



Research article

Proactive dynamic vehicle routing problems considering cooperation services for the store-depot-integrated retailer

Xianlong Ge¹, Yonghong Liang¹, Yuanzhi Jin^{2,*} and Chunbing Song³

¹ School of Economics and Management, Chongqing Jiaotong University, Chongqing 400074, China

² School of Economics and Management, Guangxi University of Science and Technology, Liuzhou 545005, China

³ Research and Development Center China Merchants Chongqing Highway Engineering Testing Center Co., LTD, Chongqing 400067, China

* **Corresponding author:** Email: 47393039@qq.com; Tel: +18223599629.

Abstract: Distribution costs remain consistently high in crowded city road networks, posing challenges for traditional distribution methods in efficiently handling dynamic online customer orders. To address this issue, this paper introduces the Proactive Dynamic Vehicle Routing Problem considering Cooperation Service (PDVRPCS) model. Based on proactive prediction and order-matching strategies, the model aims to develop a cost-effective and responsive distribution system. A novel solution framework is proposed, incorporating a proactive prediction method, a matching algorithm and a hybrid Genetic Algorithm-Simulated Annealing (GA-SA) algorithm. To validate the effectiveness of the proposed model and algorithm, a case study is conducted. The experimental results demonstrate that the dynamic scheme can significantly reduce the number of vehicles required for distribution, leading to cost reduction and increased efficiency.

Keywords: proactive prediction theory; matching algorithm; vehicle routing problem; genetic algorithm; simulated annealing algorithm

1. Introduction

With the rapid development of e-commerce platforms, the current order-driven reactive distribution model is increasingly failing to meet the requirements of e-commerce logistics for

timeliness under the increasingly fierce competition. How to design an effective dynamic demand response strategy to alleviate the conflict between distribution efficiency and cost has become an urgent problem for e-commerce logistics. The massive historical demand data accumulated by digital business enterprises in the development process can be employed to generate prior knowledge of dynamic customer demand and integrate it with the optimization process of the vehicle routing problem, which is useful for the realization of dynamic proactive response to customer demand and to improve the quality of city logistics distribution efficiency.

In the context of a “sharing economy”, the development of information technology and mobile communication networks has created favorable conditions for the employment of sharing services. Collaboration services allow participators to capture revenue from underutilized information resources. Nevertheless, the city distribution may exacerbate road traffic congestion and carbon emissions in urban areas. On this basis, some big retailers, such as Walmart, Tesco and Amazon, have begun to make new attempts to attract various resources participating in distribution [1–3]. Therefore, it is of great practical significance to consider collaborative services to design “last-mile” distribution in vehicle routing problems. At the same time, the importance and competitive value of “last-mile” and same-day delivery has led many companies to seek new solutions. One novel solution proposed by Walmart and Amazon is “crowdsourcing”, which involves using their customers to transport packages on the way to their destinations rather than hiring many drivers or using third-party logistics firms (e.g., UPS or FedEx).

With the increase of e-tailing, orders are delivered through the fleets of retailers composed of capacitated vehicles or picked up by customers in physical stores. Compared with picking up in physical stores, higher distribution costs are required in the delivery of retailers, especially in the case of small numbers and scattered orders. However, with the unwillingness or insufficient time of some customers to pick up goods in physical stores, a new solution is required to motivate some customers picking up in physical stores to make a single delivery through their private cars, which requires that the delivery location is not too far from their destination and they are willing to make a single delivery given a small compensation. On this basis, a real-life application related to the PDVRPCS model occurs. Therefore, this paper focuses on the modeling and treatment of PDVRPCS. To deal with such routing optimization problems of collaboration services, this paper designs a highly accurate proactive forecasting method, order matching algorithm and a hybrid Genetic Algorithm (GA) and Simulated Annealing (SA) to minimize the transportation cost of logistic enterprises.

The remaining parts of this paper are organized as follows. In Section 2, the relevant literature is summarized, and the innovation points of this paper are clarified. Section 3 introduces the problem studied in this paper in detail. In Section 4, a mathematical model of PDVRPCS is described, and the detailed implementation of the GA-SA for solving the model is proposed in Section 5. A case study is conducted to verify the validity of the algorithm and mathematical model in Section 6. Finally, this study concludes in Section 7.

2. Literature review

The literature review starts with proactive vehicle routing problems. Next, the existing results of vehicle routing problems based on capacity sharing are introduced, which is relevant for the research topic of this paper. Then, existing algorithms for solving vehicle routing problems are introduced briefly. Finally, the innovations of the models and algorithms proposed in this paper are emphasized.

The proactive vehicle routing problem is rooted in dynamic vehicle routing problems, referring to the vehicle routing problem with prediction and response strategies, and the possibility of demand is predicted in advance by the logistic enterprise through the customers' history of demand, attributes, etc. The distribution center of the enterprise can quickly respond to customer demand, decreasing the response time of demands. Thomas [4] designed a real-time heuristic algorithm to predict information in potential customer location and probability of demand occurrence, which can reduce the customer service delay rate. Lima et al. [5] improved the flexibility of customer response by using historical data and online data monitoring to anticipate the customer's requests from both customer needs and the enterprise itself. Based on the real-time, shared point-of-sale data, Hartzel and Wood [6] predicted future sales by using a single exponential smoothing method. Ma and Fildes [7] used a promotional decision support system to forecast demand and inventory levels by using the operational (store-level) data only. Zhu et al. [8] employed historical data to predict the demand for spare parts proactively using extreme value theory. Since the data change dynamically in real cases, Belka and Godlewski [9] focused on the aspect of ge positioning updates and their impact on the effectiveness of optimization algorithms. Ferrucci and Bock [10] proposed a space-time Poisson distribution model to cluster various customer demand patterns and realized proactive real-time control of distribution vehicles. Fathollahi-Fard et al. [11] optimized the home healthcare routing and scheduling problem considering patients' satisfaction under uncertainty by applying Jimenez's method and a modified multi-objective metaheuristic. Although all of the above papers have adopted prediction methods, there is a lack of appropriate response strategies. Therefore, fuzzy and deterministic indicators representing customers' attributes from multiple dimensions are selected in this paper. Then, the prospect value theory is used to predict the potential demands of dynamic customers, and the predicted results are employed for routing optimization.

In recent years, capacity sharing pattern design has become a hot topic in routing optimization. Fernández et al. [12] introduced the shared customer collaboration vehicle routing problem and established an integer programming model to minimize the total cost under the shared framework. The results showed that the ultimate cost savings depend on the number of shared customers and their location. Paul et al. [13] considered capacity sharing among different sales channels. Capacity sharing strategies were formulated mainly by determining transfer customers and using traditional idle capacity to reduce transportation cost. The outsourcing of 3rd Party Logistics (3PL) is a relatively mature logistics service mode, which can provide low-cost logistics services for enterprises. However, this mode requires a large number of orders as the basis of cooperation. Collaborative delivery of online orders by willing in-store customers (occasional drivers) is a low-cost solution when dealing with small, scattered orders. Archetti et al. [14] studied the vehicle routing problem with occasional drivers in logistics networks and established a static mixed integer programming model. The model is solved by the multi-start heuristic. Later, Macrina et al. [15] proposed two mixed integer programming models to extend the vehicle routing problem with occasional drivers, one allowing multiple deliveries per occasional driver and the other one allowing batch deliveries. Arslan et al. [16] studied the crowdsourcing distribution problem by proposing a variant that considers dynamic pickup and delivery options. The authors assumed that one driver can complete one or more deliveries from different sources. Mojtahedi et al. [17] proposed a sustainable vehicle routing problem by using the concept of triple bottom line for coordinating solid waste management based on the financial, environmental and social goals. Numerical experiments show that framing the coordinated logistics problem across several echelons can lead to potential overall waste disposal cost savings through increased recycling.

Fathollahi-Fard et al. [18] developed a bi-level programming model for a home health care supply chain. To solve the model, three modified versions of algorithms and two hybrid meta-heuristics were presented to solve the model. For the first time, the authors proposed outsourcing for patients' demand to receive direct services from a hospital. Wang et al. [19] presented a Genetic Algorithm and Particle Swarm Optimization (GA-PSO) heuristic for the optimization of two-echelon cooperative logistics networks and designed a negotiation process for profit allocation. Dayarian and Savelsbergh [20] proposed a crowdsourcing and same-day delivery problem under a highly dynamic and stochastic delivery environment. To address the resource sharing in two-echelon logistics networks, Luo et al. [21] proposed an improved Non-dominated Sorting Genetic Algorithm-II (NSGA-II) for minimizing the total operational cost and vehicles used. Ren et al. [22] established a multi-center joint distribution optimization model for the low-carbon vehicle routing problem with time windows and developed a modified ant colony algorithm to solve the model by using the elite strategy and three further recombination rules. Gdowska et al. [23] modeled the crowdsourcing distribution problem as a double-level stochastic problem and assumed that occasional drivers can choose to accept or reject possible assignments. In order to solve the problem, the authors designed a heuristic method to find a subset of customers to be proposed to be occasional drivers. Research on capacity sharing focuses more on how to use cooperative vehicles for transportation instead of how to match between cooperative vehicles and online orders. In this paper, a low-cost capacity sharing strategy is achieved by matching the time window and destination relevance.

The Vehicle Routing Problems Cooperation Service (VRPCS) belongs to the Nondeterministic Polynomial-hard (NP-hard) problem. In addition to the basic constraints, the time window constraints of online customers, the time window constraints of cooperative drivers and the service scope should also be satisfied, which increases the complexity of PDVRPCS [24]. Russell and Chiang [25] employed a scatter search framework to solve the vehicle routing problem with time windows. For the same problem, Zhang et al. [26] established a mathematical model of vehicle routing problems with time windows and designed a hybrid intelligent algorithm to solve it. Cheng and Wang [27] decomposed the original vehicle routing problem into a clustering problem and a set of sub-problems with time window constraints. Meanwhile, a genetic algorithm and a simple heuristic algorithm have been developed to solve the two problems. Vidal et al. [28] designed a hybrid genetic algorithm with adaptive diversity control which provides a new neighborhood movement choice and restriction method to solve time-constrained vehicle routing problems. Belhaiza et al. [29] designed a hybrid tabu search with variable neighborhood algorithm to solve the vehicle routing problem with multiple time windows. In order to deal with multiple time windows, a minimum backward time slack algorithm is designed. Aiming at minimizing the total transportation distance, Shi et al. [30] designed a novel sector combination optimization algorithm to deal with waste collection problems in urban areas. Pasha et al. [31] studied a vehicle routing problem with a factory-in-a-box in multi-objective settings and proposed to compromise the conflicting cost components. A novel customized nature-inspired evolutionary algorithm is developed to solve the problem. Fathollahi-Fard et al. [32] studied a multi-objective robust optimization of the home healthcare model, which could present multi-depot, multi-period and multi-service. Baños et al. [33] considered a multi-objective vehicle routing problem with time windows, whose objective function not only considered distance and time window punishment, but also workload balance. Finally, a Pareto-based hybrid algorithm was designed to solve the model.

In order to solve the conflict between the demand and supply imbalance of terminal distribution, which aims to reduce the distribution cost and improve the distribution efficiency, scholars have carried

out a lot of research on the mode of terminal collaborative distribution and introduced the concept of sharing into the field of distribution [34]. The public who are willing to participate receive small rewards. Jaller et al. [35] believe that crowdsourcing is a logistics mode in which traditional cargo transportation enterprises allocate the task of delivery of goods to the public who are willing to integrate it with their own schedules. This model is driven by the popularity of smartphones and the widespread adoption of mobile devices.

In a market where order arrival rates fluctuate greatly throughout the day, the advantages of crowdsourcing collaborative delivery are obvious and provide a new research perspective on collaborative vehicle routing problems. However, the volatility of customers and deliverers is an important exogenous variable in collaborative vehicle routing problems based on crowdsourced delivery. At present, a review of the relevant literature implies that few publications consider both dynamic proactive demand forecasting and order match simultaneously. Based on the studies of [13,14,36], the PDVRPCS was proposed in this paper (Figure 1), and this paper has the following four-fold scientific contributions:

- 1) Formulating a proactive dynamic vehicle routing problem considering a cooperation service model minimizing the total operational cost;
- 2) Designing a proactive customer demand forecasting method based on customer's historical data;
- 3) Developing a matching algorithm for collaborative temporary drivers and online orders;
- 4) Integrating the simulated annealing algorithm into a genetic algorithm framework to solve the PDVRPCS model.

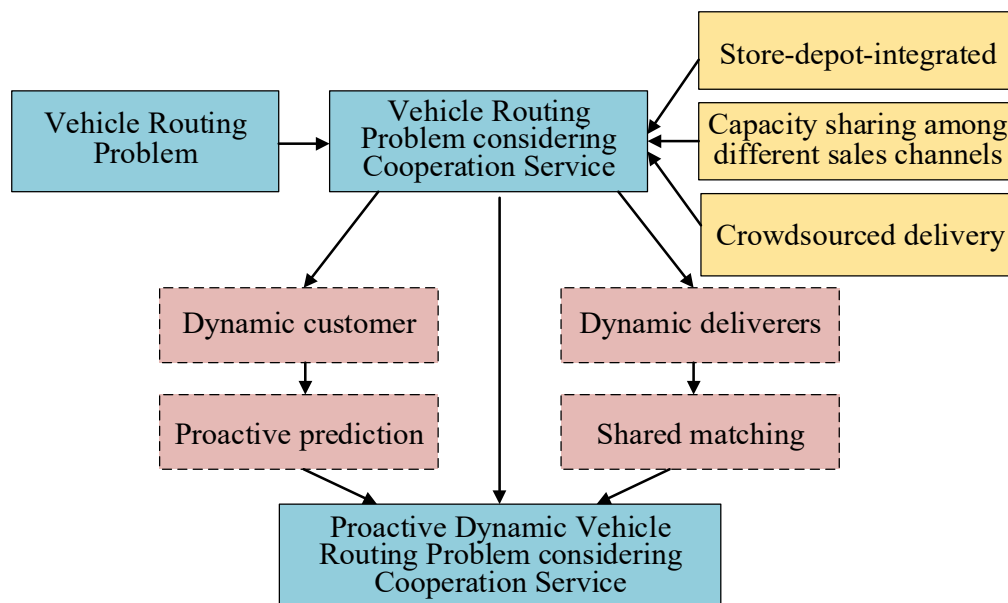


Figure 1. The research vein of PDVRPCS.

3. Problem development

In this section, the basics of PDVRPCS are described. The proactive prediction method is used based on the historical data of dynamic customers, and then the matching method is designed to realize capacity sharing.

3.1. Problem description

The research object of this paper is the store-depot-integrated retail enterprise with physical stores and online stores (the location of physical stores is denoted as depot below). This kind of enterprise has two types of customers, namely, Store Customer (SC) and Online Shopping Customer (OSC). The shared transportation service platform provided by the enterprise can achieve the resources match between SC and OSC. Store customers are willing to provide shared transportation resources to fulfill the OSC's orders by using compensation or preferential options. Capacity sharing vehicles of a SC will provide transportation services for a part of the OSC based on customer matching considering the time windows and destinations while the unserved OSC will be delivered by the enterprise's vehicles. Each capacity sharing vehicle arrives at its destination and enterprise-owned vehicles return to the warehouse after completing the service. The capacity-sharing mode considering time windows may reduce the usage frequency of enterprise-owned vehicles for serving an OSC. Therefore, distribution cost can be significantly reduced, especially for the scenario with many remote OSCs and SCs.

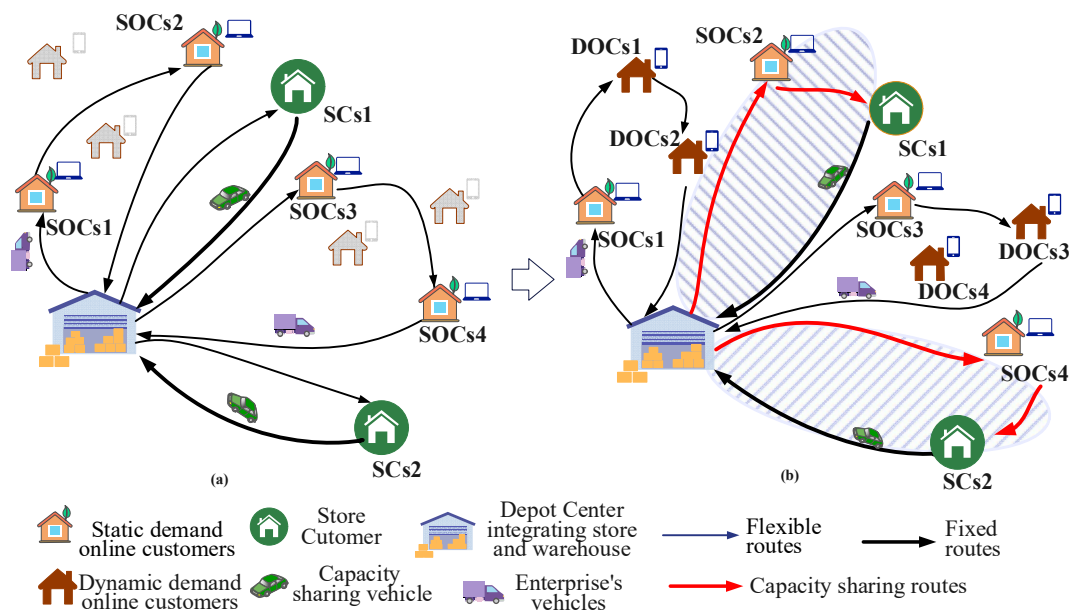


Figure 2. An illustration of PDVRPCS.

As shown in Figure 2, assuming that a retail enterprise has eight customers, among which customers SOC1 to SOC4 are OSCs, which are named static online customers. DOC1 to DOC4 are possible OSCs, which are called dynamic online customers. Customers SC1 and SC2 belong to SCs that can provide capacity sharing vehicles to provide a distribution service for dynamic online customers. Figure 2(a) shows the delivery path without the customer-matching strategy. The static online customers need to be delivered by two vehicles, and dynamic online customers have not been added to the delivery task. After predicting demands of online dynamic customers DOC1 to DOC3, the routings obtained by combining the customer matching strategy are shown in Figure 2(b), i.e., the orders of SOC2 and SC4 are distributed by SOC4 and SC2 respectively, and the remaining static and dynamic OSC are served by two vehicles.

3.2. Symbol description

The PDVRPCS model is defined as follows: a completely undirected graph $G = (N, A)$ is used to represent the distribution network with node set N and arc set A . The node set $N = \{o\} \cup N_C \cup N_H$ consists of three parts, where o represents the location of the warehouse, N_C means the location of the OSC and N_H denotes the location of the SC's destination. The parameters and variables used in this model are presented in Table 1.

Table 1. Definitions and notations related to the PDVRPCS mathematical model.

		Definitions
Sets	N	set of nodes
	o	depot integrating store and warehouse
	N_S	set of static OSCs
	N_D	set of dynamic OSCs
	N_C	set of OSCs
	N_H	set of cooperative SC's destination
	N_R	set of customers with denied service
	K	set of vehicles k
Parameters	H	set of cooperative SC's vehicles h
	d_{ij}	travel distance from node i to j
	q_i	demand of customer i
	Q	capacity of vehicles
	c_1	variable cost of vehicles, unit: /km
	c_2	initial cost of vehicles
	$c_3(t_i)$	penalty cost for soft time windows in customer i
	y_{ijk}	load of vehicle k on $arc(i, j)$
	$[a_i, b_i]$	time windows of customer i
	a_h	departure time of cooperative vehicle h
	b_h	due time of cooperative vehicle h arriving at destination
	W	waiting cost for earlier arriving, unit: /hour
	L	penalty cost for late arriving, unit: /hour
	t_i^a	arrival time to node i
	t_i^l	departure time form node i
	T_i	service time of customer i
	t_{ij}	travel time from node i to j
	ρ	compensation factor of the cooperative vehicle
	P_{ih}	payment of cooperative vehicle h for servicing customer i
	ε	degree of collaboration flexibility of the cooperative vehicle
RP_i	penalty cost for denial service to customer i	

Continued on next page

		Definitions
Variables	x_{ijk}	1 if vehicle k travels the arc (i, j) ; 0 otherwise
	z_{ik}	1 if customer i is served by vehicle k ; 0 otherwise
	ω_{ih}	1 if customer i is served by cooperative vehicle h ; 0 otherwise
	β_{ih}	1 if customer i is served by cooperative vehicle h
	x_{ok}	1 if vehicle k returns the depot after completing task; 0 otherwise

3.3. The conception of proactive predication

In order to meet the dynamic customer demands, the concept of proactive distribution is introduced. Meanwhile, the prospect theory [37] is employed to make a proactive evaluation of the dynamic customer's demands, and then judge whether to serve a dynamic customer, based on the predicted value p_{ik} , which denotes the customer's demand, and a_{ik} , which denotes the mean of distance matrix for the historical demand data. Assume that the customer historical demand data m dimensions, and all customers n can be expressed as matrix $(a_{11}, a_{12}, \dots, a_{1m}; a_{21}, a_{22}, \dots, a_{2m}; a_{n1}, a_{n2}, \dots, a_{nm})$. The triangular fuzzy number [38] and the deterministic attribute are used to describe definite and indefinite attributes in the customer demand matrix. The purpose of this expression is characterized as the dynamic customer's demand more precisely.

If the customer's attributes are determined, the logical distance d_{ik} between the average historical data and the predicted value of customer demand is:

$$d_{ik} = |p_{ik} - a_{ik}| \quad (1)$$

Using the triangular fuzzy number to describe a customer demand [36], Eq (1) can be further formulated as Eq (2):

$$d_{ik} = \sqrt{\frac{|(p_{ik}^1 - m_{ik}^1)|^2 + |(p_{ik}^2 - m_{ik}^2)|^2 + |(p_{ik}^3 - m_{ik}^3)|^2}{3}} \quad (2)$$

In Eq (2), p_{ik} can be calculated by [39]:

$$p_{ik} = \left(\max\left(\frac{l-1}{n}, 0\right), \frac{l}{n}, \min\left(\frac{l+1}{n}, 1\right) \right), l \in [1, n] \quad (3)$$

where l is the interval of the customer's attributes. A threshold is predefined by decision makers to select dynamic customers that need to be serviced, and then the selected static customers and dynamic customers will be composed into a new organization.

According to prospect theory, if the predicted value of the customer demand attribute is greater

than the actual value of customer demand ($p_{ik} > m_{ik}$), decision makers are prone to choose the option with higher risk and return. Otherwise, decision makers are prone to avoid possible risks and choose the option with low risk and stable revenue. To cope with different situations, decision makers have different degrees of understanding the risk avoidance when making decisions. Therefore, two risk factors μ_1 and μ_2 are predefined, representing the decision makers' different risk preferences. The calculation method of the prospect value considering the two risk factors is shown in Eq (4):

$$FV_i = \begin{cases} \mu_1 d_{ik}, p_{ik} > m_{ik} \\ \mu_2 d_{ik}, p_{ik} < m_{ik} \end{cases} \quad (4)$$

Finally, according to the prospect value of dynamic online customers, a threshold has to be set to select the dynamic online customers that meet the decision maker's preference. Combined with the unserved static online customers, a total unserved customer set C as shown in Eq (5):

$$\begin{cases} FV_i > 0, i \in C \\ FV_i \leq 0, i \notin C \end{cases} \quad (5)$$

3.4. Customer matching strategy

Firstly, it has to be determined whether the cooperative SCs travel time meets the OSCs time window constraints according to Eqs (6)–(8):

$$a_h + t_{oi} \geq a_i \quad (6)$$

$$a_h + t_{oi} \leq b_i \quad (7)$$

$$a_h + t_{oi} + d_{iN_h} \leq b_h \quad (8)$$

Equations (6) and (7) indicate that the departure time of the cooperative SC from the warehouse and the travel time to the OSC should not be earlier than the opening time and should not be later than the closing time of the OSC's time window. Equation (8) indicates that the collaborative SC can return to its destination no later than its acceptable latest arrival time.

Secondly, it has to be determined whether the distance between the collaboration SC and the OSC meets the collaboration requirements according to Eqs (9)–(13).

$$d_{oi} + d_{iN_h} \leq \varepsilon d_{oN_h} \quad (9)$$

$$\omega_{ih} \leq \beta_{ih} \quad \forall i \in N_s, h \in H \quad (10)$$

$$\sum_{i \in N_x} \omega_{i\phi} \leq 1 \quad \forall \phi \in \Phi \quad (11)$$

$$\omega_{ih} \in \{0,1\} \quad \forall i \in N_s, h \in H \quad (12)$$

$$\beta_{ih} \in \{0,1\} \quad \forall i \in N_s, h \in H \quad (13)$$

Equation (9) indicates that the cooperating SCs will only accept orders from the OSC's if the distance traveled from the warehouse to the OSC and to its destination is less than or equal to ε times the direct distance traveled from the warehouse to the cooperating SC's destination, and $\varepsilon \geq 1$. Equation (10) ensure that an OSC is assigned to a collaborative SC. Equation (11) ensure that a collaborative SC can serve at most one OSC. Thus, the matching results of cooperative SCs and OSC's orders can be obtained. Equations (12) and (13) are binary variables for the match.

3.5. The PDVRPCS model

The total cost includes vehicle variable cost, vehicle initial cost, soft time window penalty cost and compensation cost, denoted as: $Z = C_1 + C_2 + C_3 + C_4$, where

$$C_1 = c_1 \sum_{k \in K} \sum_{i \in N_I} \sum_{j \in N_I} x_{ijk} d_{ij} \quad (14)$$

$$C_2 = c_2 \sum_{k \in K} x_{ok} \quad (15)$$

$$C_3 = \sum_{k \in K} \sum_{i \in N_I} c_3(t_i) \quad (16)$$

$$C_4 = \sum_{i \in N_I} \sum_{h \in H} p_{ih} \omega_{ih} \quad (17)$$

Based on Table 1 and the previous description, a PDVRPCS model for minimizing the total cost can be established. The objective function is defined as:

$$\text{Min} \left(Z + \sum_{i \in N_R} RP_i \right) \quad (18)$$

s. t.

$$\sum_{h \in H, k \in K} \omega_{ih} + z_{ik} = 1, \forall i \in N_s \quad (19)$$

$$\sum_{i \in N_E, i \neq g} x_{jik} - \sum_{i \in N_E, h \neq j} x_{ijk} = 0, \forall j \in N_E, k \in K \quad (20)$$

$$\sum_{k \in K} \sum_{i \in N_s} x_{oik} - \sum_{k \in K} \sum_{j \in N_s} x_{jok} = 0 \quad (21)$$

$$\sum_{i, j \in N_s, i \neq j} x_{ijk} = \sum_{i, j \in N_s, i \neq j} x_{jik} = z_{ik}, \forall k \in K \quad (22)$$

$$y_{io} = 0, \forall i \in N_S \quad (23)$$

$$t_i^l = t_i^a + T_i, i \in N_S \quad (24)$$

$$c_3(t_i) = \begin{cases} W(a_i - t_i^a), t_i^a < a_i \\ 0, a_i \leq t_i^a \leq b_i \\ L(t_i^l - b_i), b_i < t_i^a \end{cases}, i \in N_E \quad (25)$$

$$\sum_{i,j \in N_S, i \neq j} y_{jik} - \sum_{i,j \in N_S, i \neq j} y_{ijk} = \begin{cases} q_i z_{ik} & \forall i \in N_S \\ \sum_{i \in N_C} -q_i z_{ik} & i = 0 \end{cases}, \forall k \in K \quad (26)$$

$$y_{ij} \leq Qx_{ij}, \forall i \in N_S, j \in N_S \quad (27)$$

$$\sum_{i,j \in N_S, i \neq j} x_{ijk} \leq |N_S|, \forall k \in K \quad (28)$$

The objective function (18) is to minimize the total cost, including Z and the penalty cost for denial of service. Constraint (19) guarantees that each customer is served only once. Constraint (20) guarantees the balance of vehicle flow in each customer. Constraint (21) guarantees that all vehicles start from and return to the depot. Constraint (22) guarantees that each customer is served by a vehicle at most once. Constraint (23) ensures that all vehicles must return to the depot without load. Constraint (24) represents the relationship between the departure time of vehicles from the customer, the arrival time and service time. Constraint (25) represents the penalty cost of a vehicle violating the soft time window. Constraint (26) ensures that customer requirements are met on a route. Constraint (27) defines the capacity constraint of the vehicles. Constraint (28) indicates that the number of OSCs served by each vehicle cannot exceed the total number of OSCs. Because of complexity of the PDVRPCS model, the GA-SA algorithm is designed for large-scale instances to solve this problem in the next section.

4 Solution framework of PDVRPCS

Different from other collaborative vehicle routing problems, this paper proposes a three-stage hybrid meta-heuristic algorithm to solve PDVRPCS. The first stage is a proactive prediction of dynamic OCSs, judging whether to provide the delivery service to customers according to their prospect value. In the second stage, the matching between cooperative SCs and static OCSs is solved by the Hungarian algorithm [40–42]. In the last stage, the designed GA-SA is employed to solve the static OSC that fails to match and the dynamic OSC whose prospect value is greater than zero. A special feature of the hybrid meta-heuristic algorithm is that the dynamic complexity problem is preprocessed statically by fuzzy theory before starting the meta-heuristic algorithm. When solving the main model of the vehicle routing problem, the advantages of two meta-heuristic algorithms, which are GA and SA based on neighborhood search, are used to establish a hybrid heuristic algorithm to effectively identify high-quality solutions. This algorithm is designed to balance the exploration behavior of the solution space. Figure 3 provides an overview of the framework of the hybrid algorithm.

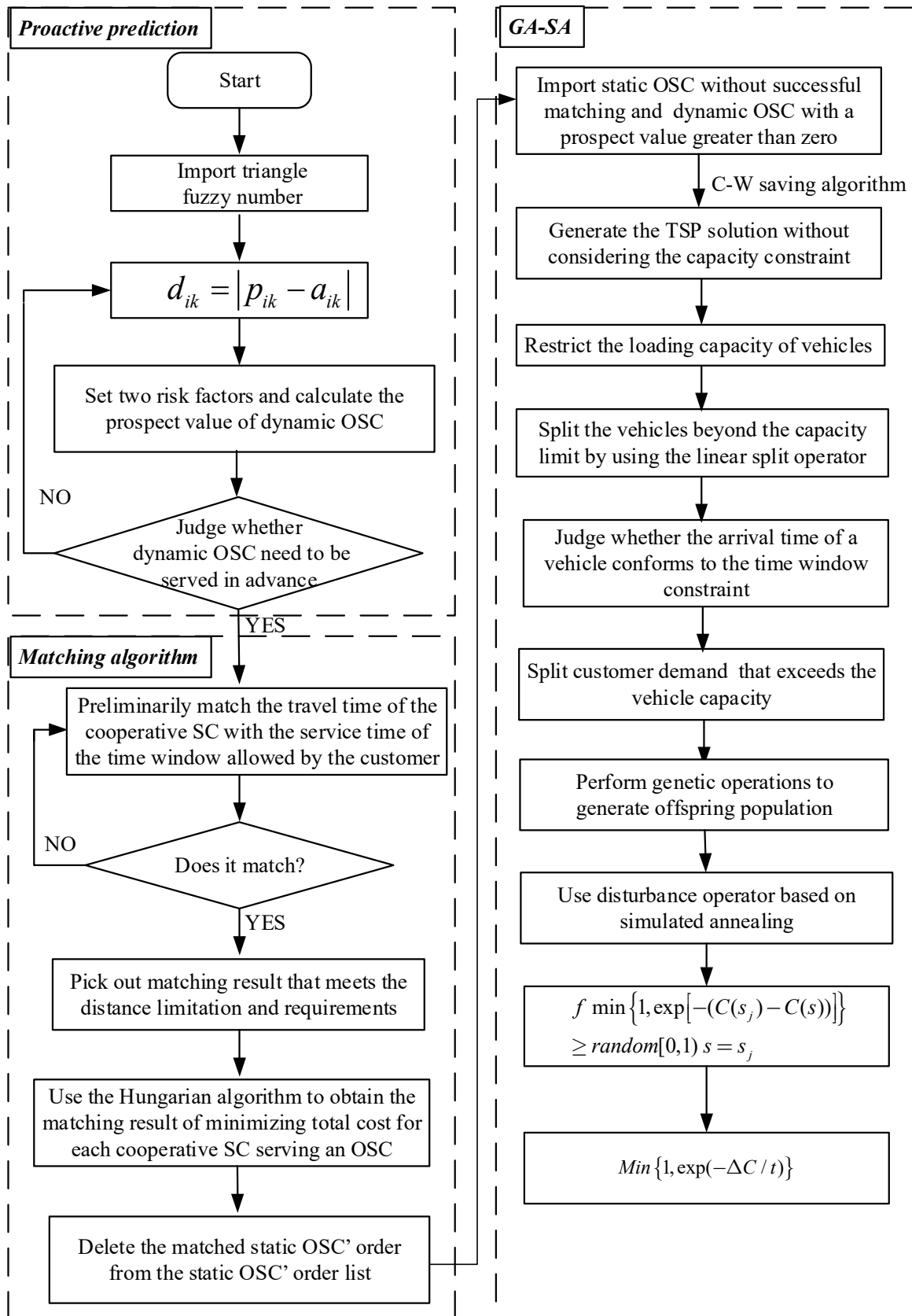


Figure 3. Solution framework of PDVRPCS.

4.1. Proactive prediction

By mining the demand regularity hidden in the demand information in the past, a proactive method can predict customer demand in advance by using correlation analysis methods and means of mathematical statistics. The method determines whether to deliver for customers based on their probability of appearing. The specific steps are as follows:

Step 1. Use triangle fuzzy number Eq (2) to calculate the logical distance of customer demand attributes;

Step 2. Judge the relationship between the predicted value of customer demand attributes and the actual value of customer demand, and determine the risk preference of decision makers;

Step 3. According to different risk preferences, set two risk factors and calculate the prospect value of dynamic OCSs considering risk factors according to Eq (4);

Step 4. Judge whether dynamic OCSs need to be served in advance according to the prospect value obtained by Eq (5).

4.2. Matching algorithm

The Hungarian algorithm is used to implement shared matching. It is a combinatorial optimization algorithm to solve the task assignment problem in polynomial time, which is widely used in the field of operations research. The specific steps are as follows:

Step 1. Preliminarily match the travel time of the cooperative SC with the service time of the time window allowed by the customer;

Step 2. Pick out matching results that meet the distance limitation and requirements;

Step 3. Use the Hungarian algorithm to obtain the matching result of minimum total cost for each cooperative SC serving an OSC;

Step 4. Delete the matched static OCS's order from the static OCS's order list.

4.3. Hybrid algorithm

SA is a simple and powerful single-point algorithm, inspired by the reduction of temperature during annealing, which is widely used to improve the search space of group-based algorithms. The hybrid GA-SA) is designed to solve the model, since the GA has strong global optimization ability and the SA has the ability to escape the local optimum. Unlike other meta-heuristics that only accept solutions that are better than the current one, the GA-SA hybrid algorithm offers the opportunity to accept new solutions that may be worse than the current one, and it initializes the search using the Chahla and Wagman (C-W) saving algorithm. The key operations of the GA-SA include the generation of an initial solution, genetic operators and a disturbance operator.

4.3.1. Solution representation and search space

In the proposed hybrid algorithm, a chromosome encoded with natural numbers is defined to represent the solution of a given optimization problem. Each natural number that codes for a chromosome is called a gene, where the number of the gene represents the number of the customer. The number of genes on the chromosome indicates the number of customers, and the order of genes indicates the order in which customers are visited. 0 indicates a garage, warehouse, or distribution center. The search space of the algorithm consists of all possible chromosomes. In order to determine the solution space, the cost function Eqs (14)–(18) and the Constraints (19)–(28) in the optimization

model are defined by five binary decision variables (x_{ijk} , z_{ik} , ω_{ih} , β_{ih} , x_{ok}). In each iteration, search operators (including crossover, mutation, destruction and repair) randomly select chromosomes from the search space to further explore the solution space.

4.3.2. Initial solution

The initial solution is generated by the following three steps:

Step 1. For static OCSs without successful matching and for dynamic OCSs with a prospect value greater than zero, use the saving method to generate the travelling salesman problem (TSP) solution without considering the capacity constraint;

Step 2. Restrict the loading capacity of the vehicles, and split the vehicles beyond the capacity limit by using the linear split operator;

Step 3. Further constrain the time window of OCSs, judge whether the arrival time of vehicles conforms to the time window constraint and, if so, generate the initial solution.

4.3.3. Design genetic operator

In heuristic algorithms, crossover, mutation, destruction and repair operators are often used to explore and exploit the search space. The traditional roulette operator is used as the selection operator [43]. Meanwhile, the strategy of retaining the best substring is employed by crossover operators. The mutation operator is to randomly select a chromosome, and randomly select a mutation position in the chromosome. If the position is not zero, a new gene will be randomly generated to replace the original gene. If the mutation location gene is zero, the mutation location is selected again.

1) Crossover operator

To explore the search space for solutions, gene segments with random min-max values are used to cross the operator to generate progeny chromosomes. For example, based on the crossover probability, two parental chromosomes are randomly selected to cross, which are parent 1: {1,6,5,2,7,8,4,3} and parent 2: {3,1,2,4,8,5,6,7}. Before crossing, two cross-indexed integer values 3 and 6, unequal and smaller than chromosome length, are first generated. Based on this, two gene segments {5,2,7,8} and {2,4,8,5} are generated by randomly selecting positions 3 to 6 from both the parent 1 and the parent 2. When crossing is performed, {5,2,7,8} and {2,4,8,5} are moved to the parental chromosome header, respectively. In the crossover operation, {5,2,7,8} and {2,4,8,5} are moved to the head of the parental chromosome, respectively, and the individuals with the same genes as {5,2,7,8} and {2,4,8,5} are deleted in the parents, resulting in two new individuals named offspring 1: {2,4,8,5,1,6,7,3} and offspring 2: {5,2,7,8,3,1,4,6} (Figure 4).

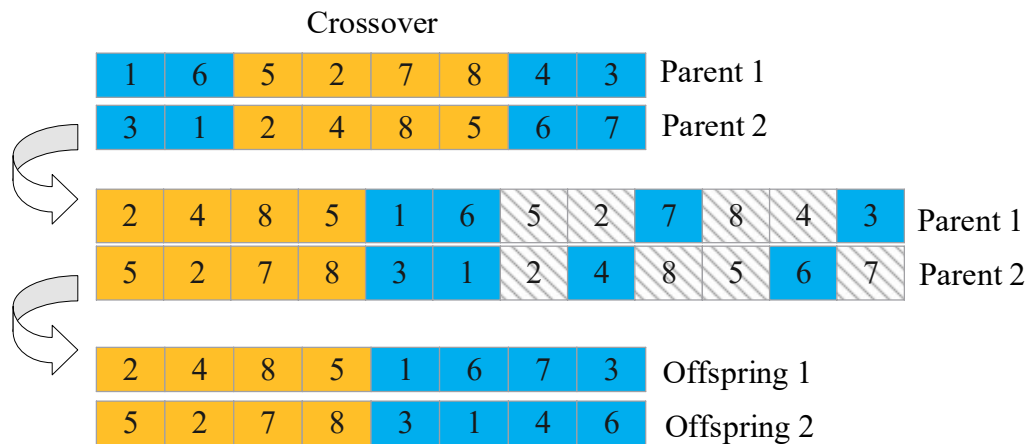


Figure 4. Crossover operator.

2) Mutation operator

In order to perform the mutation operation, a part of the population is selected using the roulette selection method and a chromosome is selected as the parent. In our study, we used three different mutation operators: two-point, insertion, reversal and two-end interval reversal operators, as shown in Figure 5. For all three operators, we use the same parent $p: \{1,6,5,2,7,8,4,3\}$ to mutate. For a selected chromosome, one of the mutation operators is randomly selected to produce two new offsprings. By using the Swap mutation operator, two mutation index integer values 3 and 6, which are unequal and smaller than the length of the chromosome, are randomly generated, and the third and sixth genes are exchanged to generate a new daughter chromosome $\{1,6,8,2,7,5,4,3\}$. A new progeny $\{1,6,2,7,8,5,4,3\}$ is generated by inserting the third gene into the sixth gene through Insertion operators. In addition, a new solution $\{1,6,8,7,2,5,4,3\}$ is created by using the Reversion operator to reverse the sequence of genes between the 3rd and 6th.

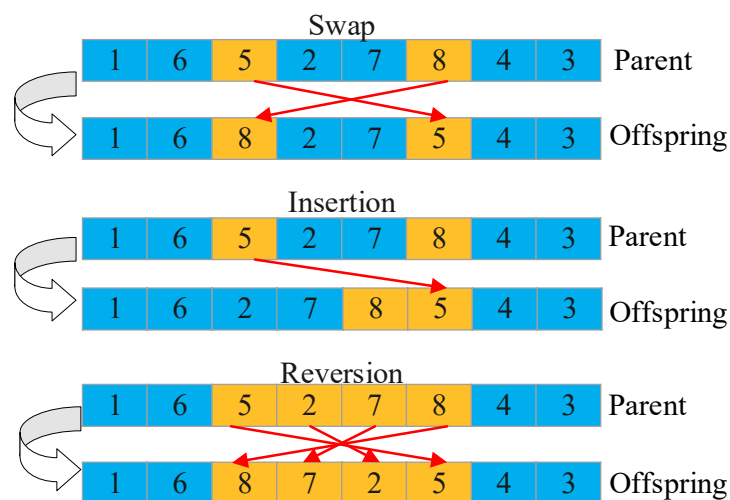


Figure 5. Mutation operators.

3) Damage and repair operators

In order to explore the space better, the damage repair operator is used to repair the chromosome $\{1,6,5,2,7,8,4,3\}$. Three gene loci 2, 7 and 8 are removed according to certain rules to get the removed

missing chromosome $\{1,6,5,4,3\}$, and then the removed points are inserted into the missing chromosome according to certain rules to get a better chromosome $\{1,6,5,2,4,7,8,3\}$ (Figure 6).

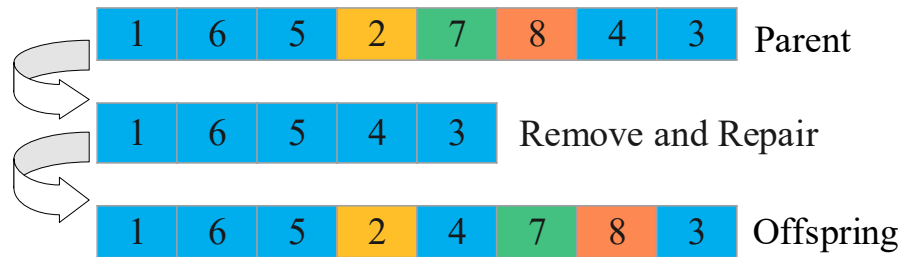


Figure 6. Remove and repair operator.

4.3.4. Disturbance operator based on simulated annealing

In the simulated annealing algorithm, the random perturbation operator is adopted, with the following steps:

Step 1. Generate new status $s_j = \text{Generate}(s)$;

Step 2. $\text{if } \min \{1, \exp[-(C(s_j) - C(s))]\} \geq \text{random}[0,1] \text{ } s = s_j$

Step 3. Accept $\text{Min} \{1, \exp(-\Delta C / t)\}$ as a state function.

4.4. Benchmark test

In order to verify the solving quality of the GA-SA algorithm, the well-known Solomon benchmark instance is used to test it. The number of nodes in the tested instances is 50 and 100. The numbers of available vehicles and vehicle capacity vary with the different instances. The data files for all the instances can be downloaded from <http://web.cba.neu.edu/~msolomon/problems.htm>, and the Best Known Solution (BKS) can be obtained from <http://web.cba.neu.edu/~msolomon/heuristi.htm>. The result is used to compare with GA-SA, which is represented by Chen and Shi (C & S), and column C&S shows the optimal value [44]. The performance of GA-SA is measured in terms of gap percentage, which is defined by Eqs (29)–(31).

$$\text{Gap}^a = \frac{(GASA - BKS)}{BKS} \times 100\% \quad (29)$$

$$\text{Gap}^b = \frac{(C \& S - BKS)}{BKS} \times 100\% \quad (30)$$

$$\text{Gap}^c = \text{Gap}^b - \text{Gap}^a \quad (31)$$

Since the benchmark instances are only simulations of realistic distribution scenarios, when applying GA-SA, the Euclidian distance is used to calculate the distance between nodes, and the travel time is assumed to be equal to the corresponding total distance, that is, the vehicle speed is 1. In

addition, since the objective function designed in this paper represents the total cost and is designed for real cases, it needs to be changed to minimize the total distance. The algorithm is run 5 times on each benchmark instance, and the best results are summarized in Table 2 ($N = 50$) and Table 3 ($N = 100$).

Table 2. Results based on the Solomon instances ($N = 50$).

	BKS	C & S	GA-SA	Gap ^a	Gap ^b	Gap ^c
C101-50	362.4	418.42	368.66	1.73%	15.46%	13.73%
C102-50	361.4	417.34	364.67	0.90%	15.48%	14.57%
C201-50	360.2	373.55	360.68	0.13%	3.71%	3.57%
C202-50	360.2	366.78	361.48	0.35%	1.83%	1.47%
R101-50	1044	1046.7	1144.71	9.65%	0.26%	-9.39%
R102-50	909	911.44	975.05	7.27%	0.27%	-7.00%
R201-50	791.9	808.53	797.20	0.67%	2.10%	1.43%
R202-50	698.5	714.19	704.99	0.93%	2.25%	1.32%
RC101-50	944	958.59	949.10	0.54%	1.55%	1.01%
RC102-50	822.5	886.47	826.84	0.53%	7.78%	7.25%
RC201-50	684.8	686.31	693.21	1.23%	0.22%	-1.01%
RC202-50	613.6	615.04	615.07	1.23%	0.23%	-0.99%
Avg				2.10%	4.26%	2.16%

As can be seen from Table 2, the average Gap^a between the results obtained by the GA-SA and the BKS for $N = 50$ is 2.10%, and the average Gap^b between the results obtained by the C & S and the BKS is 4.26%. The average Gap^c between the results obtained by the Gap^a and the Gap^b for the three types of instances is 2.16%. In the bold value of Table 2, the total distance of 8 instances, such as C101-50 and C102-50, is better than that of C & S.

Table 3. The results based on the Solomon instances ($N = 100$).

	BKS	C & S	GA-SA	Gap ^a	Gap ^b	Gap ^c
C101-100	827.3	944.32	864.1	4.45%	14.14%	9.69%
C102-100	827.3	952.65	838.7	1.38%	15.15%	13.77%
C201-100	589.1	609.22	589.1	0.00%	3.42%	3.42%
C202-100	589.1	591.56	589.1	0.00%	0.42%	0.42%
R101-100	1637.7	1692.29	1658.9	1.29%	3.33%	2.04%
R102-100	1466.6	1541.15	1623.6	10.71%	5.08%	-5.62%
R201-100	1143.2	1191.36	1206.7	5.55%	4.21%	-1.34%
R202-100	/	1084.14	1159.4	/	/	/
RC101-100	1619.8	1702.69	1697.0	4.77%	5.12%	0.35%
RC102-100	1457.4	1581.29	1498.6	2.83%	8.50%	5.67%
RC201-100	1261.8	1282.35	1261.8	0.00%	1.63%	1.63%
RC202-100	1092.3	1108.01	1104.7	1.14%	1.44%	0.30%
Avg				2.92%	5.68%	2.76%

As can be seen from Table 3, the average Gap^a between the results obtained by the GA-SA and the BKS for $N = 100$ is 2.92%, and the average Gap^b between the results obtained by the C & S and the BKS is 5.68%. The average Gap^c between the results obtained by the Gap^a and the Gap^b for the three types of instances is 2.76%. In the bold value of Table 3, the total distance of 9 instances, such as C201–100 and C202–100, is better than that of C & S.

5. Computational study

5.1. Simulation instances

In order to verify the applicability and effectiveness of the established PDVRPCS model and the designed solving framework, the data of 60 customers are randomly generated, as shown in the Appendix Tables A1–A3. For dynamic OCSs, the system first predicts whether it will generate service requests and plan delivery routes in advance. The first index depot represents the distribution center, the numbers SOC1-SOC30 represent the static OCSs that need service, and the indexes DOC1-DOC15 represent the dynamic customer that could request service.

The three parameters of customer's dependence on the enterprise a_1 , time windows of user a_2 and customer's historical demand a_3 are used to compute the prospect value for proactive evaluation of customer's demand in the current horizon. Each of them has a weight $\lambda_i, i \in \{1, 2, 3\}$, correspondingly. The value of the three fuzzy parameters is divided into five grades: poor, fair, satisfactory, good and excellent. The specific values are computed according to Eq (29), which is an exceptional case of Eq (3).

$$a_i = \left(\max\left(\frac{l-1}{4}, 0\right), \frac{l}{4}, \min\left(\frac{l+1}{4}, 0\right) \right), l \in [1, 4] \quad (32)$$

Using the customer evaluation indicator, the prospect value of customers can be calculated by Eq (33):

$$pv = \sum_{i=1}^3 \lambda_i a_i \quad (33)$$

According to the above three indicators, the prospect value of dynamic customers can be evaluated by Eq (33). The calculated results are normalized in Figure 7 ($\lambda = \{0.4, 0.3, 0.3\}$).

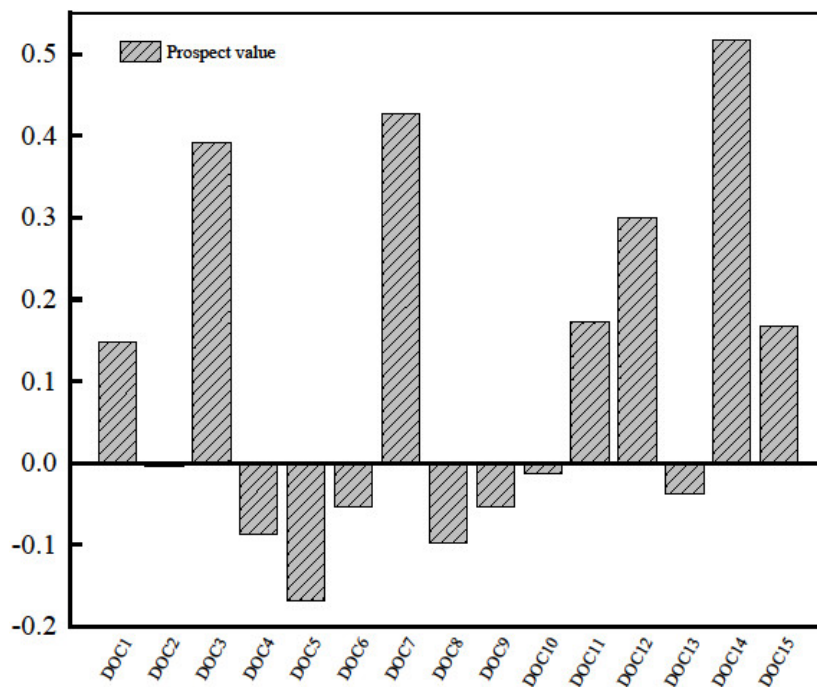


Figure 7. Predict the prospect values for dynamic customers.

The normalized data in Figure 7 show that eleven customers (DOC1, DOC3, DOC7, DOC11, DOC12, DOC14 and DOC15) have a positive prospect value. Therefore, these customers will be prioritized for planning distribution services.

The necessary parameters need to be introduced before using the proposed algorithm. Given that the capacity of the vehicle is 200, the initial cost of each vehicle is 200 Chinese Yuan (CNY), the unit variable cost is 5 CNY, the unit penalty cost of the vehicle violating the time window is 2 CNY and the driving speed is 30 km/h. If the cooperative distribution service is not considered (Scenario I), the routing plan of static OCS is carried out by GA-SA. The algorithm was run ten times, and the total cost of the optimal result is 4461 CNY. A total of five vehicles are needed to complete the task for thirty static OCSs, and the distribution plan is shown in Table 4.

Table 4. The delivery plan in Scenario I.

NO.	Tour
K1	0→SOC18→SOC10→SOC21→SOC3→SOC28→SOC14→SOC25→0
K2	0→SOC24→SOC9→SOC2→SOC20→SOC16→SOC6→SOC15→0
K3	0→SOC11→SOC29→SOC1→0
K4	0→SOC 26→SOC23→SOC5→SOC8→SOC30→SOC4→SOC19→0
K5	0→SOC13→SOC22→SOC12→SOC27→SOC17→SOC7→0

The collaborative service is not considered in Scenario II. Seven dynamic customers with positive proactive value (DOC1, DOC3, DOC7, DOC11, DOC12, DOC14 and DOC15) are renumbered SOC31, SOC32, SOC33, SOC34, SOC35, SOC36 and SOC37 as static customers, and their distribution services are fulfilled using enterprise vehicles. The total cost is 4659 CNY, and the distribution scheme is shown in Table 5.

Table 5. The dynamic delivery plan in Scenario II.

NO.	Tour
K1	0→SOC13→SOC27→SOC17→SOC35→SOC19→SOC37→SOC4→0
K2	0→SOC26→SOC23→SOC30→SOC8→SOC5→SOC10→SOC31→0
K3	0→SOC11→SOC29→SOC1→SOC22→SOC12→SOC7→SOC32→0
K4	0→SOC24→SOC28→SOC3→SOC21→SOC18→SOC36→SOC14→SOC25→0
K5	0→SOC34→SOC33→SOC9→SOC2→SOC20→SOC16→SOC6→SOC15→0

If collaborative services are not used, delivery services cannot be provided to customers who appear in the service process or fail in prediction. Therefore, collaborative services should be considered for the requirements of dynamic OCSs (Scenario III). Firstly, thirty static OCSs are matched with fifteen cooperative SCs. The matching algorithm is used under the conditions of compensation coefficient $\rho = 0.1$ and cooperation flexibility $\varepsilon = 1.5$. The result shows that eight orders of static OCSs can be served by cooperative SC. For the remaining 22 static OCS orders and the 11 dynamic OCSs with a prospect value greater than zero are renumbered 1–29. Then, GA-SA is run ten times, and the optimal distribution scheme obtained only needs four vehicles to complete the delivery tasks of these 29 customers, with a total delivery cost of 4043 CNY. The distribution plan is shown in Table 6.

Table 6. The dynamic delivery plan in Scenario III.

NO.	Tour
K1	0→SOC21→SOC14→SOC5→SOC19→SOC9→0
K2	0→SOC15→SOC3→SOC1→SOC29→SOC11→SOC27→SOC2→SOC24→0
K3	0→SOC23→SOC20→SOC6→SOC13→SOC22→SOC10→SOC28→SOC18→SOC17→0
K4	0→SOC16→SOC26→SOC25→SOC4→SOC12→SOC8→SOC7→0

The collaboration SCs are used to service new service requests that fail to predict or arise during the service process. Service is denied for new requests that cannot be successfully matched with the collaborating SCs. Assume that the system has new service requests numbered D2, D5, D8, D9 and D13. Four new requests and cooperative SC are successfully matched.

Compared with the above three scenarios, the use of collaborative services can reduce the number of vehicles on the basis of meeting the temporary requests of dynamic OCSs and better respond to the new requests generated in the delivery process. The comparison of cost for the three scenarios is shown in Table 7.

Table 7. The cost comparison under the three scenarios.

	NV	C1+C2+C3	C4	RPC	TC
Scenario I	5	4461	0	600	5061
Scenario II	5	4659	0	250	4909
Scenario III	4	4043	118	50	4211

1) NV: NO. of vehicle;

2) TC: Total cost;

3) RPC: the penalty cost for denial of service to customer.

Denial of service may reduce customer satisfaction and dependency, which is harmful to the long-term development of enterprises. Compared to these three scenarios, collaborative services can reduce distribution cost and improve delivery timeliness based on proactive predication.

In order to verify the influence of different compensation coefficients and different cooperative flexibility on the cost, multiple combinations of the two coefficients are used in the simulation case of Scenario III, As shown in Table 8, the matching number of cooperative vehicles and online orders increases if the cooperation flexibility value increases, resulting in the reduction of vehicle numbers used and total cost. In addition, the compensation cost of the cooperative vehicle increases along with the compensation coefficient increase, leading to an increase in the total cost. When that happens, total cost can no longer decrease, if $\rho = 0.3$.

Table 8. Scenario III sensitivity analysis.

	$\rho = 0.5$		$\rho = 0.1$		$\rho = 0.15$		$\rho = 0.2$	
	NV	TC	NV	TC	NV	TC	NV	TC
$\varepsilon = 1.1$	5	4351	5	4395	5	4758	5	4986
$\varepsilon = 1.3$	4	4172	4	4265	4	4527	4	4777
$\varepsilon = 1.5$	4	4031	4	4211	4	4325	4	4443

5.2. Real-world case study

To assess the potential savings of the collaborative service strategy, the proposed model and solution framework are applied for the distribution network of a large store-depot-integrated retailer in Chongqing city in the southwest of China. The retailer has 51 static OCSs, 24 dynamic OCSs that may make service requests and 37 SCs that are willing to collaborate. The detailed information has been listed in the Appendix Tables A4–A6, where the encoding of nodes is the same as in the previous section.

Similar to the result in Figure 3, the prospective value of customers is calculated from the predicted data as well as historical data for the three attributes of dynamic OCS. The results are shown in Figure 8, the customers (DOC1, DOC4, DOC5, DOC6, DOC10, DOC11, DOC13, DOC15, DOC16, DOC17, DOC18, DOC19, DOC21, DOC23 and DOC24) have a positive prospect value, and the customers (DOC2, DOC3, DOC7, DOC8, DOC9, DOC12, DOC14, DOC20 and DOC22) have a negative prospect value. Therefore, these customers will be prioritized for planning distribution services. A total of eleven static OCSs were successfully matched with cooperative SCs, namely SOC3, SOC8, SOC9, SOC12, SOC16, SOC18, SOC21, SOC24, SOC27 and SOC42, after matching between the static OCS and cooperative SC. The matching results are illustrated in Figure 9. Finally, the matched customers are removed from the static OSC set.

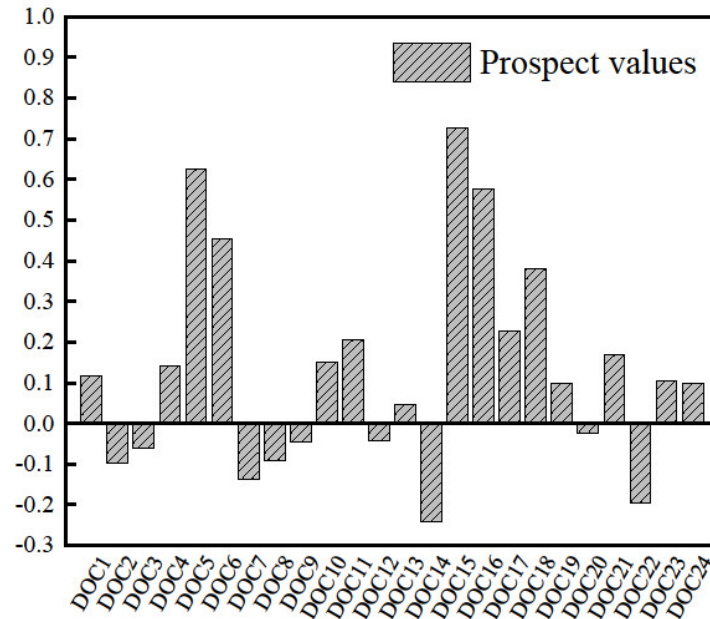


Figure 8. The prospect value for dynamic customers in the real-world case.

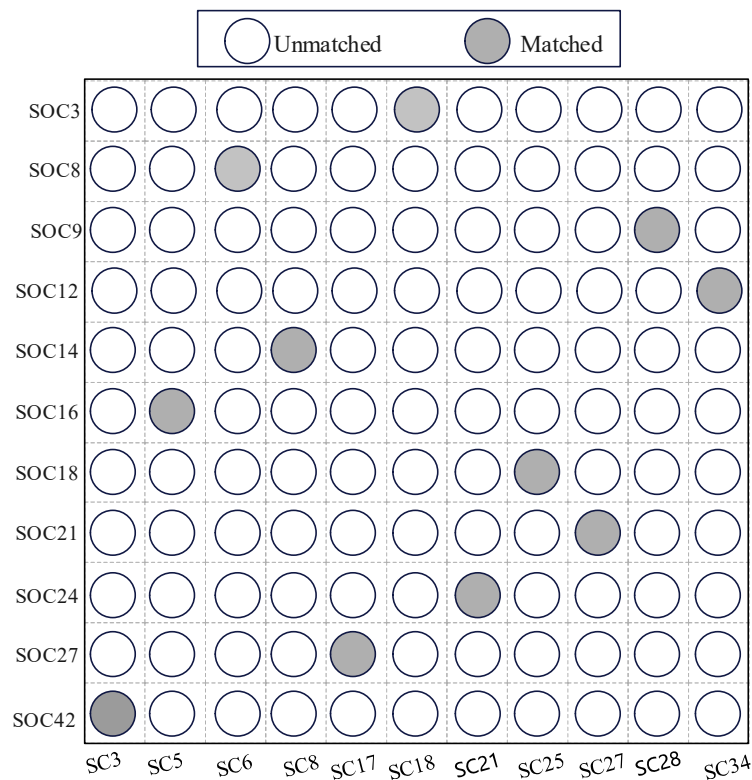


Figure 9. The matching result.

Considering the static distribution strategy of cooperative services (Scenario IV), all dynamic OCSs demands are rejected, and the penalty cost for rejections are included in total cost. The matching algorithm is employed on static customers (Figure 9) and cooperative SCs. The unmatched OCSs

(renumbered 1–40) are serviced using the enterprise vehicles.

Running GA-SA 10 times, an average of four vehicles is needed to complete the service for the 40 customers, and the average cost is 983 CNY. The total cost of the optimal result is 959 CNY, including 129 CNY of collaboration cost. The optimal distribution routes are shown in Table 9.

Table 9. The dynamic delivery plan in Scenario IV.

No.	Tour
K1	0→SOC17→SOC15→SOC39→SOC14→SOC30→SOC24→SOC12→SOC36→SOC2→SOC28→SOC1→SOC16→SOC37→0
K2	0→SOC27→SOC11→SOC31→SOC23→SOC13→SOC40→SOC4→SOC9→SOC10→SOC6→SOC18→2SOC1→SOC19→SOC20→SOC3→0
K3	0→SOC35→SOC22→SOC5→SOC8→SOC32→SOC25→SOC33→SOC34→SOC38→SOC29→26→7→0

In Scenario III, 40 static OCSs and 15 dynamic OCSs with positive prospect value are renumbered as 1–55. The solutions can be obtained by using GA-SA, in which the average number of vehicles is five, the average cost is 1124 CNY and the optimal cost is 1031 CNY. The optimal plan is shown in Table 10.

Table 10. The static delivery plan in Scenario III.

No.	Tour
K1	0→SOC17→SOC15→SOC43→SOC39→SOC14→SOC30→SOC24→SOC42→SOC12→SOC16→SOC9→SOC6→SOC46→0
K2	0→SOC36→SOC22→SOC5→SOC8→SOC51→SOC45→SOC32→SOC25→SOC54→SOC38→SOC53→SOC50→SOC29→SOC26→0
K3	0→SOC27→SOC11→SOC31→SOC49→SOC23→SOC13→SOC48→SOC40→SOC55→SOC4→SOC10→SOC52→SOC18→SOC37→0
K4	0→SOC44→SOC3→SOC47→SOC21→SOC19→SOC20→SOC1→SOC28→SOC2→SOC41→SOC35→SOC34→SOC33→SOC7→0

During the distribution process, the dynamic OCSs that could not be predicted successfully are DOC2, DOC7, DOC8, DOC14, DOC20 and DOC22. The match algorithm is used between these nine customers and cooperative SCs, and the matching result shows that DOC8, DOC14 and DOC22 are successfully matched. Therefore, the matched customers are served by collaborative SC with a compensation cost of 39 CNY. Dynamic customers (DOC2, D7 and DOC20) failed to match with cooperative SC and are rejected to be served. Finally, the total compensation cost in the real-world case is 168 CNY under Scenario III. The comparison of cost for the two scenarios is shown in Table 11.

Table 11. The cost comparison under Scenarios III and IV.

	NV	C1+C2+C3	C4	RPC	TC	Gap ^d
Scenario IV	3	830	129	1050	2009	34.64%
Scenario III	4	1124	39	150	1313	

$$1) \text{ Gap}^d = \frac{(\text{ScenarioIV} - \text{ScenarioIII})}{\text{ScenarioIV}} \times 100\%$$

As can be seen from Table 8, Scenario IV has low cost related to vehicles, because the number of vehicles used in this scenario decreases when all dynamic customers are rejected. In addition, because 21 dynamic customers are rejected in this scenario, the penalty cost is high, resulting in a higher total cost than in Scenario III. In the case that the denial cost is set to 50 CNY per customer, Scenario III saves 34.64% of the cost compared to Scenario IV, which can solve the problem of customer satisfaction and loyalty reduction caused by service rejection.

In this study, sensitivity analyses for uncertainties and key parameters have been performed in mathematical formulas. With different degree of collaboration flexibility ε of cooperative vehicle and compensation factors ρ of the cooperative vehicle, the average of all target cost functions is calculated to prove their effect. Table 12 shows the results of the analysis under different conditions, in which the variation range of cooperation flexibility degree is 1.1–1.5, and the variation range of compensation factors is 0.05–0.2. When the degree of cooperation increases, the total cost of the objective function decreases, and when the compensation factor increases, the total cost of the objective function increases. This is because when the degree of collaboration increases, more customers participate in the collaboration, resulting in cost savings. When the compensation factor increases, the subsidy to temporary deliverers will increase, which will also increase the total cost.

Table 12. The sensitivity analysis in the real-world case study.

	$\rho = 0.5$		$\rho = 0.1$		$\rho = 0.15$		$\rho = 0.2$	
	Scenario III	Scenario IV	Scenario III	Scenario IV	Scenario III	Scenario IV	Scenario III	Scenario IV
$\varepsilon = 1.1$	1347	2056	1345	2067	1358	2067	1372	2067
$\varepsilon = 1.3$	1320	2031	1323	2039	1364	2039	1369	2045
$\varepsilon = 1.5$	1317	2116	1313	2009	1327	2026	1348	2033

6. Managerial insights

Under the trend of the rapid development of a sharing economy, information technology and intra-city express business, crowdsourcing intra-city delivery decreases the distance of serving customers by integrating goods into stores and warehouses, compared with the traditional intra-city delivery problem. Due to the unpredictability of when orders will arrive, logistics companies struggle to plan ahead. To meet customers' needs as much as possible during peak demand periods, logistics companies may compensate crowd delivery workers on platforms to share the delivery pressure. However, even with this approach, the delivery capacity remains insufficient. As a result, challenges such as insufficient capacity during peak demand periods, difficulties in meeting customer time windows and balancing subsidy amounts for different stakeholders remain obstacles faced by integrated store-warehouse crowd delivery. In this situation, this paper proposes a new linear programming model to solve the problems of order demand prediction, line and time matching and compensation distribution of crowdsourcing deliverers in the process of crowdsourcing collaborative distribution.

This study raised a new mechanism for cooperative routing planning, especially for crowded cities with intense last-mile transport effort. The “crowdsourcing” approach was expected to exploit tacit capacities at low additional cost, simultaneously saving expenses and contributing to reduce

traffic and, thus, congestion. At the same time, the authors proposed a suitable forecasting method, an order matching algorithm to coordinate the different participants, and a genetic simulated annealing algorithm for the system's optimization. The approach is relevant and also up to date with current distribution developments. The solution approach, especially in its combination of methods, is novel.

7. Conclusions and future works

With the rapid development of the digital economy, online shopping has become a habit of the public. The demand for small-batch, low-weight and multi-frequency distribution has witnessed explosive growth. At the same time, distribution price, service quality and timeliness are facing unprecedented tests. To reduce the distribution cost of store-depot-integrated retailers, this study presented a PDVRPCS model and proposed a novel solution framework considering the cooperation service provided by scattered SCs. The solution framework includes three stages, i.e., customers' prospect value prediction based on historical data, a matching algorithm and a hybrid GA-SA heuristic algorithm. Finally, the feasibility and effectiveness of the model and solution framework are verified under different scenarios.

However, there are still some weaknesses in this work, and more research could be done in the future to overcome these weaknesses. First, the proposed model only considers the economic objective. Multi-objective models (e.g., carbon emissions, considering the balance between the number of deliverers and customer satisfaction) could be developed by splitting the single objective into multiple objectives. Second, more prediction algorithms (e.g., statistics-based approaches, machine learning-based approaches or deep learning-based approaches) should be considered to accurately predict the customers' demand. Third, multiple depots and heterogeneous fleets could be considered for more flexibility. When electric vehicle riders or electric private cars participate in the distribution, how to cooperate with their own fleet is worth considering.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by the humanities and social science research project of Chongqing Education Commission [grant number 20JD059] and the Chongqing Jiaotong University graduate scientific research innovation project [grant number 2022S0069]. The authors are grateful to the support of the general project of Chongqing Natural Science Foundation [grant number cstc2020jcyj-msxmX0108], and the key scientific and technological innovation project of "Construction of Chengdu-Chongqing Economic Circle" [grant number KJCXZD2020031].

Conflict of interest

The authors declare there is no conflict of interest.

References

1. N. Bose, Wal-Mart's next move against Amazon: more warehouses, faster shipping, *Technol. News*, 2016.
2. A. H. Hübner, H. Kuhn, J. Wollenburg, Last mile fulfilment and distribution in omni-channel grocery retailing: a strategic planning framework, *Int. J. Retail Distrib. Manage.*, **44** (2016). <https://doi.org/10.1108/IJRDM-11-2014-0154>
3. G. Bensinger, Amazon's next delivery drone: you, in *Wall Street Journal*, **108** (2015).
4. B. W. Thomas, Waiting strategies for anticipating service requests from known customer locations, *Transp. Sci.*, **41** (2007), 281–429. <https://doi.org/10.1287/trsc.1060.0183>
5. C. Lima, B. M. Luz, S. T. Takemoto, P. Barisson, R. Tezzin, L. Peres, et al., Strategic modeling for the characterization of the conditions that allow the anticipation of the consumer's requests, *Open J. Social Sci.*, **3** (2015), 146–160. <https://doi.org/10.4236/jss.2015.310021>
6. K. S. Hartzel, C. A. Wood, Factors that affect the improvement of demand forecast accuracy through point-of-sale reporting, *Eur. J. Oper. Res.*, **260** (2017), 171–182. <https://doi.org/10.1016/j.ejor.2016.11.047>
7. S. Ma, R. Fildes, A retail store SKU promotions optimization model for category multi-period profit maximization, *Eur. J. Oper. Res.*, **260** (2017), 680–692. <https://doi.org/10.1016/j.ejor.2016.12.032>
8. S. Zhu, R. Dekker, W. van Jaarsveld, R. W. Renjie, A. J. Koning, An improved method for forecasting spare parts demand using extreme value theory, *Eur. J. Oper. Res.*, **261** (2017), 169–181. <https://doi.org/10.1016/j.ejor.2017.01.053>
9. R. Belka, M. Godlewski, Vehicle routing optimization system with smart ge positioning updates, *Appl. Sci.*, **11** (2021), 10933. <https://doi.org/10.3390/app112210933>
10. F. Ferrucci, S. Bock, Pro-active real-time routing in applications with multiple request patterns, *Eur. J. Oper. Res.*, **253** (2016), 356–371. <https://doi.org/10.1016/j.ejor.2016.02.016>
11. A. M. Fathollahi-Fard, A. Ahmadi, F. Goodarzian, N. Cheikhrouhou, A bi-objective home healthcare routing and scheduling problem considering patients' satisfaction in a fuzzy environment, *Appl. Soft Comput.*, **93** (2020), 106385. <https://doi.org/10.1016/j.asoc.2020.106385>
12. E. Fernández, M. Roca-Riu, M. G. Speranza, The shared customer collaboration vehicle routing problem, *Eur. J. Oper. Res.*, **265** (2018), 1078–1093. <https://doi.org/10.1016/j.ejor.2017.08.051>
13. J. Paul, N. Agatz, R. Spliet, R. D. Koster, Shared capacity routing problem – an omni-channel retail study, *Eur. J. Oper. Res.*, **273** (2019), 731–739. <https://doi.org/10.1016/j.ejor.2018.08.027>
14. C. Archetti, M. Savelsbergh, M. G. Speranza, The vehicle routing problem with occasional drivers, *Eur. J. Oper. Res.*, **254** (2016), 472–480. <https://doi.org/10.1016/j.ejor.2016.03.049>
15. G. Macrina, L. Pugliese, F. Guerriero, D. Laganà, The vehicle routing problem with occasional drivers and time windows, in *ODS 2017: Optimization and Decision Science: Methodologies and Applications*, (2017), 577–587. https://doi.org/10.1007/978-3-319-67308-0_58
16. A. M. Arslan, N. Agatz, L. Kroon, R. Zuidwijk, Crowdsourced delivery: a dynamic pickup and delivery problem with Ad-hoc drivers, *Transp. Sci.*, **53** (2016), 1–318. <https://doi.org/10.1287/trsc.2017.0803>
17. M. Mojtahedi, A. Fathollahi-Fard, R. Tavakkoli-Moghaddam, S. Newton, Sustainable vehicle routing problem for coordinated solid waste management, *J. Ind. Inf. Integr.*, **23** (2021), 100220. <https://doi.org/10.1016/j.jii.2021.100220>

18. A. Fathollahi-Fard, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, N. Smith, Bi-level programming for home health care supply chain considering outsourcing, *J. Ind. Inf. Integr.*, **25** (2022), 100246. <https://doi.org/10.1016/j.jii.2021.100246>
19. Y. Wang, Y. Yuan, K. Assogba, K. Gong, H. Wang, M. Xu, et al., Design and profit allocation in two-echelon heterogeneous cooperative logistics network optimization, *J. Adv. Transp.*, **2018** (2018), 4607493. <https://doi.org/10.1155/2018/4607493>
20. I. Dayarian, M. Savelsbergh, Crowdshipping and same-day delivery: employing in-store customers to deliver online orders, *Prod. Oper. Manage.*, **29** (2020), 2153–2174. <https://doi.org/10.1111/poms.13219>
21. S. Luo, Y. Wang, J. Tang, X. Guan, M. Xu, Two-echelon multidepot logistics network design with resource sharing, *J. Adv. Transp.*, **2021** (2021), 6619539. <https://doi.org/10.1155/2021/6619539>
22. X. Ren, X. Jiang, L. Ren, L. Meng, A multi-center joint distribution optimization model considering carbon emissions and customer satisfaction, *Math. Biosci. Eng.*, **20** (2023), 683–706. <https://doi.org/10.3934/mbe.2023031>
23. K. Gdowska, A. Viana, J. P. Pedroso, Stochastic last-mile delivery with crowdshipping, *Transp. Res. Procedia*, **30** (2018), 90–100. <https://doi.org/10.1016/j.trpro.2018.09.011>
24. S. Y. Tan, W. C. Yeh, The vehicle routing problem: state-of-the-art classification and review, *Appl. Sci.*, **11** (2021), 10295. <https://doi.org/10.3390/app112110295>
25. R. A. Russell, W. C. Chiang, Scatter search for the vehicle routing problem with time windows, *Eur. J. Oper. Res.*, **169** (2006), 606–622. <https://doi.org/10.1016/j.ejor.2004.08.018>
26. Q. Zhang, T. Zhen, Y. Zhu, W. Zhang, Z. Ma, A hybrid intelligent algorithm for the vehicle routing with time windows, in *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, (2008), 47–54. https://doi.org/10.1007/978-3-540-87442-3_7
27. C. B. Cheng, K. P. Wang, Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm, *Expert Syst. Appl.*, **36** (2009), 7758–7763. <https://doi.org/10.1016/j.eswa.2008.09.001>
28. T. Vidal, T. G. Crainic, M. Gendreau, C. Prins, A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows, *Comput. Oper. Res.*, **40** (2013), 475–489. <https://doi.org/10.1016/j.cor.2012.07.018>
29. S. Belhaiza, P. Hansen, G. Laporte, A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows, *Comput. Oper. Res.*, **52** (2014), 269–281. <https://doi.org/10.1016/j.cor.2013.08.010>
30. Y. Shi, L. Lv, F. Hu, Q. Han, A heuristic solution method for multi-depot vehicle routing-based waste collection problems, *Appl. Sci.*, **10** (2020), 2403. <https://doi.org/10.3390/app10072403>
31. J. Pasha, A. L. Nwodu, A. Fathollahi-Fard, G. Tian, Z. Li, H. Wang, et al., Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings, *Adv. Eng. Inf.*, **52** (2022), 101623. <https://doi.org/10.1016/j.aei.2022.101623>
32. A. Fathollahi-Fard, A. Ahmadi, B. Karimi, Multi-objective optimization of home healthcare with working-time balancing and care continuity, *Sustainability*, **13** (2021), 12431. <https://doi.org/10.3390/su132212431>
33. R. Baños, J. Ortega, C. Gil, A. Márquez, F. de Toro, A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows, *Comput. Ind. Eng.*, **65** (2013), 286–296. <https://doi.org/10.1016/j.cie.2013.01.007>

34. E. Pourrahmani, M. Jaller, Crowdsipping in last mile deliveries: operational challenges and research opportunities, *Socio-Econ. Plann. Sci.*, **78** (2021), 101063. <https://doi.org/10.1016/j.seps.2021.101063>
35. M. Jaller, C. Otero-Palencia, A. Pahwa, Automation, electrification, and shared mobility in urban freight: opportunities and challenges, *Transp. Res. Procedia*, **46** (2020), 13–20. <https://doi.org/10.1016/j.trpro.2020.03.158>
36. W. Dai, Q. Zhong, C. Qi, Multi-stage multi-attribute decision-making method based on the prospect theory and triangular fuzzy MULTIMOORA, *Soft Comput.*, **24** (2020), 9429–9440. <https://doi.org/10.1007/s00500-018-3017-0>
37. D. Kahneman, A. Tversky, Prospect theory: an analysis of decision under risk, in *Handbook of the Fundamentals of Financial Decision Making*, World Scientific, (2013), 99–127. https://doi.org/10.1142/9789814417358_0006
38. S. Datta, N. Sahu, S. Mahapatra, Robot selection based on grey-MULTIMOORA approach, in *Grey Systems: Theory and Application*, **3** (2013). <https://doi.org/10.1108/GS-05-2013-0008>
39. X. Ge, Y. Jin, Chapter 18 - Artificial intelligence algorithms for proactive dynamic vehicle routing problem, in *Applications of Artificial Intelligence in Process Systems Engineering*, (2021), 497–522. <https://doi.org/10.1016/B978-0-12-821092-5.00011-5>
40. A. K. Jain, Y. Zhou, T. Mustufa, E. C. Burdette, G. S. Chirikjian, G. Fichtinger, Matching and reconstruction of brachytherapy seeds using the Hungarian algorithm (MARSHAL), *Med. Phys.*, **32** (2005), 3475–3492. <https://doi.org/10.1118/1.2104087>
41. A. Mirzaeinia, S. Bradley, M. Hassanalian, Drone-station matching in smart cities through Hungarian algorithm: power minimization and management, in *AIAA Propulsion and Energy 2019 Forum*, **2019** (2019), 4151. <https://doi.org/10.2514/6.2019-4151>
42. G. A. Mills-Tettey, A. Stentz, M. B. Dias, The dynamic Hungarian algorithm for the assignment problem with changing costs, 2007. Available from: https://www.ri.cmu.edu/pub_files/pub4/mills_tettey_g_ayorkor_2007_3/mills_tettey_g_ayorkor_2007_3.pdf.
43. M. R. Chen, Y. Z. Lu, A novel elitist multiobjective optimization algorithm: multiobjective extremal-optimization, *Eur. J. Oper. Res.*, **188** (2008), 637–651. <https://doi.org/10.1016/j.ejor.2007.05.008>
44. J. Chen, J. Shi, A multi-compartment vehicle routing problem with time windows for urban distribution – a comparison study on particle swarm optimization algorithms, *Comput. Ind. Eng.*, **133** (2019), 95–106. <https://doi.org/10.1016/j.cie.2019.05.008>

Appendix

Table A1. The information of SOCs in Simulation instances.

Name	Coordinate	Demand	Time windows
Depot	(50,50)	0	[8:00, 18:00]
SOC1	(02,06)	20	[10:00, 12:30]
SOC2	(96,13)	10	[10:00, 13:00]
SOC3	(86,73)	55	[12:30, 15:00]
SOC4	(07,81)	30	[10:30, 13:30]
SOC5	(30,93)	18	[11:30, 14:00]
SOC6	(70,28)	39	[11:30, 15:00]
SOC7	(14,51)	9	[11:00, 13:30]
SOC8	(22,99)	8	[12:00, 15:00]
SOC9	(99,36)	18	[8:30, 11:00]
SOC10	(70,85)	32	[10:30, 13:30]
SOC11	(45,46)	17	[9:00, 12:00]
SOC12	(07,46)	45	[12:00, 15:30]
SOC13	(35,44)	26	[8:00, 10:30]
SOC14	(78,69)	11	[13:00, 16:30]
SOC15	(60,31)	49	[9:30, 12:30]
SOC16	(72,06)	26	[12:00, 15:30]
SOC17	(02,51)	42	[8:00, 11:30]
SOC18	(86,96)	7	[10:30, 13:30]
SOC19	(17,66)	23	[9:30, 13:00]
SOC20	(87,06)	23	[8:30, 12:00]
SOC21	(88,80)	20	[9:00, 12:00]
SOC22	(10,45)	10	[12:30, 16:00]
SOC23	(37,79)	55	[9:00, 12:00]
SOC24	(64,44)	30	[8:30, 11:00]
SOC25	(62,61)	32	[8:30, 11:00]
SOC26	(45,74)	48	[10:00, 13:00]
SOC27	(03,50)	58	[11:30, 14:00]
SOC28	(88,68)	22	[9:30, 12:30]
SOC29	(32,13)	36	[11:00, 13:30]
SOC30	(48,94)	10	[9:00, 12:30]

Table A2. The information of DOCs in Simulation instances.

Name	Coordinate	Demand	Time windows
Depot	(50,50)	0	[8:00, 18:00]
DOC1	(72,54)	18	[11:00, 14:30]
DOC2	(96,05)	29	[11:00, 14:00]
DOC3	(15,51)	40	[10:30, 14:00]
DOC4	(94,47)	20	[8:00, 10:30]
DOC5	(96,94)	16	[11:30, 14:00]
DOC6	(63,13)	33	[9:30, 12:30]
DOC7	(86,36)	12	[11:30, 14:00]
DOC8	(45,67)	49	[10:00, 13:30]
DOC9	(62,36)	44	[11:30, 14:00]
DOC10	(99,22)	17	[8:00, 10:30]
DOC11	(77,41)	10	[9:00, 12:00]
DOC12	(05,65)	8	[9:00, 12:30]
DOC13	(68,38)	46	[10:30, 13:00]
DOC14	(89,99)	21	[12:00, 15:00]
DOC15	(09,80)	13	[10:00, 13:00]

Table A3. The information of cooperative SCs in simulation instances.

Name	Coordinate	Time windows
Depot	(50,50)	[8:00, 18:00]
SC1	(80,56)	[9:30, 12:00]
SC2	(86,4)	[8:30, 11:30]
SC3	(86,84)	[13:00, 16:00]
SC4	(81,37)	[11:00, 13:30]
SC5	(89,58)	[13:00, 16:00]
SC6	(11,52)	[9:30, 12:30]
SC7	(46,33)	[12:30, 15:30]
SC8	(22,9)	[8:30, 11:30]
SC9	(47,31)	[9:30, 13:00]
SC10	(99,36)	[11:00, 14:30]
SC11	(97,24)	[11:30, 15:00]
SC12	(63,11)	[12:30, 15:00]
SC13	(55,59)	[8:30, 11:00]
SC14	(25,21)	[10:00, 12:30]
SC15	(25,35)	[10:30, 14:00]

Table A4. The information of SOCs in Real-world case.

Name	Longitude	Latitude	Demand	Open time	Close time
Depot	106.522862140604	29.513198404445	0	8:00	18:00
SOC1	106.523045674727	29.523506674453	40	10:12	12:00
SOC2	106.520957260541	29.521157533104	10	10:06	11:54
SOC3	106.526957373369	29.515610121643	20	8:54	11:42
SOC4	106.524168428741	29.518607770191	30	8:00	9:48
SOC5	106.533490387547	29.518628220723	20	11:00	12:48
SOC6	106.513119904097	29.521118714733	10	9:24	11:12
SOC7	106.531411086193	29.519339320680	30	9:54	11:42
SOC8	106.516690508948	29.515822662880	20	8:12	10:00
SOC9	106.513715803590	29.513521533101	20	9:06	10:54
SOC10	106.513230719511	29.512570776048	20	9:12	11:00
SOC11	106.514390966424	29.520002913394	30	8:00	9:48
SOC12	106.530527783225	29.523585064248	20	8:54	10:42
SOC13	106.533222350287	29.523223614599	10	9:00	10:42
SOC14	106.529578789294	29.523028210778	10	10:18	12:06
SOC15	106.532520356940	29.520451088096	40	8:18	10:06
SOC16	106.533077228214	29.519665371899	10	8:54	10:42
SOC17	106.532099552091	29.512525950047	30	9:42	11:30
SOC18	106.533469279931	29.510699000361	20	9:06	10:54
SOC19	106.518132239778	29.518010871585	40	10:54	12:42
SOC20	106.536379401717	29.512294092867	30	8:12	10:00
SOC21	106.516061766446	29.521586011335	30	9:06	10:54
SOC22	106.517867222872	29.514769686522	30	9:00	10:48
SOC23	106.515184470864	29.512040253419	20	9:54	10:42
SOC24	106.525941024655	29.519036909487	10	10:18	12:06
SOC25	106.521566662263	29.519858124933	40	11:00	12:48
SOC26	106.516689073223	29.511981233502	30	8:12	10:00
SOC27	106.527958890999	29.517429091577	10	10:48	12:36
SOC28	106.529881042543	29.517142270987	30	9:12	11:00
SOC29	106.527198514963	29.517905440100	20	8:06	9:54
SOC30	106.526472357789	29.518077346732	20	10:12	12:00
SOC31	106.526544205215	29.516820173226	20	9:00	10:48
SOC32	106.515576707539	29.523090655122	40	9:48	11:36
SOC33	106.536258142184	29.511036850523	40	9:30	11:18
SOC34	106.517719878539	29.516416812887	30	10:24	12:12
SOC35	106.509812769532	29.518024165314	20	9:36	11:24
SOC36	106.512332457272	29.514727774792	40	9:24	11:12
SOC37	106.530244773521	29.511763766904	20	10:48	12:36
SOC38	106.522201331053	29.522717090721	20	10:36	12:24
SOC39	106.511429674085	29.514672831563	10	10:42	12:30
SOC40	106.518410674819	29.515319687559	30	8:24	10:12

Continued on next page

Name	Longitude	Latitude	Demand	Open time	Close time
SOC41	106.533464802465	29.512981682112	20	9:54	10:42
SOC42	106.520651838760	29.518505756413	20	9:54	11:42
SOC43	106.512080968104	29.518177226386	30	9:06	10:54
SOC44	106.511811473256	29.517265788852	20	9:36	11:24
SOC45	106.512745691435	29.516951428043	20	10:42	12:30
SOC46	106.515293722507	29.518435334626	10	10:18	12:06
SOC47	106.516048276831	29.519217085237	40	8:54	10:42
SOC48	106.525720927872	29.514848936607	20	10:36	12:24
SOC49	106.512184259663	29.516782530979	20	8:18	10:06
SOC50	106.514941953604	29.513474310386	30	8:36	10:24
SOC51	106.534547120111	29.513842080333	10	9:36	11:24

Table A5. The information of DOCs in Real-world case.

Name	Longitude	Latitude	Demand	Open time	Close time
Depot	106.522862140604	29.513198404445	0	8:00	18:00
DOC1	106.518208610162	29.520458391367	10	10:12	12:00
DOC2	106.527505262168	29.513244983224	40	9:42	11:30
DOC3	106.532836107154	29.518957262837	20	8:06	9:54
DOC4	106.515249653102	29.517477710011	10	8:00	9:48
DOC5	106.514277212228	29.512114957646	30	8:06	9:54
DOC6	106.523696855353	29.518042061637	20	8:42	10:36
DOC7	106.531985944584	29.522203204066	40	8:24	10:12
DOC8	106.515338595672	29.514176580304	20	8:00	9:48
DOC9	106.532485791656	29.515272177184	40	8:24	10:12
DOC10	106.511564456960	29.519061209647	20	10:48	12:36
DOC11	106.527706006345	29.518353294879	40	10:12	12:00
DOC12	106.526270273468	29.520025935395	40	9:30	11:18
DOC13	106.524716341324	29.518081318041	20	8:30	9:48
DOC14	106.529009779204	29.515912600921	10	10:24	12:12
DOC15	106.534837630872	29.512813693492	10	8:06	9:54
DOC16	106.535777619951	29.511995508946	40	10:54	12:42
DOC17	106.510293325074	29.514248594080	40	10:42	12:30
DOC18	106.513510650659	29.519951901632	40	10:18	12:06
DOC19	106.532081620434	29.517519396293	40	9:54	11:42
DOC20	106.518236386407	29.517391103934	40	10:00	11:48
DOC21	106.511290454740	29.516303285748	10	8:12	10:00
DOC22	106.512525600826	29.515863181463	30	8:42	10:24
DOC23	106.511339869671	29.517285464821	20	8:30	10:18
DOC24	106.533797142095	29.514851785999	40	8:06	9:54

Table A6. The information of SCs in Real-world case.

Name	Longitude	Latitude	Demand	Open time	Close time
SC1	106.523243278648	29.522241706584	21	8:06	9:54
SC2	106.517894226174	29.521503400645	29	10:18	12:06
SC3	106.519987145697	29.519700084375	31	10:06	11:54
SC4	106.525622203719	29.528001567827	14	9:54	11:42
SC5	106.533833086076	29.517786523332	39	8:54	10:42
SC6	106.517216005429	29.517052324442	26	9:48	11:36
SC7	106.511416178153	29.512755551509	23	10:30	12:18
SC8	106.530303228537	29.522492963999	24	10:42	12:30
SC9	106.531520213846	29.511952340775	28	9:48	11:36
SC10	106.519951239667	29.523730731395	11	8:24	10:12
SC11	106.529009762529	29.513747850901	25	9:36	11:24
SC12	106.531460458671	29.514966713821	11	10:48	12:36
SC13	106.524648977280	29.518427039643	37	8:48	10:36
SC14	106.525738918597	29.518054759534	38	10:48	12:36
SC15	106.522210296563	29.520018202415	36	9:00	10:48
SC16	106.529342133263	29.518666589976	26	10:30	12:18
SC17	106.527319763741	29.516640406660	40	10:54	12:42
SC18	106.528080139981	29.516242626758	30	9:30	11:18
SC19	106.531915448329	29.516419369550	29	8:06	9:54
SC20	106.525884022092	29.517464490350	28	10:54	12:42
SC21	106.526390129689	29.518683324806	19	10:36	12:24
SC22	106.535009671012	29.511264746471	37	9:54	11:42
SC23	106.513066839468	29.518216445314	14	9:12	11:00
SC24	106.530644484899	29.513559248846	33	10:36	12:24
SC25	106.533065101648	29.511771596871	21	9:54	11:42
SC26	106.515181453514	29.520560718826	24	9:36	11:24
SC27	106.517027403651	29.521578093926	31	9:12	11:00
SC28	106.513176845422	29.514590210846	31	9:06	10:54
SC29	106.517732489872	29.515359010914	16	10:24	12:12
SC30	106.520286656447	29.518557813158	39	8:36	10:24
SC31	106.510369693502	29.515517594701	12	8:12	10:00
SC32	106.515176952514	29.519166068497	33	8:42	10:30
SC33	106.516159175331	29.520384838481	37	9:00	10:48
SC34	106.528249448395	29.522635389037	30	10:24	12:12
SC35	106.516172583396	29.513179573040	33	8:06	9:54
SC36	106.534268692654	29.515555025351	35	8:48	10:36
SC37	106.535508171036	29.513119164913	12	9:36	11:24



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)