IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 January 2024, accepted 13 February 2024, date of publication 23 February 2024, date of current version 7 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3369474

RESEARCH ARTICLE

A Deep Reinforcement Learning-Based Adaptive Search for Solving Time-Dependent Green Vehicle Routing Problem

BIN YUE, JUNXU MA, JINFA SHI, AND JIE YANG
School of Management and Economics, North China University of Water Resources and Electric Power, Zhengzhou 450046, China

Corresponding authors: Jinfa Shi (shijinfa@ncwu.edu.cn) and Junxu Ma (majunxu@ncwu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China: Research on Enterprise Resource Location Optimization Based on the Internet of Things under Grant 71371172, and in part by the Major Science and Technology Project of Henan Province under Grant 232102220089.

ABSTRACT The time-dependent green vehicle routing problem with time windows is a further deepening of the research on vehicle routing problems with time windows. Its simultaneous consideration of vehicle transportation time, carbon emissions, and customer satisfaction under time-dependent variables makes it more challenging to solve than traditional vehicle routing problems. This work proposes a multi-objective optimization algorithm that combines the learnable crossover strategy and the adaptive search strategy based on reinforcement learning to overcome the local optima, poor convergence, and reduced variety of solutions that plague the multi-objective optimization algorithms when solving this problem. The proposed approach solves the problem in two stages: In the first stage, a hybrid initialization strategy is used to generate initial solutions with high quality and diversity, and crossover strategies are used to further explore the solution space and improve convergence by learning the characteristics of pareto solutions. In the second stage, the adaptive search is designed and used for learning and searching in the later stage of the algorithm. The experimental results show better solution quality obtained by the proposed approach, and the effectiveness and superiority of the proposed approach over existing methods in terms of solution convergence and diversity are demonstrated through experimental comparisons.

INDEX TERMS Multi-objective optimization, DQN, GVRPTW, time-dependent, customer satisfaction.

I. INTRODUCTION

Vehicle routing problems (VRP) belong to the typical NP-hard problems [1]. It has attracted extensive research since its formulation, and it is one of the most studied combinatorial optimization problems, with great practical implications for the logistics and transportation industries. The vehicle routing problem within a green context with the aim of optimizing economic, environmental, and social advantages has recently become a hot study area due to the strengthening of the green concept in the transportation industries.

Theoretically, the multi-objective optimization VRP is an extension of the ordinary VRP. Conflicting yet concurrent optimization objectives are typically present in a multi-objective optimization problem that represents real scenarios. Among the multiple optimization objectives of VRP, optimizing the transportation time spent by vehicles to and from customers is the key to the VRP optimization problem, which is also one of the key optimization targets of the multi-objective VRP optimization problem. Most studies simplified the transportation time between customers and set it as a constant [2], [3], [4], thereby ignoring the impact of time-dependent factors, especially the traffic condition, on the delivery time. In addition, the quality of the goods with strict requirements on the efficiency of delivery will be directly affected by the traffic congestion. Malandraki and DASKIN [5] provided the first definition of the time-dependent vehicle routing problem (TDVRP) and discussed the associated solution to this problem by proposing a velocity segmentation function. Considering that the travel

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

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

time in the TDVRP model will interfere with the principle of ''first-in, first-out'' (FIFO), Ichoua et al. [6] took the effect of time-dependent speed on the delivery time into consideration in their related research and used the speed step function to obtain a segmented linear travel time function to satisfy the principle of FIFO, which makes the TDVRP more practical for applications. Figliozzi [7] added the consideration of changing urban traffic conditions to make the time-dependent vehicle routing problem more realistic. However, most existing studies related to GVRPTW assume that the vehicle speed is a constant value, which ignores the effect of vehicle speed on delivery time. Therefore, in this paper, the GVRPTW model is constructed based on the time-dependent travel speed model.

Traffic conditions cause variations in both delivery time and vehicle carbon emissions. However, travel distance is not the only factor that influences emissions. Sawik et al. [8] analyzed the factors that impact environmental costs, total distance, carbon emissions, and fuel consumption on GVRP. The results showed that the energy consumption and carbon emissions of a vehicle are not simply positively correlated with the travel time but also affected by factors such as the load of vehicles, speeds, and road gradients [9]. Therefore, the travel time does not accurately reflect the level of vehicle carbon emissions, and when carbon emissions are one of the optimization objectives, the travel time cannot be used as the only criterion for assessing vehicle carbon emissions. In the modern logistics industry with outstanding service characteristics, customer satisfaction must also be taken into account, in addition to carbon emissions and vehicle transportation time. Related studies [10], [11], [12] show that customer satisfaction will be affected by the delivery time. Customers who require delivery service do not generally expect vehicles to arrive as soon as possible or as quickly as possible; rather, they are more likely to expect vehicles to arrive within the time windows they have specified. Therefore, time-dependent factors and factors affecting vehicle carbon emissions need to be taken into account in practical applications. The research [13], [14], [15], [16] that is pertinent to the aforementioned constraints, however, is insufficient and only covers the single-objective TDVRPTW.

Essentially, the multi-objective optimization of VRP is a specific application of multi-objective optimization theory in transportation science. As the number of optimization objectives increases, the problem becomes more complex and challenging. Mojtaba Ghasemi et al. [17], [18], [19] have employed nature-inspired algorithms to address challenges across various scales, achieving notable success in optimization outcomes. These algorithms have demonstrated particular efficacy in tackling large-scale problems, yielding impressive results. Furthermore, achieving simultaneous improvements in both convergence and diversity within the solution set of a multi-objective optimization problem presents a significant challenge [20], [21], [22]. How to obtain the relatively optimal ''equilibrium solution'' between the objectives is the focus of such problems. Currently, multi-objective optimization algorithms based on dominance relations have been mostly used to solve continuous as well as discrete multi-objective optimization problems [23], [24]. Compared with the traditional weighting and optimization methods, dominance relation-based multi-objective optimization algorithms can balance the conflicts among multiple optimization objectives and avoid weight allocation among optimization objectives. Dominance relation-based optimization algorithms use dominance relations to determine whether to retain the current solution [25], [26], [27]. In this way, the manager has more options to choose from and will be better able to make a reasonable decision according to the actual situation. However, the drawback is that, during the optimization process, multi-objective optimization algorithms are prone to settling on local optima. In this regard, the use of the crossover strategy and local search method improves the problem to a certain extent and enhances the global optimization ability of multi-objective optimization algorithms [28], [29], [30], but most existing crossover strategies are only a simple reorganization of the encoding of solutions and do not refer to the ''coding structure'' of the superior solution, making the crossover operation more random. Moreover, the equal probability selection of local search strategy ignores the search knowledge generated during the search process [31], [32], [33], which leads to the problem that the algorithm generates ''blind'' search and is prone to fall into the problem of local optimal solutions. As a result, it is necessary to refer to the encoding of superior solutions and to utilize empirical knowledge so that the selection of the local search strategy is knowledge-based rather than completely equiprobable and random. RL [34], [35] is a type of machine learning that trains an agent to optimize actions by learning from accumulating experiences. With this feature, RL meets expectations. Q-learning [36] is a prominent reinforcement learning algorithm that gets the optimal policy by updating the state and action. However, Q-learning is not suitable for the scenario that has a large state-action space. Therefore, we intend to apply DQN to address such problems.

Based on the above research analysis, the following work is done in this study: A time-dependent multi-objective GVRP optimization problem (TD-GVRPTW) is proposed, taking into account both the practical requirements and the deficiencies of the existing related research. A multi-objective optimization model that simultaneously considers three optimization objectives is constructed. The DQN-based two-stage multi-objective optimization algorithm DQMOEA is proposed to solve the model. Initially, a hybrid initialization approach is employed to preliminarily solve datasets characterized by diversity, yielding high-quality and varied initial solution sets. Two pareto front-based crossover strategies are designed to learn the location information of customers in the pareto optimal solution, which can improve the convergence performance in the later stage of the algorithm. The DQN-based adaptive search uses a tuple including routing

sequence, time information, requests about customers, and the vehicle's load to represent the current state space. One of the five heuristics is selected to execute the corresponding actions. Finally, the effectiveness of the model and algorithm established in this paper is verified by extensive experiments on the generated benchmark instances.

A. CONTRIBUTION OF THIS WORK

Based on the current relevant research, the existing literature has not adequately addressed the multi-objective optimization problem of green vehicle scheduling under time-dependent constraints. To fulfill this gap, this paper makes the following contributions:

1) A time-dependent multi-objective optimization VRP (TD-GVRPTW) is proposed. A multi-objective optimization model that simultaneously considers three optimization objectives: transportation time, carbon emissions, and customer satisfaction, is constructed.
2) We propose an efficient multi-objective optimization algorithm that solves the constructed model in two stages.
   In the first stage, we use a hybrid initialization strategy that exploits the diversity characteristics of the datasets to provide high-quality and diversified initial solutions. We also design pareto front-based crossover strategies that learn the location information of customers in the pareto optimal solutions, which can enhance the convergence performance of the algorithm.
   In the second stage, we use a novel representation of the vehicle dispatching state for the TD-GVRPTW. We develop a deep reinforcement learning method that takes state tuples as input for the reinforcement learning model, improves and searches the initial solutions obtained in the first stage, and finally achieves optimal or near-optimal solutions.
   With the proposed optimization algorithm, the following practical advantages are considered:
   - By using the hybrid initialization method, the initial solution can enhance both the efficiency and the quality of the subsequent solution in the second stage.
   - The proposed state representation in this paper enables the model inputs to be flexibly adapted based on the optimization objective.
   - The algorithm can achieve solutions with high convergence and diversity performance for multi-objective optimization problems.

II. PROBLEM DESCRIPTION AND MODEL DEVELOPMENT

A. MODEL PARAMETERS

In this paper, the TD-GVRPTW is defined as a completely directed graph $G = (C', E)$. $C' = C \cup \{c_0\}$ denotes nodes in the directed graph G. $E = \{e(i,j)|i,j \in C, i \neq j\}$ denotes all directed edges in graph G. Assume that the depot has a number of delivery vehicles. Each vehicle will leave the depot during the specified working hours, loaded with goods weighing no more than Q, and provide distribution services to a limited number of customers. Each vehicle must return to the depot immediately after completing the delivery task. Without loss of generality, this paper considers the customer's service time window as a soft time window, and the following assumptions and constraints need to be satisfied:

Assumption 1: The range of customer satisfaction is [1, 0]. It is allowed that the vehicle arrive at the customer's location

TABLE 1. The notation and meaning of variables.

Notation	Meaning
C	Customers set: $C = \{c_1, c_2, \ldots, c_n\}$
K	Vehicles set, $k = \{1,2,3,\ldots,\lvert K\rvert\}$, k is the current vehicle number
n	Number of customers
c_i	Customer i, c_0 denotes the depot
C'	$C' = C \cup \{c_0\}$
q_i	Demand for goods by customer i
st_i	Service time required for customer i
$[ee_i, ll_i]$	Customer i's allowed service time window
$[e_i, l_i]$	Ideal service time window for customer i
$e(i,j)$	Route between customer i and customer j
d_{ij}	Distance between customer i and customer j
w_{ij}	Inclination of route $e(i,j)$
t_{ij}	Transportation time of vehicles from customer i to customer j
Q	Maximum load of vehicles
z_r	Time zone $r = \{1,2,3,\ldots\}$
s_{z_r}	Vehicle travel speed in time zone r
t_{ij}^{kr}	Vehicle k departs from point i and drives to point j in time region r
a_j^{kr}	Moment when vehicle k departs in time zone r to reach customer j
ls_{ij}^{kr}	Maximum travel time of vehicle k in time zone r
x_{ij}^{kr}	Binary variable, indicates whether vehicle k departs from customer i to customer j in time zone r
c_{ij}^{kr}	Carbon emissions from vehicle k passing through $e(i,j)$ in time zone r
FC_{ij}	Fuel consumption of vehicle k on road segment $e(i,j)$
FE	Carbon emissions per unit of fuel consumed by vehicles
F_k^r	The maximum distance traveled by vehicle k within time zone r.
e_{ij}^{kr}	The actual distance traveled by the vehicle k on route $e(i,j)$ after through time zone r
J_{ij}^r	After passing through time zone r, the distance between vehicle k and customer j.
l_{ij}	the load of the vehicle when traveling on route $e(i,j)$
a	the accelerations of vehicle
C_r	the road rolling resistance coefficient
C_d	the traction coefficient
ρ	the air density
ψ	The type of vehicle
f_1	vehicle transportation time
f_2	vehicle carbon emission
f_3	customer satisfaction

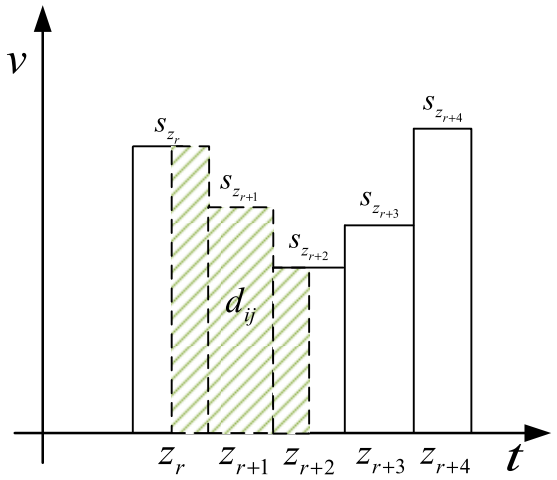

FIGURE 1. Time zones spanned from customer i to j.

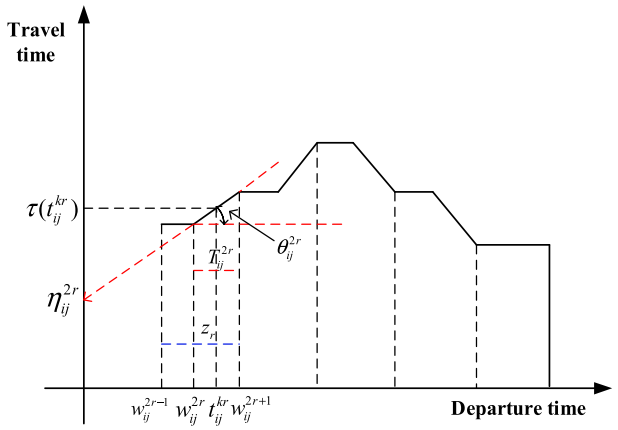

FIGURE 2. The travel time for route e(i,j).

before ee_i or after ll_i, but in this case, the customer satisfaction is 0. When the vehicle arrives at the customer's location in $[e_i, l_i]$, the customer satisfaction is 1.

Assumption 2: All vehicles are of the same type and are available in numbers to meet the delivery tasks.

Assumption 3: The route $e(i, j)$ between any customer c_i and customer c_i has a corresponding path inclination w_{ij}.

Constraint 1: The vehicle must complete the delivery task for the current distribution route and return to the depot within the specified time.

Constraint 2: The total demand $(\sum_{i=1}^{n} q_i)$ of all customers on the current distribution route does not exceed the maximum load of each vehicle.

Constraint 3: Each customer can and will only be served once.

Subject to the above constraints and assumptions, we aim to determine the distribution routes of all vehicles and, at the same time, minimize the vehicle transportation time f_1, vehicle carbon emission f_2, and maximize the value of customer satisfaction f_3.

B. FORMULATION OF TIME-DEPENDENT TRAVEL TIME

Ichoua et al. [6] assumed that the vehicle traveling speed can be considered a fixed value for a short period of time. According to this assumption, the working day can be divided into several time zones: $T = \{z_1, z_2, z_r, \ldots, z_p\}, z_r = [tt_{r-1}, tt_r]$. Define the velocity-time function as a stepwise function: $v = s(z_r)$. As shown in Fig. 1, for a directed edge $e(i, j)$ with a distance of d_{ij}, there is a probability that the time required for a vehicle to pass through edge $e(i, j)$ will span multiple time zones.

Assume that the vehicle k departs from customer i to customer j in time zone $z_r = [tt_{r-1}, tt_r]$, and denote this departure moment as t_{ij}^{kr}. Then, the traveling speed s_{z_r} and the maximum traveling time $ls_{ij}^{kr} = tt_r - t_{ij}^{kr}$ of vehicle k in this time zone can be obtained.

Based on the research of Ichoua et al. [6], the vehicle transportation time function can be modeled as a segmented linear function. It is assumed that the transportation time through the edge $e(i, j)$ spans a maximum of two time zones. As shown in Fig. 2, divide the time zone r into two parts: $T_{ij}^{2r-1} = \left[w_{ij}^{2r-1}, w_{ij}^{2r}\right], T_{ij}^{2r} = \left[w_{ij}^{2r}, w_{ij}^{2r-1}\right]$. $w_{ij}^1, w_{ij}^2, w_{ij}^3, \ldots, w_{ij}^{2r}, w_{ij}^{2r+1}$ are function breaks.

When the departure time of the vehicle is within T_{ij}^{2r-1}, the vehicle passes through only one time zone for the route $e(i, j)$. In contrast, when the vehicle's departure time is within T_{ij}^{2r}, the vehicle will cross two time zones, and the corresponding transportation time will change and will no longer be a fixed value. The slope θ_{ij}^{2r} of the function and the corresponding intercept η_{ij}^{2r} in the time zone z_r are calculated as follows:

$$\theta_{ij}^{2r} = \frac{\tau(w_{ij}^{2r+1}) - \tau(w_{ij}^{2r})}{w_{ij}^{2r+1} - w_{ij}^{2r}} \tag{1}$$

$$\eta_{ij}^{2r} = \frac{w_{ij}^{2r+1}\tau(w_{ij}^{2r}) - w_{ij}^{2r}\tau(w_{ij}^{2r+1})}{w_{ij}^{2r+1} - w_{ij}^{2r}} \tag{2}$$

Thus, given an arbitrary vehicle departure time t_{ij}^{kr} in the time zone z_r, the transportation time of a vehicle can be calculated using the following equation:

$$\tau(t_{ij}^{kr}) = \theta_{ij}^{r} t_{ij}^{kr} + \eta_{ij}^{r} \tag{3}$$

Let $x_{ij}^{kr} = 1$ denote that vehicle k departs from customer i to customer j in the time zone z_r. Conversely, $x_{ij}^{kr} = 0$. Then the time required for vehicle k to pass through the edge $e(i, j)$ can be calculated by the following equation:

$$t_{ij} = \sum_{k \in V} \sum_{r=1}^{p} \tau(t_{ij}^{kr}) x_{ij}^{kr} \tag{4}$$

Finally, travel time for all vehicles can be calculated using the following formula:

$$f_1 = \min \sum_{i \in C'} \sum_{j \in C', j \neq i} (t_{ij} + st_j) \tag{5}$$

The following pseudocode can be used to calculate the travel time of a vehicle from customer i to customer j.

input: t_{ij}^{kr}, $d_{ij}F_k^r = ls_{ij}^{kr} \cdot s_{z_r}$
$J_{ij}^r = d_{ij} - e_{ij}^{kr}$, $J_{ij}^r \geq 0$
1: if $F_k^r \geq d_{ij}$:
2: $J_{ij}^r = 0$; $e_{ij}^{kr} = d_{ij}$; $t_{ij} = \frac{d_{ij}}{s_{zr}}$
3: else
4: $J_{ij}^r = d_{ij} - F_k^r$; $e_{ij}^{kr} = F_k^r$; $t_{ij} += ls_{ij}^{kr}$
5: ζ =1
6: While True:
7: $F_k^{r+\zeta} = s_{z_{(r+\zeta)}}\tau(t_{ij}^{kr})$
8: If $F_k^{r+\zeta} \leq J^{r+\zeta-1}$:
9: $J^{r+\zeta} = J_{ij}^r - F_k^{r+\zeta}$; $e_{ij}^{k,r+\zeta} = F_k^{r+\zeta}$; $t_{ij} += \tau$; $\zeta += 1$
10: Else:
11: $J^{r+\zeta} = 0$; $e_{ij}^{k,r+\zeta} = J_{ij}^{r+\zeta-1}$; $t_{ij} += \frac{J_{ij}^{r+\zeta-1}}{s_{z_{(r+\zeta)}}}$
12: Break
13: End if
14: End while
15: End if
output: travel time t_{ij}

C. CARBON EMISSION MODEL

Vehicles powered by fossil fuels such as petroleum will generate carbon emissions, and the vehicle's carbon emissions are affected by factors such as vehicle load, travel time, and vehicle speed. According to Hoen et al. [37], the carbon emission of a vehicle from customer i to customer j can be calculated by the following equation:

$$EC_{ij} = FE \cdot FC_{ij} \tag{6}$$

where FE is the carbon emissions per unit of oil consumed by the vehicle, based on the European carbon emission calculation standard, this paper sets FE as 2621 g/L. Referring to Bektas and Laporte [38], combined with the research content of this paper, the following formula is used to calculate the fuel consumption FC_{ij} when the vehicle travels the distance d_{ij}, and the factors affecting the vehicle's carbon emissions are considered comprehensively.

$$FC_{ij} = \begin{cases} (a_{ij}(w + l_{ij}) + \psi s_{z_r}^2)d_{ij}, \\ \qquad t_{ij}^{kr} \in z_r, t_{ij}^{kr} + \tau(t_{ij}^{kr}) \in z_r \\ (a_{ij}(w+l_{ij})+\psi s_{z_r}^2)e_{ij}^{kr} + (a_{ij}(w+l_{ij})+\psi s_{z_{r+1}}^2)(d_{ij}-e_{ij}^{kr}) \\ \qquad t_{ij}^{kr} \in z_r, t_{ij}^{kr} + \tau(t_{ij}^{kr}) \notin z_r \end{cases} \tag{7}$$

$$a_{ij} = a + g\sin w_{ij} + gC_r \cos w_{ij}, \beta = 0.5C_dA\rho \tag{8}$$

where w denotes the empty vehicle mass, l_{ij} denotes the load of the vehicle when traveling on route $e(i, j)$. a_{ij} denotes the parameter related to route $e(i, j)$, which is determined by factors such as road gradient, rolling resistance, etc., and ψ denotes the parameter related to the vehicle type. Where a denotes the accelerations (m/s^2) of the vehicle, the velocity difference between two consecutive time zones ($s_{z_{r+1}} - s_{z_r}$) is defined as accelerations in this paper. g denotes the gravitational acceleration constant, C_r denotes the road rolling resistance coefficient, C_d denotes the traction coefficient, A denotes the frontal surface area (m^2) of the vehicle, and ρ denotes the air density (kg/m^3). According to the research of Bektas and Laporte [38], the traction coefficient C_d of a general loaded vehicle is taken to be 0.7, windward area A is 5m^2, the air density ρ is taken as 1.204 kg/m^3, and the rolling resistance coefficient for a typical concrete road is 0.012. Bringing the above parameters into the equation yields the value of β as 2.107 and the value of a_{ij} as $s_{z_{r+1}} - s_{z_r} + 9.81 \times (\sin w_{ij} + 0.012\cos w_{ij})$. The carbon emissions of all vehicles are calculated as follows:

$$f_2 = \sum_{i\in C'} \sum_{j\in C'\setminus\{i\}} \sum_{k\in V} \sum_{r\in\{1,\ldots,p\}} x_{ij}^{kr} \cdot e_{ij}^{kr} \cdot EC_{ij} \tag{9}$$

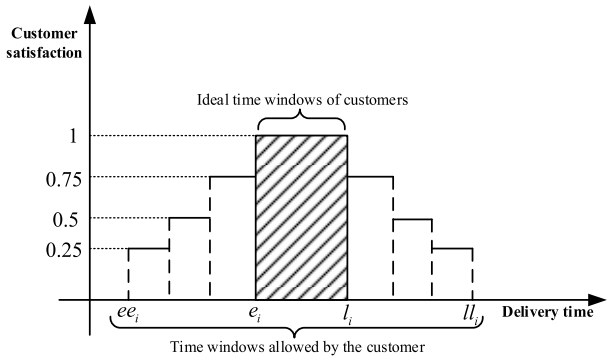

FIGURE 3. Customer satisfaction function.

D. CUSTOMER SATISFACTION

Customer satisfaction is an important metric for assessing distribution effectiveness in actual logistics systems. This research models customer satisfaction as a segmented function using the rating approach in order to measure the customer satisfaction more accurately. As shown in Fig. 3, the service time windows allowed for each customer are divided into two cases: Ideal time windows $[e_i, l_i]$; feasible time windows $[ee_i, ll_i]$. Categorize customer satisfaction into five levels: $le \in \{0, 0.25, 0.5, 0.75, 1\}$.

If the vehicle arrives within the customer's ideal time window, customer's satisfaction is 1. Otherwise, the customer's satisfaction varies gradually with the delivery time. Customer satisfaction $cs(a_j^{kr})$ is calculated as follow:

$$cs(a_j^{kr}) = \begin{cases} 0, a_j^{kr} < ee_i || a_j^{kr} > ll_i \\ 1, e_i \leq a_j^{kr} \leq l_i \\ 1 - \frac{1}{4} \cdot \zeta, e_i - le \cdot \zeta < a_j^{kr} < e_i - le \cdot (\zeta - 1) \\ 1 - \frac{1}{4} \cdot \zeta, l_i - le \cdot \zeta < a_j^{kr} < l_i - le \cdot (\zeta - 1) \end{cases} \tag{10}$$

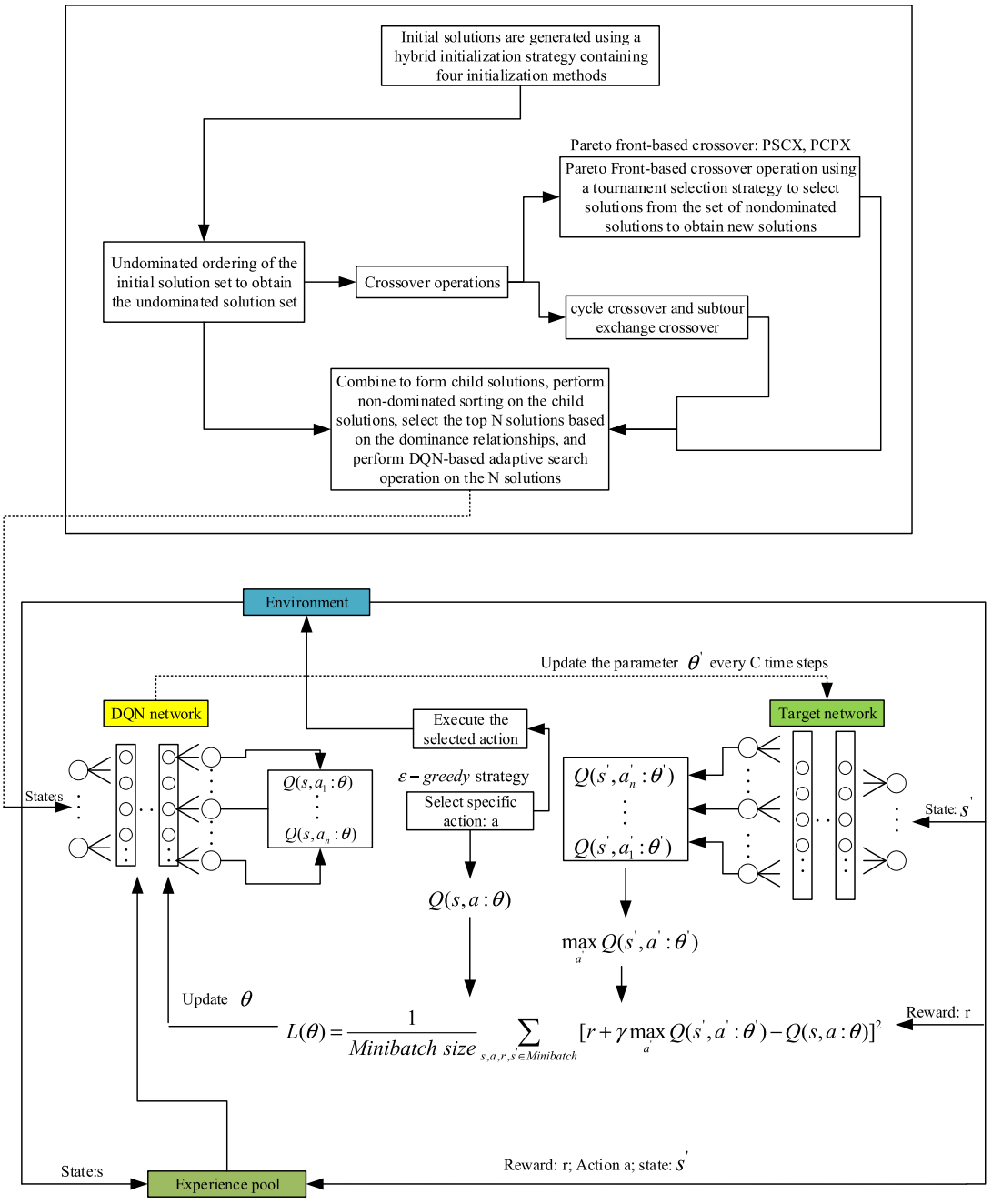

FIGURE 4. The overall process of algorithm.

where $le = \frac{e_i - ee_i}{3}$, $\zeta \in \{1, 2, 3\}$. Maximizing overall customer satisfaction f_3 means minimizing $\frac{1}{f_3}$. The calculation is shown in Equations 11 and 12:

$$a_j^{kr} = \tau\left(\sum_{j=1}^{j-1}\tau\left(t_{j-1,j}^{kr}\right) + st_j\right) \tag{11}$$

$$\frac{1}{f_3} = \min\frac{1}{\sum_{k\in V}\sum_{i\in C}\sum_{j\in C'}\sum_{r\in\{1,\ldots,p\}} cs_i\left(a_j^{kr}\right)} \tag{12}$$

III. ALGORITHM DESCRIPTION

In this section, we propose a two-stage optimization algorithm that includes the diversity solution initialization phase and the DQN-based adaptive search to solve the above three-objective optimization model. The overall framework is shown in Fig. 4.

A. INITIALIZATION OF SOLUTIONS

Each initial solution is represented by a two-dimensional vector with length n. The first dimension vector represents the order in which the vehicle serves n customers. The second dimension vector represents the vehicle corresponding to each served customer. Fig. 5 shows the encoding of a feasible solution containing 11 customers and 3 delivery vehicles, generating a total of 3 distribution routes.

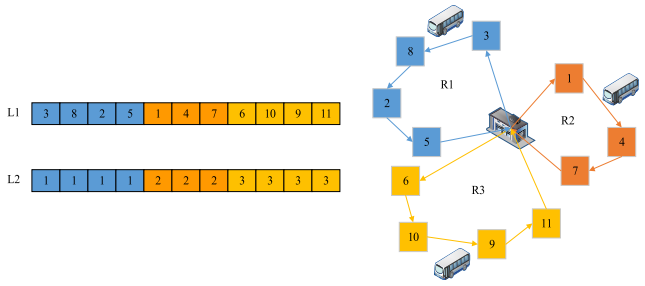

FIGURE 5. Encoding of initial solutions.

In combinatorial optimization problems, high quality and diversity of initial solutions can result in more satisfying optimization results [39]. In reality, the geographic location between customers will show different characteristics, and the sequence of vehicle delivery to customers is closely related to the final delivery time. In addition, customers will also have different requirements for delivery time. In order to generate initial solutions that match the characteristics of the customer's location distribution as well as diversity, this paper uses a hybrid initialization strategy to initialize the population solutions, which include the random generation method, the k-nearest neighbor heuristic (K-NNH), the improved push forward insertion heuristic (IPFIH), and the ideal time window selection heuristic (ITH), respectively. Each initialization method generates one-fourth of the total number of population solutions.

1) GENERATE INITIAL SOLUTIONS USING RANDOMIZED GENERATION METHODS

The random generation method is more suitable for the case where customer locations are scattered and random. First, a delivery route is initialized, and customers to be delivered are randomly added to the current path until constraints 1∼3 are no longer fully satisfied. Then another delivery vehicle is activated to continue to complete the delivery tasks of the remaining customers until all the delivery tasks are completed.

2) K-NNH (K-NEAREST NEIGHBOR HEURISTIC) INITIALIZATION

Considering the type of clustering customers, a k-nearest neighbor heuristic is used to initialize partial solutions. First, a delivery route is initialized, assuming that customer i is the current customer served by the vehicle. The next customer to be served is one of the k-closest customers to customer i. The rest of the customers will be continually added to the current distribution route until the delivery needs of all customers are met. Otherwise, generate another distribution route.

3) ITH INITIALIZATION

There may be cases in which the customer has more stringent requirements for the service time windows, for which the ITH initialization part is used to solve the problem.

The pseudocode of ITH.

Input: the number of customers (N), service time windows
1: Sort all customers by ideal service time windows and obtained the sorted sequence S.
2: **for** $i = 1$ to N **do:**
3: Select the customers $S(i)$ based on the order of non-incremental ideal time windows
4: **if** constraint 1 and constraint 2 are satisfied **do:**
5: Insert this customer into the current route and update the current load and transit time for the current vehicle.
6: **else:**
Mark that the current vehicle's delivery task is completed
7: Initializing a new vehicle and new route
8: **end if**
9: **end for**
10: Output the initial solutions.

4) IPFIH INITIALIZATION METHOD

The push forward insertion heuristic [40] (PFIH) was proposed by Solomon and is an effective construction heuristic for the VRPTW problem. In this paper, we use a new initialization method (IPFIH) based on PFIH for selecting the initial customer and K-NNH to select the next customers to be served in the current distribution route. Initial customers are selected according to the following equations:

$$h_i = -\omega d_{0i} + \xi ll_i + \sigma \cdot (\frac{|p_i|}{360}) \tag{13}$$

$$h_j = -\omega d_{ji} + \xi ll_j + \sigma \cdot (\frac{|p_j - p_1|}{360}) \tag{14}$$

where d_{0i} denotes the distance between customer i and the depot, p_i denotes the polar coordinates of customer i with respect to the depot. The larger the value of h, the greater the probability that customer i will be the initial node.

B. CROSSOVER STRATEGIES

Multi-objective optimization algorithms are prone to low search efficiency and high volatility of the optimal solutions obtained. A large part of the reasons that cause the local optima is that the search strategy carries a large degree of randomness, and the new generating solutions do not reference the quality solutions that have already been generated. The pareto-optimal solution is the optimal equilibrium solution obtained through multi-objective optimization. Therefore, the Pareto optimal solutions contain the encoding features that can be learned for reference. In this section, we use cycle crossover and subtour exchange crossover and propose two crossover strategies based on Pareto front: similar customer order crossover (PSCX) and customer pair order crossover (PCPX) to balance global search and solution quality. The operational details are as follows:

PSCX: First, the temporary route is constructed based on the customers with the highest frequency of occurrence at

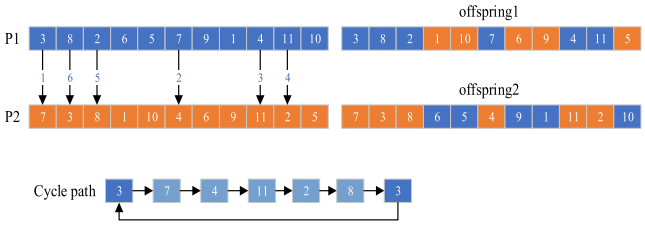

FIGURE 6. The cycle crossover operation.

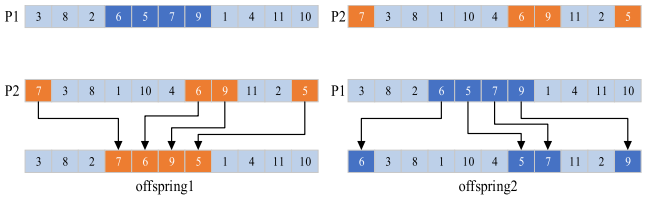

FIGURE 7. The subtour exchange crossover.

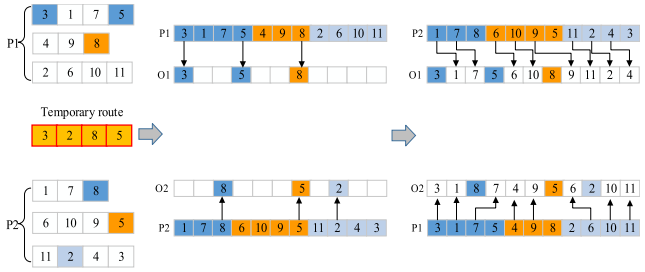

FIGURE 8. The operation process of PSCX.

each location in the pareto solution. Then two parent solutions P1 and P2 are randomly selected from the non-pareto solutions, and the customer position on each route of the parent solution is compared with the customers on the temporary route. If the customer and their positions are the same, this customer is placed in the same position as its offspring solution, and the vacancy in the offspring solution is filled by the different customers of another parent solution. The specific operation is shown in Fig. 8.

PCPX: As shown in Fig. 9, for each customer i, select the neighboring "customer pairs" [i, j] that appear for the first time in each Pareto optimal solution to construct the reference set. Two parent solutions, P1 and P2, are then randomly selected and compared with the constructed reference set. The "customer pairs" with the same comparison result are placed in the corresponding positions of their offspring solutions. Similarly, the remaining blank node of the offspring solution is filled by different customer nodes in the other parent solution.

The pseudocode for creating the temporary route and reference set:

C. DQN-BASED ADAPTIVE SEARCH

This subsection attempts to find a way to overcome the shortcomings of the algorithm's blind search by learning

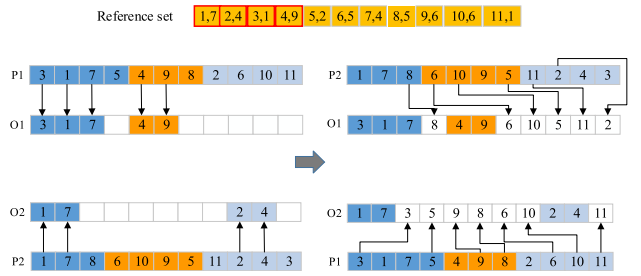

FIGURE 9. The operation process of PCPX.

local search actions and states through reinforcement learning methods. DQN is a deep reinforcement learning method that uses a function approximator to approximate Q functions. Compared to Q-learning, which uses a Q-table to record state-action pairs and then computes the Q-value for all state-action pairs, DQN is more feasible for solving problems with large combinatorial spaces.

1) STATE SPACE DESCRIPTION

DQN training of agents involves multiple episodes; every episode has multiple time steps. In an episode, the agent takes an action at each time step. At each time step t, the state of the VRP environment is described by a tuple $s_t = \{s^l, s^r, s^c\}$. The specific representations and definitions are as follows:

$$s^l = \{n_i^c, n_{pj}^c, n_{bj}^c, n_k^c; \forall p \in C', b \in C, k \in K\} \tag{15}$$

where n_i^c denotes the coordinates of the depot, n_{pi}^c denotes the coordinates of the predecessor node of the served customer node j, and n_{bi}^c denotes the coordinates of the successor node of the served customer node j. n_k^c denotes the coordinates of the first customer served by the vehicle k in the current distribution route. The description provides information about the current route delivery sequence and the location of each customer.

$$s^r = \{\varpi_j, b_j, o_j; j \in C\} \tag{16}$$

where $\varpi_j = (ll_j - ee_j) - t_{ij}$, the smaller value of ϖ_j indicates that the service demand of customer j is more urgent, and the measure of urgency can prioritize the customers with a more urgent demand for delivery. $b_j = ll_j - a_j^{kr}$, where b_j represents the difference between the latest delivery time specified by customer j and the actual delivery time of vehicle k. A larger b_j implies a higher feasibility of adjusting customer j to other locations. $o_j = a_j^{kr} - t_{ij}^{kr} + st_j$, where o_j denotes the time required for vehicle k to complete the distribution task of customer j, which is the sum of the transportation time from customer i to customer j and the unloading time of vehicle k. s^r describes the time-related information about customers.

$$s^c = \{\delta_k, \chi_k, \mu_k; \forall k \in K\} \tag{17}$$

δ_k represents the time overhead for vehicle k to complete all tasks of the current distribution route. μ_k represents the remaining available time. $\mu_k = (depot_{end} - depot_{start}) - \delta_k > 0$, Where $depot_{start}$ is the start time of the distribution activity

Input: Pareto solutions (PS)
Find the longest delivery route R from pareto solutions and set the temporary route's length to $|R|$
Initialize two zero matrix of size $|C|\cdot|R| \rightarrow \boldsymbol{M}, \boldsymbol{N}$
1:**For** $i = 1$ **to** K **do:**
2: **For** $j = 1$ **to** $|$PS$|$ **do:** #$|$ PS$|$**:** the number of pareto solutions
3: for each delivery route, the following is carried out.
4: **Switch** (PS[j].route[k])**: # route[k]: the kth customer**
5: **case** C_1**:** M [1][K] $+=1$;
6: **case** C_2**:** M [2][K] $+=1$;
7: **case** C_3**:** M [3][K] $+=1$;
...
8: **case** C_n**:** M[n][K] $+=1$; #n denotes the number of customers
9: **end** switch
10:**end** for
11:**end** for
12:output the matrix's greatest element's index for each column. (Form the temporary route)

1:**For** $i = 1$ to $|$PS$|$ **do:**
2: **For** $j = 1$ to $|C|$ do:
3: **If** index(C_j)+1 > index(C_j) **and** index(C_j)+1 <= **len**($curent\_route \epsilon PS_i$) **do:**
4: N[j][index(C_j)+1] = True
5: **Else**
6: Continue
7: **End** if
8: **End** for
9:**End** for
10:Output the changed matrix
11:Selecting the first customer with True value from each row to construct reference set.

of the depot, $depot_{end}$ is the end time of the distribution activity. Obviously, $(depot_{end} - depot_{start})$ is the time allowed for the distribution activity of the depot.

χ_k denotes the cumulative time violation of vehicle k in the distribution process, which is calculated as follow:

$$\chi_k = \sum_{j \in J^k} \max(a_j^{kr} - ll_j, 0) \tag{18}$$

where J^k denotes the set of delivery tasks for vehicle k. if $a_j^{kr} < ll_j$, the time violation of vehicle k is 0. s^r and s^c represent the time-related information about customers and the depot, which can help agents make decisions.

Based on the tuple's description, the agent is able to have a more complete understanding of the information related to the current distribution route, and it can utilize the time-related information to perform the actions guided by the heuristics.

2) ACTION DEFINITION

The combinatorial nature of this VRP means that a large number of different actions can be taken to construct and improve the order of the distribution routes. However, it is impractical to enumerate all possible actions during the training of the agent. Combining the algorithm training requirements and practical feasibility, this paper abstracts the actions into five heuristics, namely inter 2-opt, inter or-opt, external exchange, external 2-opt, external move. At each time step, the agent explores the development of a new routing scheme using the ε-greedy strategy. At the beginning of training, when ε is set to 1, the agent adopts a completely random action pattern, and then as the training iterates, the agent gradually increases the probability of utilizing the learned actions. The value of ε gradually decreases according to a certain decay rate: $\varepsilon_{t+1} = \varepsilon_t(1 - decay\_rate)$. The agent selects one of the actions to change the existing routing sequence, and the feasibility of the action execution has to satisfy the constraints 1∼3. The operation contents and procedures for the five types of actions are shown below:

a: INTER 2-OPT

in-route exchange. Select the route with the maximum remaining available time (μ_k). Select the customer with the largest b_j in this route and perform the 2-opt operation on customer j and the customer after it, checking the routing cost and customer satisfaction after the exchange. If the routing cost is less than the original cost and customer satisfaction is greater than or equal to the original value, then this transformation is taken; otherwise, discard this move. As shown in Fig. 10 (a), the reference state for executing this action is the value of μ_k; a larger μ_k means that this route has a higher adjustment value, and the route with the largest μ_k is selected for this action.

b: INTER OR-OPT

in-route exchange. Select the route with the maximum remaining available time (μ_k). As shown in Fig. 10(b), select m (0< m < the number of customers of this route) neighboring customers in this route. This operation is an extension of Inter 2-opt for routes with a large number of nodes. The new route is obtained by inserting neighboring customers from the original position into different positions. Calculating whether the customer satisfaction is better than the original customer satisfaction. If no move yields better customer satisfaction, maintain the original route. The reference state for executing this action is the largest remaining available time (μ_k) and the smallest ϖ_j.

c: EXTERNAL EXCHANGE

inter-route exchange. exchange the locations of customers on two different distribution routes. Select the customer with the largest b_j in each of the two routes and exchange their positions. Perform Inter 2-opt after exchange. The exchange behavior is not considering feasibility. This is done in order to help the search escape local optima. As shown in Fig. 10 (c), customers from two different routes are exchanged to obtain two new routes. Selected customers are required to

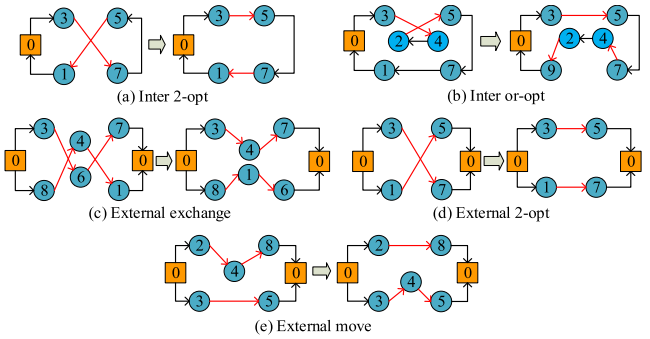

FIGURE 10. Local search operations.

meet the maximum service time windows remaining (b_j) on their respective routes.

d: EXTERNAL 2-OPT

inter-route exchange. As shown in Fig. 10 (d), Similar to Inter 2-opt, but aims to adjust two edges between different routes. Select two different paths that have the maximum value of χ_k and the minimum value of χ_k, and then perform this action. For the exchanged routes, perform inter 2-opt. If the exchanged routes have a lower routing cost, accept this exchange. Otherwise, keep the original route.

e: EXTERNAL MOVE

inter-route move. As shown in Fig. 10 (e), select the first route with the smallest μ_k, and select the customer with the largest o_j. Select the second route with the largest μ_k. Move customer j from the first route to the end of the second route. For the changed second route, perform Inter 2-opt on the customer j forward. If this exchange obtains a lower routing cost, keep it. Otherwise, keep the original route.

D. REWARD FUNCTION

Given a state s_t, the agent selects an action a_t and performs it, and subsequently, the agent receives a reward r_t for performing the action a_t. Combined with the multi-objective optimization problem studied in this paper, multiple objectives are considered simultaneously with the same priority. A new reward calculation method is used, which takes into account the relative changes between the optimization objectives of the parent and child generations. The specific calculation method is as follows:

$$r = \sum_{i=1}^{n} \frac{p_i - o_i}{p_i} \tag{19}$$

where p_i and o_i denote the ith objective function value for the original and new solutions, respectively. The larger the value of r, the more effective the selected action is.

E. COMPUTATION COMPLEXITY ANALYSIS

In this section, the time complexity of the proposed algorithm in this paper is analyzed. From the structure of the algorithm, it becomes evident that the computational burden predominantly arises from the search mechanism employed in the latter stage. Consequently, a focused analysis of the time complexity of the algorithm's second stage is deemed adequate.

To analyze the time complexity of this algorithm, it is essential to first determine the dimension of the elements within the state tuple $s_t = \{s^l, s^r, s^c\}$. Where s^l includes the following:

1) The coordinates of the depot.

2) The coordinates of the predecessor node of the served customer node j.

3) The coordinates of the successor node of the served customer node j.

4) The coordinates of the first customer served by the vehicle k in the current distribution route.

Thus, the dimension of s^l is $2 \times (|C| + |C| + |K|)$. Given that the quantity of warehouses is singular, the coordinate dimension of the warehouse is deemed invariant and, consequently, is not considered in the analysis. The multiplication of the preceding equation by a factor of 2 is necessitated by the presence of both horizontal and vertical coordinates for each node along the distribution route.

Where s^r includes the following:

1) Slack time of each customer's request.

2) The gap between the latest delivery time and the actual delivery time.

3) The time required for vehicle k to complete the distribution task of customer j.

Thus, the dimension of s^r is $|C| + |C| + |C|$.

Where s^c includes the following:

1) The time overhead for vehicle k to complete all tasks of the current distribution route.

2) Remaining available time of a distribution route.

3) The total delivery time violation of requests assigned to vehicle k.

Thus, the dimension of s^c is $|C| + |C| + |C|$.

Inter 2-opt contains three steps. For step 1, the remaining time of all dispatching routes needs to be sorted, so the complexity is $O(|K| \log |K|)$. For step 2, all customers on the selected route need to be sorted, so the maximum complexity of step 2 is $O(|C| \log |C|)$. For step 3, the maximumm complexity of 2-opt is $O((|C| - 1)^2)$.

By the same token, it can be inferred that the maximum complexity of the remaining four heuristic actions is uniformly $O((|C| - 1)^2)$.

For the DQN training, the complexity depends on the number of parameters to be trained. Since the state space has a dimension of $10\,|C| + 2\,|K|$. Thus, the first layer has $(10\,|C| + 2\,|K|) \times e$ parameters to be trained (e denotes the number of neurons in each hidden layer). The subsequent hidden layer has up to e^2 parameters. Given that the network yields a maximum of five actions, the number of parameters present in the output layer is $5e$. Further recognizing that $|C| > |K|$, the complexity of a time step of DQN training, is $O(|C| \times e + (l-2) \times e^2 + (|C| - 1)^2)$, which l denotes total layers of DQN. In this paper, the quantity of customers

($|C|$) surpasses the number of vehicles ($|K|$). Given that DQN training has n episodes, each with up to T time steps, the overall complexity of the DQN training can be approximated as $O(n \cdot T \cdot |C|^2)$.

F. IMPLEMENTATION OF DQN-BASED ADAPTIVE SEARCH

DQN is an off-policy reinforcement learning method that combines the advantages of DNN and Q-learning. DQN is trained utilizing multiple episodes, just like DNN. Actions described in 3.3.2 are used in each episode to improve solutions. The optimal policy (π_θ) is finally obtained through the continuous improvement of the weighting parameters (θ) during the learning process.

In order to improve the sampling efficiency and stability of the algorithm, DQN introduces an empirical playback mechanism and samples in a stochastic, equiprobable manner. DQN consists of three main components, including the objective prediction network, the target network, and the experience playback mechanism.

In reinforcement learning, sample data are often correlated and non-static among each other, which can lead to difficult model convergence and continuous fluctuation of loss values if the correlated data are directly used for model training. Therefore, in the initial stage of the algorithm, DNN models are not trained until the experience pool has E (where E represents the size of the experience pool) experiences. Agent stores the experience samples ($e_t = (s_t, a_t, r_t, s_{t+1})$) obtained from interacting with the environment at each step into the experience pool, and after executing several steps, a small batch of samples is randomly drawn from the experience pool and fed into the neural network as discrete data. DQN uses two network models containing DNN for learning, including the prediction network $Q(s, a, \theta)$ and the target network $Q(s, a, \theta')$. Therefore, the loss function under the dual network architecture is shown below:

$$L(\theta) = E_{\pi_\theta}[(r + \gamma \max_{a'} Q(s', a', \theta') - Q(s, a, \theta))^2] \quad (20)$$

Then calculating the semi-gradient of parameter θ:

$$\nabla_\theta L(\theta) = E_{\pi_\theta}[(r + \gamma \max_{a'} Q(s', a', \theta') - Q(s, a, \theta)) \nabla Q(s, a, \theta)] \quad (21)$$

Update the weighting parameter θ using MBSGD:

$$\theta = \theta - \alpha \nabla_\theta L(\theta) \quad (22)$$

The pseudocode of DQN-based adaptive search:

1. Initialize size of (experience pool) $= E$; Episodes $= M$; Frequency of target network updates: C
2. Initialize the minibatch size: n
3. Initialize the experience pool empty
4. Randomly initialize the weight parameters θ
5. Initialize the target network's parameter $\theta' = \theta$
6. for $i = 1$ to M do:
7. initialize state s_1
8. for $t = 1$ to T do: # an episode has T time steps
9. select a random action a_t with $\varepsilon - greedy$ strategy otherwise select an action $a_t = argmax_a Q(s_t, a : \theta)$
10. execute action a_t and get reward r_t; $s_t \rightarrow s_{t+1}$
11. store $e_t = (s_t, a_t, r_t, s_{t+1})$ in the experience pool
12. if $E > n$ do:
13. if $NR_t > \psi$ do: #NR_t denotes the total negative reward in an episode
14. randomly select a minibatch from experience pool
15. for e_j in the selected minibatch do:
16. calculate $r_j + \gamma \underset{a'}{max} Q(s_{j+1}, a' : \theta')$
17. end for
18. calculate loss by equation
19. update weighting parameter θ
20. update target network parameter $\theta' = \theta$
21. else:
22. break
23. end if
24. end if
25. end for
26. end for

IV. EXPERIMENTAL AND COMPARATIVE ANALYSIS

In this section, we use the proposed algorithm to solve the generated instances based on Solomon benchmark instances and conduct comparative experiments based on the idea of the control variable method to verify the effectiveness of the initialization strategy and crossover strategy. Finally, the proposed algorithm is compared with four other multi-objective optimization algorithms. All algorithms and tests are performed on a computer with an Intel Core i7-12700KF CPU @ 3.5GHz and 64GB RAM, and NVIDIA RTX GPU.

TABLE 2. Basic information about instances.

Instances	Customer number	Vehicle capacity	Service time
C1	25/50/100	200	900
C2	25/50/100	700	900
R1	25/50/100	200	100
R2	25/50/100	1000	100
RC1	25/50/100	200	100
RC2	25/50/100	1000	100

A. DATASETS DESCRIPTION AND PARAMETER SETTING

The generated instances are used to validate the effectiveness of the proposed algorithm. Based on the type of distribution, customers can be categorized into three types: C (clustering customers), R (random distribution of customers), and RC (the combination of clustering and random distribution). There are three types of customer numbers, and the number of customers, vehicle capacity, and service time are shown in Table 2. The working day T is divided into five time zones, and the road inclination between customer i and customer j is generated by a uniform distribution U [15, 0]. Table 3 shows the speed information for five time zones.

According to Tanvir Ahamed et al. [41], the hyperparameter values of DQN are selected by an informal search in this paper. It should be noted that these values are obtained in

TABLE 3. Speed information about each time zones.

Time zones	z_1	z_2	z_3	z_4	z_5
Period	[0, 0.2T]	[0.2T, 0.3T]	[0.3T, 0.7T]	[0.7T, 0.8T]	[0.8T, T]
speed	1.17	0.67	1.33	0.83	1

TABLE 4. Hyperparameter values.

hyperparameter	value
Experience pool size: E	10000
Minibatch size: n	200
Target network update frequency: C	1000(25), 2500(50, 100)
Discount rate: γ	0.95
Learning rate: α	0.0002
Decay rate: ε	0.001
Episode termination threshold: ψ	-15

our research context and may not be fully applicable to other problems. The hyperparameter settings are shown in Table 4.

B. INDICATORS FOR PERFORMANCE EVALUATION

In this section, three assessment indicators—hypervolume, inverted generational distance, and proposed indicator—RPD are used to assess the algorithm's performance. Since it is the first time to solve this multi-objective optimization problem, there is no real PF (pareto front) available in practice. Therefore, in order to achieve the approximative PF, we begin with repeated trials utilizing the existing multi-objective algorithm [27]. The details of the three evaluation indicators are as follows:

The evaluation indicators used in this paper are shown below:

Hypervolume (HV): an indicator to evaluate the algorithm's performance. Given a reference point, the convergence and diversity of the solutions will be measured by the reference point and PS (pareto set) obtained by algorithm [42]. The higher the hypervolume values are, the better the solution.

$$HV = \delta\left(\cup_{i=1}^{|S|} v_i\right) \tag{23}$$

where δ denotes the Lebesgue measure, which is used to measure the volume. $|S|$ denotes the number of non-dominated solution sets. v_i denotes the hypervolume formed by the reference point and the ith solution in the solution set.

Inverted generational distance (IGD): an indicator to evaluate the distance between the approximate PF and PF obtained by the algorithm. The lower the IGD, the better the solution.

$$IGD(P, P^*) = \frac{\sum_{x \in P^*} \min_{y \in P} dis(x, y)}{|P^*|} \tag{24}$$

where P is the solution obtained by the algorithm and $P*$ is a set of uniformly distributed reference points. dis(x, y) denotes the Euclidean distance between the reference point x and the point y in P.

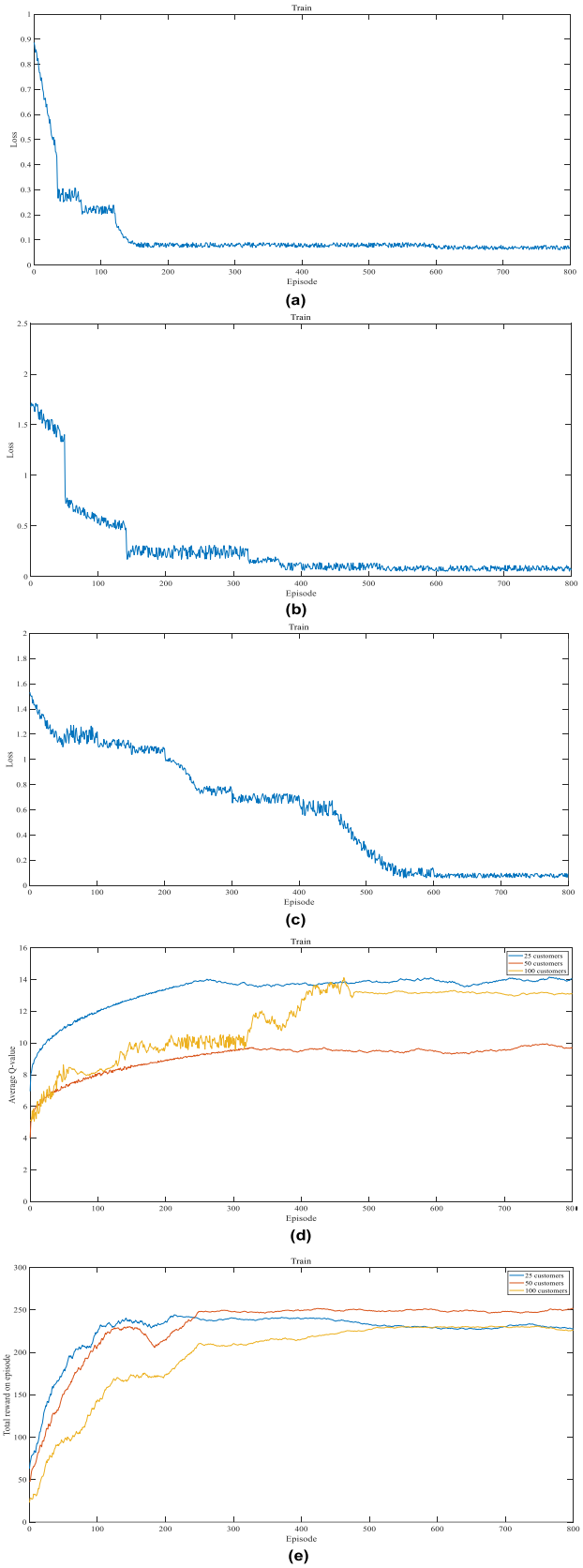

FIGURE 11. (a) The loss on each episode(25 customers). (b) The loss on each episode(50 customers). (c) The loss on each episode(100 customers). (d) The average Q value on episode. (e) The total reward on episode.

TABLE 5. Comparison of partial computational results.

Instances	HV		IGD	
	DQMOEA-X	DQMOEA-NX	DQMOEA-X	DQMOEA-NX
C101_25	7.58E-01	**7.94E-01**	1.31E-02	**1.17E-02**
C102_25	**8.33E-01**	7.84E-01	**3.31E-02**	3.99E-02
C103_25	7.22E-01	**7.25E-01**	1.74E-02	**1.71E-02**
C104_25	7.51E-01	**7.62E-01**	7.59E-03	**7.07E-03**
C105_25	**7.02E-01**	6.81E-01	**8.31E-03**	2.67E-02
C106_25	**7.72E-01**	7.70E-01	**8.15E-03**	1.12E-02
C107_25	**7.48E-01**	7.24E-01	**8.12E-03**	1.68E-02
C108_25	**8.02E-01**	7.58E-01	**8.11E-03**	2.39E-02
C109_25	**7.37E-01**	7.13E-01	**6.11E-03**	1.21E-02
C101_50	**6.89E-01**	6.51E-01	**5.37E-03**	2.67E-02
C102_50	**7.76E-01**	**7.83E-01**	6.07E-03	**5.97E-03**
C103_50	**7.42E-01**	6.97E-01	**5.70E-03**	2.43E-02
C104_50	6.92E-01	**7.15E-01**	**1.69E-02**	2.91E-02
C105_50	**7.43E-01**	7.14E-01	**1.42E-02**	5.09E-02
C106_50	**6.88E-01**	6.73E-01	**1.93E-02**	3.17E-02
C107_50	**7.75E-01**	7.55E-01	**1.00E-02**	4.39E-02
C108_50	**7.79E-01**	7.47E-01	**1.32E-02**	2.72E-02
C109_50	**8.09E-01**	7.83E-01	**1.13E-02**	2.69E-02
C101_100	**7.04E-01**	6.96E-01	**6.73E-03**	2.21E-02
C102_100	**6.77E-01**	6.63E-01	**1.51E-02**	4.33E-02
C103_100	**8.07E-01**	6.76E-01	**1.33E-02**	2.74E-02
C104_100	7.57E-01	**8.27E-01**	1.39E-02	**1.21E-02**
C105_100	6.14E-01	**7.01E-01**	1.54E-02	**1.14E-02**
C106_100	**7.41E-01**	6.47E-01	**7.73E-03**	1.12E-02
C107_100	**7.30E-01**	6.61E-01	**7.17E-03**	1.13E-02
C108_100	6.94E-01	**7.19E-01**	8.07E-03	**7.93E-03**
C109_100	**8.31E-01**	7.39E-01	**1.37E-02**	6.02E-02

TABLE 6. Overall comparison of computational results.

Instances	HV		IGD	
	DQMOEA-X	DQMOEA-NX	DQMOEA-X	DQMOEA-NX
C1_25	6	3	6	3
C2_25	2	6	3	5
R1_25	10	2	10	2
R2_25	6	5	5	6
RC1_25	7	1	7	1
RC2_25	3	5	1	7
C1_50	7	2	8	1
C2_50	4	4	3	5
R1_50	10	2	9	3
R2_50	7	4	6	5
RC1_50	5	3	6	2
RC2_50	2	6	5	3
C1_100	6	3	6	3
C2_100	5	3	7	1
R1_100	6	6	8	4
R2_100	5	6	6	5
RC1_100	6	2	7	1
RC2_100	6	2	6	2
Total	103	65	109	59

Additionally, in order to compare the performance between various algorithms, the relative percentage difference (RPD) is used to analyze the compared algorithms in the same instance. The RPD value is calculated as follows:

$$RPD = \begin{cases} \dfrac{D_b - D_c}{D_b}, & HV - basedRPD \\ \dfrac{D_c - D_b}{D_b}, & IGD - basedRPD \end{cases} \tag{25}$$

where D_c represents the HV or IGD value acquired from the comparing algorithms, D_b represents the best HV or IGD value. The lower the RPD, the better the current algorithm.

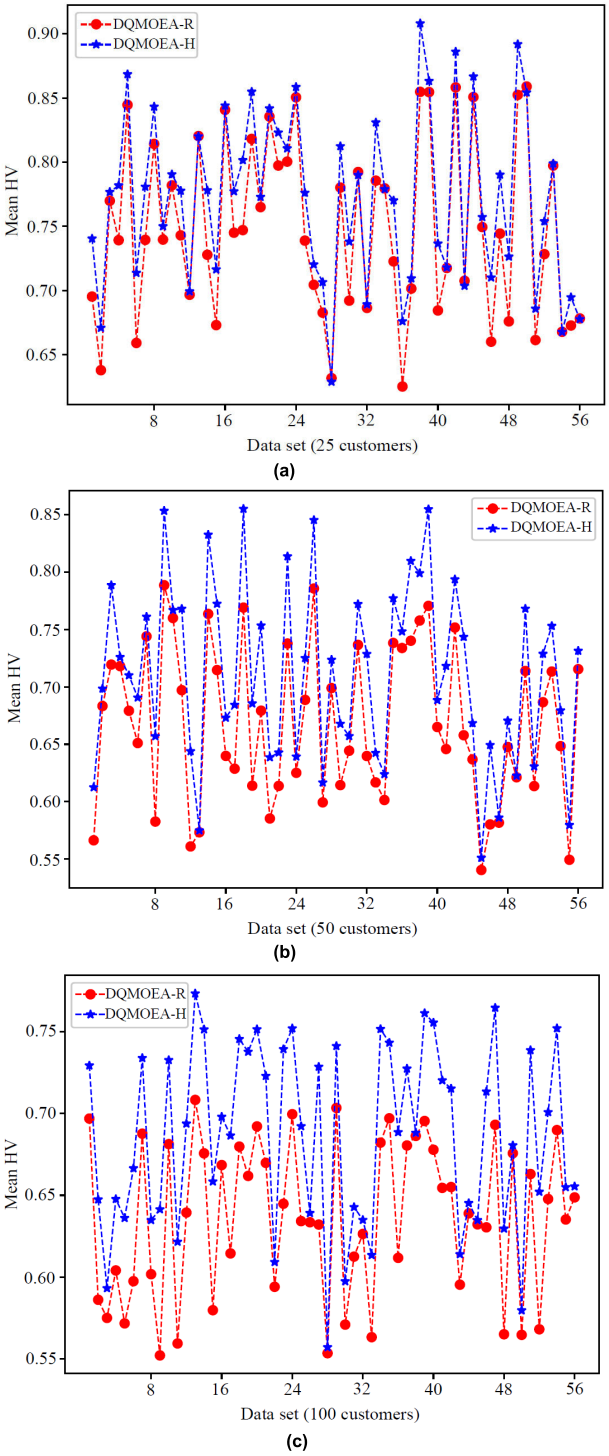

FIGURE 12. (a) The mean HV value of solutions (25 customers). (b) The mean HV value of solutions (50 customers). (c) The mean HV value of solutions (100 customers).

C. TRAINING RESULTS AND EVALUATION OF ALGORITHM'S EFFECTIVENESS

1) TRAINING RESULTS

In this section, we train the DQN model using instances with 25, 50, and 100 customers, respectively, and obtain the following training result plots:

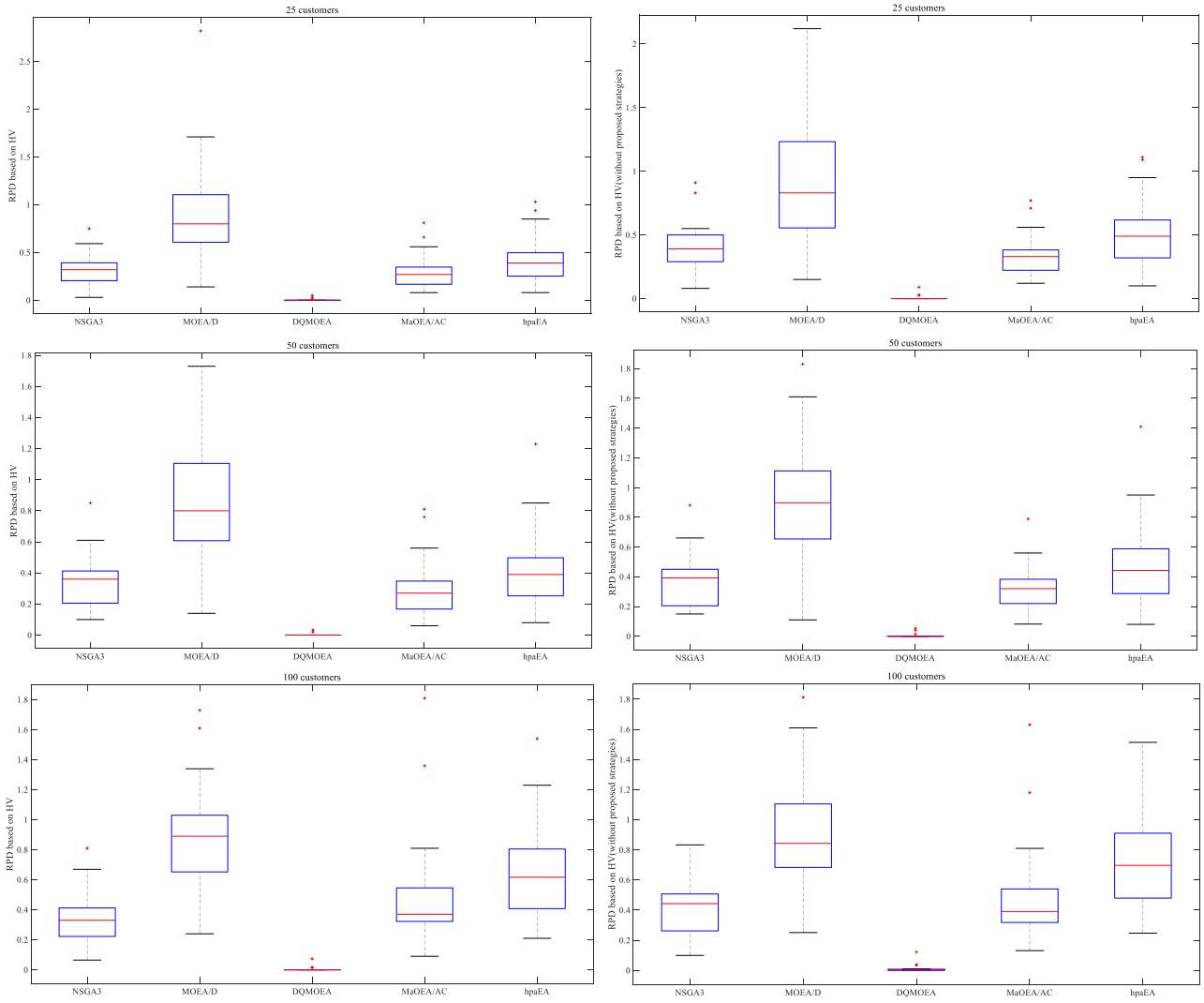

FIGURE 13. The comparisons of two groups experiments

Fig. 11(a) to Fig. 11(c) show the training loss of datasets with 25, 50, and 100 customers, respectively. Fig. 11(d) plots the average Q value on each episode for datasets with 25, 50, and 100 customers. Fig. 11(e) plots the total reward on each episode for datasets with 25, 50, and 100 customers. On each episode, termination occurs when the accumulated negative reward is less than the episode termination threshold. Overall, the training loss for all datasets will no longer improve significantly after 600 episodes.

As shown in Fig. 11(a) to Fig. 11(c), in the early stage training of datasets with 25, 50, and 100 customers, the training loss jumps down about every 1000 time steps and 2500 time steps, which corresponds to the parameter θ' update frequency of the target network, respectively. As a result of the gradual acquisition of experience by the agent in the early state, DQN training has acute jumps of loss for datasets with 25 and 50 customers before 200 episodes. However, on the dataset with 100 customers, this phenomenon of drop in loss values occurs up to about episode 550. Fig. 11(d) shows the average Q value on each episode for datasets with 25 and 50 customers steadily increasing and stabilizing after 300 episodes. For datasets with 100 customers, it is more sensitive to the updating of θ', the average Q-value is changing about every 50 episodes (2500 time steps), and this change continues until about episode 500. The magnitude of change decreases over time steps, suggesting that the marginal improvement of DQN is diminishing as training continues. Fig. 11(e) illustrates the gradual stabilization of the total reward value of each episode after several steps of training. During post-training, the actions executed by the agent have little effect on the value of the objective function to be optimized.

2) EFFECTIVENESS OF PROPOSED STRATEGIES

In order to verify the effectiveness of the hybrid initialization strategy, two types of DQMOEA are used for comparison in this section: DQMOEA-R: DQMOEA using only

TABLE 7. Computational results of algorithms (25 customers).

Instance	HV					IGD				
	NSGA3	MOEA/D	DQMOEA	MaOEA/AC	hpaEA	NSGA3	MOEA/D	DQMOEA	MaOEA/AC	hpaEA
C101_25	**0.8013**	0.6855	0.7921	0.7345	0.733	3.74E-02	1.67E-01	**4.23E-03**	4.43E-02	9.03E-02
C102_25	0.7392	0.72	**0.7754**	0.7581	0.6977	4.01E-02	8.56E-02	**3.34E-02**	5.40E-02	6.81E-02
C103_25	0.7051	0.6343	**0.7192**	0.6845	0.6573	3.36E-02	7.40E-02	**1.47E-02**	3.87E-02	4.89E-02
C104_25	0.6219	**0.6472**	0.6267	0.6459	0.6291	5.08E-02	4.82E-02	**1.03E-02**	1.36E-02	2.96E-02
C105_25	0.8167	0.6561	**0.8173**	0.7936	0.7642	6.25E-02	1.10E-01	5.99E-02	6.94E-02	**5.90E-02**
C106_25	0.7579	0.6832	**0.7782**	0.7482	0.7528	5.11E-02	1.06E-01	**4.30E-02**	4.63E-02	6.15E-02
C107_25	0.8313	0.6239	**0.8439**	0.8102	0.7408	4.05E-02	1.27E-01	**4.02E-02**	8.13E-02	6.57E-02
C108_25	0.7438	0.589	**0.7655**	0.7243	0.6859	8.46E-02	1.19E-01	**3.30E-02**	5.03E-02	6.74E-02
C109_25	0.7322	0.5857	0.7588	0.724	**0.7624**	4.84E-02	1.00E-01	**3.25E-02**	3.95E-02	8.20E-02
C201_25	0.7867	0.5653	**0.8765**	0.7471	0.7283	7.47E-02	2.46E-01	**4.05E-03**	4.85E-02	1.03E-01
C202_25	0.806	0.7101	**0.8471**	0.7687	0.8123	6.15E-02	1.40E-01	**2.50E-02**	7.05E-02	7.63E-02
C203_25	0.7751	0.6864	**0.8346**	0.7908	0.7649	**2.71E-02**	1.14E-01	4.66E-02	3.59E-02	8.09E-02
C204_25	0.6932	0.6004	**0.7144**	0.6994	0.7055	5.89E-02	1.17E-01	**1.80E-02**	3.34E-02	6.23E-02
C205_25	0.8274	0.6544	0.8091	**0.8705**	0.7542	3.84E-02	1.55E-01	**3.32E-02**	9.64E-02	1.20E-01
C206_25	0.823	0.6193	**0.87**	0.8202	0.771	5.24E-02	1.49E-01	**3.68E-02**	6.76E-02	1.22E-01
C207_25	**0.8705**	0.7456	0.8187	0.804	0.826	9.22E-02	1.30E-01	**1.28E-02**	8.53E-02	8.47E-02
C208_25	0.7971	0.649	**0.8763**	0.8218	0.8045	7.16E-02	1.58E-01	**3.13E-02**	8.05E-02	1.14E-01
R101_25	**0.8441**	0.6218	0.815	0.7795	0.7475	4.92E-02	1.16E-01	**2.46E-02**	5.64E-02	8.42E-02
R102_25	0.7868	0.7074	**0.8051**	0.7524	0.7269	6.30E-02	1.30E-01	**2.68E-02**	6.16E-02	8.63E-02
R103_25	0.754	0.6757	**0.7813**	0.7469	0.7846	6.12E-02	1.07E-01	**2.62E-02**	6.26E-02	6.65E-02
R104_25	0.6795	0.6708	**0.7496**	0.6797	0.7308	6.86E-02	9.78E-02	**2.93E-02**	5.03E-02	9.74E-02
R105_25	0.7884	0.6481	**0.8406**	0.8092	0.8076	5.95E-02	1.29E-01	**2.25E-02**	5.12E-02	7.31E-02
R106_25	0.759	0.6942	**0.8416**	0.7302	0.7678	6.06E-02	1.33E-01	**2.18E-02**	8.16E-02	7.86E-02
R107_25	0.5607	0.5019	**0.6986**	0.5889	0.6207	6.86E-02	2.18E-01	**1.87E-02**	8.69E-02	8.20E-02
R108_25	0.6636	0.6224	**0.7305**	0.7049	0.7064	6.15E-02	9.72E-02	**1.43E-02**	7.17E-02	5.16E-02
R109_25	0.759	0.6709	**0.8099**	0.7516	0.7284	4.95E-02	1.08E-01	**1.88E-02**	6.01E-02	6.39E-02
R110_25	0.7339	0.6855	**0.796**	0.7583	0.7494	5.59E-02	1.28E-01	**2.12E-02**	4.04E-02	1.02E-01
R111_25	0.7	0.6273	**0.8096**	0.7191	0.7516	6.35E-02	1.11E-01	**2.47E-02**	5.33E-02	7.19E-02
R112_25	0.6941	0.7189	**0.7937**	0.7099	0.7394	5.93E-02	8.94E-02	**1.68E-02**	4.62E-02	6.39E-02
R201_25	0.746	0.5529	**0.8371**	0.719	0.7705	**2.74E-02**	2.11E-01	3.33E-02	4.44E-02	4.20E-02
R202_25	0.8319	0.6473	**0.8319**	0.7889	0.8012	5.65E-02	1.08E-01	**5.43E-02**	5.51E-02	6.97E-02
R203_25	0.7727	0.7117	**0.8022**	0.7782	0.7622	**5.14E-02**	9.04E-02	5.28E-02	5.66E-02	7.83E-02
R204_25	0.7159	0.6489	**0.7313**	0.7156	0.7054	3.49E-02	8.38E-02	**1.21E-02**	2.39E-02	4.14E-02
R205_25	0.8075	0.6473	**0.8754**	0.8245	0.7956	4.29E-02	1.20E-01	**4.13E-02**	6.54E-02	1.20E-01
R206_25	0.7869	0.6403	0.809	0.7917	**0.8328**	3.58E-02	1.17E-01	**1.90E-02**	4.73E-02	8.11E-02
R207_25	0.7447	0.6329	**0.7859**	0.7308	0.7214	4.41E-02	1.05E-01	**1.74E-02**	4.95E-02	7.01E-02
R208_25	0.693	0.6189	**0.7512**	0.7033	0.7261	5.73E-02	9.88E-02	**5.44E-03**	3.57E-02	5.60E-02
R209_25	0.7595	0.6325	**0.8317**	0.7295	0.8111	5.04E-02	1.57E-01	**2.17E-02**	5.73E-02	1.00E-01
R210_25	0.7496	0.654	0.7946	0.7212	**0.8072**	5.36E-02	1.10E-01	**3.40E-02**	6.62E-02	8.18E-02
R211_25	0.7885	0.7368	**0.8266**	0.7664	0.8202	8.87E-02	9.83E-02	**2.83E-02**	5.85E-02	1.21E-01
RC101_25	0.8017	0.7345	**0.8658**	0.8478	0.8319	6.58E-02	1.14E-01	**1.39E-02**	5.69E-02	8.19E-02
RC102_25	0.6629	0.6016	**0.81**	0.7033	0.6643	7.33E-02	2.07E-01	**1.40E-02**	6.77E-02	1.06E-01
RC103_25	0.6638	0.703	**0.818**	0.758	0.7598	8.26E-02	1.25E-01	**2.25E-02**	5.59E-02	9.04E-02
RC104_25	0.7203	0.5631	**0.7597**	0.6933	0.733	4.32E-02	1.32E-01	**1.41E-02**	3.25E-02	7.21E-02

TABLE 7. *(Continued.)* **Computational results of algorithms (25 customers).**

RC105_25	0.8461	0.6616	**0.8462**	0.8064	0.8224	5.79E-02	1.31E-01	**2.36E-02**	4.52E-02	8.85E-02
RC106_25	0.7205	0.6824	**0.8103**	0.7208	0.7647	5.63E-02	1.73E-01	**1.68E-02**	5.94E-02	1.06E-01
RC107_25	0.7252	0.6461	**0.7704**	0.722	0.7395	4.70E-02	1.83E-01	**2.04E-02**	4.38E-02	1.09E-01
RC108_25	0.7283	0.6116	**0.7709**	0.7452	0.703	7.19E-02	1.21E-01	**9.73E-03**	4.96E-02	7.57E-02
RC201_25	**0.8054**	0.5429	0.7717	0.7612	0.691	6.13E-02	2.71E-01	**4.72E-03**	7.45E-02	1.32E-01
RC202_25	0.7656	0.6905	**0.8451**	0.757	0.8078	4.11E-02	1.22E-01	**2.22E-02**	5.49E-02	8.88E-02
RC203_25	0.7621	0.708	**0.8253**	0.7601	0.7671	**3.13E-02**	1.09E-01	3.31E-02	3.79E-02	9.23E-02
RC204_25	**0.7756**	0.6234	0.7639	0.7474	0.7307	**1.82E-02**	1.03E-01	2.13E-02	3.47E-02	8.81E-02
RC205_25	0.799	0.6471	**0.8694**	0.8	0.8268	6.58E-02	1.55E-01	**2.75E-02**	4.74E-02	9.59E-02
RC206_25	0.7726	0.6755	**0.8632**	0.7739	0.7404	9.43E-02	1.70E-01	**2.83E-02**	6.40E-02	9.19E-02
RC207_25	0.8085	0.6542	**0.8405**	0.7952	0.7908	3.41E-02	1.25E-01	**3.04E-02**	5.54E-02	8.70E-02
RC208_25	0.7998	0.6715	**0.8506**	0.7627	0.7339	6.15E-02	1.10E-01	**2.89E-02**	7.23E-02	8.10E-02
Average	0.757	0.651	0.803	0.752	0.752	0.055	0.129	0.025	0.055	0.082
Gap	0.056	0.189	0	0.063	0.063	-1.217	-4.201	0	-1.213	-2.299

the random initialization strategy; DQMOEA-H: DQMOEA using a hybrid generation strategy with four initialization methods. The two algorithms were used to solve all datasets five times, and the HV values of the nondominated solutions were averaged separately. Comparison results are obtained as shown in Fig. 12.

From Fig. 12(a) to Fig. 12(c), it can be seen that DQMOEA-H performs significantly better than DQMOEA-R. This indicates that the hybrid initialization strategy can generate high-quality and diverse initialization solutions. Moreover, as the number of customers increases, the difference between the hybrid strategy and the randomized strategy becomes more and more significant, and the superiority of DQMOEA using the hybrid initialization strategy becomes more apparent.

Two different forms of DQMOEA are designed to test the effectiveness of the two crossover strategies. (1) DQMOEA-X integrates both PSCX and PCPX crossover strategies. (2) DQMOEA-NX does not include PSCX as well as PCPX and uses only random crossover strategies. The solving ability of the two algorithms on different datasets is compared through HV values as well as IGD values.

As shown in Table 5 and Table 6, the combined performance of the two crossover strategies on the C1-type dataset is optimal and stable. The amount of QMOEA-X's dominance in HV values among the 27 C1-type instances solved by the two algorithms is 19, which is a 70% dominance. The corresponding IGD value is 74% at the same time. Out of the 168 datasets solved, DQMOEA-X achieved a favorable number of 103 on the HV values and an overall favorable rate of 61.3%. The number of advantages on the IGD values reached 109, and the overall favorable rate reached 64.8%. In addition, the overall trend of the two indicators in different instances is consistent, indicating that the proposed crossover strategy can improve the convergence of solutions as well as the diversity of solutions. The experimental results verify the effectiveness of the proposed crossover strategies.

3) PERFORMANCE EVALUATION OF PROPOSED ALGORITHM

In this section, four well-known MOEAs: NSGA3, MOEA/D, MaOEA/AC, and hpaEA are used to compare with the DQMOEA for further validation of the effectiveness of the DQMOEA. For the purpose of objective and fair comparison, we conducted two sets of experiments for comparative analysis. The first group of trials does not include the hybrid initialization and pareto-based crossover strategies; as a control, the second group of studies includes the hybrid initialization and pareto-based crossover strategies. Each algorithm was repeated 10 times independently, and the average values of HV as well as IGD were obtained. Using RPD values to compare and assess the performance of different algorithms, the results are shown as follows:

Fig. 13 shows the RPD values of five algorithms for each of the three instances. In general, DQMOEA has the optimal RPD values, which suggests that the solutions obtained by DQMOEA have better convergence and diversity. The overall comparison between the first and second columns indicates that the proposed strategies not only help DQMOEA achieve better solution convergence and diversity but also improve the performance of the other four control algorithms as well. However, the comparison of the algorithms still favors the DQMOEA, which indeed demonstrates the effectiveness of the DQN-based adaptive search and strategies suggested in this research. In a longitudinal comparison, for control algorithms, the median as well as the upper quartile in the box plot of RPD values are gradually increasing, which means that the quality of solutions will decrease as the size of the problem gets larger. However, the results obtained by DQMOEA only fluctuate slightly, implying that as the number of customers increases, the conflict between convergence and diversity of

TABLE 8. Comparison of the computational results of the five algorithms (100 customers).

Instance	HV					IGD				
	NSGA3	MOEA/D	DQMOEA	MaOEA/AC	hpaEA	NSGA3	MOEA/D	DQMOEA	MaOEA/AC	hpaEA
C101_100	0.7472	0.5746	**0.7943**	0.7083	0.6606	3.69E-02	1.66E-01	**8.91E-04**	4.42E-02	8.74E-02
C102_100	0.7024	0.6054	**0.7519**	0.6797	0.6713	3.97E-02	8.37E-02	**3.24E-02**	5.39E-02	6.60E-02
C103_100	0.6456	0.5655	**0.7006**	0.6487	0.6176	3.22E-02	7.33E-02	**1.29E-02**	3.69E-02	4.78E-02
C104_100	0.5837	0.5691	**0.6414**	0.6018	0.5944	4.96E-02	4.63E-02	**8.80E-03**	1.23E-02	2.84E-02
C105_100	0.7504	0.6163	**0.8252**	0.7405	0.7449	7.19E-02	1.08E-01	**5.70E-02**	6.91E-02	5.81E-02
C106_100	**0.8159**	0.6084	0.8143	0.7352	0.7113	**3.01E-02**	1.05E-01	4.25E-02	4.43E-02	6.08E-02
C107_100	**0.8114**	0.5937	0.8111	0.7308	0.6983	**4.59E-02**	1.26E-01	4.82E-02	8.05E-02	6.53E-02
C108_100	0.6695	0.5635	**0.7709**	0.6705	0.6754	8.30E-02	1.18E-01	**3.11E-02**	4.97E-02	6.72E-02
C109_100	0.7131	0.5728	**0.7587**	0.6927	0.6699	4.72E-02	9.97E-02	**3.12E-02**	3.91E-02	8.16E-02
C201_100	0.7287	0.4925	**0.829**	0.7086	0.6353	7.46E-02	2.44E-01	**2.90E-03**	4.74E-02	1.02E-01
C202_100	0.7687	0.641	**0.8375**	0.7543	0.7365	6.07E-02	1.40E-01	**2.12E-02**	6.90E-02	7.50E-02
C203_100	0.7364	0.5972	**0.7912**	0.7496	0.7134	3.51E-02	1.13E-01	4.37E-02	**2.52E-02**	8.04E-02
C204_100	0.6599	0.5426	**0.703**	0.6653	0.6703	5.89E-02	1.16E-01	**1.53E-02**	3.15E-02	5.95E-02
C205_100	0.8394	0.6363	**0.8863**	0.7538	0.7357	3.73E-02	1.55E-01	**3.05E-02**	9.57E-02	1.19E-01
C206_100	0.7763	0.6023	**0.8501**	0.7756	0.7408	5.05E-02	1.48E-01	**3.29E-02**	6.63E-02	1.22E-01
C207_100	0.8076	0.6934	**0.8609**	0.7797	0.7757	9.07E-02	1.28E-01	**1.04E-02**	8.51E-02	8.34E-02
C208_100	0.7742	0.6085	**0.8547**	0.7313	0.7263	7.05E-02	1.57E-01	**2.75E-02**	7.94E-02	1.12E-01
R101_100	**0.7859**	0.6064	0.7948	0.7177	0.6978	4.85E-02	1.15E-01	**2.12E-02**	5.47E-02	8.41E-02
R102_100	0.7149	0.6074	**0.8006**	0.7162	0.7009	6.18E-02	1.28E-01	**2.45E-02**	6.11E-02	8.54E-02
R103_100	0.7087	0.6232	**0.7941**	0.7295	0.715	5.98E-02	1.06E-01	**2.44E-02**	6.12E-02	6.39E-02
R104_100	0.6455	0.5808	**0.7199**	0.6723	0.6523	6.67E-02	9.67E-02	**2.77E-02**	4.84E-02	9.47E-02
R105_100	0.7444	0.642	**0.8404**	0.7738	0.7515	5.85E-02	1.28E-01	**2.09E-02**	5.06E-02	7.03E-02
R106_100	0.7093	0.5932	**0.8047**	0.7175	0.7064	6.02E-02	1.31E-01	**1.90E-02**	7.99E-02	7.57E-02
R107_100	0.503	0.4245	**0.6699**	0.5642	0.577	6.84E-02	2.16E-01	**1.49E-02**	8.50E-02	8.06E-02
R108_100	0.627	0.5738	**0.7366**	0.674	0.6637	6.01E-02	9.67E-02	**1.29E-02**	6.99E-02	5.15E-02
R109_100	0.6994	0.614	**0.7939**	0.7212	0.6982	4.81E-02	1.07E-01	**1.86E-02**	5.82E-02	6.11E-02
R110_100	0.6617	0.591	**0.7659**	0.7132	0.6737	5.40E-02	1.27E-01	**1.88E-02**	3.95E-02	1.01E-01
R111_100	0.6314	0.6118	**0.7692**	0.7013	0.6886	6.30E-02	1.11E-01	**2.26E-02**	5.29E-02	6.89E-02
R112_100	0.6652	0.6289	**0.7737**	0.7026	0.697	5.78E-02	8.80E-02	**1.36E-02**	4.53E-02	6.13E-02
R201_100	0.7237	0.5023	**0.813**	0.695	0.6927	3.74E-02	2.10E-01	**6.00E-03**	4.39E-02	4.00E-02
R202_100	0.7755	0.6306	**0.8259**	0.7604	0.7304	6.33E-02	1.07E-01	6.23E-02	**5.27E-02**	6.77E-02
R203_100	0.7202	0.6185	**0.7741**	0.752	0.7232	5.08E-02	8.85E-02	**4.91E-02**	5.53E-02	7.74E-02
R204_100	0.6486	0.6104	**0.7131**	0.6879	0.6726	5.29E-02	8.33E-02	**1.09E-02**	2.20E-02	4.06E-02
R205_100	0.7787	0.6159	**0.8613**	0.799	0.7606	6.23E-02	1.19E-01	**3.87E-02**	6.53E-02	1.19E-01
R206_100	0.7348	0.6045	**0.8043**	0.7779	0.7487	5.40E-02	1.16E-01	**1.50E-02**	4.63E-02	7.84E-02
R207_100	0.6688	0.5784	**0.756**	0.6914	0.6812	4.38E-02	1.03E-01	**1.49E-02**	4.92E-02	6.82E-02
R208_100	0.6576	0.5669	**0.719**	0.6684	0.6529	5.63E-02	9.77E-02	**5.00E-03**	3.39E-02	5.41E-02
R209_100	0.7246	0.5699	**0.7964**	0.7182	0.7359	5.88E-02	1.56E-01	**1.89E-02**	5.64E-02	9.77E-02
R210_100	0.7102	0.5935	**0.8026**	0.7108	0.7148	5.33E-02	1.08E-01	**3.16E-02**	6.56E-02	7.90E-02
R211_100	0.7417	0.6351	**0.823**	0.7558	0.7259	8.75E-02	9.67E-02	**2.63E-02**	5.76E-02	1.18E-01
RC101_100	0.7839	0.6786	**0.8918**	0.8239	0.7659	6.51E-02	1.13E-01	**1.22E-02**	5.66E-02	8.01E-02
RC102_100	0.611	0.4946	**0.7706**	0.6794	0.6349	7.32E-02	2.05E-01	**1.37E-02**	6.59E-02	1.05E-01
RC103_100	0.6435	0.6038	**0.7875**	0.728	0.6728	8.19E-02	1.23E-01	**2.17E-02**	5.49E-02	8.82E-02
RC104_100	0.6334	0.5523	**0.7239**	0.6843	0.6378	4.13E-02	1.31E-01	**1.03E-02**	3.20E-02	6.94E-02

TABLE 8. *(Continued.)* **Comparison of the computational results of the five algorithms (100 customers).**

RC105_100	0.763	0.6216	**0.8406**	0.7783	0.7276	5.67E-02	1.31E-01	**2.18E-02**	4.34E-02	8.81E-02
RC106_100	0.6737	0.5669	**0.791**	0.7137	0.6797	5.57E-02	1.72E-01	**1.56E-02**	5.79E-02	1.03E-01
RC107_100	0.6609	0.5423	**0.7608**	0.7123	0.641	4.50E-02	1.81E-01	**1.84E-02**	4.20E-02	1.08E-01
RC108_100	0.6317	0.5843	**0.7673**	0.7355	0.6858	7.09E-02	1.20E-01	**9.40E-03**	4.77E-02	7.28E-02
RC201_100	0.6888	0.4337	**0.7809**	0.6512	0.6017	6.09E-02	2.71E-01	**4.30E-03**	7.26E-02	1.30E-01
RC202_100	0.7234	0.6005	**0.8133**	0.7364	0.7209	3.95E-02	1.20E-01	**1.85E-02**	5.39E-02	8.79E-02
RC203_100	0.7149	0.6013	**0.796**	0.7266	0.6977	4.03E-02	1.09E-01	2.98E-02	**2.76E-02**	9.07E-02
RC204_100	**0.7911**	0.6024	0.7423	0.7049	0.6714	**1.64E-02**	1.03E-01	1.90E-02	3.40E-02	8.67E-02
RC205_100	0.7602	0.6139	**0.8508**	0.7765	0.7395	6.49E-02	1.54E-01	**2.47E-02**	4.69E-02	9.47E-02
RC206_100	0.7445	0.5793	**0.8485**	0.7474	0.7163	9.37E-02	1.69E-01	**2.80E-02**	6.35E-02	9.07E-02
RC207_100	0.7868	0.6156	**0.8349**	0.7628	0.7278	**3.26E-02**	1.23E-01	3.40E-02	5.36E-02	8.70E-02
RC208_100	0.7308	0.6236	**0.813**	0.7367	0.6963	5.97E-02	1.08E-01	**2.53E-02**	7.09E-02	7.81E-02
Average	0.712	0.589	0.790	0.718	0.693	0.056	0.128	0.022	0.053	0.080
Gap	0.099	0.254	0	0.090	0.122	-1.464	-4.648	0	-1.361	-2.544

note: the Gap value of each column is calculated as $(Average_{DQMOEA} - Average_{selected})/Average_{DQMOEA}$

the solution schemes becomes more severe, and the superiority of DQMOEA becomes more significant.

Tables 7 and 8 show the results of the five algorithms for a small-size instance with 25 customers and a large-size instance with 100 customers, respectively. Overall, the solutions obtained by DQMOEA have optimal convergence and distribution. When comparing the solution results of the two instances, it can be seen that the gap value increases as the size of the instance increases. The slight change in the gap value based on the IGD value, compared to the more significant change in the gap value based on the HV, shows that DQMOEA has better convergence in solving large instances.

V. CONCLUSION

In this paper, a multi-objective optimization problem of time-dependent green vehicle scheduling with time windows is investigated. and the multi-objective optimization mathematical model is established by considering vehicle transportation time, customer satisfaction, and carbon emissions as the optimization objectives. We propose a novel deep reinforcement learning-based two-stage optimization algorithm that consists of a hybrid initialization and DQN-based adaptive search. Four initialization methods are used to initialize solutions based on the distribution types of customers. Aiming at the blind searching of the multi-objective optimization algorithm in the process of running, two crossover strategies (PSCX and PCPX) based on pareto front are designed to learn the structure of the pareto optimal solution. The DQN-based adaptive search uses a new way to describe the state space of delivery and selects the heuristics to execute the actions for local searching. The experimental results show that the hybrid initialization and crossover strategy used in this paper can further explore the solution space and improve the algorithm's global search capability and convergence. DQN-based adaptive search is able to learn to perform higher-quality local searching and obtain better approximate optimal solutions.

In the future, we will explore the following aspects: (1) combining production scheduling and vehicle scheduling to carry out joint scheduling research, establish the corresponding scheduling model, and design the algorithm to solve the problem; (2) considering more realistic and complex constraints, such as customers with pick-up and delivery needs; and (3) considering such multi-objective optimization problems in the dynamic case.

REFERENCES

[1] Y.-L. Lan, F. Liu, W. W. Y. Ng, J. Zhang, and M. Gui, "Decomposition based multi-objective variable neighborhood descent algorithm for logistics dispatching," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 5, pp. 826–839, Oct. 2021, doi: 10.1109/TETCI.2020.3002228.

[2] J. Wang, T. Weng, and Q. Zhang, "A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2467–2478, Jul. 2019, doi: 10.1109/TCYB.2018.2821180.

[3] X. Wang, T.-M. Choi, Z. Li, and S. Shao, "An effective local search algorithm for the multidepot cumulative capacitated vehicle routing problem," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 12, pp. 4948–4958, Dec. 2020, doi: 10.1109/TSMC.2019.2938298.

[4] Y. Lyu, J. H. Yuan, and Y. Sun, "Optimization of vehicle routing problem in military logistics on wartime," *Control Decis.*, vol. 34, no. 1, pp. 121–128, Jan. 2019, doi: 10.13195/j.kzyjc.2017.0983.

[5] C. Malandraki and M. S. Daskin, "Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms," *Transp. Sci.*, vol. 26, no. 3, pp. 185–200, Aug. 1992, doi: 10.1287/trsc.26.3.185.

[6] S. Ichoua, M. Gendreau, and J.-Y. Potvin, "Vehicle dispatching with time-dependent travel times," *Eur. J. Oper. Res.*, vol. 144, no. 2, pp. 379–396, Jan. 2003, doi: 10.1016/s0377-2217(02)00147-9.

[7] M. Andres Figliozzi, "The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics," *Transp. Res. E, Logistics Transp. Rev.*, vol. 48, no. 3, pp. 616–636, May 2012, doi: 10.1016/j.tre.2011.11.006.

[8] B. Sawik, J. Faulin, and E. Pérez-Bernabeu, "A multicriteria analysis for the green VRP: A case discussion for the distribution problem of a Spanish retailer," *Transp. Res. Proc.*, vol. 22, pp. 305–313, Jan. 2017, doi: 10.1016/j.trpro.2017.03.037.

[9] L. Cai, W. Lv, L. Xiao, and Z. Xu, "Total carbon emissions minimization in connected and automated vehicle routing problem with speed variables," *Exp. Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113910, doi: 10.1016/j.eswa.2020.113910.

[10] X. Ren, X. Jiang, L. Ren, and L. Meng, "A multi-center joint distribution optimization model considering carbon emissions and customer satisfaction," *Math. Biosciences Eng.*, vol. 20, no. 1, pp. 683–706, Dec. 2022, doi: 10.3934/mbe.2023031.

[11] H. Cui, J. Qiu, J. Cao, M. Guo, X. Chen, and S. Gorbachev, "Route optimization in township logistics distribution considering customer satisfaction based on adaptive genetic algorithm," *Math. Comput. Simul.*, vol. 204, pp. 28–42, Feb. 2023, doi: 10.1016/j.matcom.2022.05.020.

[12] V. S. Nguyen, Q. D. Pham, T. H. Nguyen, and Q. T. Bui, "Modeling and solving a multi-trip multi-distribution center vehicle routing problem with lower-bound capacity constraints," *Comput. Ind. Eng.*, vol. 172, Oct. 2022, Art. no. 108597, doi: 10.1016/j.cie.2022.108597.

[13] M. K. Mehlawat, P. Gupta, A. Khaitan, and W. Pedrycz, "A hybrid intelligent approach to integrated fuzzy multiple depot capacitated green vehicle routing problem with split delivery and vehicle selection," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 6, pp. 1155–1166, Jun. 2020, doi: 10.1109/TFUZZ.2019.2946110.

[14] H. Fan, Y. Zhang, P. Tian, Y. Lv, and H. Fan, "Time-dependent multi-depot green vehicle routing problem with time windows considering temporal-spatial distance," *Comput. Oper. Res.*, vol. 129, May 2021, Art. no. 105211, doi: 10.1016/j.cor.2021.105211.

[15] B. Pan, Z. Zhang, and A. Lim, "A hybrid algorithm for time-dependent vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 128, Apr. 2021, Art. no. 105193, doi: 10.1016/j.cor.2020.105193.

[16] B. Pan, Z. Zhang, and A. Lim, "Multi-trip time-dependent vehicle routing problem with time windows," *Eur. J. Oper. Res.*, vol. 291, no. 1, pp. 218–231, May 2021, doi: 10.1016/j.ejor.2020.09.022.

[17] M. Ghasemi, A. Rahimnejad, R. Hemmati, E. Akbari, and S. A. Gadsden, "Wild geese algorithm: A novel algorithm for large scale optimization based on the natural life and death of wild geese," *Array*, vol. 11, Sep. 2021, Art. no. 100074, doi: 10.1016/j.array.2021.100074.

[18] M. Ghasemi, I. F. Davoudkhani, E. Akbari, A. Rahimnejad, S. Ghavidel, and L. Li, "A novel and effective optimization algorithm for global optimization and its engineering applications: Turbulent flow of water-based optimization (TFWO)," *Eng. Appl. Artif. Intell.*, vol. 92, Jun. 2020, Art. no. 103666, doi: 10.1016/j.engappai.2020.103666.

[19] M. Ghasemi, S. Ghavidel, J. Aghaei, E. Akbari, and L. Li, "CFA optimizer: A new and powerful algorithm inspired by Franklin's and Coulomb's laws theory for solving the economic load dispatch problems," *Int. Trans. Electr. Energy Syst.*, vol. 28, no. 5, p. e2536, May 2018, doi: 10.1002/etep.2536.

[20] W. Zhang, D. Yang, G. Zhang, and M. Gen, "Hybrid multiobjective evolutionary algorithm with fast sampling strategy-based global search and route sequence difference-based local search for VRPTW," *Exp. Syst. Appl.*, vol. 145, May 2020, Art. no. 113151, doi: 10.1016/j.eswa.2019.113151.

[21] Y. Zhou and J. Wang, "A local search-based multiobjective optimization algorithm for multiobjective vehicle routing problem with time windows," *IEEE Syst. J.*, vol. 9, no. 3, pp. 1100–1113, Sep. 2015, doi: 10.1109/JSYST.2014.2300201.

[22] F. E. Zulvia, R. J. Kuo, and D. Y. Nugroho, "A many-objective gradient evolution algorithm for solving a green vehicle routing problem with time windows and time dependency for perishable products," *J. Cleaner Prod.*, vol. 242, Jan. 2020, Art. no. 118428, doi: 10.1016/j.jclepro.2019.118428.

[23] J. Wang, W. Ren, Z. Zhang, H. Huang, and Y. Zhou, "A hybrid multi-objective memetic algorithm for multiobjective periodic vehicle routing problem with time windows," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4732–4745, Nov. 2020, doi: 10.1109/TSMC.2018.2861879.

[24] J.-Y. Ji and M. L. Wong, "Decomposition-based multiobjective optimization for nonlinear equation systems with many and infinitely many roots," *Inf. Sci.*, vol. 610, pp. 605–623, Sep. 2022, doi: 10.1016/j.ins.2022.07.187.

[25] W. Zhang, H. Li, W. Yang, G. Zhang, and M. Gen, "Hybrid multiobjective evolutionary algorithm considering combination timing for multi-type vehicle routing problem with time windows," *Comput. Ind. Eng.*, vol. 171, Sep. 2022, Art. no. 108435, doi: 10.1016/j.cie.2022.108435.

[26] Y. Cai, M. Cheng, Y. Zhou, P. Liu, and J.-M. Guo, "A hybrid evolutionary multitask algorithm for the multiobjective vehicle routing problem with time windows," *Inf. Sci.*, vol. 612, pp. 168–187, Oct. 2022, doi: 10.1016/j.ins.2022.08.103.

[27] G. Srivastava, A. Singh, and R. Mallipeddi, "NSGA-II with objective-specific variation operators for multiobjective vehicle routing problem with time windows," *Exp. Syst. Appl.*, vol. 176, Aug. 2021, Art. no. 114779, doi: 10.1016/j.eswa.2021.114779.

[28] X. B. Yan, Y. W. Fang, and W. S. Peng, "Multi-objective Harris hawk optimization algorithm based on adaptive Gaussian mutation," *J. Beijing Univ. Aeronaut. Astronautics.*, pp. 1–14, Jan. 2023, doi: 10.13700/j.bh.1001-5965.2022.0686.

[29] P. X. Zhao, W. H. Luo, and X. Han, "Time-dependent and bi-objective vehicle routing problem with time windows," *Adv. Prod. Eng. Manag.*, vol. 14, no. 2, pp. 201–212, Jun. 2019, doi: 10.14743/apem2019.2.322.

[30] Z. Wang, K. Ye, M. Jiang, J. Yao, N. N. Xiong, and G. G. Yen, "Solving hybrid charging strategy electric vehicle based dynamic routing problem via evolutionary multi-objective optimization," *Swarm Evol. Comput.*, vol. 68, Feb. 2022, Art. no. 100975, doi: 10.1016/j.swevo.2021.100975.

[31] Z. Zhang, H. Qin, and Y. Li, "Multi-objective optimization for the vehicle routing problem with outsourcing and profit balancing," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1987–2001, May 2020, doi: 10.1109/TITS.2019.2910274.

[32] J. Long, Z. Sun, P. M. Pardalos, Y. Hong, S. Zhang, and C. Li, "A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem," *Inf. Sci.*, vol. 478, pp. 40–61, Apr. 2019, doi: 10.1016/j.ins.2018.11.006.

[33] J.-Q. Li, Y. Du, K.-Z. Gao, P.-Y. Duan, D.-W. Gong, Q.-K. Pan, and P. N. Suganthan, "A hybrid iterated greedy algorithm for a crane transportation flexible job shop problem," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 2153–2170, Jul. 2022, doi: 10.1109/TASE.2021.3062979.

[34] Z. Zhang, Z. Wu, H. Zhang, and J. Wang, "Meta-learning-based deep reinforcement learning for multiobjective optimization problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7978–7991, Oct. 2023, doi: 10.1109/TNNLS.2022.3148435.

[35] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?bknumber=6267343

[36] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, May 1992, doi: 10.1007/bf00992698.

[37] K. M. R. Hoen, T. Tan, J. C. Fransoo, and G. J. van Houtum, "Effect of carbon emission regulations on transport mode selection under stochastic demand," *Flexible Services Manuf. J.*, vol. 26, pp. 170–195, Jun. 2012, doi: 10.1007/s10696-012-9151-6.

[38] T. Bektas and G. Laporte, "The pollution-routing problem," *Transp. Res. B, Methodol.*, vol. 45, no. 8, pp. 1232–1250, Sep. 2011, doi: 10.1016/j.trb.2011.02.004.

[39] M. Bruglieri, M. Paolucci, and O. Pisacane, "A matheuristic for the electric vehicle routing problem with time windows and a realistic energy consumption model," *Comput. Oper. Res.*, vol. 157, Sep. 2023, Art. no. 106261, doi: 10.1016/j.cor.2023.106261.

[40] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, Apr. 1987, doi: 10.1287/opre.35.2.254.

[41] T. Ahamed, B. Zou, N. P. Farazi, and T. Tulabandhula, "Deep reinforcement learning for crowdsourced urban delivery," *Transp. Res. B, Methodol.*, vol. 152, pp. 227–257, Oct. 2021, doi: 10.1016/j.trb.2021.08.015.

[42] R. J. Kuo, M. F. Luthfiansyah, N. A. Masruroh, and F. Eva Zulvia, "Application of improved multi-objective particle swarm optimization algorithm to solve disruption for the two-stage vehicle routing problem with time windows," *Exp. Syst. Appl.*, vol. 225, Sep. 2023, Art. no. 120009, doi: 10.1016/j.eswa.2023.120009.

BIN YUE received the B.E. degree in computer science and technology from Nanyang Institute of Technology, Nanyang, China, in 2020. He is currently pursuing the Ph.D. degree in management science and engineering with the North China University of Water Resources and Electric Power. His research interests include operations optimization, decision-making, and intelligent optimization algorithms.

JUNXU MA received the M.S. degree from Guangxi University, Nanning, China, in 2009, and the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2017. He is currently a Lecturer with the North China University of Water Resources and Electric Power. He has published multiple articles in the fields of CNC machine tools and intelligent manufacturing. His research interests include mechanical manufacturing systems and improving manufacturing precision.

JINFA SHI received the Ph.D. degree from Chongqing University, Chongqing, China, in 1994. He received the Postdoctoral Certificate at the Postdoctoral Mobile Station, Beijing Institute of Technology, Beijing, China, in 1996. He is currently a Professor, a Ph.D. Supervisor, and the Vice President of the North China University of Water Resources and Electric Power. He has published nearly 200 academic articles. His research interests include advanced manufacturing technology and management, industrial engineering and integrated management, information management, and system simulation.

JIE YANG received the B.E. and M.E. degrees from the North China University of Water Resources and Electric Power, Zhengzhou, China. He has published nearly 60 academic articles. His research interests include management science and engineering, mechanical engineering, and intelligent manufacturing and management.