

Research Article

Spatiotemporal-Dependent Vehicle Routing Problem Considering Carbon Emissions

Ziqi Liu, Yeping Chen , Jian Li , and Dongqing Zhang

College of Information Management, Nanjing Agricultural University, Nanjing 210031, China

Correspondence should be addressed to Jian Li; lijianzh@njau.edu.cn

Received 21 June 2021; Revised 15 August 2021; Accepted 19 August 2021; Published 24 September 2021

Academic Editor: Dragan Pamučar

Copyright © 2021 Ziqi Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming to improve the timeliness of logistics distribution and render the optimized route scheme effective under the real traffic network, we study the green vehicle routing problem with dynamic travel speed from both dimensions of time and space. A discrete formulation is proposed to calculate the travel time based on periods and arcs, which allows a vehicle to travel across an arc in multiple periods. Then, we establish a mixed-integer nonlinear programming model with minimum distribution costs including transportation costs, carbon emissions costs, and penalty costs on earliness and tardiness. A hybrid adaptive genetic algorithm with elite neighborhood search is developed to solve the problem. In the algorithm, a neighborhood search operator is employed to optimize elite individuals so that the algorithm can stimulate the intensification and avoid falling into a local optimum. Experimental instances are constructed based on benchmark instances of vehicle routing problem. The numerical results indicate that the proposed algorithm is rather effective in global convergence. Compared with the routing schemes in which travel speed merely varies with time periods or locations, the vehicle route optimized on spatiotemporal-varying speed outperforms them in terms of carbon emissions and timeliness. The research can provide a scientific and reasonable method for logistics enterprises to plan the vehicle schedule focusing on spatiotemporal-dependent speed of the road network.

1. Introduction

Most freight companies ignore the varying speed of the vehicle when planning the distribution routes (i.e., Çam and Sezen [1]), so that the preoptimized route scheme is hard for vehicles to service customers within the time windows. In order to ensure the timeliness of distribution, the vehicle route planning should apply the traffic information about the road network to calculate the travel time more precisely. At the same time, with the increasing carbon emissions, reducing vehicle fuel consumption has become the main trend of logistics distribution. Thus, the objective of distribution should additionally take carbon emissions influenced by travel speed on arcs into account. Therefore, we focus on the green vehicle routing problem with the spatiotemporal-varying vehicle speed and soft time windows, which is a variant of vehicle routing problem (VRP).

In classical VRP, a fleet of vehicles at one depot must deliver known demands to a set of customers [2]. The

objective is to minimize the travel cost with a set of specified routes for the vehicles under various constraints. VRP with time windows (VRPTW) can be generalized by imposing additional constraints that customers are only available during a specified time interval, called time window [3]. In most VRPTW, it is assumed that the distribution cost is proportional to the travel distance; however, the cost of carbon emission is not proportional to the travel distance. Therefore, considering the minimum distance or the minimum distribution cost cannot reduce carbon emissions. The cost of carbon emission is directly related to vehicle fuel consumption, which is associated with other factors, such as vehicle speed, current vehicle load, and road inclination [4]. With the heavy traffic congestion, vehicle speed has become an important factor that affects carbon emissions. Travel speed varies with different moments and locations, and such dynamic travel speed affects carbon emission and the arrival time at the customer [5]. Therefore, this study investigates the spatiotemporal-dependent green vehicle routing

problem with time windows. The fixed costs of distribution, the variable cost of transportation, penalty costs on time windows, and carbon emission cost are considered in the objective function.

We focus on the spatiotemporal-dependent green vehicle routing problem with time windows which is related to the time-dependent vehicle routing problem with time windows (TDVRPTW) and green vehicle routing problem with time windows (GVRPTW). VRP has been proven to be an NP-hard problem, so it is difficult to obtain the optimal solution in the finite amount of computation time. Liu et al. [6] applied ant colony algorithm, a metaheuristic algorithm, to solve time-dependent vehicle routing problem with time windows (TDVRPTW). Time-dependent green vehicle routing problem with time windows (TDGVRPTW) studied in our work is more complicated than TDVRPTW which needs to calculate the carbon emissions of each period and road section. Thus, we apply the genetic algorithm, a classical metaheuristic algorithm to find the near-optimal solutions of the problem. The contributions of this work are summarized as follows:

- (1) Due to the real traffic network speed distribution, the calculation of travel time is a discrete function of periods and arcs, which meets the first-in and first-out rules. We adopt the fuel consumption introduced by Esteves-Booth et al. [7] with the spatiotemporal speed considered. This work establishes a mixed-integer nonlinear planning model (MINP) with the total distribution costs.
- (2) A hybrid adaptive genetic algorithm with elite neighborhood search (HAGA_ELS) is proposed. In the algorithm, the elite neighborhood search is employed to avoid premature convergence and balance the intensification and diversification of the solution.

The rest of the article is as follows. Section 2 introduces relevant research. Section 3 describes the research problem and introduces the calculation method of travel time with spatiotemporal dependence, upon which the mixed-integer nonlinear planning model is formulated. Section 4 presents HAGA_ELS. Section 5 assesses the algorithm through modified benchmark instances and analyzes the sensitivity of the model to different parameter settings. Section 6 presents the conclusions.

2. Literature Review

VRPTW was developed based on classic VRP. Russell [8] was the first to propose an effective heuristic algorithm to solve the problem of travelers with time windows. Solomon [9] established a mixed-integer nonlinear planning problem for VRPTW. Most studies about VRPTW used a fixed speed, while the dynamics of speed in real traffic networks was ignored. At the same time, to reduce the increasing carbon emissions, the objective of distribution should additionally take carbon emissions influenced by travel speed into account. Therefore, on the basis of VRPTW, this work mainly reviews the time-dependent vehicle routing problem with

time windows (TDVRPTW) and time-dependent green vehicle routing problem with time windows (TDGVRPTW).

2.1. Research on TDVRPTW. The traditional VRP assumes that the speed is constant, but this assumption is inconsistent with the real situation because the vehicle is running in the real traffic network [3]. Thus, time-dependent VRP (TDVRP) has arisen to describe more real network optimization problems. TDVRP accounts for the real traffic conditions by setting travel time based on the path between two customers and the time of day. Generally, time-dependent issues can be addressed in two ways: one is that travel time changes over time, and the other is that travel speed varies with moment.

The first type of research on time-dependent travel time directly assumes that travel time is a function related to the time in a day. The travel time function can be piecewise linear form (Sun et al. [10]) or distribute function form (Hu et al. [11] and Babaei and Rajabi-Bahaabadi [12]). Sun et al. [10] applied a travel time function that is piecewise linear and continuous based on the change of speed. As for the distribution function of travel time, Hu et al. [11] built a travel time distribution function and considered the uncertainty of travel time by adding uncertainty budget into the function. Babaei and Rajabi-Bahaabadi [12] assumed that travel time for each time of day satisfies a distribution function and the travel time between specific customers can be calculated by the departure time. The travel time can also be acquired from intelligent transportation system. Saint-Guillain et al. [13] retrieved the travel time between customers by Google Map's API before scheduling.

The second research on time-dependent travel speed studies the relationship between travel speed and time to simulate the real traffic network. Travel speed is mostly a stepwise function (Xiao and Konak [14], Wu and Ma [15], Zhu and Hu [16], Duan et al. [17], Gruler et al. [18], and Gmira et al. [19]), or a continuous function (Xu et al. [20] and Fan et al. [21]), or retrieved from historical data (Franceschetti et al. [22]). Gmira et al. [19] assumed travel speed of each time and arc as a stepwise function, from which a piecewise travel time function can be derived. As for the continuous speed function, Xu et al. [20] used a sine function to simulate time-dependent speed and a non-dominated sorting genetic algorithm to solve the problem. Fan et al. [21] also used the sine function in [20] to describe the travel speed but segmented it based on the road grade. Some research studies directly retrieved travel speed from historical data rather than build a function. Franceschetti et al. [22] assumed that the travel speed and time period are constants that can be extracted from historical data, and then travel time can be calculated from the speed and distance.

Most of the above literature studies only consider the speed varying in time, ignoring the speed varying in space. Table 1 lists the methods of time-dependent issue processing of the above literature studies.

2.2. Research on TDGVRPTW. In recent years, reducing carbon emissions has become an important issue in vehicle routing optimization. Kwon et al. [23] aimed at the green

TABLE 1: Literature studies in TDVRPTW.

Instances	Year	Travel time				Travel speed	
		Piecewise linear function	Distribution function	Historical data	Stepwise function	Continuous function	Historical data
Sun et al. [10]	2018	√					
Hu et al. [11]	2018		√				
Babaei and Rajabi-Bahaabadi [12]	2019		√				
Saint-Guillain et al. [13]	2021			√			
Xiao and Konak [14]	2016				√		
Wu and Ma [15]	2017				√		
Zhu and Hu [16]	2019				√		
Duan et al. [17]	2019				√		
Gruler et al. [18]	2020				√		
Gmira et al. [19]	2021				√		
Xu et al. [20]	2019					√	
Fan et al. [21]	2021					√	
Franceschetti et al. [22]	2013						√
Our work		Temporal axis: travel speed varies with a stepwise function converted from continuous function Spatial axis: travel speed obeys a distribution in accordance with historical data					

vehicle routing problem to set carbon emission as a ratio function for driving distance, thus equating the problem of reducing carbon emission to minimizing driving paths, but which ignored the impact of speed on fuel consumption.

In the subsequent research studies, time-varying speed was taken into account in TDGVRPTW. Some of them only targeted carbon emissions. For example, Qian and Eglese [24] established a model that aims to minimize fuel consumption under time-dependent travel speed and used the tabu search algorithm to solve the problem. And Xiao and Konak [25] formulated a model for a fleet of heterogeneous vehicles operating, in which carbon emissions are calculated by dynamic payload weights and time-varying speed. While some research studies considered the biobjective of minimizing carbon emissions and costs (see Wang et al. [26] and Poonthalir and Nadarajan [27]), in which Poonthalir and Nadarajan [27] adopted triangular distribution to describe varying speed.

Further studies focused on more factors affecting carbon emissions. Xu et al. [20] calculated the fuel consumption by the load of a vehicle and its capacity, time-varying speed, driving distance, and traffic congestion. Fan et al. [21] established a travel speed function that considers three road conditions and analyzed the impact of vehicle type, speed, load, and road gradient on fuel consumption, proposing a hybrid variable neighborhood genetic algorithm for solving the problem.

Table 2 summarizes and compares the formulation in the stream of TDGVRPTW.

From the literature presented above, we can conclude that most of the studies on TDGVRPTW did not consider the difference in speed or travel time on different arcs. Among them, Fan et al. [21] considered the time dependence of travel speed under different road grades and established travel speed function of three road grades. However, the stochastic speed distribution was not fully considered, and the spatial difference in speed was only reflected in the road

classification. Given that a spatial speed difference also exists in the same road grade section, the treatment cannot fully reflect the spatial difference in real speed in the wide distribution problem.

In the current study, a spatiotemporal-dependent speed division pattern (T&S) is used. In this pattern, each arc has different speeds in different periods, and the speed of arcs is determined by periods and arcs. The travel speed in different dimensions obeys a certain distribution. This study analyzes the limitations of the time-dependent speed division pattern (T) and the space-dependent speed division pattern (S).

3. Problem and Mathematical Model

TDGVRPTW studied in this paper can be described as follows. In the network graph $G = (V, E)$, the set of nodes is $V = \{0\} \cup V_0$, where 0 is the distribution center and $V_0 = \{1, \dots, n\}$ is the set of known customers. The set of edges is $E = \{(i, j) | i \neq j \in V\}$. The location (x_i, y_i) , nonnegative demand w_i , service time ST_i , and delivery time of the arrival window $[ET_i, LT_i]$ of customer i are known. The shortest path between two nodes, namely, edge $(i, j) \in E$, has a fixed distance of d_{ij} . The distribution road network has a time-varying traffic state, and the distribution time range is divided into a fixed number of time intervals $L = \{1, 2, \dots, h\}$. The different arcs $(i, j) \in E$ in each period $l \in L$ have various travel speed v_{ij}^l , and travel speed v_{ij}^l is known in advance in accordance with the travel speed function. Distribution center has a certain number of distribution vehicles $K = \{1, 2, \dots, m\}$, and all customers V_0 go back to the distribution center after the completion of the distribution. Distribution vehicle $k \in K$ has the same maximum load Q . The fuel consumption formula $FUEL(v, q, d)$ is given based on travel speed v , load weight q , and travel distance d , and fuel-to-CO² coefficient ρ is known. On the premise of the constraints of vehicle loading and customer time window,

TABLE 2: Literature studies in TDGVRPTW.

References	Year	Problem features			Objectives	Solution
		Temporal speed	Spatial speed	Factors in fuel consumption/ carbon emission		
Kwon et al. [23]	2013			Distance	Operation costs and carbon emission trading costs	Simulated annealing algorithm
Qian and Eglese [24]	2016	*		Speed	Carbon emissions	Tabu search algorithm
Xiao and Konak [25]	2015	*		Speed, load, and vehicle type	Fuel consumption	Partial mixed-integer planning and iterated neighborhood search
Wang et al. [26]	2019	*		Speed and distance	Carbon emission and operating costs	A multiphase hybrid heuristic algorithm
Poonthair and Nadarajan [27]	2018	*		Speed and distance	Carbon emission and operating costs	Particle swarm optimization
Xu et al. [20]	2019	*		Speed, load, vehicle capacity, and distance	Fuel consumption and customer satisfaction	Nondominated sorting genetic algorithm II
Fan et al. [21]	2021	*	* (Divided by three road grades)	Speed, load, vehicle type, and road gradient	Total costs: fixed costs, penalty costs, and fuel costs	Hybrid genetic algorithm
Our work		*	* (Stochastic speed distribution)	Speed, load, vehicle parameters, and distance	Total costs: transportation costs, carbon emission costs, and penalty costs	HAGA_ELS

the vehicle distribution routing is optimized considering the carbon emission of the vehicle. Thus, the total costs including the fixed costs of distribution, transportation costs, carbon emission costs, and penalty costs of the time window are the lowest. In the TDGVRPTW, the following constraints need to be met:

- (1) The distribution center is unique, and all vehicles from the distribution center go back to the distribution center after completing the distribution task.
- (2) Each customer can only be served once and by one vehicle only.
- (3) Only one type of vehicle is available in the distribution center, and the number is unlimited.
- (4) Customer demands for goods, location, and service time window are known in advance.
- (5) Vehicles in service time do not produce fuel consumption.

3.1. Spatiotemporal-Dependent Travel Time Function. In a time-varying road network, the vehicle speed changes continuously with time. According to the historical traffic data, the morning rush hour is usually from 7:00 am to 9:00 am, and the evening rush hour is from 17:00 pm to 19:00 pm. Many vehicles emerge during the rush hour, and the average speed on the road decreases. For example, the relationship between vehicle speed v and time t in an entire day proposed by Xu et al. [20] can be approximated by the trigonometric function in equation (1), where φ_p , γ_p , and δ_p are constants related to the road conditions of the day:

$$v(t) = \begin{cases} \varphi_1 \sin(\gamma_1 t) + \delta_1, & t \in [0, T_1] \\ \vdots & \\ \varphi_p \sin(\gamma_p t) + \delta_p, & t \in [T_{p-1}, T_p] \end{cases} \quad (1)$$

TDVRPTW is an NP-hard problem, and it greatly increases the difficulty of solving the problem by using the function of continuous varying speed. To reduce the computational complexity caused by nonlinear speed change, this study transforms the trigonometric function of travel speed into a piecewise function in accordance with the period and takes the average value of velocity in each period as the constant velocity in that period (Figure 1). The finer the time division is, the closer the speed segmentation function is to the continuous change in speed. The time interval is divided into periods $L = \{1, 2, \dots, h\}$, the time of each period is recorded as $\Gamma_l (l \in L)$, and the time interval is ΔI . The travel speed of the vehicle in any period l is

$$v^l = \frac{\int_l^{l+\Delta I} v(t) dt}{\Delta I}, \quad l \in L. \quad (2)$$

In some studies (e.g., Kwon et al. [23]), the travel time calculation of arc (i, j) depends on the departure time of vehicles from node i and it assumes that the characteristics of arcs are the same. However, in a real road network, the traffic congestion of different arcs varies due to the influence of geographic location, traffic flow, and other factors. The all-day travel speed of each arc is not completely equal, instead the travel speed is spatiotemporal-dependent; that is, at the same time $l \in L$, the traveling speed v_{ij}^l of different arc (i, j) is various. Therefore, the travel speed and travel time in the

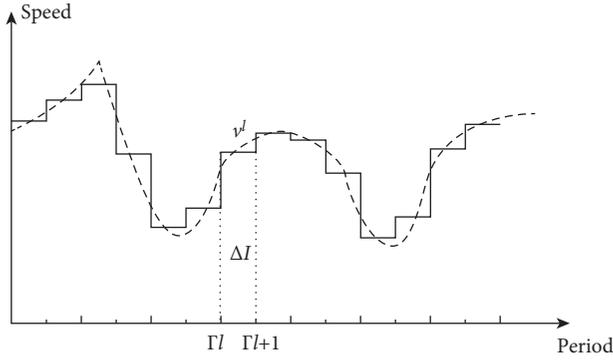


FIGURE 1: Travel speed variation function.

process of vehicle distribution not only depend on the time when the vehicle leaves node i but also relate to the arc from node i to the next node j . According to the historical traffic data on all-day arcs in a province of China in the study of Yang [28], the travel speed of different arcs in the same period obeys a normal distribution; that is, the travel speed v_{ij}^l of any arc $(i, j) \in E$ in period l obeys normal distribution $N(v^l, \sigma^2)$, and the variance σ^2 reflects the difference degree of travel speeds of different arcs. On this basis, the non-negative random velocity matrix \tilde{V} of all arcs in the road network in different periods can be obtained. A small network with 6 nodes, 21 arcs, and 9 periods is taken as an example, as shown in Figure 2. Figure 3 presents the mean value of the speed distribution of the small network, including 189 (21×9) random velocity variables.

In the distribution process, vehicles are likely to cross from a period to the next period or even cross multiple periods when passing through the arc (i, j) . Hill and Benton [29] and others did not consider the FIFO criterion when dealing with time-varying speeds across periods; that is, the vehicles that start first should arrive first starting from one point to the same next point and driving on the same arc [30]. Several studies (e.g., Ioannou et al. [31]) ensured the continuity of travel time by adjusting the speed across periods, but the calculation was complicated for solving problems. Therefore, this study refers to the work of Li et al. [32] to improve the calculation of travel time across periods, which satisfies the FIFO criterion. The vehicle is assumed to travel from node i to node j at time a_i across S periods, so road section length d_{ij} is divided into d_1, d_2, \dots, d_{S+1} . The speed in the segment s ($s = 1, 2, \dots, S + 1$) is $v_s \in \tilde{V}$, and the time of division is Γ_s , as shown in Figure 4.

The travel time of road section d_1 is $\Gamma_1 - a_i$, and the travel time of subsequent road section d_s is $\Gamma_s - \Gamma_{s-1}$ ($s = 2, 3, \dots, S$). Thus, an equation between section length and time is available:

$$d_1 = v_1(\Gamma_1 - a_i), d_s = v_s(\Gamma_s - \Gamma_{s-1}) \quad s = 2, 3, \dots, S. \quad (3)$$

The driving distance of the $S + 1$ section is

$$d_{S+1} = d_{ij} - \sum_{s=1}^S d_s = d_{ij} - v_1(\Gamma_1 - a_i) - \sum_{s=2}^S v_s(\Gamma_s - \Gamma_{s-1}). \quad (4)$$

Therefore, the arrival time t_j of node j is

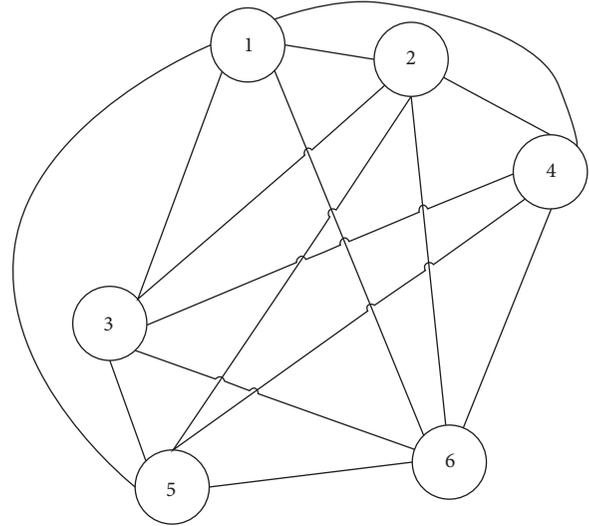


FIGURE 2: Small-size network.

	period									
	1	2	3	4	5	6	7	8	9	
1	61	55	43	37	29	23	29	31	14	= \tilde{V}
2	57	53	48	47	33	26	26	30	28	
3	68	46	35	51	21	24	31	20	42	
...	
10	49	38	51	55	36	25	22	24	47	
11	52	47	44	36	22	40	13	20	44	
12	63	58	45	29	48	30	31	13	32	
13	47	50	46	40	26	34	29	34	50	
14	65	43	45	36	36	23	25	42	40	
...	
20	50	49	56	37	17	31	35	30	39	
21	52	48	38	33	21	17	24	14	38	

FIGURE 3: Velocity matrix.

$$t_j = \Gamma_S + \frac{d_{S+1}}{v_{S+1}} = \frac{v_1}{v_{S+1}} a_i + \Gamma_S + \frac{d_{ij} - v_1 \Gamma_1 - \sum_{s=2}^S v_s (\Gamma_s - \Gamma_{s-1})}{v_{S+1}}. \quad (5)$$

According to equation (5), the time t_j when the vehicle arrives at node j is an increasing function of time a_i when the vehicle departs from node i , which meets the FIFO criterion.

When the arc (i, j) spans S periods, the following conditions should be met when the vehicle starts from node i : the vehicle starts from the first period and has not finished driving in arc (i, j) from the first period to the S th period. The corresponding expression is as follows:

- (a) $a_i < \Gamma_1$,
- (b) $v_1(\Gamma_1 - a_i) + \sum_{s=2}^S v_s(\Gamma_s - \Gamma_{s-1}) \leq d_{ij}$.

The abovementioned arrangement yields

$$\Gamma_1 - \frac{d_{ij}}{v_1} + \sum_{s=2}^S \frac{v_s}{v_1} (\Gamma_s - \Gamma_{s-1}) \leq a_i < \Gamma_1. \quad (6)$$

The calculation formula of the arrival time t_j of the next node j where any arc (i, j) crosses S ($S < h$) periods is as follows:

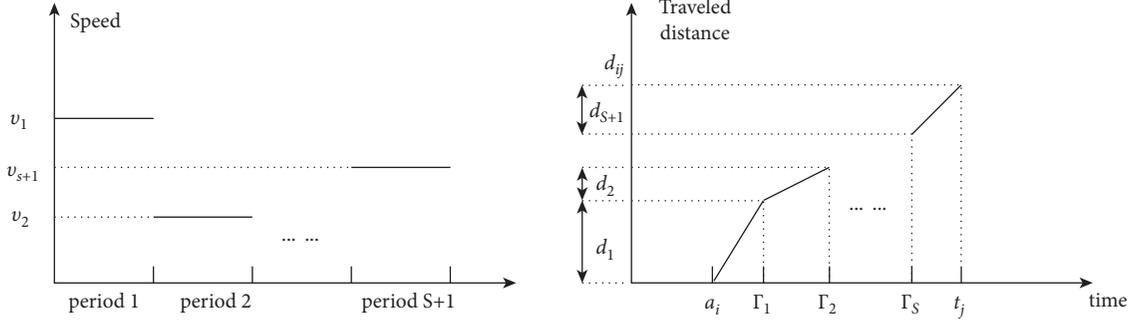


FIGURE 4: Example of a time path without the nonstopping assumption.

$$t_j = \begin{cases} t_i + \frac{d_{ij}}{v_{ij}^l}, & a_i < \Gamma_{l+1} - \frac{d_{ij}}{v_{ij}^l}, \quad l = 1, \dots, L, \\ \frac{v_{ij}^l}{v_{ij}^{l+S}} a_i + \Gamma_{l+S} + \frac{d_{ij} - v_{ij}^l \Gamma_{l+1} - \sum_{s=l+1}^{l+S-1} v_{ij}^s (\Gamma_s - \Gamma_{s-1})}{v_{ij}^{l+S}}, & \Gamma_{l+1} - \frac{d_{ij}}{v_{ij}^l} + \sum_{s=l+1}^{l+S-1} \frac{v_{ij}^{l+S}}{v_{ij}^s} (\Gamma_s - \Gamma_{s-1}) \leq a_i < \Gamma_{l+1}, \quad l = 1, \dots, L-S. \end{cases} \quad (7)$$

3.2. *Calculation of the Amount of Fuel Consumption.* Carbon emissions are directly affected by vehicle fuel consumption. Hence, this study applies vehicle fuel consumption to calculate carbon emissions. Considering that most distribution vehicles are heavy types, a carbon emission model is established in combination with the comprehensive modal emission model (CMEM) [7]. The objective function of fuel consumption when driving in different sections is as follows:

$$fuel = \lambda \left(\alpha_1 d v^2 + \frac{\alpha_2 d}{v} + \alpha_3 Q d \right). \quad (8)$$

In equation (8), $\lambda = \zeta / \kappa \varphi$, ζ is the fuel-air mass ratio, κ is a typical diesel calorific value, φ is the fuel conversion coefficient, engine module coefficient $\alpha_1 = 0.5 C_d A_f \rho / (1000 \epsilon \omega)$, speed module coefficient $\alpha_2 = k N V$, and vehicle load module coefficient $\alpha_3 = \rho_0 + (\rho_{max} - \rho_0) / Q_0$. The meanings and values of specific parameters are presented in Appendix A.

In summary, throughout the distribution, the total fuel consumption after passing through all customer nodes is

$$FUEL = \sum_{(i,j) \in E} \sum_{k \in K} x_{ijk} fuel_{ij} = \sum_{(i,j) \in E} \sum_{k \in K} x_{ijk} \sum_{l \in L} \lambda \left(\alpha_1 d_{ij}^l (v_{ij}^l)^2 + \alpha_2 \frac{d_{ij}^l}{v_{ij}^l} + \alpha_3 Q_{ij} d_{ij}^l \right). \quad (9)$$

In equation (9), x_{ijk} is a binary variable indicating whether arc (i, j) is traveled by vehicle k ($x_{ijk} = 1$) or not ($x_{ijk} = 0$). $fuel_{ij}$ is the fuel consumption of the vehicle from node i to node j . d_{ij}^l is the length of the arc (i, j) during time l . v_{ij}^l represents the travel speed of the arc (i, j) during time l . And Q_{ij} represents the loading weight of the vehicle from node i to node j .

Figure 5 shows the relationship between fuel consumption and travel speed for a fixed vehicle type. With the increase of travel speed, the fuel consumption decreases at first. However, when the travel speed reaches a certain value, here 48.14 km/h, the fuel consumption begins to increase.

The specific process of calculating carbon emission in consideration of spatiotemporal-dependent travel speed is shown in Algorithm 1.

3.3. *Mathematical Model.* On the basis of the analysis above, the TDGVRPTW model is established, and the model parameters and decision variables are shown in Table 3.

In the above decision variables, x_{ijk} is the variable related to the arc of vehicle routing. y_{ki} represents the connection between vehicles and customers. Q_{ij} and t_i are variables in vehicle driving process. Q_{ij} indicates the connection between vehicle load and arcs. t_i is calculated by

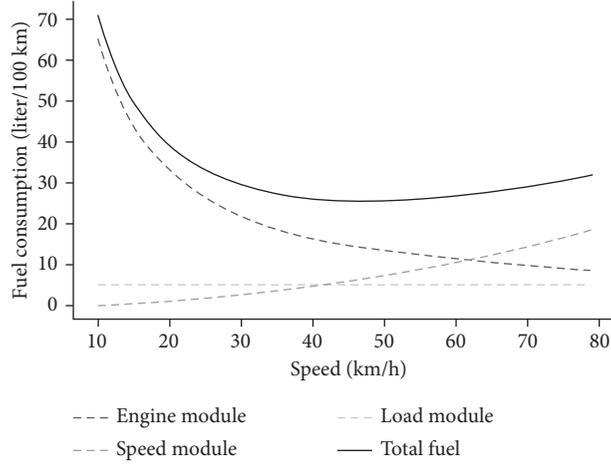


FIGURE 5: Fuel consumption function of travel speed (with 2000 kg of load).

```

//Depart from  $i$  to  $j$ , assume that the period is divided into  $L$  segments
(1) Input  $d_{ij}, a_i, Q_{ij}$ , Velocity matrix  $\tilde{V}$ , Time slot division  $\Gamma$ , parameters  $\lambda, \alpha_1, \alpha_2, \alpha_3$ ;
(2)  $\Gamma_0 \leftarrow a_i, v \leftarrow \tilde{V}(\text{arc}(i, j))$ , where  $\Gamma > a_i$ ,  $\text{fuel} \leftarrow 0, d' \leftarrow d_{ij}$ ;
//Judging whether to span  $S$  periods
(3) If the criteria of crossing  $S$  periods is satisfied. //i.e. The path is divided into  $S + 1$  segments
(4) For  $k = 1$  to  $S$  do begin //Recursive method
(5)  $d_k \leftarrow v_k (\Gamma_k - \Gamma_{k-1})$ ; //The distance traveled in each period
(6)  $\text{fuel} \leftarrow \text{fuel} + \lambda (\alpha_1 d_k) (v_k^2 + \alpha_2 (\Gamma_k - \Gamma_{k-1}) + \alpha_3 Q_{ij} d_k)$ ;
(7)  $d' \leftarrow d' - d_k$ ;
(8) End for
(9)  $d_{S+1} \leftarrow d'$ ; //the length of the  $S + 1$  section
//Calculate the arrival time at node  $j$  and fuel consumption of traveling  $\text{arc}(i, j)$ 
(10)  $t_j \leftarrow \Gamma_S + d_{S+1}/v_{S+1}$ ;
(11)  $\text{fuel} \leftarrow \text{fuel} + \lambda (\alpha_1 d_{S+1} (v_{S+1}^2) + \alpha_2 (t_j - \Gamma_S) + \alpha_3 Q_{ij} d_{S+1})$ ;
(12) Else
(13)  $t_j \leftarrow a_i + d_{S+1}/v_{S+1}$ ;
(14)  $\text{fuel} \leftarrow \text{fuel} + \lambda (\alpha_1 d_{ij} (v_1^2) + \alpha_2 (t_j - a_i) + \alpha_3 Q_{ij} d_{ij})$ ;
(15) End if

```

ALGORITHM 1: Calculation of travel time and fuel consumption.

spatiotemporal-dependent speed, which is the crucial variable considered for TDGVRPTW. The mathematical

optimization model of this paper is formulated to minimize the total distribution costs, as follows:

$$\begin{aligned}
 F = & \sum_{i \in V_0} \max\{ET_i - t_i, 0\} \beta_1 + \sum_{i \in V_0} \max\{0, t_i - LT_i\} \beta_2 + \sum_{k \in K} \sum_{i \in V_0} f_1 y_{ki} + \sum_{i \in V} \sum_{j \in V} x_{ijk} f_2 d_{ij} \\
 & + \sum_{(i,j) \in E} \sum_{k \in K} x_{ijk} \sum_{l \in L} \lambda \left(\alpha_1 d_{ij}^l (v_{ij}^l)^2 + \alpha_2 \frac{d_{ij}^l}{v_{ij}^l} + \alpha_3 Q_{ij} d_{ij}^l \right) \rho \times \text{tax},
 \end{aligned} \tag{10}$$

$$\sum_{i \in V_0} y_{ki} w_i \leq Q \quad \forall k \in K, \tag{11}$$

$$\sum_{k \in K} y_{ki} = 1 \quad \forall i \in V_0, \tag{12}$$

TABLE 3: Notations of sets, parameters, and variables.

Notations for customer locations and requirements	
i	Index of nodes, $i = 0, 1, 2, \dots, n$ (the depot is represented by 0)
n	Total number of nodes (including the depot)
V	Set of nodes including the depot
V_0	Set of nodes excluding the depot, $V_0 = V/\{0\}$
E	Set of arcs formed by all pairs of nodes, $E = \{(i, j): \forall i \in V, j \in V, i \neq j\}$
d_{ij}	Distance of arc (i, j)
w_i	The demand of customer i
ST_i	Service time of customer i
$[ET_i, LT_i]$	Time window for starting to serve customer i
Notations for vehicle parameters	
k	Index of vehicles, $k = 0, 1, 2, \dots, m$
K	Set of vehicles
Q	Carrying capacity of a vehicle
Notations for time-varying traffic conditions	
l	Index of period, $l = 1, 2, \dots, h$
h	Number of periods, $h = \text{card}(L)$
L	Set of periods, $l \in L$
$[\Gamma_l, \Gamma_{l+1}]$	Beginning and ending time of period l
v_{ij}^l	Travel speed on arc (i, j) in period l
d_{ij}^l	Distance of arc (i, j) in period l
t_{ij}	Travel time spent on arc (i, j)
Notations for fuel consumption and CO ² emission	
α_1	Engine power module coefficient
α_2	Engine power speed module coefficient
α_3	Vehicle load module coefficient
λ	Coefficient of fuel consumption
ρ	Fuel-to-CO ² coefficient
Notations for total costs	
f_1	Fixed cost of vehicle
f_2	Drive cost rate
β_1	Penalty coefficient of arrival before the time window
β_2	Penalty coefficient of arrival after the time window
tax	Carbon tax
The binary and continuous decision variables of the model are defined as follows:	
x_{ijk}	$\{0, 1\}$ variable indicating whether arc (i, j) is traveled (1) or not (0) by vehicle k
y_{ki}	$\{0, 1\}$ variable indicating whether customer i is served (1) or not (0) by vehicle k
Q_{ij}	Continuous variable indicating the load of vehicle on arc (i, j)
t_i	Continuous variable indicating the arrival time (or service starting time) at node i

$$\sum_{i \in V_0} x_{ijk} - \sum_{i \in V_0} x_{jik} = 0 \quad \forall j \in V_0, \forall k \in K, \quad (13)$$

$$\sum_{i \in V_0} x_{ijk} = y_{kj} \quad \forall j \in V_0, \forall k \in K, \quad (14)$$

$$\sum_{j \in V_0} x_{ijk} = y_{ki} \quad \forall i \in V_0, k \in K, \quad (15)$$

$$\sum_{j \in V_0} x_{0jk} = \sum_{i \in V_0} x_{i0k} \leq 1, \quad (16)$$

$$t_j \geq t_i + ST_i + t_{ij} - M(1 - x_{ijk}) \quad \forall i \in V, j \in V_0, k \in K, \quad (17)$$

$$t_j \leq t_i + ST_i + t_{ij} + M(1 - x_{ijk}) \quad \forall i \in V, j \in V_0, k \in K, \quad (18)$$

$$\sum_{i \in V} Q_{ij} - \sum_{p \in V} Q_{jp} = w_j \quad \forall j \in V_0. \quad (19)$$

Equation (10) is the objective function that represents the sum of penalty costs on earliness and tardiness, vehicle fixed costs, transportation costs, and carbon emission tax. Formulation (11) represents the vehicle maximum load constraint. Equation (12) indicates that each customer must be delivered by one vehicle. Equation (13) expresses the vehicle flow balance constraint in the node. Equations (14) and (15) represent the uniqueness constraint of node distribution vehicles. Formulation (16) represents the constraint that vehicle departures and returns to the distribution center. Formulations (17) and (18) express the relationship between the arrival time of a node and that of the previous node, where M is a positive big number and t_{ij} can be deduced from equation (7). Equation (19) indicates that the node delivery quantity is equal to the difference between the loading quantity when entering and the loading quantity when leaving.

4. Problem-Solving Method

TDGVRPTW is an NP-hard problem. In this study, the spatiotemporal division of travel speed is added to the classical VRP, thus making the solution increasingly difficult. A metaheuristic algorithm is usually used to solve this kind of problem. The genetic algorithm has strong global search ability and robustness, and it is mature enough for application in vehicle scheduling problems. However, it easily falls into local convergence. Therefore, this study designs a hybrid adaptive genetic algorithm with elite local search (HAGA_ELS). The basic idea is to obtain the initial population by using the Clarke and Wright Saving (CWS) heuristic in the study of Marković et al. [33] and then perform crossover and mutation on individual populations in the evolution process. The elite individuals in the original population are optimized by neighborhood search and reinserted into the new population to keep the population size unchanged and increase the population diversity. Meanwhile, the crossover and mutation rates are adaptively adjusted according to the iteration times and individual fitness changes. The operation flow of HAGA_ELS is shown in Figure 6.

4.1. Initial Solution. In this study, population size is set as N , and each chromosome is composed of n (n is the number of customers) integers. The sequence of gene bits indicates the order in which distribution vehicles visit customers. To improve the quality of the initial population, this study uses the CWS heuristic and random permutation operator to generate the initial population. The steps of the random permutation operator are as follows: an integer sequence containing customers is randomly generated, and vehicles are assigned to the route in accordance with the vehicle load constraint. When the vehicle cannot meet the next customer demand, another vehicle is called, and the current gene of the chromosome is set to 0. The driving sequence of a vehicle appears as subrouting. If the number of required vehicles exceeds the number of existing

vehicles, then the route is an infeasible solution. The steps above are repeated until the chromosome number reaches population size N .

4.2. Fitness Function. The fitness of each chromosome in the population is constructed by the model objective function equation (10). In this study, the pressure principle is used to deepen the gap of fitness between individuals, thus increasing the probability of excellent individuals entering the next generation [34]. The fitness function is shown in equation (20), where a is a constant greater than 1, f_i is the fitness of chromosome i , and z_i is the objective function value corresponding to chromosome i :

$$f_i = a \left\{ \max_n z_i \right\} - z_i. \quad (20)$$

4.3. Selection Operator. In this study, elite and roulette strategies are combined to implement the selection operation. The specific steps are as follows: with the descending order of chromosome fitness value, $N(1 - \Phi)$ of the top individuals is selected for elite evolution operation (Φ is the generation gap of population evolution) to ensure that the outstanding individuals in the parent will not be lost due to crossover and mutation. Then, the roulette strategy is adopted to select $N\Phi$ individuals from their parents. The selection probability of each chromosome is $p_i = f_i / (\sum_{i \in N} f_i)$, and the greater the fitness is, the greater the probability of being selected is. The selected individuals are subjected to crossover mutation operation, and the elite evolutionary individuals are reinserted into the population after crossover and mutation so that the population size is always N .

4.4. Improved Crossover and Mutation Operators. For the traditional adaptive genetic algorithm, when the larger fitness value of the crossover operation individual is equal to the maximum fitness value of the population, the crossover and mutation rates are zero; additionally, the influence of evolution time on crossover and mutation rate is not considered. Therefore, the improved method of cloud adaptive adjustment proposed by Dai et al. [35] is adopted. An adaptive adjustment algorithm of crossover and mutation rate is designed based on the robustness and efficiency of cloud theory. In the early stage of evolution, large crossover and mutation rates make the individuals with low fitness participate in the crossover and mutation operation with high probability to produce excellent individuals. In the late stage of evolution, small crossover and mutation rates make individuals with high fitness participate in the crossover and mutation operation with low probability to protect outstanding individuals from being destroyed. The algorithm for generating crossover rate P_c and mutation rate P_m is as follows:

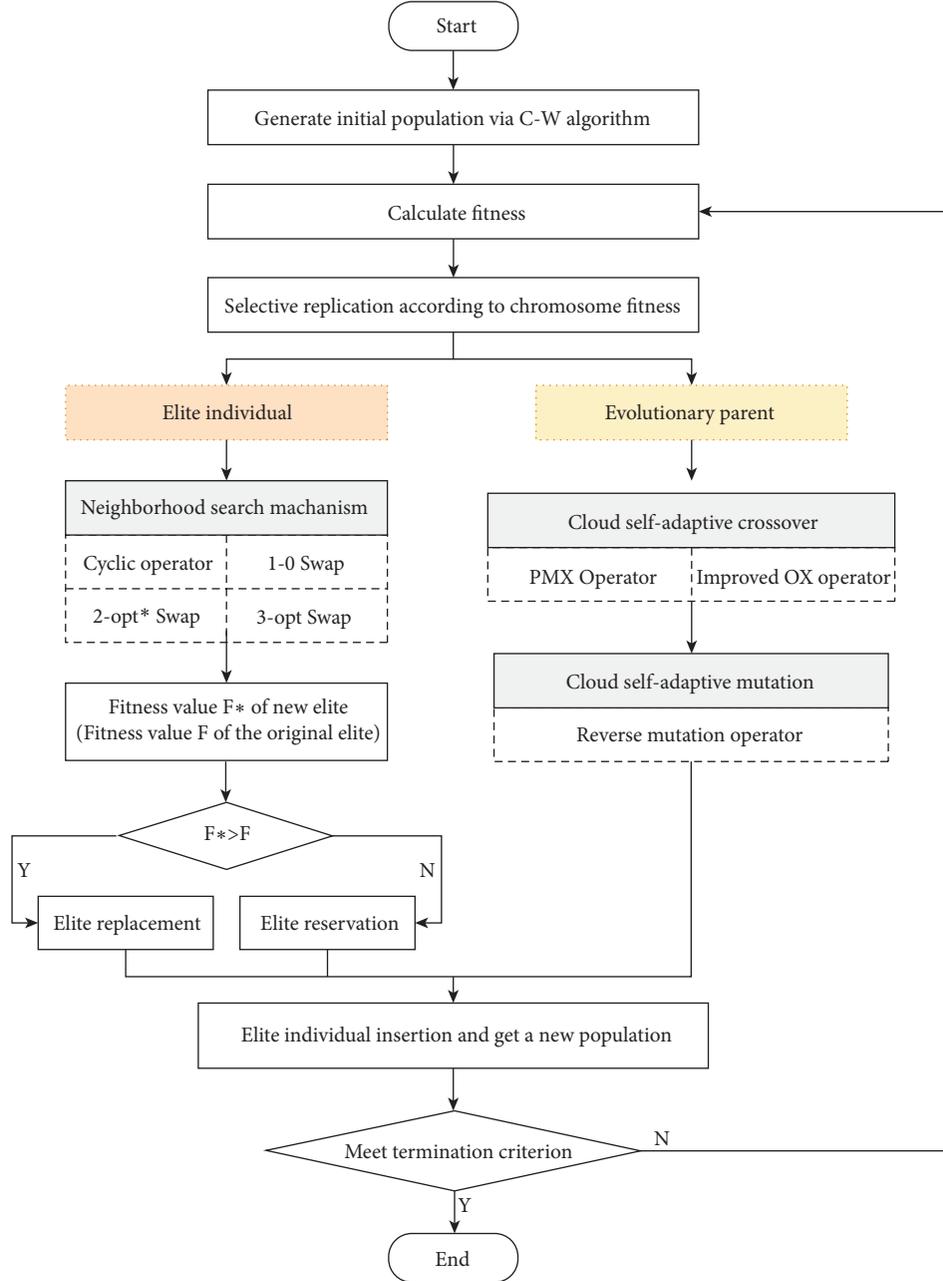


FIGURE 6: HAGA_ELS algorithmic flow.

$$\begin{aligned}
 P_c &= \begin{cases} k_1 e^{-\left(\frac{(f-f_{avg})^2}{2\text{Norm}[f_{\max}-f_{avg}/C_1, f_{\max}-f_{avg}/C_1 C_2]}\right)}, & f \geq f_{avg}, \\ k_3, & f \leq f_{avg}, \end{cases} \\
 P_m &= \begin{cases} k_2 e^{-\left(\frac{(f'-f_{avg})^2}{2\text{Norm}[f_{\max}-f_{avg}/C_3, f_{\max}-f_{avg}/C_3 C_4]}\right)}, & f' \geq f_{avg}, \\ k_4, & f' \leq f_{avg}, \end{cases}
 \end{aligned} \tag{21}$$

where f is the larger fitness value among the crossover individuals, f' is the fitness value of the mutant individual, f_{\max} is the maximum fitness value in the population, and f_{avg} is the average fitness value. Norm is a normal random number generator. $C_1 \sim C_4$ are the control coefficients

($C_1 = C_3 = (T+1)/6N$, $C_2 = C_4 = 15 - (T - N/2)^2$), and $k_1 \sim k_4$ are constants in the range of $[0, 1]$. The control coefficient in the algorithm is appropriately adjusted in accordance with evolutionary generation T , search accuracy, and other factors.

4.4.1. Hybrid Crossover Operator. To avoid the ineffectiveness of crossover caused by the same crossover genomes, two hybrid crossover operators are used for crossover operations to increase individual diversity. When the crossover genomes are different, the Partial-Mapped Crossover (PMX) crossover operator is employed to exchange the crossover genomes of parents A and B, and a mapping relationship is established. The conflicting genes in offspring A and B are replaced in accordance with the mapping relationship to ensure that the coding of offspring obtained by crossover is nonrepetitive. When the cross genomes are the same, we adopt the improved Order Crossover (OX) crossover operator in the work of Zhang and Li [34]. The crossover regions of parents A and B are placed at the front and back of the other parent, respectively, and the genes in parents A and B that are duplicated in the inserted cross genomes are deleted to form two new children. The crossover process is shown in Figure 7.

4.4.2. Reverse Mutation Operator. In this study, the mutation operation is improved by the reversal operator adopted by Barma et al. [36]. Two nodes in the mutated chromosome are randomly selected and then inserted reversely after the selected two nodes break. If the fitness value of the new chromosome after reversal mutation is better than the fitness value of the original chromosome, the reversal mutation operation is effective. The new chromosome after mutation is retained, and the original chromosome is deleted so that the individual is always mutated in an improved direction. The specific mutation operation is shown in Figure 8.

4.5. Elite Local Search. To avoid the premature convergence by the traditional elite retention strategy, the neighborhood search operator is designed to further optimize elite individuals and improve the local optimum caused by direct retention of elite individuals. To search the new solution space effectively, the four neighborhood operators in the study of Xu et al. [37] are employed to search the elite individuals. These four operators are cyclic operator, 1-0 exchange, 2-opt* exchange, and 3-opt exchange, respectively. If the new individual after the neighborhood search is better than the original elite individual, then the elite optimization operation is effective. Afterward, the improved elite individual is recombined with the individuals after crossover and mutation operations to ensure that the optimal individual of the new population is always better than or equal to the optimal individual of the previous generation (see Algorithm 2).

- (1) Cyclic operator: two node positions in a subrouting are randomly selected, and the genes between the two node positions are circularly moved. As shown in Figure 9(a), the positions of nodes 6 and 5 are selected, and the intermediate part $\underline{6} \underline{2} \underline{5}$ is sequentially moved forward, i.e., $\underline{2} \underline{5} \underline{6}$.
- (2) 1-0 exchange: it is also known as swap move, which means inserting a node in a subrouting into another

subrouting. The exchange operation is illustrated in Figure 9(b).

- (3) 2-opt* exchange: a node position in the two sub-routings is selected, and the tails of the two sub-routings are exchanged in accordance with the node position, as shown in Figure 9(c).
- (4) 3-opt exchange: four continuous points $(i, i + 1, j, j + 1)$ in a certain subrouting are transformed into $(i + 1, i, j + 1, j)$, as shown in Figure 9(d).

4.6. Termination Condition. As the process of the meta-heuristic algorithm is uncertain, it is difficult to determine a number of terminal iterations to stop the calculation. Considering the change of the fitness value would get smaller, it is reasonable to stop the process when the best solution is not improved in successive generations Ω .

5. Computational Experiment and Analyses

In this section, the experimental design is described in Section 5.1. Parameter setting is provided in Section 5.2. The performance of the algorithm is evaluated in Section 5.3. In Sections 5.4 and 5.5, three speed patterns are compared and the sensitivity of the model is assessed. All experiments are implemented in Matlab R2018b in a PC with Intel Core i5 2.4 GHz processor, 8 GB RAM, and Windows 10 professional operating system.

5.1. Experimental Design. Nine benchmark instances are used to assess the effectiveness of HAGA_ELS (see, e.g., Augerat et al. [38], Breedam [39], and Christofides and Eilon [40]). In these instances, the geographical distribution of customers in E instances is randomly and evenly distributed, the geographical location of customers in A instances has semiclustering characteristics, and the geographical location of customers in B instances is in accordance with the clustering characteristics. The time windows are set to [60,600], which allows early arrivals but does not allow delayed arrivals. n represents the number of nodes in the instances, NV represents the number of vehicles used, Capacity represents the maximum load of vehicles, and r_0 represents the ratio of the total demand to the total capacity. Table 4 lists the characteristics and basic information of the instances.

The instance for sensitivity analysis is designed following specific distributions. $n(n \in \{10, 20, 40\})$ customers are generated randomly, where the location (x, y) obeys the uniform distribution of $U(-50, 50)$, customer demand D_i obeys the uniform distribution of $U(0, 10)$, and the customer time window (ET_i, LT_i) is set to (60,600). On the basis of travel speed function $v(t)$ of equation (1), travel speed v_{ij}^t of different arcs in different periods is randomly generated so that v_{ij}^t obeys $Normal(v(t), \sigma^2)$.

5.2. Parameter Setting. The parameters, for instance, are set as follows: fixed vehicle cost $f_1 = 110$ yuan/vehicle; time penalty cost $\beta_1 = 50$ yuan/hour and $\beta_2 = 100$ yuan/hour; fuel

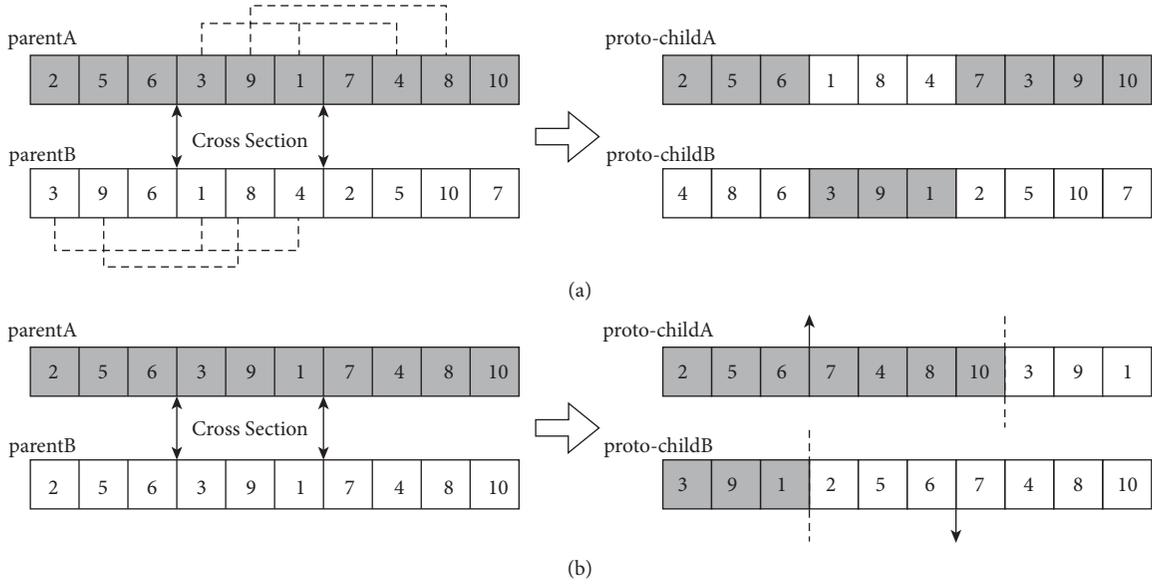


FIGURE 7: Crossover operator. (a) PMX crossover operator. (b) Improved OX crossover operator.

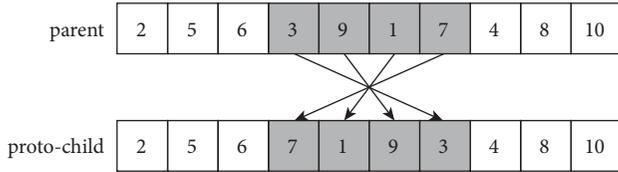


FIGURE 8: Reverse mutation operator.

cost rate $f_2 = 0.8$ yuan/L; fuel consumption parameters are set as $a_1 = 0.111$, $a_2 = 36.0302$, and $a_3 = 0.000059$ [41]; fuel-to- CO_2 coefficient $\rho = 2.61$ kg/L; and carbon tax = 1.5 yuan/t. The parameters for the CMEM model are given in Appendix A.

The parameters to be determined for the algorithm include the cloud model $k_1 \sim k_4$, the population size N , and successive generations Ω . For the parameters $k_1 \sim k_4$ and N , the trial and error are executed to achieve the best configuration of HAGA_ELS in a set of experiments. We test the population size of 10, 50, 100, and 200 and use crossover related probabilities k_1, k_3 ranging from 0 to 1 with an increment of 0.5 and mutation related probabilities k_2, k_4 ranging from 0 to 1 with an increment of 0.5. Thus, we have a 3D matrix holding all the possible combinations. Because finding the appropriate configuration over all the problem instances is time consuming, we opt to perform on a particular instance B_n78_k10 with maximum nodes. The algorithm has been run 10 times for each combination within fixed generations of 500. Detailed results are coming in Appendix B. The results show that when the population size increases to 200, the solution does not improve significantly in general, but the CPU time increases by 2-3 times, so the population size could be defined less than 200. We also find that in most experiments the difference between solutions in last consecutive iterations of 50 final is insignificant. From all

the test runs, the best configuration has a population size of 100, crossover related probabilities of 1 ($k_1 = k_3 = 1$), mutation related probabilities of 0.5 ($k_2 = k_4 = 0.5$), and successive generations of 50 (for HAGA_ELS).

5.3. *Evaluating the Performance of HAGA_ELS on TDVRPTW.* To evaluate the proposed algorithm HAGA_ELS, two GA-based algorithms and two classical constructive heuristics are used for comparison. Classical constructive heuristics, sequential insertion algorithm (SI), and Clarke and Wright Saving (CWS) heuristic are, respectively, discussed in Avdoshin and Beresneva [42] and Marković et al. [33]. GA-based algorithms, standard adaptive genetic algorithm (SAGA), and elite strategy genetic algorithm (EGA) were coded as follows:

- (1) SAGA is the same as HAGA_ELS, except for the elite local search described in Section 4.5.
- (2) EGA adopts elite local search in HAGA_ELS, but the crossover and mutation are the same as the standard genetic algorithm (SGA).

Table 5 lists the results obtained by the three metaheuristic algorithms. Each instance is tested on the three metaheuristic algorithms 10 times. Avg represents the average value of 10 calculation results, CV represents the deviation coefficient ($CV = ST D / Mean$) under each algorithm. $Gap_s\%$ and $Gap_e\%$, respectively, represent the optimized percentage of the average values of SAGA and EGA, for example, $Gap_s\% = (SAGA_{Avg} - HAGA_ELS_{Avg}) / HAGA_ELS_{Avg} * 100$.

As shown in Table 5, the average optimization percentage compared with SAGA and EGA is 2.59% and 1.67%, respectively. The average results of the nine instances obtained by HAGA_ELS are better than those obtained by the

```

(1)Input  $X_{Good}$ ,  $fitness$ 
(2)For  $i = 1$  to  $size(X_{Good}, 1)$  do begin
(3)  $X_{Good}' \leftarrow X_{Good}(i, :)$  //Update  $X_{Good}'$  after each step of operation
    //Cyclic operator. Choosing  $r_1$  and  $r_2$  at random, which are in the same subrouting
(4)  $New\_X_{Good} d(r_2) \leftarrow X_{Good} d'(r_1)$ 
(5) For  $k = r_1$  to  $r_2 - 1$  do begin
(6)  $New\_X_{Good} d(k) \leftarrow X_{Good} d'(k + 1)$ 
(7) End for
    //1-0 swap. Choosing  $r_1$  and  $r_2$  in different sub routings
(8) Remove  $New\_X_{Good} d(i, r_1)$ 
(9)  $New\_X_{Good} d(i, r_2) \leftarrow X_{Good} d'(i, r_1)$ ;  $New\_X_{Good} d(i, r_2 + 1 : end) \leftarrow X_{Good} d'(i, r_2 : end)$ 
    //2-opt swap. Choosing  $r_1$  and  $r_2$  in different sub routings
(10)  $New\_X_{Good} d(i, r_1 : where\ node\ is\ 0) \leftarrow X_{Good} d'(i, r_2\ where\ node\ is\ 0)$ 
(11) Similarly for the other subrouting where  $r_2$  is located
    //3-opt swap. Choosing  $r_1$  and  $r_2$  in the same subrouting
(12)  $New\_X_{Good} d(i, r_1) \leftarrow X_{Good} d'(i, r_1 + 1)$ ;  $New\_X_{Good} d(i, r_1 + 1) \leftarrow X_{Good} d'(i, r_1)$ 
(13)  $New\_X_{Good} d(i, r_2) \leftarrow X_{Good} d'(i, r_2 + 1)$ ;  $New\_X_{Good} d(i, r_2 + 1) \leftarrow X_{Good} d'(i, r_2)$ 
    //Judging the effectiveness of elite evolution
(14)  $new\_fitness \leftarrow F(New\_X_{Good} d)$ 
(15) If  $new\_fitness > fitness(i)$ 
(16)  $X_{Good} d(i, : ) \leftarrow New\_X_{Good} d$ 
(17) End if
(18)End for
    
```

ALGORITHM 2: Procedure of ELS.

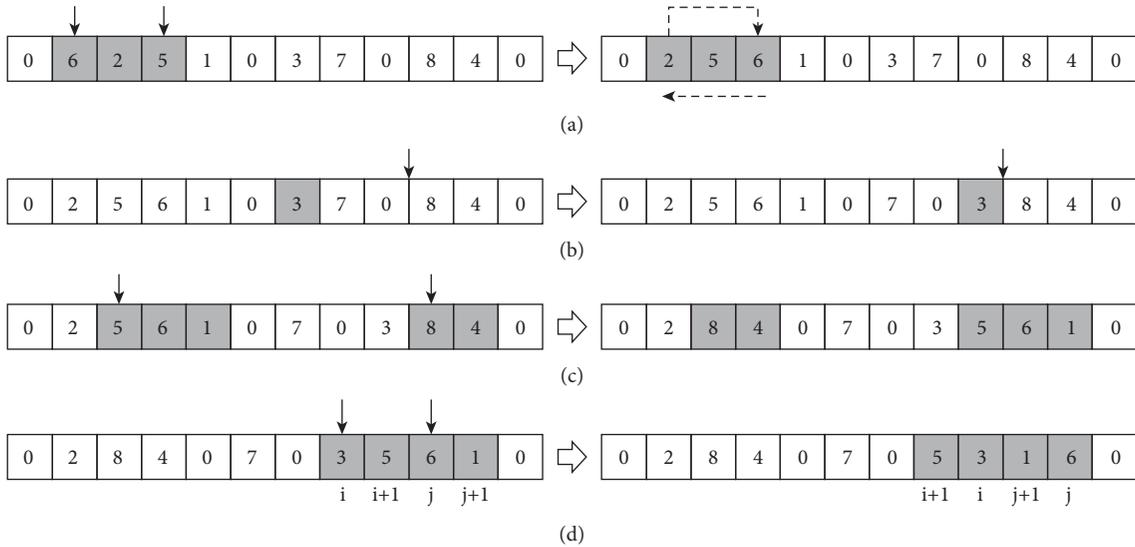


FIGURE 9: Elite neighborhood search operation. (a) Cyclic operator. (b) 1-0 swap. (c) 2-opt* swap. (d) 3-opt swap.

two other algorithms. HAGA_ELS has an average deviation coefficient of 4.91% for nine instances, which is much smaller than other two algorithms compared to 6.32% of SAGA and 5.98% of EGA. The result shows that HAGA_ELS is rather effective and stable.

Table 6 lists the results of the three metaheuristic algorithms and the known optimal solution (OPT). The iterations represent the number of average iterations to obtain the optimal solution in 10 experiments, and $Gap_h\%$, $Gap_s\%$, and $Gap_e\%$, respectively, represent the gap between the solution of three algorithms and the known

optimal solution. For example, $Gap_h\% = (HAGA_ELS_{Best} - opt) / opt * 100$. As shown in Table 6, the average gap between HAGA_ELS and OPT is 3.87%, while the average gap of SAGA and EGA is 6.61% and 5.54%, respectively. Given that a time window is added to the calculation instance in this study, the deviation is still acceptable. Therefore, HAGA_ELS has firm solving capability for instances with different characteristics and sizes.

Figure 10 shows the convergence graph of the three metaheuristic algorithms for large instances. Figures 10(a)–10(c) response to E_n51_k5, A_n65_k9, and B_n78_k10

TABLE 4: Characteristics of benchmark instances.

No.	Instances	n	NV	Capacity	r_0
1	E_n33_k4	33	4	8000	0.918
2	E_n45_k4	45	4	2010	0.898
3	E_n51_k5	51	5	160	0.971
4	A_n37_k6	37	6	100	0.950
5	A_n55_k9	55	9	100	0.932
6	A_n65_k9	65	9	100	0.974
7	B_n35_k5	35	5	100	0.874
8	B_n56_k7	56	7	100	0.880
9	B_n78_k10	78	10	100	0.937

TABLE 5: Comparison of three GA-based algorithms.

Instances	HAGA_ELS		SAGA		EGA		Gap_s%	Gap_e%
	Avg	CV	Avg	CV	Avg	CV		
E_n33_k4	863.56	4.22	876.98	3.60	879.91	2.44	1.55	1.89
E_n45_k4	740.76	3.02	755.05	7.24	753.27	6.23	1.93	1.69
E_n51_k5	579.68	6.32	610.58	8.59	597.28	7.37	5.33	3.04
A_n37_k6	986.34	4.03	1009.91	5.40	1004.58	5.83	2.39	1.85
A_n55_k9	1125.33	8.14	1154.33	5.97	1147.96	5.17	2.58	2.01
A_n65_k9	1276.16	5.36	1330.86	6.82	1304.31	8.67	4.29	2.21
B_n35_k5	962.90	2.79	966.10	6.02	965.39	3.90	0.33	0.26
B_n56_k7	724.17	3.96	736.26	7.02	730.38	6.57	1.67	0.86
B_n78_k10	1344.55	6.32	1388.13	6.26	1360.61	7.62	3.24	1.19
Average	955.94	4.91	980.91	6.32	971.52	5.98	2.59	1.67

TABLE 6: Comparison of optimal solutions of GA-based algorithms.

Instances	OPT	HAGA_ELS		SAGA		EGA		Gap_h%	Gap_s%	Gap_e%
		Best	Iterations	Best	Iterations	Best	Iterations			
E_n33_k4	835	853.44	208	871.79	201	874.73	173	2.21	4.41	4.76
E_n45_k4	728	737.18	453	747.40	420	735.44	448	1.26	2.66	1.02
E_n51_k5	521	575.58	375	601.58	342	590.74	311	10.48	15.47	13.39
A_n37_k6	949	979.71	202	1001.31	168	1001.31	134	3.24	5.51	5.51
A_n55_k9	1073	1114.10	248	1145.74	207	1143.64	189	3.83	6.78	6.58
A_n65_k9	1174	1262.02	417	1322.44	329	1286.07	270	7.50	12.64	9.55
B_n35_k5	955	957.11	204	959.08	197	961.72	176	0.22	0.43	0.70
B_n56_k7	707	715.06	295	728.36	289	721.13	280	1.14	3.02	2.00
B_n78_k10	1266	1328.63	525	1373.95	381	1346.38	364	4.95	8.53	6.35
Average	912	946.98	325	972.41	282	962.35	260	3.87	6.61	5.54

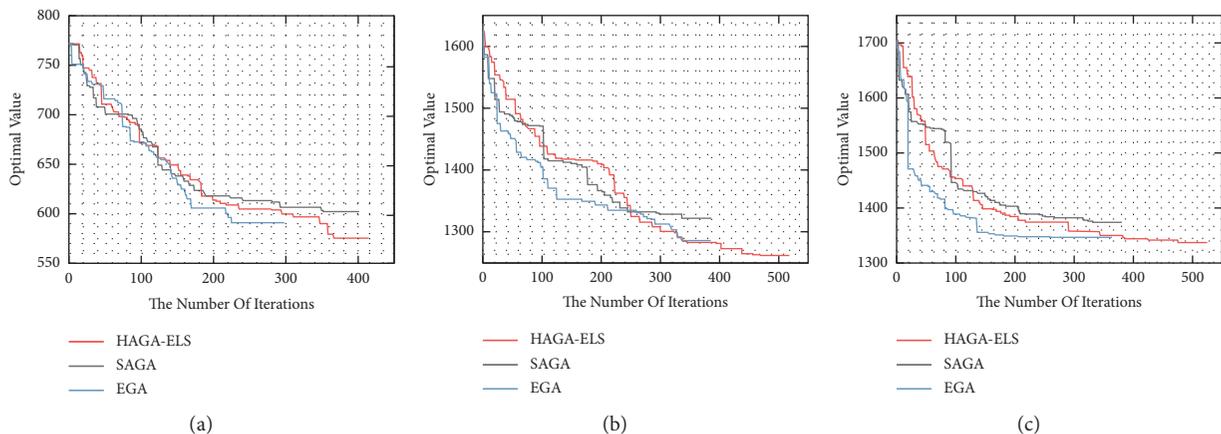


FIGURE 10: Convergence comparison.

instances, respectively. According to the graph, HAGA_ELS can obtain more qualified solution within more iterations.

The computational results of HAGA_ELS and classical constructive heuristics are shown in Table 7. $Gap_h\%$ is much smaller than $Gap_{si}\%$ and $Gap_{cws}\%$, which means HAGA_ELS performs better than SI and CWS in any instances. The average gap of HAGA_ELS is 3.87%, while the average gaps of SI and CWS are all over 15%. Thus, compared with constructive heuristics, HAGA_ELS has ability to obtain the better solution.

5.4. Comparison of Three Patterns. The second experiment is a small-size experiment with 10 customers to verify the validity of the model, and it is carried out under three patterns (T&S, T, and S). Each experiment is run 10 times. Table 8 shows a cost comparison for the three patterns. Given that the fixed costs are the same, they are omitted in the Table 8, in which $Gap_T\%$ and $Gap_S\%$, respectively, represent the difference between T&S pattern and T and or S patterns. The optimization percentage of each kind of costs between S pattern and T&S pattern is calculated as $Gap\% = (cost' - cost)/cost$. Figure 11 shows the distribution scheme under the three patterns with the corresponding carbon emission costs and average travel speed. Figure 12 shows the travel speed of each road section under the three patterns.

Table 8 shows the comparison of costs for the three patterns. Although the transportation cost under the T&S pattern is slightly higher, the total cost of T and S patterns is, respectively, 10.51% and 26.96% higher than that of the T&S pattern, mainly because the time penalty cost is greatly reduced under the T&S pattern. This finding shows that applying the spatiotemporal-varying speed facilitates to reduce the default of the customer required.

Figure 11 shows the distribution scheme under the three patterns, and Figure 12 shows the average travel speed of each arc under the three patterns. As shown in Figure 11, the scheme obtained by the T&S pattern has the lowest carbon emission cost and the fastest average travel speed. The average travel speeds under T&S, T, and S patterns are 26.77, 24.38, and 23.17 km/h, respectively. Figure 12 indicates that under the T&S pattern, the travel routing can avoid traffic congestion, so that the distribution has a high travel speed. The average travel speed of arcs can be seen in Figure 11. The maximum travel speed under the three patterns does not reach the inflection point of the carbon emission cost function (Equation (9)). Hence, the carbon emission cost drops with the increase in speed. The T&S pattern improves travel speed by avoiding congestion, thus reducing the carbon emission cost.

As shown in Figure 12, the T&S pattern mainly selects the route to avoid congestion through the node sequence. Thus, its travel speed is generally at a high level. Detailed analysis of Figure 11 indicates that the T&S pattern avoids the congestion in the fourth arc (a→c) and the fifth arc (c→b). In these arcs, the order of node arrival is c→a→b and b→a→c in T and S patterns, respectively, and the order of node arrival is changed to a→c→b in the

T&S pattern. This finding shows that the T&S pattern can help vehicles plan the node arrival sequence and avoid the traffic congestion, thus reducing carbon emission cost and improving distribution speed.

5.5. Comparison of Different Factor Impacts on Three Patterns. To assess the effectiveness of the T&S pattern further, we simulate and analyze different customer sizes in Section 5.5.1 and different speed distributions in Section 5.5.2.

5.5.1. Analysis of Different Customer Size. We select three customer groups with different sizes ($n = 10$, $n = 20$, and $n = 40$) for testing and adopt three dynamic speed patterns, namely, T&S, T, and S. Nine groups of experiments are performed, and each group of experiments is run 10 times. We obtain the best results of the three dynamic speed patterns with different sizes and compare them in terms of cost, travel time, and speed. Under $\sigma = 5$, we generate speed data that obey a normal distribution under different arcs in the same period to ensure a random difference of speed in road network. At the same time, the travel speed on each arc obeys the function law of the period, which reflects the real traffic network to a certain extent.

We report the computational results of the nine groups of experiments in Table 9, in which $Gap_T\% = |Item_T - Item_{T\&S}|/Item_{T\&S} \times 100$, $Gap_S\% = |Item_S - Item_{T\&S}|/Item_{T\&S} \times 100$. $Gap_T\%$ and $Gap_S\%$ are greater than zero under different customer sizes n ; that is, the best result is obtained by using the spatiotemporal travel speed pattern (T&S). The $Gap_T\%$ in cost is over 10%, and that of $Gap_S\%$ is over 15%. The routing optimization under the T&S pattern is superior to those under T and S patterns in terms of cost.

Figure 13(a) shows the total cost deviation between T&S pattern and T (or S) pattern under different customer sizes, and Figure 13(b) presents the total travel time deviation between T&S pattern and T (or S) pattern. With the increase in customer size, the total cost deviation and total travel time deviation also increase. Table 9 also indicates that the average optimization percentage $Gap\%$ appears an upward trend with customers increasing. $Gap_T\%$ increases from 7.33% to 12.92%, and $Gap_S\%$ increases from 15.81% to 18.39%. Therefore, with the increase in customer size, the T&S pattern performs much better than T and S patterns by reducing the distribution cost and improving the distribution efficiency.

5.5.2. Analysis of Different Speed Deviation. Three different standard deviations of speed distribution ($\sigma = 10$, $\sigma = 20$, and $\sigma = 40$) are selected to randomly generate a spatiotemporal speed matrix. The travel speed v_{ij}^l of different arcs in different periods obeys the normal distribution of $Normal(v(t), \sigma^2)$. ($v(t)$ is the time travel speed function (equation (1)). By changing the variance of the normal distribution, the difference of travel speed in arcs can be adjusted. The greater the variance is, the greater the difference in travel speed is among different arcs. In the

TABLE 7: Comparison of optimal solutions of HAGA_ELS and constructive heuristics.

Instances	OPT	HAGA-ELS	SI	CWS	Gap-h%	Gap-si%	Gap-cws%
E_n33_k4	835	853.44	888.82	937.77	2.21	6.45	12.31
E_n45_k4	728	737.18	881.00	964.60	1.26	21.02	32.50
E_n51_k5	521	575.58	638.28	771.85	10.48	22.51	48.15
A_n37_k6	949	979.71	1175.90	1065.50	3.24	23.91	12.28
A_n55_k9	1073	1114.10	1394.10	1257.70	3.83	29.93	17.21
A_n65_k9	1174	1262.02	1571.70	1639.60	7.50	33.88	39.66
B_n35_k5	955	957.11	1007.50	1245.10	0.22	5.50	30.38
B_n56_k7	707	715.06	869.39	861.50	1.14	22.97	21.85
B_n78_k10	1266	1328.63	1424.40	1704.20	4.95	12.51	34.61
Average	912	946.98	1094.57	1160.87	3.87	19.85	27.66

TABLE 8: Cost item deviations of the different patterns at a small-size instance.

Cost item	T&S	T	S	Gap_T%	Gap_S%
Total cost	670.74	741.21	851.54	10.51	26.96
Transportation cost	203.66	197.52	191.27	-3.01	-6.08
Carbon cost	170.58	182.19	199.93	6.81	17.21
Time penalty cost	76.49	141.5	240.34	84.99	214.21

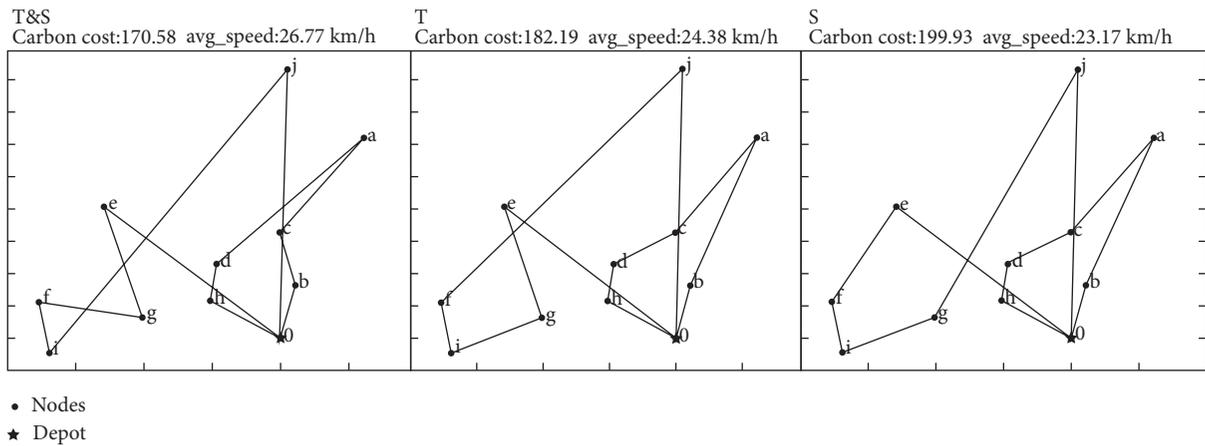


FIGURE 11: Comparison of paths under three patterns.

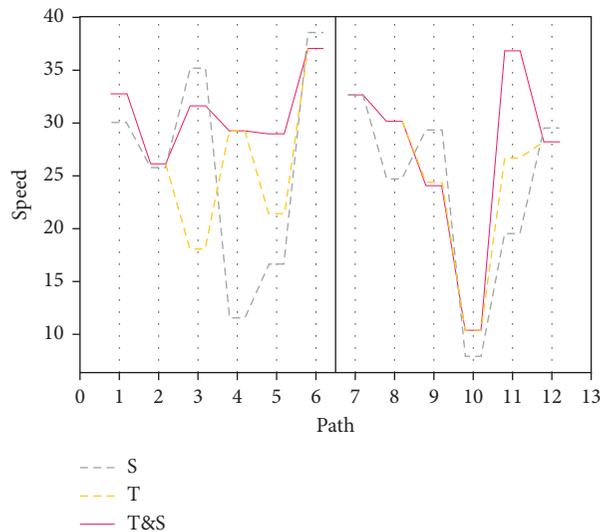


FIGURE 12: Speeds of small-size instance under three patterns.

TABLE 9: Numerical results of different patterns under different sizes.

Customer size	Item	T&S	T	S	Gap_T%	Gap_S%
$n = 10$	Cost	670.74	741.21	851.54	10.51	26.96
	Travel time	1168.38	1198.30	1250.51	2.56	7.03
	Average speed	26.77	24.38	23.17	8.93	13.45
	Average	—	—	—	7.33	15.81
$n = 20$	Cost	1026.02	1226.8	1222.52	19.57	19.15
	Travel time	2245.09	2430.23	2456.08	8.25	9.40
	Average speed	28.39	25.80	25.30	9.12	10.88
	Average	—	—	—	12.31	13.14
$n = 40$	Cost	2730.46	3254.57	3650.34	19.19	33.69
	Travel time	5332.41	5813.8	5855.05	9.03	9.80
	Average speed	31.03	27.76	27.41	10.54	11.67
	Average	—	—	—	12.92	18.39

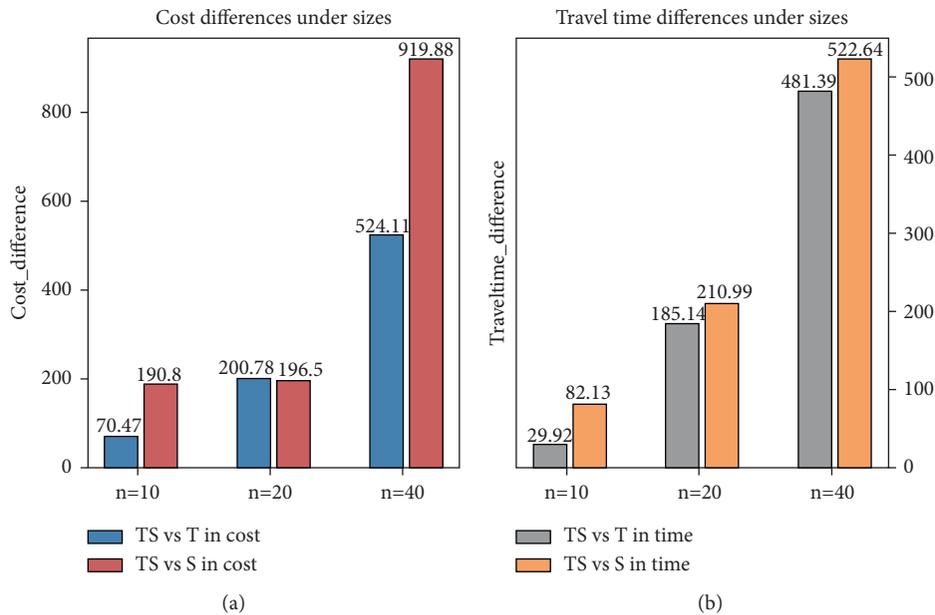


FIGURE 13: Total cost and total time deviation of different patterns under different sizes.

same manner, three different spatiotemporal-varying speed matrices are tested with the three dynamic speed patterns. To avoid the compound effect caused by the change in customer group size, we set the customer group size to be constant ($n = 20$), and each group of experiments is run 10 times to obtain the optimal results of three speed patterns under different speed matrices. A comparative analysis is performed in terms of cost, travel time, and speed.

The numerical results of each group of experiments are shown in Table 10, which illustrates that $Gap_T\%$ and $Gap_S\%$ are greater than or equal to 0 under the velocity matrix generated by different variances σ^2 . Thus, the spatiotemporal speed pattern (T&S) is the best method under the velocity matrix generated by different variances. When the variance of the distribution is small, that is, $\sigma = 2$, the difference in spatial speed is small. In this case, the running results of the T&S and T patterns are consistent, so there is no obvious difference whether speed spatiality is

considered or not. However, with the increase in speed distribution variance σ^2 , $Gap_T\%$ increases continuously and reaches at 17.53% under $\sigma = 8$. Meanwhile, $Gap_S\%$ increases from 3.55% to 17.53%. When the difference of spatial speed amplifies, the T&S pattern becomes superior to T and S patterns, and the deficiency of the T pattern that does not consider speed spatiality becomes increasingly apparent.

Figure 14(a) shows a comparison of the total cost of the three patterns under different speed distribution variances. In the same customer group, when the speed difference changes, the total cost of distribution of the T&S pattern changes slightly, but the total cost of T and S patterns increases drastically. The total cost deviations keep growing, $Gap_T\%$ increases from 0.00% to 25.31%, and $Gap_S\%$ increases from 3.14% to 25.31%. The change in the total cost of carbon emissions is similar to the change trend of the total cost, as indicated in Figure 14(b). With the increase in the speed difference, the carbon cost

TABLE 10: Numerical results of different patterns under different standard deviation of speeds.

Standard deviation	Item	T&S	T	S	Gap_T%	Gap_S%
$\sigma = 2$	Cost	998.46	998.46	1029.80	0.00	3.14
	Travel time	2236.65	2236.65	2361.43	0.00	5.58
	Average speed	28.57	28.57	28.02	0.00	1.93
	Average	—	—	—	0.00	3.55
$\sigma = 5$	Cost	1026.02	1226.8	1222.52	19.57	19.15
	Travel time	2245.09	2430.23	2456.08	8.25	9.40
	Average speed	28.39	25.80	25.30	9.12	10.88
	Average	—	—	—	12.31	13.14
$\sigma = 8$	Cost	1017.01	1274.39	1274.39	25.31	25.31
	Travel time	2202.18	2467.53	2467.53	12.05	12.05
	Average speed	31.19	26.44	26.44	15.23	15.23
	Average	—	—	—	17.53	17.53

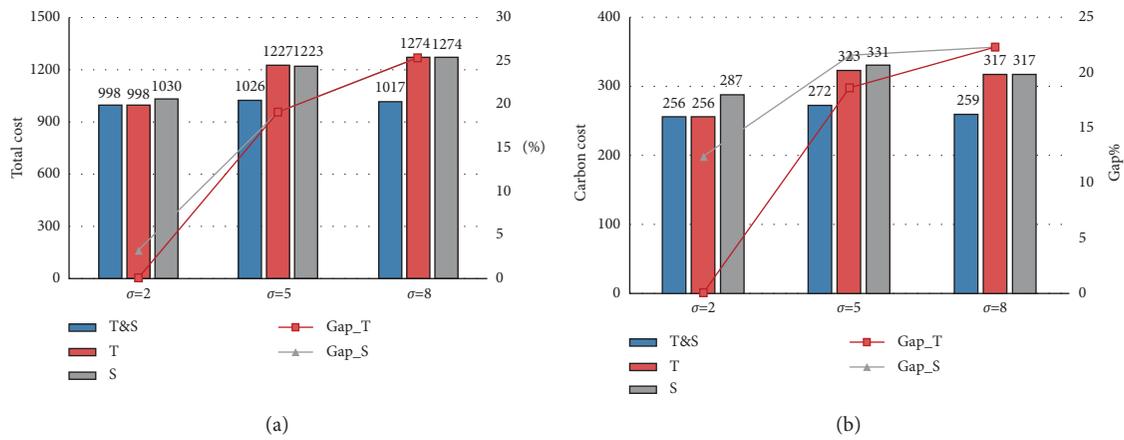


FIGURE 14: Cost item differences of different patterns under different standard deviation.

TABLE 11: CMEM model parameters.

Group	Notation	Description	Value
Vehicle-related parameters	$A_f (m^2)$	Windward area of vehicle	5.03
	ε	Vehicle transmission coefficient	0.4
	ω	Efficiency parameters of diesel engine	0.9
	k	Engine friction coefficient	0.2
	$N (rev/s)$	Engine speed	38.33
	$V (l)$	Engine displacement	4.6
	ρ_0	Fuel consumption coefficient when the vehicle is unloaded	0.169
	ρ_{max}	Fuel consumption coefficient when the vehicle is fully loaded	0.377
	Q_0	Car capacity	20
Other parameters	δ	Fuel-air mass ratio	1
	κ	Typical diesel calorific value	44
	φ	Fuel conversion coefficient	737

TABLE 12: Algorithm performance for parameter combinations.

Parameter combinations	Computational time (s)				Optimal solution			
	10	50	100	200	10	50	100	200
(0,0,0)	1.99	11.77	20.04	57.68	1481.44	1413.69	1389.69	1391.41
(0,0,0.5)	1.99	14.16	21.15	60.65	1416.25	1394.56	1365.38	1366.13
(0,0,1.0)	2.03	12.12	23.02	59.43	1539.01	1446.55	1387.92	1369.75
(0.5,0,0)	2.01	13.54	24.88	60.97	1483.91	1421.87	1367.36	1361.67
(0.5,0,5)	1.98	14.47	22.46	59.99	1429.89	1365.41	1353.71	1347.60
(0.5,1,0)	2.09	11.93	25.51	61.43	1557.80	1469.35	1383.04	1379.74
(1,0,0)	2.07	11.85	23.57	61.97	1507.70	1435.00	1357.79	1349.13
(1,0,0.5)	2.12	12.07	23.70	60.98	1444.75	1349.81	1342.85	1340.74
(1,0,1.0)	2.11	13.20	26.23	64.31	1623.76	1470.49	1428.64	1416.77

deviation $Gap_T\%$ and $Gap_S\%$ also increases. The carbon cost of the T&S pattern is always lower than those of T and S patterns, and the greater σ is, the greater the advantage of the T&S pattern is. According to Figure 14 and Table 10, compared with T and S patterns, the T&S pattern can effectively reduce the carbon emission cost and improve the timeliness of distribution.

6. Conclusion

Complex traffic networks and ever-changing traffic flow in cities increase the uncertainty in urban logistics distribution businesses. Route optimization considering traffic changes has become an important measure to reduce the cost and carbon emission of freight enterprises. In this study, we establish a mixed-integer nonlinear programming model that considers carbon emission and propose a spatiotemporal-dependent travel time calculation method. A hybrid adaptive genetic algorithm is developed, in which an elite neighborhood search operator is applied to ensure the obtainment of the best solution and to increase individual diversity, thus avoiding falling into the local optimum. The effectiveness of this algorithm is evaluated by comparing it with other algorithms on benchmark instances. A sensitivity analysis based on different sizes and speed variances assesses the effectiveness of this pattern. The following conclusions are drawn:

- (1) Compared with previous algorithms, HAGA_ELS can obtain a high-quality solution. It can avoid falling into the local optimal solution and the premature problem.
- (2) Compared with those patterns that only consider time or space dependence of speed, it will obtain more reasonable routings that consider spatiotemporal changes in speed. The rationality specifically reflects in less time penalty cost and carbon emission cost.
- (3) The customer size and speed differences have a significant influence on the effect of spatiotemporal pattern. With the increase of customer size or speed difference, this pattern will bring lower total cost and shorter travel time compared with other patterns that only consider time or space dependence of speed.

Based on the comparison of three patterns, we give the following recommendations to freight enterprises: freight enterprises should make use of the intelligent transportation system to scheme preoptimized route, which provides the possibility of obtaining traffic information about the road network. Especially in the case of scattered customers resulting in considerable speed difference, preoptimized scheme under spatiotemporal traffic information obtains performance gain.

The future research will focus on speeding up the algorithm. For further research, one direction is to speed up the algorithm to solve the large size problem more efficiently.

Appendix

(A). CMEM Model Parameters in Table 11

(B). Algorithm Performance for Various Population Sizes

In Table 12, the obtained average optimal solution values and the computational time for considered parameter combinations are presented. Parameter combinations consists of the population size N and the cloud model settings (k_1, k_2) or (k_3, k_4) .

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Ziqi Liu and Yeping Chen contributed to the work equally.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2018YFB1403100) and the Humanities and Social Sciences Youth Foundation of Ministry of Education of China (19YJC630079).

References

- [1] Ö. N. Çam and H. K. Sezen, "Linear programming formulation for vehicle routing problem which is minimized idle time," *Decision Making: Applications in Management and Engineering*, vol. 3, no. 1, pp. 22–29, 2020.
- [2] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.
- [3] C. Lin, K. L. Choy, G. T. S. Ho, S. H. Chung, and H. Y. Lam, "Survey of green vehicle routing problem: past and future trends," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1118–1138, 2014.
- [4] M. Turkensteen, "The accuracy of carbon emission and fuel consumption computations in green vehicle routing," *European Journal of Operational Research*, vol. 262, no. 2, pp. 647–659, 2017.
- [5] T. Bektaş and G. Laporte, "The pollution-routing problem," *Transportation Research Part B: Methodological*, vol. 45, no. 8, pp. 1232–1250, 2011.
- [6] C. Liu, G. Kou, X. Zhou, Y. Peng, H. Sheng, and F. E. Alsaadi, "Time-dependent vehicle routing problem with time windows of city logistics with a congestion avoidance approach," *Knowledge-Based Systems*, vol. 188, Article ID 104813, 2020.
- [7] A. Esteves-Booth, T. Muneer, J. Kubie, and H. Kirby, "A review of vehicular emission models and driving cycles," *Proceedings of the Institution of Mechanical Engineers - Part C*:

- Journal of Mechanical Engineering Science*, vol. 216, no. 8, pp. 777–797, 2002.
- [8] R. A. Russell, “Technical note—an effective heuristic for the M-tour traveling salesman problem with some side conditions,” *Operations Research*, vol. 25, no. 3, pp. 517–524, 1977.
- [9] M. M. Solomon, *Vehicle Routing and Scheduling with Time Window Constraints: Models and Algorithms (Heuristics)*, University of Pennsylvania, Philadelphia, PA, USA, 1984.
- [10] P. Sun, L. P. Veelenturf, M. Hewitt, and T. Van Woensel, “The time-dependent pickup and delivery problem with time windows,” *Transportation Research Part B: Methodological*, vol. 116, pp. 1–24, 2018.
- [11] C. Hu, J. Lu, X. Liu, and G. Zhang, “Robust vehicle routing problem with hard time windows under demand and travel time uncertainty,” *Computers & Operations Research*, vol. 94, pp. 139–153, 2018.
- [12] M. Babaei and M. Rajabi-Bahaabadi, “School bus routing and scheduling with stochastic time-dependent travel times considering on-time arrival reliability,” *Computers & Industrial Engineering*, vol. 138, Article ID 106125, 2019.
- [13] M. Saint-Guillain, C. Paquay, and S. Limbourg, “Time-dependent stochastic vehicle routing problem with random requests: application to online police patrol management in Brussels,” *European Journal of Operational Research*, vol. 292, no. 3, pp. 869–885, 2021.
- [14] Y. Xiao and A. Konak, “The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 88, pp. 146–166, 2016.
- [15] Y. Wu and Z. Ma, “Time-dependent production-delivery problem with time windows,” *Systems Engineering—Theory & Practice*, vol. 37, no. 1, pp. 172–181, 2017.
- [16] L. Zhu and D. Hu, “Study on the vehicle routing problem considering congestion and emission factors,” *International Journal of Production Research*, vol. 57, no. 19, pp. 6115–6129, 2019.
- [17] Z. Duan, Z. Lei, and S. Sun, “Multi-objective robust optimization method for stochastic time-dependent vehicle routing problem,” *Journal of Southwest Jiaotong University*, vol. 54, no. 3, pp. 565–572, 2019.
- [18] A. Gruler, A. Perez-Navarro, L. Calvet, and A. A. Juan, “A simheuristic algorithm for time-dependent waste collection management with stochastic travel times,” *SORT-Statistics and Operations Research Transactions*, vol. 44, no. 2, pp. 285–309, 2020.
- [19] M. Gmira, M. Gendreau, A. Lodi, and J.-Y. Potvin, “Tabu search for the time-dependent vehicle routing problem with time windows on a road network,” *European Journal of Operational Research*, vol. 288, no. 1, pp. 129–140, 2021.
- [20] Z. Xu, A. Elomri, S. Pokharel, and F. Mutlu, “A model for capacitated green vehicle routing problem with the time-varying vehicle speed and soft time windows,” *Computers & Industrial Engineering*, vol. 137, Article ID 106011, 2019.
- [21] 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,” *Computers & Operations Research*, vol. 129, Article ID 105211, 2021.
- [22] A. Franceschetti, D. Honhon, T. Van Woensel, T. Bektaş, and G. Laporte, “The time-dependent pollution-routing problem,” *Transportation Research Part B: Methodological*, vol. 56, pp. 265–293, 2013.
- [23] Y.-J. Kwon, Y.-J. Choi, and D.-H. Lee, “Heterogeneous fixed fleet vehicle routing considering carbon emission,” *Transportation Research Part D: Transport and Environment*, vol. 23, pp. 81–89, 2013.
- [24] J. Qian and R. Eglese, “Fuel emissions optimization in vehicle routing problems with time-varying speeds,” *European Journal of Operational Research*, vol. 248, no. 3, pp. 840–848, 2016.
- [25] Y. Xiao and A. Konak, “A simulating annealing algorithm to solve the green vehicle routing & scheduling problem with hierarchical objectives and weighted tardiness,” *Applied Soft Computing*, vol. 34, pp. 372–388, 2015.
- [26] Y. Wang, K. Assogba, J. Fan, M. Xu, Y. Liu, and H. Wang, “Multi-depot green vehicle routing problem with shared transportation resource: integration of time-dependent speed and piecewise penalty cost,” *Journal of Cleaner Production*, vol. 232, pp. 12–29, 2019.
- [27] G. Poonthallir and R. Nadarajan, “A fuel efficient green vehicle routing problem with varying speed constraint (F-GVRP),” *Expert Systems with Applications*, vol. 100, pp. 131–144, 2018.
- [28] H. Yang, *Evolution of Urban Frequent Traffic Congestion Based on Taxi GPS Data*, Harbin Institute of Technology, Harbin, China, 2018.
- [29] A. V. Hill and W. C. Benton, “Modelling intra-city time-dependent travel speeds for vehicle scheduling problems,” *Journal of the Operational Research Society*, vol. 43, no. 4, pp. 343–351, 1992.
- [30] S. Ichoua, M. Gendreau, and J.-Y. Potvin, “Vehicle dispatching with time-dependent travel times,” *European Journal of Operational Research*, vol. 144, no. 2, pp. 379–396, 2003.
- [31] 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.
- [32] Y. Li, Z. Gao, and J. Li, “Vehicle routing optimization in urban dynamic network based on real time traffic information,” *Theory and practice of system engineering*, vol. 33, no. 7, pp. 1813–1819, 2013.
- [33] D. Marković, G. Petrović, Ž. Čojbašić, and A. Stanković, “The vehicle routing problem with stochastic demands IN an urban area—a case study,” *Facta Universitatis—Series: Mechanical Engineering*, vol. 18, no. 1, pp. 107–120, 2020.
- [34] J. Zhang and C. Li, “Research on dynamic distribution vehicle routing optimization under the influence of carbon emission,” *Chinese Journal of Management Science*, 2020.
- [35] C. H. Dai, Y. F. Zhu, and W. R. Chen, “Adaptive probabilities of crossover and mutation in genetic algorithms based on cloud model,” in *Proceedings of the 2006 IEEE Information Theory Workshop—ITW ‘06 Chengdu*, pp. 710–713, Chengdu, China, October 2006.
- [36] P. S. Barma, J. Dutta, and A. Mukherjee, “A 2-opt guided discrete antlion optimization algorithm for multi-depot vehicle routing problem,” *Decision Making: Applications in Management and Engineering*, vol. 2, no. 2, pp. 112–125, 2019.
- [37] M. Xu, G. Yu, X. Zhou, and X. Ge, “Low carbon vehicle scheduling problem and algorithm with minimum comprehensive cost,” *Computer integrated manufacturing system*, vol. 21, no. 7, pp. 1906–1914, 2015.
- [38] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi, “Computational results with a branch and cut code for the capacitated vehicle routing problem,” Technical Report 1 RR949-M, Institut National Polytechnique, Grenoble, France, 1995.
- [39] A. V. Breedam, “Comparing descent heuristics and metaheuristics for the vehicle routing problem,” *Computers & Operations Research*, vol. 28, no. 4, pp. 289–315, 2001.

- [40] N. Christofides and S. Eilon, "An algorithm for the vehicle-dispatching problem," *Journal of the Operational Research Society*, vol. 20, no. 3, pp. 309–318, 1969.
- [41] S. Li, B. Dan, and X. Ge, "Optimization model and algorithm of low carbon vehicle routing problem under multi-graph time-varying network," *Computer Integrated Manufacturing Systems*, vol. 25, no. 2, pp. 454–468, 2019.
- [42] S. M. Avdoshin and E. N. Beresneva, "Constructive heuristics for capacitated vehicle routing problem: a comparative study," *Proceedings of the Institute for System Programming of the RAS*, vol. 31, no. 3, pp. 145–156, 2019.