2	$1.8862e+05$	5.16	1872.5	0	1121.1	41.04
	f_3	$1.8829e+05$	4.98	2433.6	41.46	794.86	0
RC205	f_1	$1.798e+05$	0	2667.6	11.06	1303.9	5.47
	f_2	$1.8922e+05$	5.24	2402	0	1717.5	38.92
	f_3	$1.8874e+05$	4.97	2684.5	11.76	1236.3	0
RC206	f_1	$1.7971e+05$	0	2321	18.59	1214.9	5.08
	f_2	$1.8881e+05$	11.23	1957.1	0	1313.4	13.6
	f_3	$1.8866e+05$	11.05	2170.6	10.91	1156.2	0
RC207	f_1	$1.7975e+05$	0	2038	11.59	1247.6	7.94
	f_2	$1.8877e+05$	5.02	1826.3	0	1266.5	9.58
	f_3	$1.8866e+05$	4.96	1952.1	6.89	1155.8	0
RC208	f_1	$1.7948e+05$	0	1678.6	4.64	978.5	3.53
	f_2	$1.8845e+05$	5	1604.1	0	954.91	1.03
	f_3	$1.8845e+05$	4.6	1795.7	11.94	945.14	0

Table 10: The result of the four optimal objectives.

Instances	OBJ	TC	TI	FC	Instances	OBJ	TC	TI	FC
RC201	f_1	180116	2458.9	1611.8	RC208	f_1	179485	1678.6	978.5
	f_2	189253	2335.3	1751		f_2	188456	1604.1	954.91
	f_3	189055	2428.2	1548.9		f_3	188458	1795.7	945.14
	MO-TC	180155	2459	1655		MO-TC	179439	1663	939
	MO-TI	189233	2332	1733		MO-TI	188446	1599	946
	MO-FC	188141	2350	1641		MO-FC	188415	1775	915

Single objective: f_1, minimum total cost, f_2, minimum delivery time, and f_3, minimum fuel consumption.

The results are in Table 9.

In Table 9, OBJ is the objective, TC is total cost, TI is travel time, and FC is fuel consumption; RD_i is the difference percentage calculated by

$$RD_1 = \frac{TC_i - TC_{\min}}{TC_{\min}} \times 100,$$
$$RD_2 = \frac{TI_i - TI_{\min}}{TI_{\min}} \times 100, \quad (32)$$
$$RD_3 = \frac{FC_i - FC_{\min}}{FC_{\min}} \times 100.$$

The results show that the optimal solutions are different under different optimization objectives. Compared with the f_2 and f_3 objectives, the total cost obtained with the f_1 objective is the minimum, but the delivery time and fuel consumption are not the minimum. The values of RD indicate the deviation percentage between the current value and the minimum value. The larger the value, the greater the deviation. When taking travel time as the optimization objective, the RD values get the maximum seven times, indicating that when taking travel time as the optimization objective, although the total travel time gets the optimization, the total cost and fuel consumption are high. Therefore, it is not appropriate to take travel time as a separate optimization objective. When f_1, the total cost is the optimization objective; the average value of RD is 6.62, and when f_3, the fuel consumption is the optimization objective; the average value of RD is 5.30. Therefore, the solution obtained under the f_3 objective is better.

4.6. Comparison between Single Objective and Multiobjective. Instances: RC201 and RC208

Optimization objectives: f_1, minimum total cost, f_2, minimum travel time, f_3, minimum fuel consumption, and MO, multiobjective.

Algorithm: ABC + DA

Results are shown in Table 10.

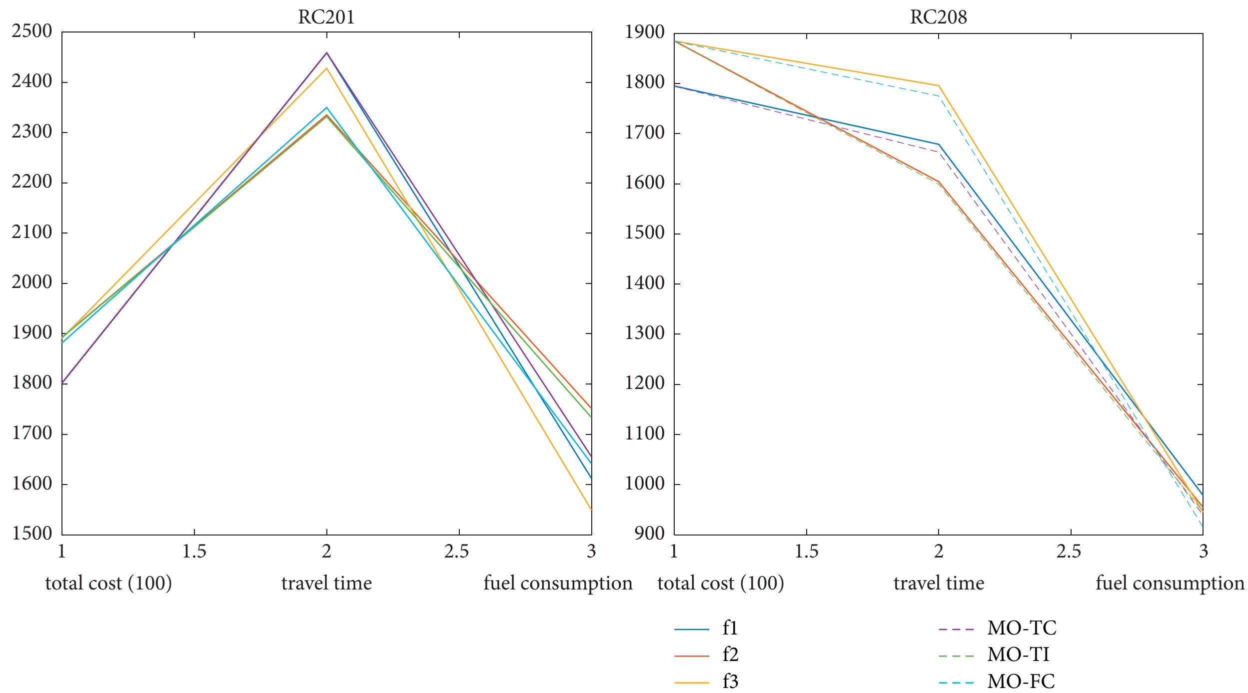

FIGURE 6: The curves of the combined values of RC201 and RC208.

TABLE 11: The results of C1~C2.

Ben	NP	MC	MT	MF	AC	AT	AF	AV	Ben	NP	MC	MT	MF	AC	AT	AF	AV
C101	4	87009	9555	951	88980	9570	980	10	R112	2	46460	1599	960	46466	1599	966	9
C102	8	89881	10206	1577	92647	10301	1736	11	R201	11	180941	2240	1340	87473	276	1427	3
C103	6	90138	9946	1138	90144	9973	1144	11	R202	2	182692	2110	1192	182704	2156	1204	3
C104	7	90383	10030	1383	90390	10119	1390	10	R203	7	192774	1967	1274	192847	2048	1347	3
C105	6	88060	9697	1185	91946	9833	1289	11	R204	4	180340	1949	840	180549	2137	1049	3
C106	4	87954	9643	1154	87961	9844	1161	11	R205	15	180478	2132	978	180677	2258	1177	3
C107	6	87141	9647	1141	88100	9712	1158	10	R206	3	185731	1914	1231	185749	1946	1249	3
C108	5	87936	9558	1003	89534	9597	1068	11	R207	8	180375	1811	875	180537	2014	1037	3
C109	6	87307	9657	1111	89847	9693	1181	10	R208	3	180357	1616	857	180396	1747	896	3
C201	18	196375	11983	575	196409	12059	609	4	R209	4	185836	1823	1219	189408	1872	1308	3
C202	14	196416	9568	616	196524	11463	724	4	R210	13	180542	1939	1042	180700	2238	1200	3
C203	11	196551	11966	751	196808	12603	1008	4	R211	4	180283	1520	783	180285	1520	785	3
C204	8	196676	10549	876	196789	11057	989	4	RC101	6	42060	2028	1660	42072	2048	1672	12
C205	9	196348	9435	548	196418	9816	618	4	RC102	6	41841	1908	1441	41851	1933	1451	12
C206	9	196338	9379	538	196378	9624	578	4	RC103	5	42015	1995	1515	42048	2037	1548	10
C207	11	196361	9727	561	196442	10009	642	4	RC104	8	42675	1818	1175	42705	1850	1205	10
C208	6	196403	10606	603	196733	11030	933	4	RC105	7	42267	1897	1417	42297	1983	1447	13
R101	7	48145	2276	1603	49021	2290	1664	15	RC106	3	41543	1886	1397	42005	1957	1555	11
R102	5	49603	2142	1603	49719	2151	1719	14	RC107	7	40809	1788	1309	40832	1792	1332	10
R103	10	48335	2126	1335	48419	2167	1419	12	RC108	4	40784	1779	1284	40786	1779	1286	10
R104	7	47694	1820	1194	47732	1925	1232	11	RC201	4	180155	2332	1641	185120	2384	1704	3
R105	7	44762	1960	1388	46336	2068	1622	12	RC202	2	179834	2543	1334	179867	2560	1367	3
R106	6	48236	1805	1236	48257	1819	1257	12	RC203	13	179630	1977	1130	179674	2051	1174	3
R107	6	47611	1789	1111	47645	1937	1145	11	RC204	12	187535	1924	1035	187964	1940	1131	3
R108	6	46544	1736	1044	46582	1781	1082	9	RC205	10	188797	2474	1297	189007	2535	1507	3
R109	4	45688	1707	1188	45694	1709	1194	11	RC206	8	179694	2096	1172	183089	2166	1214	3
R110	10	44268	1682	1048	46519	1730	1119	10	RC207	7	185754	1829	1224	186313	1931	1280	3
R111	7	43772	1688	1092	46430	1750	1216	11	RC208	4	179439	1585	915	186090	1594	923	3

In Table 10, MO-TC is the solution with the minimum total cost, MO-TI is the solution with the minimum delivery time, and MO-FC is the solution with the minimum fuel consumption in the Pareto solution set.

In the single objective function of RC201, the minimum total cost is 180116, the minimum travel time is 2335.3, and the minimum fuel consumption is 1548.9. In the multi-objective function of RC201, the minimum total cost is

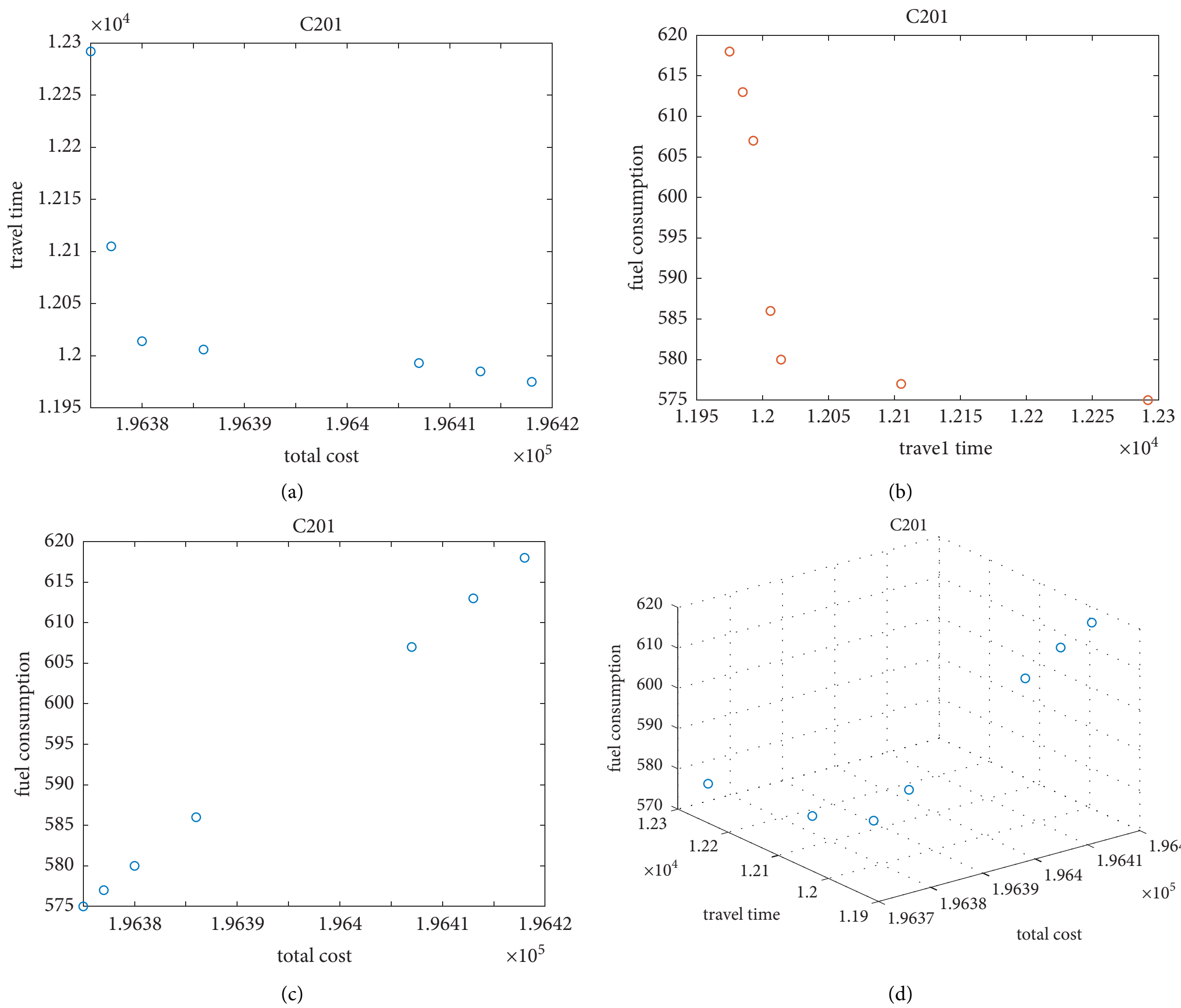

Figure 7: The solutions of C201.

$180155 > 180116$, the deviation is $RD = 0.02$, the minimum travel time is $2332 < 2335.3$, and the deviation is $RD = -0.14$; the minimum fuel consumption is $1641 > 1548.9$, and the deviation is $RD = 5.95$. In the single objective function of RC208, the minimum total cost is 179485, the smallest travel time is 1604.1, and the smallest fuel consumption is 945.14. In the multiobjective function of RC208, the minimum total cost is $179439 < 179485$, the deviation is $RD = -0.02$, the minimum travel time is $1599 < 1604.1$, and the deviation $RD = -0.32$; the minimum fuel consumption is $915 < 945.14$, and the deviation $RD = -3.2$. Four of the six RD values are less than 0, indicating that the combination of cost, time, and fuel consumption under multiobjective function is better than that under single objective function. Figure 6 is the curves of the combined values of RC201 and RC208, respectively. In the figure, the solid lines are the combined curve obtained by single objective and the dotted lines are the combined curve under multiobjective. The dotted lines are mostly below the solid lines, which also indicate that the combined values obtained by the multiobjective function are better.

4.7. Results and Analysis of All Benchmarks. In this section, we use ABC + DA algorithm to solve all benchmarks and analyze the results. The number of iterations is set to 200. All results are in Table 11 in which "MC" is the minimum total cost, "MT" is the minimum travel time, and "MF" is the minimum fuel consumption. "AC," "AF," and "AT," respectively, represent the average cost, average fuel consumption, and average time of the whole Pareto solution set. "AV" means the average number of activated vehicles.

The data in Table 11 indicate the following:

(1) In general, due to the high quality of solutions, the number of solutions of every benchmark is not a huge amount.

(2) Because of the cost of the depots, the values of MCs are too large which indicate that the ratio of depots' cost to logistic cost is greater than the ratio of fuel consumption and travel time cost. However, as the number of customers increases, the fuel consumption and travel time cost increase, leading the change in selection depots.

(3) More vehicles need to be enabled to participate in distribution to meet customer time window needs. Taking the RC201 case as an example, the total demand of all customer is 1724 and the capacity of vehicles is 1000. Theoretically, only two vehicles need to be enabled, but in the MOGTDLRP, three vehicles need to be enabled. Due to the use of large capacity vehicles, the number of multiobjective optimal solutions of C2, R2, and RC2 is more than that of corresponding C1, R1, and RC1. The average number of Pareto optimal solutions for C1 is 5.78, while C2 is 10.75, which is about twice that of C1. The average number of Pareto optimal solutions for R1 is 6.42; while R2 is 6.72, which is greater than 6.42. The average number of Pareto optimal solutions for RC1 is 5.75; while RC2 is 7.5, which is greater than 5.75. From a practical point of view, logistic companies want to use fewer vehicles, so when designing the algorithm, the number of vehicles is the priority condition for the solution. On the contrary, using fleets of different capacities is also an effective way for logistic networks to achieve a low-carbon economy.

(4) Figures 7(a)–7(c) are the two-dimensional projections of C201 total cost and travel time, travel time and fuel consumption, and total cost and fuel consumption. Figure 7(d) is the three-dimensional diagrams of the solutions of C201. The relationship between cost and fuel consumption is linear because the fuel consumption is part of the total cost. Figures 7(a), 7(b), and 7(d) solution distribution is relatively uniform, indicating that the quality of the solution is higher.

5. Conclusion and Outlook

In this study, we proposed the MOGTDLRP model with three objectives and a time-dependent speed function with dividing one day into four periods and definition five road types. Fifty six instances were obtained based on the Solomon benchmarks by referring to the relevant literature. In every benchmark, there are 100 customers whose coordinates, requirements, and time windows are known and 10 depots whose coordinates, capacity, and time window are also known. The benchmarks are solved by nine efficient HHs with the high-quality initial solution obtained by IMTT. The results indicated that the MOGTDLRP could effectively reduce the fuel consumption and the total costs.

According to the analysis of experimental results, this study can get the following conclusions and suggestions:

(1) The proposed HHs have high accuracy in solving the MOGTDLRP model and can obtain high-quality solution in reasonable computing time. The MOGTDLRP model can effectively reduce logistic costs, fuel consumption, and travel times.

(2) Factors such as customer distribution and customer time windows should be the same issues that logistic companies need to pay attention to as depot costs.

(3) Heterogeneous fleet can reduce the distribution cost.

(4) The traffic speed limit affects the delivery time and vehicle fuel consumption, so the nearest delivery scheme does not necessarily have the least delivery time or fuel consumption.

In summary, the time-dependent network is closer to the realistic transportation network than the static network. However, the route from node to node is unique in the time-dependent network. However, there are multiple routes between nodes in real road networks. Furthermore, the actual urban road network is complex, and the speed of vehicles on the same road at the different time periods is random and fluctuating. At the same time, when a vehicle moves from one node to another, it often passes through multiple nodes, and the routes between the nodes are not unique. Therefore, how to solve the optimization problem of green location-path of random road network and multinode road network is the next research direction.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Hua-xing Zhang conceptualized the study; Chun-miao Zhang curated the data;Chun-miao Zhang carried out formal analysis; Hua-xing Zhang carried out funding acquisition; Hua-xing Zhang investigated the study; Chun-miao Zhang developed the methodology; Hua-xing Zhang administrated the project; Hua-xing Zhang collected resources; Chun-miao Zhang helped with software; Hua-xing Zhang wrote the original draft; Hua-xing Zhang reviewed and edited the study.

Acknowledgments

This work was supported by the Zhejiang Provincial Natural Science Foundation of China, under Grant Y18F030036.

References

[1] A. Sbihi and R. W. Eglese, "Combinatorial optimization and green logistics," *4OR*, vol. 5, no. 2, pp. 99–116, 2007.

[2] R. Dekker, J. Bloemhof, and I. Mallidis, "Operations research for green logistics—an overview of aspects, issues, contributions and challenges," *European Journal of Operational Research*, vol. 219, no. 3, pp. 671–679, 2012.

[3] R. B. Lopes, C. Ferreira, B. S. Santos, and S. Barreto, "A taxonomical analysis, current methods and objectives on location-routing problems," *International Transactions in Operational Research*, vol. 20, 2013.

[4] O. Dukkanci, B. Y. Kara, and T. Bektaş, "The green location-routing problem," *Computers & Operations Research*, vol. 105, pp. 187–202, 2019.

[5] Ç. Koç, T. Bektas, O. Jabali, and G. Laporte, "The impact of depot location, fleet composition and routing on emissions in city logistics," *Transportation Research Part B: Methodological*, vol. 84, pp. 81–102, 2016.
[6] V. F. Yu, N. M. E. Normasari, and W.-H. Chen, "Location-routing problem with time-dependent demands," *Computers & Industrial Engineering*, vol. 151, Article ID 106936, 2021.
[7] K. Alamatsaz, A. Ahmadi, and S. M. J. Mirzapour Al-E-Hashem, "A multiobjective model for the green capacitated location-routing problem considering drivers' satisfaction and time window with uncertain demand," *Environmental Science and Pollution Research*, vol. 29, no. 4, pp. 5052–5071, 2021.
[8] C. Zhang, Y. Zhao, and L. Leng, "A hyper-heuristic algorithm for time-dependent green location routing problem with time windows," *IEEE Access*, vol. 8, 2020.
[9] R. Pitakaso, K. Sethanan, and C. Theeraviriya, "Variable neighborhood strategy adaptive search for solving green 2-echelon location routing problem," *Computers and Electronics in Agriculture*, vol. 173, Article ID 105406, 2020.
[10] X. Wang and X. Li, "Carbon reduction in the location routing problem with heterogeneous fleet, simultaneous pickup-delivery and time windows," *Procedia Computer Science*, vol. 112, pp. 1131–1140, 2017.
[11] R. Vakili, M. Akbarpour Shirazi, and H. Gitinavard, "Multi-echelon green open-location-routing problem: a robust-based stochastic optimization approach," *Scientia Iranica*, vol. 28, 2020.
[12] E. M. Toro, J. F. Franco, M. G. Echeverri, and F. G. Guimarães, "A multi-objective model for the green capacitated location-routing problem considering environmental impact," *Computers & Industrial Engineering*, vol. 110, pp. 114–125, 2017.
[13] M. Rabbani, M. Davoudkhani, and H. Farrokhi-Asl, "A new multi-objective green location routing problem with heterogonous fleet of vehicles and fuel constraint," *International Journal of Strategic Decision Sciences*, vol. 8, no. 3, pp. 99–119, 2017.
[14] J. Tang, S. Ji, and L. Jiang, "The design of a sustainable location-routing-inventory model considering consumer environmental behavior," *Sustainability*, vol. 8, no. 3, p. 211, 2016.
[15] M. Rabbani, F. Navazi, E. Niloofar, and F.-A. Hamed, "A green transportation location-inventory-routing problem by dynamic regional pricing," *Journal of Industrial Engineering and Management Studies*, vol. 7, pp. 35–58, 2020.
[16] Y. Wang, S. Peng, X. Zhou, and M. L. Mahmoudi, "Green logistics location-routing problem with eco-packages," *Transportation Research Part E: Logistics and Transportation Review*, vol. 143, Article ID 102118, 2020.
[17] F. Faraji and B. Afshar-Nadjafi, "A bi-objective green location-routing model and solving problem using a hybrid metaheuristic algorithm," *International Journal of Logistics Systems and Management*, vol. 30, 2018.
[18] M. Tavana, H. Tohidi, M. Alimohammadi, and R. Lesansalmasi, "A location-inventory-routing model for green supply chains with low-carbon emissions under uncertainty," *Environmental Science and Pollution Research*, vol. 28, no. 36, pp. 50636–50648, 2021.
[19] L. Leng, Y. Zhao, Z. Wang, H. Wang, and J. Zhang, "Shared mechanism-based self-adaptive hyperheuristic for regional low-carbon location-routing problem with time windows," *Mathematical Problems in Engineering*, vol. 2018, Article ID 8987402, 21 pages, 2018.
[20] Z. Ebrahim Qazvini, A. Haji, and H. Mina, "A fuzzy solution approach for supplier selection and order allocation in green supply chain considering location-routing problem," *Scientia Iranica*, vol. 28, 2019.
[21] M. Sadati and B. Atay, "A hybrid variable neighborhood search approach for the multi-depot green vehicle routing problem," *Transportation Research Part E Logistics and Transportation Review*, vol. 149, no. 4, Article ID 102293, 2021.
[22] Z. Zarrat Dakhely Parast, H. Haleh, S. Avakh Darestani, and H. Amin-Tahmasbi, "Green reverse supply chain network design considering location-routing-inventory decisions with simultaneous pickup and delivery," *Environmental Science and Pollution Research*, vol. 14, pp. 1–22, 2021.
[23] M. E. Tayebi Araghi, R. Tavakkoli-Moghaddam, F. Jolai, and S. M. Hadji Molana, "A green multi-facilities open location-routing problem with planar facility locations and uncertain customer," *Journal of Cleaner Production*, vol. 282, no. 3, Article ID 124343, 2021.
[24] S. Khalafi and Z. Mahnaz, "Design of a green supply chain network by considering the possibility of storage and location-routing problem under uncertainty," *International Journal of Logistics Systems and Management*, vol. 34, pp. 253–268, 2019.
[25] T. Qiannan, P. N. Ram Kumar, L. Wenli, and L. Kunpeng, "Simultaneous product and service delivery vehicle routing problem with time windows and order release dates," *Applied Mathematical Modelling*, vol. 89, pp. 669–687, 2021.
[26] M. Rahbari, A. A. Khamseh, Y. Sadati-Keneti, and M. J. Jafari, "A risk-based green location-inventory-routing problem for hazardous materials: NSGA II, MOSA, and multi-objective black widow optimization," *Environment Development and Sustainability*, vol. 24, 2012.
[27] K. M. Huang, C. W. Lu, and M. J. Lian, "Modeling and algorithm for multi-echelon location-routing problem," *Kongzhi yu Juece/Control and Decision*, vol. 32, no. 10, pp. 1803–1809, 2017.
[28] A. Rybičková, D. Mockova, and D. Teichmann, "Genetic algorithm for the continuous location-routing problem," *Neural Network World*, vol. 29, no. 3, pp. 173–187, 2019.
[29] I. Moon, S. Salhi, and X. Feng, "The location-routing problem with multi-compartment and multi-trip: formulation and heuristic approaches," *Transportmetrica: Transportation Science*, vol. 16, no. 3, pp. 501–528, 2020.
[30] X.-Y. Li, P. Tian, and S. C. H. Leung, "An ant colony optimization metaheuristic hybridized with tabu search for open vehicle routing problems," *Journal of the Operational Research Society*, vol. 60, no. 7, pp. 1012–1025, 2009.
[31] H. Raoofpanah and V. Ghezavati, "Extended hybrid tabu search and simulated annealing algorithm for location-inventory model with multiple products, multiple distribution centers and multiple capacity levels," *Production Engineering*, vol. 13, pp. 649–663, 2019.
[32] F. M. Müller and I. S. Bonilha, "Hyper-heuristic based on ACO and local search for dynamic optimization problems," *Algorithms*, vol. 15, 2021.
[33] N. R. Sabar, X. J. Zhang, and A. Song, "A math-hyper-heuristic approach for large-scale vehicle routing problems with time windows," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC)*, Sendai, Japan, May 2015.
[34] T. N. Ferreira, J. A. P. Lima, A. Strickler, J. N. Kuk, S. R. Vergilio, and A. Pozo, "Hyper-heuristic based product

selection for software product line testing," *IEEE Computational Intelligence Magazine*, vol. 12, no. 2, pp. 34–45, 2017.

[35] A. Strickler, J. A. Prado Lima, S. R. Vergilio, and A. Pozo, "Deriving products for variability test of feature models with a hyper-heuristic approach," *Applied Soft Computing*, vol. 49, pp. 1232–1242, 2016.

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

[37] G. Ioannou, M. Kritikos, and G. Prastacos, "A greedy look-ahead heuristic for the vehicle routing problem with time windows," *Journal of the Operational Research Society*, vol. 52, no. 5, pp. 523–537, 2001.