An Improved Particle Swarm Optimization Algorithm for the Distribution of Fresh Products

Yane Hou, Chunxiao Wang, Weichuan Dong and Lanxue Dang

Abstract—**In the application field of fresh products distribution, it is necessary to use multi-compartment vehicles for distribution because of their particular demands for temperature. This paper studied the multi-compartment vehicle routing problem with soft time windows for multiple fresh products distribution. Firstly, a mathematical model of the issued problem was built, which aims to minimize the total cost including vehicle cost, delivery cost, refrigeration cost, damage cost and penalty cost of delivery time. Then, we presented an improved particle swarm optimization algorithm to solve this problem. In the process of particle updating, the sequential crossover operator usually used in genetic algorithm was introduced to enhance the diversity of particles. Finally, the proposed algorithm was evaluated on some benchmark instances, and the experiment results demonstrate its effectiveness and good stability, when compared with genetic algorithm and simulated annealing algorithm. It can draw a conclusion that the proposed algorithm can provide a reliable and stable solution approach for the distribution of fresh products.**

Index Terms—**fresh products distribution, multi-compartment vehicle, particle swarm optimization, sequential crossover operator, soft time window.**

I. Introduction

WITH the continuous improvement of people's living standard, high-quality fresh products is the most important thing for people. People are more and more concerned about the quality of fresh products. Because the fresh products are easy to deteriorate, it is a challenging job to build a high-quality fresh products distribution and logistics system. Generally speaking, fresh goods are generally divided into room, refrigeration and freezing three types according by their requirements of temperature. The traditional vehicles with a single compartment may no longer meet the needs of actual distribution. It has become a trend to use the vehicles with multiple compartments to delivery fresh goods. Therefore, it is of great significance to plan the delivery routes of multi-compartment vehicles for fresh products to reduce transportation costs and energy consumption.

As a variant of vehicle routing problem (VRP) [1], multi-compartment vehicle routing problem (MCVRP) [2] is a NP-hard problem with extremely high solving complexity. MCVRP uses multi-compartment vehicles to transport multiple incompatible products simultaneously, which is different from the classical VRP. For each compartment, it has a certain capacity constraint. Meanwhile, one type of product is only placed on a compartment of the multi-compartment vehicles. MCVRP has been widely used in many fields, such as garbage classification [3],[4], retail store cargo transportation [5], cold chain distribution [6], etc. The recent review of MCVRP was summarized in [7].

Fresh products distribution belongs to an application of cold chain transportation, which uses a set of cold chain vehicles to deliver multiple products with different temperature requirements simultaneously, and each type of product is placed in different compartments on the same vehicle. Chen et al. [8] assumed that the relation between vehicle compartment and product was one-to-one corresponding, and then established a mathematical model of the multi-compartment cold chain distribution of fresh products. For the retail store cargo distribution, the weight and volume constraints were considered by [9] and then proposed a method based on Lagrange Relaxation to minimize the routing cost and refrigeration cost. After that, a hybrid algorithm based on metaheuristics known as greedy randomized adaptive search procedure and variable neighborhood search for the vehicle routing problem with time window(VRPTW) with fuzzy travel times were developed by [10] for the multiple fuzzy time windows in cold chain transportation. These researches promote the development of cold chain transportation research to a certain extent.

However, with the increasing demand of customers, the time windows of customers becomes a new constraint in the research of cold chain transportation. For MCVRP, it derives a new variant of MCVRP, which is multi-compartment vehicle routing problem with time windows. The time window constraint is generally divided into hard time window and soft time window. If the customers have hard time windows, it means that the vehicle must serve the customer within its specified time window. While for the soft time constraint of the customer, it does not require the vehicle visit the customer within its time window. But if the vehicle arrives at the customer outside the time window, there will produce penalty costs. There are some literatures about the MCVRP with hard time windows [11],[12]. Because of its complexity, the research about MCVRP with soft time window is still rare. Therefore, it is necessary to carry out a research on multi-compartment vehicle routing problem with soft time window (MCVRPSTW).

In view of this, we take the MCVRPSTW used in fresh product distribution as the research object in this paper. The mathematical model of the issued problem is firstly built. And then, inspired by the good performance of particle swarm optimization for combinatorial optimization problems

Manuscript received November 2, 2022, revised February 9, 2023. This research has been supported by key project of colleges and universities of Henan Province (23A520014).

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

Chunxiao Wang is a graduate student of College of Computer and Information Engineering, Henan University, Kaifeng, 475004, China (e-mail:henuwcx@henu.edu.cn)

Weichuan Dong is a doctor of Department of Geography, Kent State University, Kent, OH, 44240, USA (e-mail:wdong@kent.edu)

Lanxue Dang is a professor of College of Computer and Information Engineering, Henan University, Kaifeng, 475004, China (Corresponding author,e-mail:danglx@vip.henu.edu.cn)

[13],[14], this paper designed an improved particle swarm algorithm combined with sequential crossover operator (named SCOPSO) to solve the MCVRPSTW problem. The developed algorithm was tested on classical instances and compared with two existing metaheuristics to verify its effectiveness.

The rest of this paper is organized as follows. Section II gives the description of the problem and its mathematical model. Section III describes the design of the proposed algorithm. Section IV evaluates and analyzes the performance of SCOPSO algorithm. The fifth section gives the conclusion and points out the research direction in future.

II. Problem Description and Model

A. Problem Description

In the multi-compartment vehicles logistics distribution system, different fresh products should be placed in the different compartments according to its temperature requirements. For the three types of fresh products, there are room temperature goods, refrigerated good and frozen goods. It is assumed that the fresh products at room temperature and refrigerated are placed in a compartment, where the refrigerated products are packed with refrigerated foam boxes, and the frozen fresh products are stored in separate compartments. In this way, different kinds of fresh products are placed in different compartments, which can not only deliver different kinds of fresh products by the same vehicle, but also improve the utilization rate of vehicles and reduce the distribution cost.

In this paper, the research problem is described as a multi-compartment vehicle distribution problem with soft time window. Suppose there is a distribution center (that is depot) and a set of customer nodes needed to be served, which both have the location information. A series of multi-compartment vehicles with the same compartment and capacity are located at the center. Each customer contains three types of fresh product to be delivered, which includes normal temperature, refrigerated and frozen products. Each customer has a soft time window and it can only be served by a multi-compartment vehicle. The vehicle must arrive within the customer's specified time window, otherwise the penalty cost will be generated. At any time, the total demand of all the fresh products on a multi-compartment vehicle shall not exceed the capacity of the compartment in which they are located. Each vehicle starts from the center, and then returns to the distribution center after serving the customer points. The optimization objective is to minimize the total distribution cost.

B. Cost Analysis

In order to establish the mathematical model of the problem, the first step is to define the types of parameters and decision variables that are used to build the model. They are defined in Table I.

This paper considers several costs, such as vehicle cost, distribution cost, vehicle refrigeration cost, fresh product freshness loss cost, and time penalty cost. We decide the linear sum of these five costs as the objective function of the addressed problem. Every cost of the problem are described in the following.

(1) vehicle cost

It is the total fixed cost of vehicles starting from the distribution center, which is defined in equation (1).

$$W_1 = \sum_{k=1}^{K} \sum_{j=1}^{N} F_k x_{0jk} \tag{1}$$

(2) distribution cost

It is composed of the total distances of all the vehicles and the cost used per unit distance. The distance cost d_{ij} between any two points is calculated by formula (2), and the delivery cost is defined in equation (3).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2}$$

$$W_2 = c \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{K} d_{ij} x_{ijk} \tag{3}$$

(3) refrigeration cost

For refrigeration and frozen products, there will be a certain refrigeration cost due to the use of refrigeration equipment in vehicles. For three types of fresh products, only the refrigerated goods and frozen goods need to be considered this cost. The refrigerated goods are transported in foam freezers, resulting lower refrigeration cost. While for frozen products, it need have a freezer, which accounts for the main refrigeration cost in the process of distribution. Thus, the definition of refrigeration cost is shown in equation (4).

$$W_3 = \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{p=1}^{P} f_p q_{ip} y_{ik} \tag{4}$$

(4) freshness damage cost

Fresh products will cause damage in the process of distribution, there is a certain economic loss, so we consider the cargo loss cost of fresh products. First of all, fresh products preservation input coefficient ε_p is introduced. Meanwhile, considering the influence of different refrigeration methods on the freshness in the distribution process, the fresh product preservation effort coefficient $f(\varepsilon_p)$ is also introduced and defined in equation (5), where ε_p and $f(\varepsilon_p)$ is positively correlated. When the fresh food material type p reaches customer i, its freshness function per kilogram is shown in equation (6). So, the final damage cost of fresh products is defined as equation (7).

$$f(\varepsilon_p) = 1 - e^{-\varepsilon_p} \tag{5}$$

$$\theta_{ip} = \theta_0{}^{(\sigma_p - f(\varepsilon_p))} \tag{6}$$

$$W_4 = \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{p=1}^{P} y_{ik} G_p q_{ip} (\theta_0 - \theta_{ip}) \tag{7}$$

(5) time window penalty cost

The time window penalty cost is the penalty that needs to be accepted, when the vehicle exceeds the customer's time window. It are used to reflect customer satisfaction, because early and late arrival will affect customer satisfaction. If the vehicle arrives at the customer earlier than the earliest service

TABLE I
DEFINITION OF RELEVANT PARAMETERS

parameter	description
V	set of depot and customers, $V = \{0, 1, 2, 3, ..n\}$, where 0 is the depot
N	set of customers, $N = \{1, 2, 3, ..n\}$
K	set of vehicles, $K = \{1, 2, ..k\}$
P	fresh types set, $P = \{1, 2, ..p\}$
F_k	fixed cost of vehicle k
c	distance cost per kilometer
v	average speed of vehicle per kilometer
d_{ij}	the distance from customer i to customer j
t_{ij}	travel time between two customers i and j, t_{ij}=d_{ij}/v
ET	the earliest service time of the depot
LT	the latest service time of the depot
a_i	the earliest service time of customer i
b_i	the latest service time of customer i
w_i	the service time of customer i
α	penalty cost for early arrival of vehicle per minute
β	penalty cost for late arrival of vehicle per minute
q_{ip}	customer i demand for good p
s_{ik}	the arrival time of vehicle k at customer i
θ_0	initial freshness value
σ_p	product p freshness decay coefficient
θ_{ip}	the freshness function when product p arrives at customer i
G_p	the value coefficient of product p
f_p	refrigeration cost coefficient of product p
ε_p	fresh-keeping investment of fresh product p
y_{ik}	if customer i is served by vehicle k, then y_{ik}=1; otherwise y_{ik}=0.
x_{ijk}	if vehicle k travels directly from customer i to customer j , then x_{ijk} =1; otherwise x_{ijk}=0.

time, the waiting cost will be $a_i - s_{ik}$. If the vehicle arrives at the customer later than the latest service time, the late cost will be $s_{ik} - b_i$. Otherwise, the penalty cost will be 0, as shown in equation (8). Thus, the total penalty cost is defined in equation (9).

$$F(s_{ik}) = \begin{cases} \alpha(a_i - s_{ik}), s_{ik} < a_i \\ 0, a_i \le s_{ik} \le b_i \\ \beta(s_{ik} - b_i), s_{ik} > b_i \end{cases} \tag{8}$$

$$W_5 = \sum_{\mathrm{i}=1}^{N} \sum_{k=1}^{K} F(s_{ik}) \tag{9}$$

Based on the above analysis, the objective function of the problem is defined as shown in equation (10), which consists of vehicle cost, distribution cost, refrigeration cost, freshness damage cost and time window penalty cost.

$$F = W_1 + W_2 + W_3 + W_4 + W_5 \tag{10}$$

C. Mathematical Model

This section, we build the mathematical model of the multi-compartment vehicle with soft time windows problem.

Minimize F

Subject to:

$$\sum_{k=1}^{K} y_{ik} = 1, \forall i \in N \tag{11}$$

$$\sum_{i=1}^{n} x_{ijk} = y_{jk}, \forall j \in N, k \in K \tag{12}$$

$$\sum_{\mathrm{i}=1}^{N} x_{0jk} = \sum_{i=1}^{N} x_{i0k}, \forall k \in K \tag{13}$$

$$\sum_{p=1}^{P} \sum_{i=1}^{N} q_{ip} y_{ik} \le Q_{\mathrm{k}}, \forall \mathrm{k} \in K \tag{14}$$

$$ET \le s_{ik} \le LT \tag{15}$$

$$\mathrm{s_{ik}}+\mathrm{w}_i + t_{ij} - s_{jk} \le (1-\mathrm{x_{ijk}})M \tag{16}$$

$$\sum_{i \in S} \sum_{\mathrm{j} \in S} x_{kij} \le |S| - 1, \forall S \subseteq N (S \ne \emptyset), i \ne j, \forall k \in K \tag{17}$$

$$x_{ijk} \in \{0, 1\}, \forall i \in N, \forall j \in N, \forall k \in K \tag{18}$$

$$y_{ik} \in \{0, 1\}, \forall i \in N, \forall k \in K \tag{19}$$

The objective function is defined in equation (10), that is, the optimization objective is the minimum value of the sum of each cost. Equation (11) indicates that each customer can only be served by one vehicle. Equations (12) and (13) indicate that only one vehicle is allowed to arrive and leave at each customer point. Equation (14) represents the capacity constraint of the vehicle. It ensures that the sum of all cabin demands of the customers delivered by each vehicle cannot exceed the maximum capacity of the vehicle. Equation (15) means that the service time of the vehicle is in the time window of the distribution center. If it is not in the corresponding time window, it will be punished accordingly. Equation (16) represents the continuity of vehicle delivery time, where M is a very large positive integer. Equation (17) eliminates sub-tour constraints. Equations (18) and (19) are decision variables.

III. Algorithm Design

A. Basic Description of PSO

Inspired by the group behavior of birds, Kennedy and Eberhart [15] proposed the famous particle swarm optimization (PSO) algorithm in 1995. PSO is a random search algorithm based on group cooperation developed by simulating the predation behavior of birds. Additionally, PSO algorithm has the characteristics of easy implementation, high precision, and fast convergence.

The fundamental core of PSO is to use the information sharing of individuals in the population to make the movement of the whole population in the problem space to obtain the optimal solution. Each Particle in the search space has a certain speed, and its direction and speed are adjusted in the iteration according to the individual extreme value P_{best} and group extreme value G_{best}, which forms a positive feedback mechanism. In the optimization process of PSO, the position of particle represents the solution of the solving problem, and each particle's performance depends on the number of optimization adaptation values of the evaluation function. Each particle uses a velocity vector to determine its travel direction based on its size and speed. According to the current optimal particle position and the memory of meeting the optimal solution position, the particle velocity and position are constantly changed to complete the search in the solution space. In each iteration, the historical optimal position of the population and the individual historical optimal position are recorded and updated by the equations (20) and (21).

$$v_{id} = \omega v_{id} + c_1 r_1 (P_{best} - x_{id}) + c_2 r_2 (G_{best} - x_{id}) \quad (20)$$

$$x_{id} = x_{id} + v_{id} \quad (21)$$

where x_{id} represents the position of particle i in d dimension; v_{id} denotes the velocity value corresponding to particle i; ω is the inertia weight factor; c_1 and c_2 are individual learning factor and group learning factor, respectively; r_1 and r_2 are two random numbers in the range of [0,1].

B. Overall Flow of the Proposed Algorithm

This paper proposed a particle swarm optimization algorithm with sequential crossover operator (called SCOPSO), which aims to overcome the shortcoming of falling into local optimum easily for the traditional PSO algorithm. SCOPSO introduces the sequential crossover operator from the genetic algorithm into PSO to jump out of the local optimum and expand the solution space of particles. The basic flow of the algorithm is in the following.

(1) Set the values of the parameters of the proposed algorithm, which includes population size N, the maximum number of iterations T_{max}, inertia factor ω, cognitive factor c_1, and social factor c_2. In additional, let the iteration variable t be 0.

(2) The initial particles S are generated by the nearest neighbor algorithm in section 3.3. The population of particles is denoted as $S = \{S_i | i=1, 2, ..., N\}$. The optimal individual is recorded as S_{best}.

(3) The optimal local solution P_{best} and optimal global solution G_{best} are initialized, P_{best} = G_{best} = S_{best}.

(4) For each particle in the particle population, a new particle population R is obtained by using the order crossover operator. In the execution process of this step, the inverse particle S_v of the current particle S_i, is first obtained. Then, the set of particles $plist = \{S_v, P_{best}, G_{best}\}$ is constructed and the selection probability is assigned to each particle in $plist$. Finally, a particle S_a is selected by the roulette method from $plist$. The S_a and the current particle S_i are applied the sequential crossover operator to generate a new particle S_n. For each particle in S, do this step until a new population of particles R is generated.

(5) Update the local and global optimal solutions in the new particle population R. It supposes that the best particle in the particle population R is R_{best}. If R_{best} is better than P_{best}, then P_{best} = R_{best}. If the updated local optimal solution P_{best} outperforms the global optimal solution G_{best}, then G_{best} = P_{best}.

(6) If $t > T_{max}$, then jump to (7); Otherwise, execute t++ and jump to (4).

(7) Return G_{best}.

C. Generating Initial Population

For each particle of the initial population, this paper adopted the nearest neighbor method to construct a whole route from the depot, including all stations and then back to the depot. The route is a traveling salesman problem (TSP) route including all stations. The specific generation process of the TSP route is descried as follows. First, a customer was randomly selected from a particle as the current point, and then added to the TSP route. Then, a customer was inserted into the route by selecting the nearest customer based on the current node. This step will be repeatedly executed until all the customers were inserted into the route. Finally, a particle, which is the whole TSP route excluding the two depots, was generated. When all the particles were generated, the initial solution set of particle populations was obtained.

D. Particle Decoding

For a particle, it was split into the set of routes by the thought form [16]. In other words, the sequence of customers in a particle was decoded to several routes by the capacity constraints and time window constraints. First, we construct a route starting from the depot and ends to the depot, and get the customers list of the particle by their location in sequence. Then, the customers in the customers list of the particle are traversed one by one. If the current customer can be inserted into the route without violating the constraints of capacity and time windows, it will be added to the route. If the current customer cannot be inserted into the route, a new route only including depots is created and the customer is inserted into this new route. When a customer is inserted into a route, the customer will be removed from the customers list of the particle. This process is did until the customers list is null.

For example, a particle including 10 customers is denoted as {1, 2, 7, 3, 10, 4, 8, 5, 6, 9}. We split the particle by vehicle capacity constraints and time window constraints. Finally, three routes are obtained, which are {0-1-2-7-0}, {0-3-10-4-8-0} and {0-5-6-9-0}.

E. Sequential Crossover Operator

In genetic algorithm (GA), the crossover operation combines the genetic information of the parents to produce the offspring of the parents. The crossover operators include single-point crossover, uniform crossover, and sequential crossover. We employed the sequential crossover operator in our proposed algorithm. Sequential crossover operator retains the relative order of the genes of the parents as much as possible, which helps to inherit the chromosome fragments of the excellent individuals to the offspring. At the same time, the sequential crossover operator plays a global search role and increases the search solution space.

In the proposed algorithm, we used the sequential crossover operator to generate the new particle. First, a particle R1 was chosen randomly as one parent, and a new particle was obtained by the R1 in reverse order. Then, the particles list was created by the new particle, the local optimal solution P_{best} and the global optimal solution G_{best}. The selection probability of the particles list was set to $\omega/(\omega + c_1 + c_2)$, $c_1/(\omega + c_1 + c_2)$, and $c_2/(\omega + c_1 + c_2)$, respectively. Another parent R2 was selected from the particles list by the roulette method. Finally, the sequential crossover operation between R1 and R2 are performed to get the new particle.

Fig.1 shows an example of operation process of the sequential crossover operator. The particles R1 and R2 are denoted as {1, 5, 3, 4, 2, 6, 7} and {2, 5, 3, 6, 7, 4, 1} respectively. First of all, three customers 5, 3, 4 at positions from 2 to 4 are randomly selected in the parent particle R1 for crossover. In the offspring particle, customers 5, 3, and 4 are kept in the same position and place them at positions from 2 to 4. Then, the customers in R2 are inserted into the offspring in sequence and skipped if an existing customer is encountered. First, customer 2 in R2 is inserted into position 1 of the offspring. Then customers 5 and 3 are no longer inserted. Next, customer 6 is inserted behind customer 4, and then customer 7 is inserted. For the customer 4, it does not be inserted because it has existed in the offspring. Finally, the last customer 1 is inserted into the position 7 of the offspring, and the final offspring partial is {2, 5, 3, 4, 6, 7, 1}.

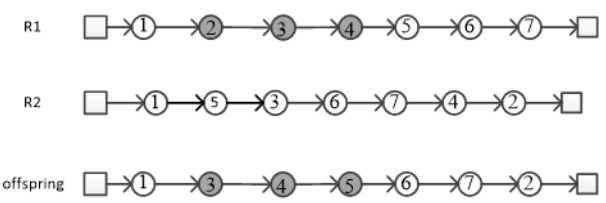


Fig. 1. an example of sequential crossover operator crossover

IV. Experimental Results and Analysis

A. Experimental Environment and Parameter Setting

The proposed algorithm was implemented in PyCharm by python 3.7. All the experiments were run on a personal computer, which has a processor with Intel(R).core(TM).i5-11400H and 16.0GB RAM memory. The parameters of the SCOPSO are set as follows. The iteration number and population size are 1000 and 50. The inertia factor ω, self-cognition factor c_1, and social cognition factor c_2 were set to 0.4, 0.5, and 0.5 respectively. Each instance was executed 10 times.

TABLE II
THE VALUES OF PARAMETERS USED IN OBJECTIVE FUNCTION

parameter	value
F_k	20
v	1
c	1
α	2
β	3
θ_0	0.98
G_p	normal temperature:1.5; cold storage:2.5; frozen:5.0
f_p	normal temperature:0.1; cold storage:0.3; frozen:0.5
σ_p	normal temperature2.0; cold storage:2.2; frozen:2.5
ε_p	normal temperature:0.2; cold storage:1.8; frozen:5.0

B. Test Instances

In order to verify the effectiveness of the proposed algorithm for solving the MCVRPSTW problem, we used it to solve some benchmark instances. The standard instances for VRPTW proposed by Solomon [17] are adopted to test the algorithm. These instances are divided into six sets, which are C1, C2, R1, R2, RC1 and RC2. The problem scale of the instances are 25, 50, and 100 customers, and the total number of instances is 56.

According to the construction method for MCVRPTW proposed by [3], we selected 24 instances to test the performance of the SCOPSO algorithm. The construction method for the instances used in this paper are described as follows. We divide the single product demand of customer into three types of product demand including normal temperature, cold storage and frozen products by a certain proportion according to the coordinate position of customer. When the customers coordinate X and Y meet the conditions, which are $X_{min} \leq X < X_{min} + (X_{max} - X_{min})/2$ and $Y_{min} \leq Y < Y_{min} + (Y_{max} - Y_{min})/2$, its demand is split in a ratio of 2:1, that is, the ratio of refrigeration and freezing is 2:1. The values of X_{min} and Y_{min} are both set to 0. X_{max} and Y_{max} represent the maximum values of all the customers coordinate X and coordinate Y respectively. When the customers coordinate locates in other area, the demand of the demand is split by 3:1, which means that the ratio of room temperature and frozen is 3:1. Table II shows the values of parameters used in the objective function.

C. Performance of the Improvement Strategy of the SCOPSO algorithm

In order to evaluate the advantages of the improved PSO, we compare the SCOPSO algorithm with the PSO algorithm without a sequential crossover operator to solve the instances with 50 customers respectively. Table III shows the results obtained by the two algorithms. Columns $Best$, Avg and Std represent the best solution,average solution and standard deviation value obtained by the algorithm respectively. Columns g_1 and g_2 indicates the percentage improvement of SCOPSO on the best solution and average solution compared with the PSO algorithm.

As reported in Table III, SCOPSO improved the best solution and average solution increased by 12.18% and 15.03% on average respectively, when compared with PSO. For instances C201,C202,C204, and RC203, the cost of the best solution is decreased by 26.11%, 26.16%, 28.23%,and 22.50% respectively, all of which are more than 20%. The

TABLE III
RESULTS OF THE TWO ALGORITHMS ON INSTANCES WITH 50 CUSTOMERS

Instance	PSO			SCOPSO			Gap(%)	
	Best	Avg	Std	Best	Avg	Std	g_1	g_2
C101	935.93	960.49	6.04	888.74	919.35	6.92	5.04	4.28
C102	972.76	1007.45	8.57	843.18	927.09	17.26	13.32	7.98
C103	983.72	1034.16	23.11	871.50	918.65	7.71	11.41	11.17
C104	911.67	997.42	9.79	864.19	905.01	7.98	5.21	9.26
C201	1063.05	1493.37	95.17	785.49	869.25	14.11	26.11	41.79
C202	1068.00	1345.12	58.18	788.63	894.52	20.47	26.16	33.50
C203	1001.29	1194.46	44.12	816.27	849.15	7.39	18.48	28.91
C204	1154.44	1402.10	58.00	828.57	877.98	7.97	28.23	37.38
R101	1229.24	1309.63	12.52	1129.64	1175.34	10.11	8.10	10.25
R102	1186.28	1231.28	11.23	1050.19	1103.83	14.37	11.47	10.35
R103	1143.59	1199.74	12.40	1064.85	1112.42	10.03	6.89	7.28
R104	1039.08	1093.19	6.81	945.96	1027.92	9.93	8.96	5.97
R201	1011.81	1326.93	38.16	934.75	1016.85	16.48	7.62	23.37
R202	1043.83	1156.62	25.52	888.40	1018.42	24.36	14.89	11.95
R203	1044.40	1135.53	33.12	902.29	962.76	13.20	13.61	15.21
R204	994.66	1176.23	72.77	898.67	955.83	11.62	9.65	18.74
RC101	1298.84	1400.85	13.52	1229.07	1261.18	6.93	5.37	9.97
RC102	1065.39	1170.47	20.57	1074.29	1123.97	7.38	-0.84	3.97
RC103	1149.12	1251.66	13.73	1071.33	1127.42	14.70	6.77	9.93
RC104	1142.32	1159.89	2.72	1087.42	1118.95	6.27	4.81	3.53
RC201	970.05	1087.09	18.68	884.08	947.43	13.10	8.86	12.85
RC202	1044.25	1125.36	11.05	908.10	979.62	13.90	13.04	12.95
RC203	1176.16	1232.26	12.17	911.54	1009.21	23.32	22.50	18.10
RC204	994.70	1006.58	1.48	829.06	884.69	10.52	16.65	12.11
Average	1067.69	1187.41	25.39	937.34	999.45	12.33	12.18	15.03

instance C204 has the maximum improvement percentages, which is 28.23%. While for the average solutions of the instances, the maximum improvement of the SCOPSO algorithm is 41.79%, and the minimum improvement is 3.53%. The results reveal that the SCOPSO algorithm outperforms the PSO algorithm. The reason can be explained that the sequential crossover operator can preserve the relative order of genes of both parents as much as possible, which is helpful to inherit the excellent individual fragments to the offspring. Moreover, it also plays the role of enhancing the global search and expanding the search solution space. Moreover, the SCOPSO algorithm has a smaller average standard deviation value than PSO, which demonstrates the stability of the SCOPSO algorithm.

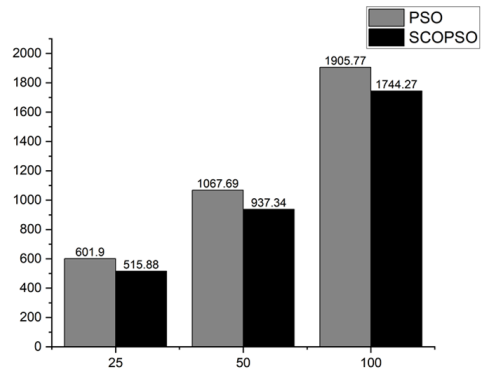


Fig. 2. comparison of average values of best solutions

Furthermore, the PSO algorithm and SCOPSO algorithm were used to solve other instances with 25 customers and 100 customers. The average value of the best solutions of the two algorithms on three types of instances is shown in Fig.2. Seen from Fig.2, the best solutions of SCOPSO was improved by 13.79%, 12.01%, and 8.15% on average respectively, compared with the PSO algorithm. The experimental results prove that the proposed algorithm has better performance than the PSO algorithm.

D. Algorithm Comparison

To further evaluate the performance of the proposed algorithm, we compared the SCOPSO algorithm with two state-of-the-art metaheuristics, which are simulated annealing algorithm (SA) and genetic algorithm (GA) for the MCVRPSTW problem. To have fair comparison, these three algorithms were executed on the same machine. They also used the same nearest neighbor algorithm to construct the initial solution, and the sequential crossover operator. For SA, the initial temperature and termination temperature are 100 and 0.1, and the cooling rate is 0.99. For the GA algorithm, the number of iterations of GA is 500, and the crossover probability and the mutation probability are set to 0.92 and 0.1. The population size of GA is set to 50.

Table IV, Table V and Table VI show the results found by the three algorithms on different instances with 25, 50, and 100 customers. In these tables, $Best$, Avg, and Std represent the best solution, average solution and standard variance obtained by the algorithm, respectively. The execution times of three algorithms are no longer compared, because the computation time of them are not much different.

As shown in these three tables, it can be seen that the SCOPSO algorithm is superior to the SA and GA algorithms

TABLE IV
RESULTS OF 25 CUSTOMERS OBTAINED BY THREE ALGORITHMS

Instance	SA			GA			SCOPSO		
	Best	Avg	Std	Best	Avg	Std	Best	Avg	Std
C101	440.61	494.91	13.05	444.61	505.32	16.01	423.87	493.15	10.57
C102	437.64	462.84	7.95	435.69	461.54	6.41	416.39	440.23	6.41
C103	441.06	493.85	10.39	432.25	463.83	10.19	420.88	456.51	7.90
C104	490.16	507.53	4.77	472.29	495.90	6.21	434.62	471.69	6.07
C201	456.37	541.02	11.14	449.74	523.48	18.80	472.11	507.37	8.68
C202	448.27	514.03	12.89	443.05	485.31	6.72	448.02	479.49	6.11
C203	503.41	544.38	7.49	465.22	504.48	9.26	393.40	469.62	11.19
C204	465.38	510.26	9.96	427.28	492.68	14.51	450.37	488.02	5.93
R101	619.02	662.96	10.21	652.28	690.24	6.57	605.92	643.23	4.90
R102	640.65	657.80	5.56	628.31	659.43	5.22	593.15	637.94	8.64
R103	624.43	649.58	5.25	606.02	641.32	7.07	562.14	613.20	8.12
R104	565.13	604.34	5.37	550.83	602.48	8.06	533.32	562.03	5.51
R201	571.48	640.71	13.21	530.59	615.87	22.94	519.31	579.18	9.94
R202	567.46	626.80	16.68	550.27	609.72	17.53	475.43	553.32	12.94
R203	600.22	644.24	13.64	509.96	599.23	15.03	541.11	568.66	5.90
R204	623.54	692.24	14.41	624.83	665.97	9.48	608.42	631.75	5.99
RC101	657.89	680.93	4.29	633.99	685.76	6.47	635.69	654.56	3.59
RC102	656.41	685.50	5.36	635.62	668.26	5.09	628.27	640.78	4.58
RC103	657.18	669.54	2.37	637.22	663.07	4.74	623.79	641.02	4.33
RC104	632.07	645.96	2.56	616.98	653.92	7.41	605.09	631.64	7.37
RC201	544.75	604.62	10.26	526.48	620.06	20.62	520.67	564.81	8.04
RC202	491.67	617.80	15.17	505.09	635.63	21.99	473.37	549.68	17.96
RC203	467.41	596.48	18.18	480.87	591.15	14.64	475.43	547.56	10.06
RC204	577.37	601.72	6.54	501.10	555.30	9.00	520.36	552.64	5.13
Average	549.15	597.92	9.45	531.69	587.08	11.25	515.88	557.42	7.74

in solving instances with different scale problems.

For small-scale instances, compared with SA and GA, the best solution of the SCOPSO algorithm was reduced by 6.68% and 2.86% on average, respectively. At the same time, the SCOPSO algorithm improved the average solution by 6.76% and 4.91%, on average respectively. In addition, the SCOPSO algorithm found 17 best solutions, GA found 6 best solutions, SA found 1 best solutions. The SCOPSO algorithm found 24 the best average solutions. The experimental results show that the SCOPSP algorithm has outstanding performance on small-scale case instances.

The SCOPSO algorithm still shows good performance in a medium-scale case with 50 customers. Compared with SA and GA, the best solution of the SCOPSO algorithm was improved on average by 6.81% and 3.15%. Among the 24 problems, the SCOPSO algorithm found 17 best solutions, while the SA and the GA algorithms found 1 and 6 best solutions respectively. In terms of average solution, the SCOPSO algorithm found 23 better solutions, and it was improved by 7.32% and 3.40% on average when compared with SA and GA respectively. Despite doubling the size of the problem, the SCOPSO algorithm still achieves quite good results.

From the results in Table VI, the SCOPSO algorithm has an average improvement of 5.89% and 2.62% in the best solution compared with SA and GA. In addition, the SCOPSO algorithm, the SA algorithm and the GA algorithm found 22, 1, and 1 best solutions, respectively. The ability of SCOPSO to find the best solution is still better than SA and GA in large-scale cases. Moreover, the SCOPSO algorithm has an average improvement of 4.84% and 2.17% compared with SA and GA in terms of average solutions.

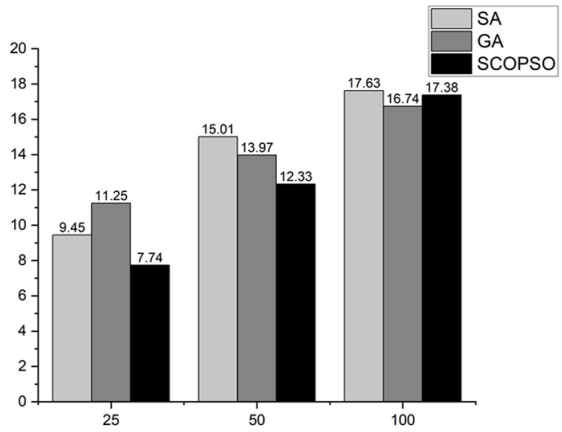


Fig. 3. average standard deviation of three algorithms

In summary, the SCOPSO algorithm is more competitive than the SA and GA algorithms in solving the quality of small, medium, and large scale cases. The experimental results also prove that the SCOPSO algorithm has better optimization performance and stability.

E. Stability Analysis of the SCOPSO algorithm

This section, we analyse the stability of the SCOPSO algorithm. We calculated the average standard deviation of three algorithms on three instances with different customers, and the results are shown in Fig.3. Seen from Fig.3, the SCOPSO algorithm has the smallest average standard deviation values in three types of problems on the whole.

TABLE V
RESULTS OBTAINED BY THREE ALGORITHMS ON INSTANCES WITH 50 CUSTOMERS

Instance	SA			GA			SCOPSO		
	Best	Avg	Std	Best	Avg	Std	Best	Avg	Std
C101	946.63	977.34	3.28	911.57	946.04	4.69	888.74	919.35	6.92
C102	916.30	969.09	11.75	918.57	936.46	4.65	843.18	927.09	17.26
C103	924.34	958.40	7.46	891.84	928.23	5.01	871.50	918.65	7.71
C104	903.92	965.98	10.64	869.01	913.32	8.46	864.19	905.01	7.98
C201	844.19	971.65	32.15	866.67	963.54	25.7	785.49	869.25	14.11
C202	864.26	961.99	11.45	870.61	955.78	20.28	788.63	894.52	20.47
C203	868.20	948.22	19.45	837.50	925.62	14.33	816.27	849.15	7.39
C204	913.85	1064.75	26.45	848.96	940.27	24.61	828.57	877.98	7.97
R101	1199.40	1266.48	11.87	1154.64	1216.90	14.32	1129.64	1175.34	10.11
R102	1145.66	1208.60	12.82	1092.60	1161.41	15.08	1050.19	1103.83	14.37
R103	1125.28	1186.56	9.37	1085.65	1128.28	10.94	1064.85	1112.42	10.03
R104	1030.21	1088.91	7.43	1021.89	1057.78	7.49	945.96	1027.92	9.93
R201	1011.87	1105.08	27.54	914.88	1030.00	20.25	934.75	1016.85	16.48
R202	982.59	1080.75	24.30	976.52	1074.77	24.77	888.40	1018.42	24.36
R203	921.63	1050.99	20.72	891.87	998.41	24.93	902.29	962.76	13.20
R204	991.44	1017.23	6.97	895.35	955.51	13.88	898.67	955.83	11.62
RC101	1299.48	1358.53	10.45	1210.78	1291.28	14.25	1229.07	1261.18	6.93
RC102	1112.64	1180.42	15.16	1123.11	1150.87	5.48	1074.29	1123.97	7.38
RC103	1136.97	1180.75	11.10	1126.97	1180.44	10.97	1071.33	1127.42	14.7
RC104	1124.68	1154.99	4.59	1126.64	1152.52	3.41	1087.42	1118.95	6.27
RC201	961.68	1038.18	12.61	916.52	969.37	12.21	884.08	947.43	13.10
RC202	981.67	1055.21	16.49	904.11	1001.56	12.82	908.10	979.62	13.90
RC203	909.76	1102.85	32.39	951.11	1049.49	21.44	911.54	1009.21	23.32
RC204	889.24	978.54	13.70	821.98	899.60	15.31	829.06	884.69	10.52
Average	1000.25	1077.98	15.01	967.89	1034.48	13.97	937.34	999.45	12.33

TABLE VI
RESULTS FOUND BY THREE ALGORITHMS ON INSTANCES WITH 100 CUSTOMERS

Instance	SA			GA			SCOPSO		
	Best	Avg	Std	Best	Avg	Std	Best	Avg	Std
C101	2195.02	2292.20	14.24	2085.90	2213.26	25.23	2039.35	2168.15	16.74
C102	2072.20	2176.23	24.90	2038.11	2171.37	26.84	2004.51	2112.76	20.77
C103	1979.66	2044.80	10.22	1935.31	1995.83	11.71	1846.50	1972.43	16.59
C104	1946.04	1995.65	5.62	1899.91	1944.49	9.27	1878.00	1956.61	12.16
C201	1853.02	2075.27	30.65	1886.99	1988.46	14.07	1709.32	1814.40	21.26
C202	1662.23	1767.42	29.95	1652.78	1706.60	25.37	1582.79	1634.50	15.32
C203	1604.44	1716.56	28.54	1609.73	1685.13	25.10	1559.05	1631.93	15.47
C204	1679.50	1732.96	24.45	1582.50	1666.89	14.43	1567.42	1622.69	12.43
R101	2088.85	2153.33	7.73	1979.42	2078.95	15.99	1943.99	2012.85	15.05
R102	2040.86	2088.74	9.79	1969.98	2032.65	10.81	1864.95	1983.78	16.25
R103	1910.77	2009.51	12.82	1896.32	1943.22	8.74	1874.70	1923.88	11.67
R104	1890.20	1899.32	4.65	1787.87	1848.26	9.06	1774.74	1817.37	11.28
R201	1646.64	1713.51	24.50	1579.19	1660.57	14.24	1505.87	1626.75	22.00
R202	1548.79	1660.56	17.21	1487.88	1617.33	24.47	1447.90	1583.29	25.80
R203	1622.57	1739.84	23.63	1585.28	1700.11	18.69	1494.65	1610.94	27.10
R204	1370.10	1481.65	21.17	1349.54	1420.95	14.24	1441.41	1496.15	11.91
RC101	2235.84	2310.28	13.92	2130.56	2236.83	16.00	2127.92	2213.96	16.76
RC102	2133.58	2209.95	15.47	2116.01	2175.09	10.36	2085.66	2146.09	10.67
RC103	2028.53	2047.82	6.27	2016.26	2061.11	10.68	1972.02	2027.23	12.15
RC104	2001.40	2090.37	15.96	2034.88	2093.17	9.69	1994.22	2049.12	13.38
RC201	1752.31	1891.03	32.81	1658.40	1806.64	30.60	1577.56	1712.74	21.11
RC202	1705.12	1833.77	20.78	1638.68	1772.72	24.48	1570.74	1747.85	31.17
RC203	1719.65	1777.68	12.00	1586.26	1697.80	14.87	1533.75	1667.43	24.42
RC204	1590.91	1651.41	15.73	1497.39	1598.30	16.73	1465.52	1579.36	15.59
Average	1844.93	1931.66	17.63	1791.88	1879.84	16.74	1744.27	1838.01	17.38

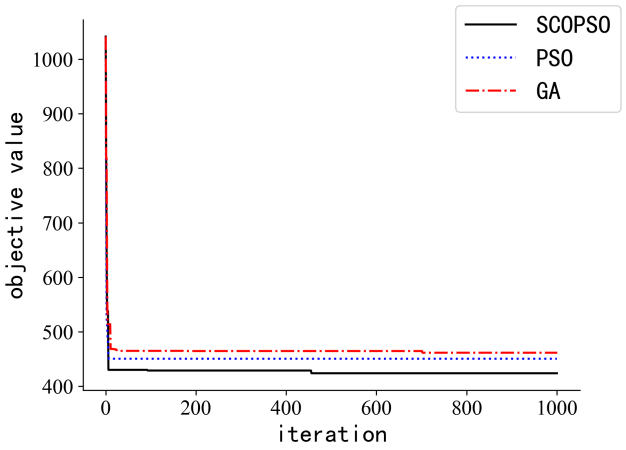
For small-scale cases, the average standard deviation of the SCOPSO algorithm is 7.74, which is 1.71 and 3.51 lower than SA and GA algorithms, respectively. Compared with SA and GA, the SCOPSO algorithm decreases the average standard variance by 2.68 and 1.64 respectively at the case of medium scale. For large-scale cases, the average standard variance of the SCOPSO algorithm is reduced by 0.25 and -0.64 respectively compared with SA and GA algorithms. Although the standard deviations of SA, GA and SCOPSO algorithms show an increasing trend with the increasing of the scale, the average deviations of SCOPSO algorithm were still obviously smaller than SA and GA algorithms. The results demonstrate that the SCOPSO algorithm has better stability than the SA and GA algorithms.

F. Convergence Analysis of the SCOPSO algorithm

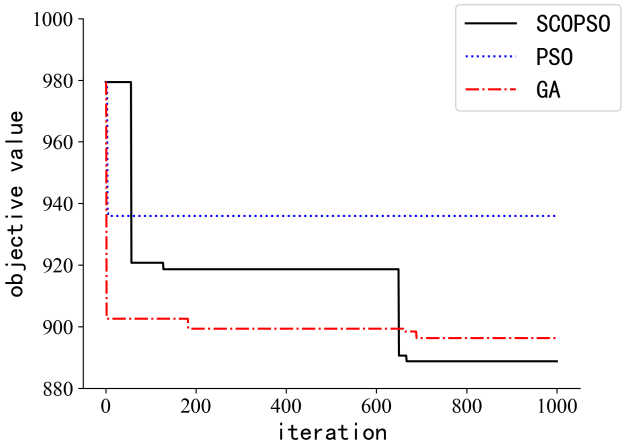
In this section, we take the instance C101 as an example to analyze the convergence of the proposed algorithm. For GA, PSO and SCOPSO three algorithms, we draw the change trends of its objective function values with the increasing of the iteration numbers respectively, when the instance C101 has 25, 50 and 100 customers. Fig.4 gives the convergence results of different algorithms.

As shown in Fig.4, SCOPSO has good convergence and the ability to find the best solution. Among three algorithms, PSO has the fastest convergence, but its performance is very poor. GA and SCOPSO both have a certain convergence rate, but the SCOPSO algorithm does not converge prematurely like GA. For the small-scale instance, as shown in Fig.4(a), all the algorithms converge very fast in the early stage. GA and PSO tend to be stable soon after early convergence, while the SCOPSO algorithm tends to be stable about 450 iterations. When the number of customers is 50, PSO converges very quickly. GA and SCOPSO both have a fixed objective values about 200 iterations, but with the increase of the iterations, SCOPSO finds the smaller objective value about 650 iterations and then tends to be stable. When the number of customers reaches 100, GA and SCOPSO tend to be stable about 200 iterations. When the number of iterations is greater than about 760, SCOPSO can obtain the better solution.

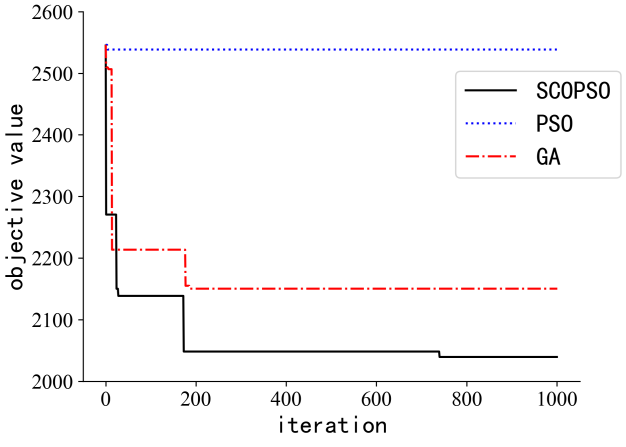
On the whole, the SCOPSO algorithm converges fast in the early stage, which is in line with the characteristics of the particle swarm optimization algorithm. With the increase of iterations, the SCOPSO algorithm can jump out of the local optima so as to find better solutions from a global perspective, because the sequential crossover operator is used to expand the search space of the particles.



(a) 25 customers



(b) 50 customers



(c) 100 customers

Fig. 4. Convergence curves of C101 obtained by different algorithms

V. Conclusion

The routes planning of fresh products distribution is very interesting and challenging. This paper deals with the multi-compartment vehicle routing problem with soft time windows for the distribution of fresh products. A mathematical model was first established considering the key factors, which occur in the distribution of the fresh products. Then, we proposed a hybrid particle swarm optimization algorithm to solve the problem. The algorithm adopted the sequential crossover operator in genetic algorithm to expand the solution space of the improved particle swarm, so that the enhanced algorithm can jump out of the local optimum. The performance of the algorithm was verified by a set of classic instances. The experimental results show that the sequential crossover operator has the advantage of enhancing the global search and expanding the solution space. Furthermore, the proposed algorithm was compared with the simulated annealing algorithm and genetic algorithm, which revealed the effectiveness of the proposed algorithm for the

multi-compartment vehicle routing problems with a soft time windows. Meanwhile, the algorithm has a smaller average standard deviation and good convergence.

In the future, there are some promising research topics about the new variants and solution methods of the MCVRP. The new problem attributes will be considered into the research of MCVRP, such as carbon emission, fuzzy time window, stochastic demand and flexible variable compartment and so on. Additionally, more efficient approaches need to be developed, due to the complexity of the multi-compartment vehicle routing problem.

REFERENCES

[1] Dantzig G B and Ramser J H, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80-91, 1959.

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

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

[4] Michael W L, "Linear Program for System Optimal Parking Reservation Assignment," *Journal of Transportation Engineering, Part A: Systems*, vol.145, no.12, 0000280, 2019.

[5] Ostermeier M and Huebner A, "Vehicle selection for a multi-compartment vehicle routing problem," *European Journal of Operational Research*, vol. 269, no. 2, pp. 682-694, 2018.

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

[7] Ostermeier M, Henke T, Hübner A and Wäscher G, "Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions," *European Journal of Operational Research*, vol. 292, no. 3, pp. 799-817, 2021.

[8] Chen J M, Nan Z and Wang Y, "Route Optimization of Multi-compartment Cold Chain Distribution Vehicle for Fresh Agricultural Products," *System Engineering*, vol. 36, no. 8, pp. 106-113, 2018.

[9] Emmanual D C and Monique G, "Scheduling Deliveries in Vehicles with Multiple Compartments," *Journal of Global Optimization*, vol. 26, no. 1, pp. 43-78, 2003.

[10] Brito J, Martinez F J, Moreno J A and Verdegay J L, "Fuzzy optimization for distribution of frozen food with imprecise times," *Fuzzy Optimization and Decision Making*, vol. 11, no. 3, pp. 337-349, 2012.

[11] Chen J M, Dan B and Shi J, "A variable neighborhood search approach for the multi-compartment vehicle routing problem with time windows considering carbon emission," *Journal of Cleaner Production*, vol. 277, 123932, 2020.

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

[13] Zhao F Q, Qin S and Yang G Q, Ma W M, Zhang C and Song H B, "A factorial based particle swarm optimization with a population adaptation mechanism for the no-wait flow shop scheduling problem with the makespan objective," *Expert Systems With Applications*, vol. 126, pp. 41-53, 2019.

[14] Liu R C, Li J X, Fan J, Mu C H and Jiao L H, "A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization," *European Journal of Operational Research*, vol. 261, no. 3, pp. 1028-1051, 2017.

[15] Kennedy J and Eberhart R, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[16] Prins C, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Computers and Operations Research*, vol. 31, no. 12, pp. 1985-2002, 2004.

[17] Solomon M, "Algorithms for the vehicle routing and scheduling problems with time window and constraints," *Operations Research*, vol. 35, no. 2, pp. 166-324, 1987.