

Real Environment-Aware Multisource Data-Associated Cold Chain Logistics Scheduling: A Multiple Population-Based Multiobjective Ant Colony System Approach

Li-Jiao Wu, *Student Member, IEEE*, Zong-Gan Chen[✉], *Member, IEEE*, Chun-Hua Chen, *Member, IEEE*, Yun Li[✉], *Fellow, IEEE*, Sang-Woon Jeon[✉], *Member, IEEE*, Jun Zhang[✉], *Fellow, IEEE*, and Zhi-Hui Zhan[✉], *Senior Member, IEEE*

Abstract—Cold chain logistics (CCL) scheduling is important for smart cities as it directly affects the service quality and operating profits of logistics companies. However, traditional CCL models seldom reflect the real transportation environment, making the solutions hardly applicable to the real CCL scenes. Hence, this paper attempts to establish a multisource data-associated CCL model oriented to the real transportation environment. This environment is considered by employing the real-captured driving duration and distance between any two places. Three scheduling objectives (namely, quality losses, personnel and vehicle costs, and transportation costs) are taken into account. To efficiently solve the proposed multisource data-associated multiobjective CCL model, a multiple population-based multiobjective ant colony system (MPMOACS) approach is proposed. Based on

the multiple populations for multiple objectives framework, the MPMOACS approach can optimize multiple objectives sufficiently, and thus obtain promising solutions distributed along the entire Pareto front. To further enhance the performance of the MPMOACS, a ranking-based local search strategy is also designed. Experiments are conducted on not only the existing benchmark instances but also a real environment-aware multi-source dataset that is built based on real-captured transportation data of Guangzhou and Shenzhen, China. Compared with six state-of-the-art and very recent well-performing multiobjective optimization approaches, the proposed MPMOACS approach exhibits the overall best performance.

Index Terms—Evolutionary computation, multisource data association, cold chain logistics scheduling, multiobjective optimization, ant colony system.

Manuscript received 27 July 2021; revised 21 December 2021, 8 March 2022, and 7 June 2022; accepted 4 August 2022. Date of publication 16 September 2022; date of current version 5 December 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102102; in part by the National Natural Science Foundations of China under Grant 62206100, Grant 62176094, and Grant 61873097; in part by the Guangdong Natural Science Foundation under Grant 2020B010166002, Grant 2018B030312003, and Grant 2021B1515120078; in part by the Ministry of Science and Technology of China under Grant 111001842920228002; in part by the Natural Science Foundation of Guangdong Province through ‘Key Project’ under Grant 2021B1515120078; in part by the Ministry of Science and Technology of China under Grant G2022032012L; in part by the National Research Foundation of Korea under Grant NRF-2021H1D3A2A01082705; and in part by the Korea Institute of Ocean Science and Technology under Grant PE99732. (*Corresponding authors: Zong-Gan Chen; Zhi-Hui Zhan.*)

Li-Jiao Wu and Zhi-Hui Zhan are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, also with the Pazhou Laboratory, Guangzhou 510330, China, and also with the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, Guangzhou 510006, China (e-mail: zhanapollo@163.com).

Zong-Gan Chen is with the School of Computer Science, South China Normal University, Guangzhou 510631, China (e-mail: charleszg@qq.com).

Chun-Hua Chen is with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China.

Yun Li is with the Industrial Artificial Intelligence Centre, Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518110, China, and also with i4AI Ltd., London WCIN3AX, U.K.

Sang-Woon Jeon is with the Department of Electronics and Communication Engineering, Hanyang University, Ansan 15588, South Korea.

Jun Zhang is with Zhejiang Normal University, Jinhua 321004, China, and also with Hanyang University, Ansan 15588, South Korea.

This article has supplementary downloadable material available at <https://doi.org/10.1109/TITS.2022.3203629>, provided by the authors.

Digital Object Identifier 10.1109/TITS.2022.3203629

I. INTRODUCTION

COLD chain logistics (CCL) is concerned with the transportation of perishable goods that require low-temperature storage [1], [2], [3]. It has attracted increasing research attention in recent years since smart logistics forms a fundamental part of smart cities [4], [5], [6]. Nevertheless, the quality of service in CCL is still limited by the quality losses of goods and expensive distribution costs. Specifically, the acute quality losses of goods that occur during the long distribution duration may lead to serious waste and even safety problems [7], [8]. Furthermore, the high distribution costs may weaken the competitiveness of the logistics companies. Hence, effective CCL scheduling should be conducted to optimize both quality losses and distribution costs.

Having developed from the vehicle routing problem (VRP), most of the existing CCL models derive driving duration by assuming a constant driving speed [9], [10], but the driving speed between two places is not a constant since it varies with time (i.e., the driving speed is time-dependent). For example, the driving speed in peak hours is much lower than that in other hours. Thus, the scheduling results of the existing CCL models are often inadequate to a real-world environment. To integrate the time-dependent speed into the VRP model, researchers have studied the time-dependent VRP (TDVRP) [11], [12], [13], [14], [15]. Nevertheless, only a simple and uniform step function is used to describe the time-dependent

driving speed in the TDVRP [16], [17], which is insufficiently accurate to reflect the real transportation environment. Besides, the TDVRP hypothesizes that there is a straight path between any two places and adopts the corresponding Euclidean distance to calculate the driving duration. However, such a hypothesis does not necessarily hold due to the complex urban layout. Therefore, the TDVRP still has limitations and is inefficient when the scheduling results are returned to the real CCL scenes. In other words, although the time-dependent speed is considered, the TDVRP model is not suitable for real-world CCL scheduling.

To address the problem that the solutions of the traditional CCL models may fail to work in real CCL scenes, a real environment-aware multisource data-associated CCL model is developed in this paper. Multisource data refer to the data obtained from multiple sources, consisting of order data, service data, and transportation data. The first two are gathered from customers and logistics companies, respectively, while the transportation data are captured through map API (e.g., the Baidu map API of China). In detail, the order data include customers information; the service data include information about personnel and vehicle resources; and transportation data include the driving distance, duration, and speed in real environments. By associating order data, service data, and transportation data in our model, the gap between the problem model and the real CCL scenes can be narrowed, and thus the feasibility of scheduling results can be improved.

To satisfy the requirements of different stakeholders, the proposed CCL model considers three objectives, resulting in the multisource data-associated multiobjective CCL model. The first objective is to minimize the quality losses of goods. The second objective is to minimize the personnel and vehicle costs. The third objective is to minimize the transportation costs measured by the total driving distance and driving duration.

Recently, many scheduling approaches for solving various multiobjective VRP models have been proposed [18], [19], [20]. For example, a coevolutionary constrained multiobjective optimization (CCMO) framework to solve a multiobjective VRP with time windows (VRPTW) model that aims to minimize the number of vehicles and the total traveled distance was proposed in [21]. A multi-depot VRPTW model with five objectives was solved by a two-stage multiobjective evolutionary algorithm (TS-MOEA) [22]. The first stage of the TS-MOEA finds extreme solutions using the extremized crowded nondominated sorting genetic algorithm II (EC-NSGA-II) [23], which is developed from the nondominated sorting genetic algorithm II (NSGA-II) [24]. The second stage of the TS-MOEA obtains more Pareto solutions based on extreme solutions through a multiobjective evolutionary algorithm based on decomposition (MOEA/D) [25]. However, the quality losses of goods are not considered in these multiobjective VRP models. Hence, the scheduling approaches proposed for these multiobjective VRP models may not obtain satisfying results when being used to solve the multisource data-associated multiobjective CCL model.

In addition, multiobjective CCL scheduling has seldom been studied in previous work, which indicates a lack of existing

multiobjective scheduling approaches to multiobjective CCL scheduling problems. Note that although multiple objectives are considered in the CCL model proposed in [10], their work combined multiple objectives into one objective and solved the model using a single-objective optimization approach. In such case, the multiple objectives may not be sufficiently optimized. Therefore, it is necessary to develop an effective multiobjective scheduling approach to efficiently optimize the multisource data-associated multiobjective CCL model.

Owing to the promising performance of ant colony optimization (ACO) in solving the VRP [26], [27], [28], [29] and other combinatorial optimization problems [30], [31], [32], [33], a widely used ACO variant, i.e., the ant colony system (ACS) [34], [35] is regarded as an appropriate approach for solving our model. However, the simultaneous optimization of multiple objectives is difficult due to the conflict among objectives, which challenges many existing multiobjective techniques [36], [37]. Thus, the famous and popular multiple populations for multiple objectives (MPMO) framework proposed by Zhan *et al.* [38] is employed and combined with the ACS, resulting in a multiple population-based multiobjective ACS (MPMOACS) approach. Furthermore, a ranking-based local search (RBLs) strategy is designed to further enhance the search efficiency of MPMOACS.

Including the problem modeling aspect and the algorithm designing aspect, the contributions of the work reported in this paper are as follows.

First, a multisource data-associated multiobjective CCL model oriented to the real transportation environment is established. By using real transportation data, our model can reflect the real-world situation more accurately, and thus improve the feasibility of scheduling results. Furthermore, three objectives (i.e., quality losses, personnel and vehicle costs, and transportation costs) are considered in the proposed model, which can satisfy various real-world demands.

Second, the new MPMOACS approach is proposed to solve the proposed model. According to the suggestion of the MPMO framework [39], [40], the MPMOACS employs three populations. In this way, all the three objectives can be sufficiently searched to obtain promising solutions distributed along the entire Pareto front.

Third, the RBLs strategy is designed to strengthen the performance of the MPMOACS. By combining the ranking of all objectives and an effective local search operator, the RBLs strategy can adjust the distribution sequence of customers reasonably and thus improve the solution quality.

The remainder of this paper is organized as follows. Section II describes and formulates the multisource data-associated multiobjective CCL model. Section III develops the MPMOACS approach. Section IV illustrates the experiments, with results discussed. Finally, Section V concludes the work.

II. MULTISOURCE DATA-ASSOCIATED MULTI-OBJECTIVE CCL MODEL

A. Model Description

1) *Model Information*: To begin with, all variables involved in this model are presented in Table I. Given one depot and n

TABLE I
 TERMINOLOGY NOTATION

A) Multisource Data	
a) Order data	
i, j	the indexes of the vertices
n	the number of customers
b_i	the earliest time for i to accept service, i.e., the lower bound of time window
e_i	the latest time for i to accept service, i.e., the upper bound of time window
q_i	the demand of i
st_i	the service duration of i , $st_0=st_{n+1}=0$
b) Service data	
k	the index of vehicles
Cap	the maximum capacity of vehicles
K	the set of vehicles
c_0	fixed cost of a vehicle in use, 100
c_1	salary cost of unit distribution duration, 1
d_1	deterioration rate during driving and waiting duration, 0.002
d_2	deterioration rate during service duration, 0.003
c) Transportation data	
dis_{ij}	the driving distance from i to j
$dur_{ij}(t)$	the driving duration from i to j when departing from i at time t
$sp_{ij}(t)$	the driving speed from i to j at time t
C_{ij}	the sequence of the constant speeds from i to j of each hour in a day
B) Decision Variables	
$x_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ travels from } i \text{ to } j \text{ (} i \neq j \text{)} \\ 0, & \text{otherwise} \end{cases}$	$i, j \in V, k \in K$
$y_k = \begin{cases} 1, & \text{if vehicle } k \text{ is used} \\ 0, & \text{otherwise} \end{cases}$	$k \in K$
C) Other Variables	
$t_1(i)$	the driving and waiting duration that the goods of customer i have experienced when they are delivered
$t_2(i)$	the total service duration of customers that are served by the same vehicle with customer i and are served before i
ql_i	the quality losses of unit goods when i receives the goods
a_{ijk}	the time when vehicle k arrives at j from i
s_{ik}	the time when vehicle k begins the service of i
l_{ik}	the time when vehicle k leaves i

customers, the search space of the multisource data-associated multiobjective CCL model can be represented as a directed graph, $G = (V, A)$. $V = \{0, 1, 2, \dots, n, n + 1\}$ represents the vertex set of depot and customers. Herein, 0 is the depot vertex that vehicles depart from, and $n + 1$ is the depot vertex that vehicles return to after finishing distribution tasks, both of which refer to the same depot but use different indexes for clarity. Besides, the remaining vertices represent customers, and the customer vertex set is denoted as N ($N \subseteq V$). The set of all edges between any two vertices is denoted as A .

Three objectives are considered in this model, which are denoted as quality losses, personnel and vehicle costs, and transportation costs. Meanwhile, the hard time window constraint and capacity constraint are considered in this model. Given a customer i and its time window $[b_i, e_i]$, the service of i cannot begin before b_i or after e_i . Furthermore, each vehicle departs from the depot at b_0 and must return to the depot before e_{n+1} , which can be regarded as the hard time window constraint of the depot. To satisfy the capacity constraint, the total demand of goods distributed by a vehicle cannot exceed the maximum capacity of this vehicle.

2) *Multisource Data Association*: As our CCL model is oriented to a real transportation environment, three types of

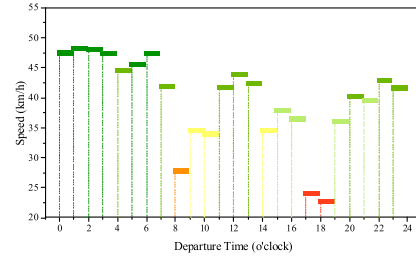


Fig. 1. Schematic diagram of the step function between the departure time and driving speed.

data are used in our CCL model. The first type of data is order data that contain all necessary information of each customer, including their location, demand, time windows, and service duration. The second type of data is the service data, including the number of vehicles that can be used, the maximum capacity of each vehicle, the fixed cost of vehicle usage, the salary cost of unit time, and the deterioration rates of goods. The third type of data is transportation data, including the driving distance, driving duration, and driving speed. These three types of data are regarded as multisource data because they are collected from different sources. Multisource data association is a promising way to build a more practical model [41]. Therefore, the associations among these three types of data are constructed when they are used as the input of our model.

3) *Real Transportation Environment Consideration*: To introduce the real transportation environment information in our model, the driving duration between any two places at different times and the driving distance between any two places are captured. Then, the average captured driving duration in one hour is adopted to calculate the driving speed in that hour. In this way, the step function between the departure time and driving speed can be constructed, and the driving duration from one place to another place at any time can be derived through the step function.

Specifically, the step function based on real-captured transportation data can be formulated as

$$sp_{ij}(t) = C_{ij}(\lfloor t \rfloor) \quad (1)$$

where $sp_{ij}(t)$ is the driving speed from vertex i to vertex j at time t ($t \in [0, 24)$). C_{ij} contains the constant speeds from i to j of each hour in a day, i.e., $\{C_{ij}(0), C_{ij}(1), \dots, C_{ij}(23)\}$, which is calculated based on real-captured data. For clarity, an example is provided in Table S.I in the supplementary material and Fig. 1 to help explain the step function. As a part of Table S.I, Table II shows the captured driving duration between two places during the first 9 hours in a day. The driving duration is captured twice every hour in the example and the captured data are shown as the columns named num1 and num2, respectively; and the average duration is shown as the column named Average. Given the real distance, the time-dependent speed in each hour can be obtained, leading to the step function shown in Fig. 1 for each hour in a day.

Moreover, unlike the TDVRP in which any two places share the same step function, each pair of places has its step function in our model. Note that the step function can be established between the departure time and either the driving speed or driving duration. The driving speed is chosen in our model

TABLE II
EXAMPLE OF REAL-CAPTURED TRANSPORTATION DATA

Departure Time(o'clock)	Duration (h)			Distance (km)	Speed (km/h)
	num1	num2	Average		
[0:00, 1:00)	0.627	0.620	0.624	29.665	47.540
[1:00, 2:00)	0.622	0.609	0.616	29.665	48.157
[2:00, 3:00)	0.616	0.620	0.618	29.665	48.002
[3:00, 4:00)	0.634	0.618	0.626	29.665	47.388
[4:00, 5:00)	0.653	0.681	0.667	29.665	44.475
[5:00, 6:00)	0.623	0.676	0.650	29.665	45.638
[6:00, 7:00)	0.618	0.634	0.626	29.665	47.388
[7:00, 8:00)	0.758	0.654	0.706	29.665	42.018
[8:00, 9:00)	0.900	1.238	1.069	29.665	27.750

because it can ensure the satisfaction of the first-in-first-out (FIFO) property [17]. Herein, the FIFO property refers to the fact that given the same origin and destination, a vehicle that departs later cannot arrive at the destination before a vehicle that departs earlier.

B. Model Formulation

Three objectives are considered in the multisource data-associated multiobjective CCL model. The first objective is the total quality losses of goods when customers receive them, which is calculated as

$$f_1 = \sum_{k \in K} y_k \times \left(\sum_{j \in V} \sum_{i \in N} x_{jik} \times q_i \times ql_i \right) \quad (2)$$

where q_i is the demand of customer i , and ql_i is the quality losses of unit goods for customer i . Since the quality of goods at time t is always measured as $Q(t) = Q(0) \times e^{-d \times t}$ ($Q(0) = 1$, d is the deterioration rate) [42], the delivered quality losses of unit goods for customer i is measured as

$$ql_i = 1 - e^{-(d_1 \times t_1(i) + d_2 \times t_2(i))} \quad (3)$$

where the distribution duration of goods for customer i is decomposed into the duration of driving and waiting (i.e., $t_1(i)$) and the duration of serving other customers that are served by the same vehicle with i and are served before customer i (i.e., $t_2(i)$). In addition, the deterioration rates of $t_1(i)$ and $t_2(i)$ are denoted as d_1 and d_2 , respectively. Two different deterioration rates are considered because the vehicle carriage is opened when serving customers, which can result in a higher deterioration rate when compared with the situation in which the vehicle carriage is closed (i.e., the vehicle is driving or waiting). In this way, the longer distribution duration of perishable goods (including both the driving and waiting duration $t_1(i)$ and the service duration $t_2(i)$) will lead to a larger value calculated by (3), reflecting worse quality losses.

The second objective is the personnel and vehicle costs, which are measured by the sum of the fixed costs of vehicle usage and the salary costs for the distribution duration. Thus, the formulation of the second objective is

$$f_2 = \sum_{k \in K} y_k \times c_0 + \sum_{k \in K} y_k \times (s_{(n+1)k} - s_{0k}) \times c_1 \quad (4)$$

where the first term of (4) is the fixed costs of vehicle usage, the second term of (4) is the salary costs, and the distribution

duration of vehicle k is computed as the difference between $s_{(n+1)k}$ and s_{0k} . Herein, c_0 is the fixed cost of a vehicle in use, and c_1 is the salary cost of unit distribution duration.

The third objective is the transportation costs that depend on the driving distance and driving duration of vehicles. Hence, the third objective can be measured as

$$f_3 = \sum_{k \in K} y_k \times \left(\sum_{i \in V} \sum_{j \in V} x_{ijk} \times dis_{ij} \right) + \sum_{k \in K} y_k \times \left(\sum_{i \in V} \sum_{j \in V} x_{ijk} \times dur_{ij}(l_{ik}) \right) \quad (5)$$

where the first term is used to compute the driving distance, and the second term is used to compute the driving duration. The variable l_{ik} refers to the time that vehicle k departs from i .

In summary, f_1 is the minimization of the quality losses, while f_2 and f_3 are related to the distribution costs. It should be noted that given enough vehicles, each customer can be served at b_i ($1 \leq i \leq n$) (i.e., f_1 is minimized), but it may lead to too much waiting time, expensive vehicle costs, and redundant driving distance and duration. That is, the minimization of f_1 may lead to increases in f_2 and f_3 . Hence, f_1 conflicts with f_2 and f_3 . Furthermore, with the constraint of hard time windows, the total distribution duration of a path with the shortest driving distance may be longer than that of a path with a longer driving distance if the waiting duration of the former is much greater than that of the latter. In this case, the minimization of f_2 may result in an increase in f_3 and vice versa. Hence, f_2 conflicts with f_3 to some extent.

To optimize the three objectives mentioned above, the multisource data-associated multiobjective CCL model can be formulated as

$$\min f = \begin{cases} f_1 \\ f_2 \\ f_3 \end{cases} \quad (6)$$

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1, \quad \forall i \in N \quad (7)$$

$$\sum_{i \in N} x_{0ik} - y_k = 0, \quad \forall k \in K \quad (8)$$

$$\sum_{i \in N} x_{i(n+1)k} - y_k = 0, \quad \forall k \in K \quad (9)$$

$$\sum_{j \in V} \sum_{i \in N} x_{jik} \times q_i - Cap \leq 0, \quad \forall k \in K \quad (10)$$

$$a_{ijk} = x_{ijk} \times (l_{ik} + dur_{ij}(l_{ik})), \quad \forall k \in K, \quad \forall (i, j) \in A \quad (11)$$

$$l_{ik} = s_{ik} + st_i, \quad \forall k \in K, \quad \forall i \in V \quad (12)$$

$$s_{ik} = \max \left\{ b_i, \max_{j \in V} \{ x_{jik} \times a_{jik} \} \right\}, \quad \forall k \in K, \quad \forall i \in V \quad (13)$$

$$b_i \leq s_{ik} \leq e_i, \quad \forall k \in K, \quad \forall i \in V \quad (14)$$

$$t_1(i) = \begin{cases} \max_{k \in K, j \in V} \{x_{jik} \times (t_1(j) + s_{ik} - l_{jk})\}, & \text{if } i \in V \setminus 0 \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

$$t_2(i) = \begin{cases} \max_{k \in K, j \in V} \{x_{jik} \times (t_2(j) + st_j)\}, & \text{if } i \in V \setminus 0 \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

$$x_{ijk}, y_k \in \{0, 1\}, \quad \forall i, j \in V, \quad \forall k \in K \quad (17)$$

where (6) is the objective of the CCL model. Constraint (7) means that each customer can only be served once by one vehicle. Constraints (8) and (9) indicate that each vehicle in use should depart from the depot at the beginning of the distribution and return to the depot after finishing all the distribution tasks, respectively. Constraint (10) is the capacity constraint. Constraints (11), (12), and (13) provide the calculations of a_{ijk} , l_{ik} , and s_{ik} . The time window constraint is specified as constraint (14). Finally, the distribution duration of goods for customer i for quality losses calculation is formulated as constraints (15) and (16). Constraint (17) shows that both decision variables can either be 0 or 1.

III. MPMOACS APPROACH

In this section, the proposed MPMOACS approach that is used to solve the CCL scheduling problem is described first. Then, as rescheduling is required when disruption events happen during the distribution process, a rescheduling strategy is introduced at the end of this section, i.e., in Section III-F.

The MPMOACS is an approach developed from the ACS and the MPMO framework. MPMO is a concise but effective framework for multiobjective optimization, which adopts M ($M \geq 2$) populations for M objectives and one population focuses on one objective [38]. Besides, an information-sharing mechanism is designed to achieve the coevolution of multiple populations. Hence, three populations are required for our CCL model because there are three objectives. Herein, these three populations are denoted as three colonies since our approach is an ACS-based approach. The first colony for f_1 (i.e., quality losses) is named *Qcolony*, the second colony for f_2 (i.e., personnel and vehicle costs) is named *Pcolony*, and the third colony for f_3 (i.e., transportation costs) is named *Tcolony*.

The MPMOACS procedure is shown in **Algorithm 1**. First, in line 2, an archive named *Archive* is initialized as an empty set, which is used to store the Pareto solutions, and three pheromone matrixes for three colonies are constructed and initialized. Note that each colony constructs and updates its pheromone matrix individually in order to optimize its objective more effectively. Second, the solution construction for each ant of the three colonies is shown in lines 5 to 8. Moreover, after each ant (i.e., solution) is obtained, all its three objective fitness values are calculated in line 9. Third, the pheromone is updated locally in line 10. Fourth, in line 11, the solutions obtained from the three colonies are used to update the *Archive* through non-dominated sort. Fifth, the pheromone of each colony is updated globally in lines 12 to 14. Finally,

Algorithm 1: Procedure of the MPMOACS

Input: Order data, service data, transportation data

Output: All Pareto solutions in *Archive*

1 Begin

2 Initialization of *Archive* and Pheromone;

3 $pop \leftarrow 10$;

4 **while** *terminal condition is not satisfied* **do**

5 $ColonySet = \{Qcolony, Pcolony, Tcolony\}$

6 **foreach** *colony in ColonySet* **do**

7 **for** $i = 1 : pop$ **do**

8 Solution construction of ant i ;

9 Calculate the tree objective fitness values of ant i ;

10 Local update for pheromone matrix of *colony*;

11 *Archive* update;

12 **foreach** *colony in ColonySet* **do**

13 **foreach** *solution in Archive* **do**

14 Global update for pheromone matrix of *colony*;

15 Local search of *Archive* using RBLS;

16 **End**

as shown in line 15, the RBLS strategy is applied to improve the solutions in the *Archive*.

A. Solution Encoding

The encoding of a solution should specify the distribution route of each vehicle and the distribution sequence of each customer. Besides, the encoding format must be concise. Thus, a solution of MPMOACS is encoded as a two-dimensional array, and each row of the array represents the distribution route of a vehicle. That is, if m vehicles are used for distribution in a solution, the solution will be encoded as an array with m rows.

An example of a solution with 5 customers and 2 vehicles is shown in Fig. 2. Route 1 is the distribution route of vehicle 1, and Route 2 is the distribution route of vehicle 2. According to Fig. 2, Route 1 is $0 \rightarrow 3 \rightarrow 1 \rightarrow 0$, and Route 2 is $0 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 0$. That is, vehicle 1 serves customer 3 first after departing from the depot, then it serves customer 1; and finally, it returns to the depot. Meanwhile, vehicle 2 serves customer 2 first, and then it serves customers 4 and 5 in order. Identically, vehicle 2 returns to the depot once the service of customer 5 is finished. Therefore, the encoding result of the solution *sol* in Fig. 2 is

$$sol = \begin{bmatrix} 0 & 3 & 1 & 6 & 6 \\ 0 & 2 & 4 & 5 & 6 \end{bmatrix} \quad (18)$$

where the last element of each row represents the depot, which is denoted as $n + 1$. Herein, the last element is 6 because n is 5. The first row of *sol* is the distribution route of vehicle 1, and the second row is for vehicle 2.

B. Solution Construction

To construct a solution, an ant first initializes a solution as a two-dimensional array with an element 0 at the beginning of

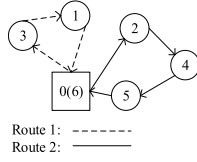


Fig. 2. Schematic diagram of a solution.

the first row, which means that the first vehicle is scheduled to depart from the depot. Then, the ant constructs a complete solution by iteratively determining the next customer to be served following the state transition rule. The state transition rule is formulated as

$$j = \begin{cases} \arg \max_{u \in J} \left\{ \tau^*(i, u) \times [\eta^*(i, u)]^\beta \right\}, & \text{if } q \leq q_0 \\ \text{roulette wheel selection,} & \text{otherwise} \end{cases} \quad (19)$$

where i is the current customer vertex visited by the current vehicle, and β is a parameter. J is the set of customers that have not been assigned to any vehicle until now and can be served by the current vehicle without constraint violation. If J is empty, another empty vehicle is put into use. In this case, the solution (i.e., the array) sets the next row to present that a new vehicle is used. The $\tau^*(i, u)$ and $\eta^*(i, u)$ represent the pheromone information and heuristic information between the customers i and u , respectively.

According to the MPMO framework, each colony has its pheromone matrix and the heuristic information tailored for its objective. To distinguish the pheromone matrix and heuristic information of a colony from the other colonies, a variable denoted as $*$ is used. If the ant comes from $Qcolony$, $*$ is $Qcolony$. Similarly, $*$ is $Pcolony$ when the ant comes from $Pcolony$, and $*$ is $Tcolony$ when the ant comes from $Tcolony$.

Moreover, a parameter q_0 ($0 \leq q_0 \leq 1$) is introduced to help determine the next customer to be served. If a random number q ranging from 0 to 1 is less than or equal to q_0 , customer u with the greatest value of $\tau^*(i, u) \times [\eta^*(i, u)]^\beta$ is chosen as the next customer. Otherwise, the next customer is determined through the roulette wheel selection, and the selection probability of customer j is calculated by

$$p(i, j) = \begin{cases} \frac{\tau^*(i, j) \times [\eta^*(i, j)]^\beta}{\sum_{u \in J} \tau^*(i, u) \times [\eta^*(i, u)]^\beta}, & \text{if } j \in J \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

C. Heuristic Information Design

To determine Pareto solutions with high quality, different heuristic information should be designed for different colonies according to their objectives. For the ants of $Qcolony$, the formulation of heuristic information is

$$\eta^{Qcolony}(i, j) = h_1(j) \times h_2(j) \times h_3(j) \quad (21)$$

First, to minimize the quality losses, the goods should be delivered to customers as soon as customers can accept the services. Hence, the ants of $Qcolony$ should prefer to select customer j if j can be served as early as possible once j is ready to accept service. In this case, the first term of

$\eta^{Qcolony}(i, j)$ is formulated as

$$h_1(j) = e^{-(s_{jk} - b_j)} \quad (22)$$

where the maximum value of $h_1(j)$ is 1, which occurs when s_{jk} equals b_j . The value of $h_1(j)$ decreases as the difference between s_{jk} and b_j increases. Meanwhile, considering that the deterioration rate is identical for goods of different customers, ants may prefer to serve customers with greater demand first because the distribution duration of more goods can be reduced, which is conducive to reducing the total quality losses of goods. Hence, the second term of $\eta^{Qcolony}(i, j)$ can be represented as

$$h_2(j) = \frac{q_j}{Cap - \sum_{r \in V_k} q_r} \quad (23)$$

where the denominator term is the remaining capacity of vehicle k , and V_k represents the set of customers that have been assigned to vehicle k . In this way, $h_2(j)$ is proportionate to q_j . Nevertheless, if only $h_1(j)$ and $h_2(j)$ are considered, few feasible solutions can be found by $Qcolony$ since the number of vehicles required by the ants exceeds the number of vehicles that are provided. This case occurs because the driving and waiting duration is ignored by $h_1(j)$ and $h_2(j)$, which may mislead ants to make vehicle k wait at the destination of a customer and assign other customers to new vehicles even though the waiting duration is sufficient for vehicle k to serve other customers. Therefore, the duration of driving and waiting should be considered to reduce the wastage of time so that the number of vehicles used can be reduced. Based on the analysis mentioned above, the last term of $\eta^{Qcolony}(i, j)$ is formulated as

$$h_3(j) = (s_{jk} - l_{ik})^{-1} \quad (24)$$

which is the reciprocal of the duration of driving and waiting required for vehicle k from leaving customer i at l_{ik} to start serving customer j . A greater value of $h_3(j)$ indicates less driving duration or less waiting duration.

To minimize f_2 (i.e., the personnel and vehicle costs), the number of vehicles being used and the total distribution duration should be optimized. In other words, customers with less waiting or driving duration should be served first. Meanwhile, customers with higher urgency should have higher priority. The motivation of this heuristic information comes from the previous work [43] on the VRPTW. Thus, the formulation of heuristic information for f_2 is

$$\eta^{Pcolony}(i, j) = \frac{1}{s_{jk} - l_{ik}} \times \frac{1}{e_j - l_{ik} - dur_{ij}(l_{ik})} \quad (25)$$

where the first term of (25) is the same as (24), and the second term is the measurement of urgency.

As for f_3 , the objective of minimizing the transportation costs, the heuristic information should focus on the minimization of the driving distance and driving duration. One note is that both the distance and duration should be considered since the driving distance is fixed while the driving duration is time-dependent. Thus, the heuristic information of f_3 is

simply determined as

$$\eta^{Tcolony}(i, j) = \frac{1}{dur_{ij}(l_{ik})} + \frac{1}{dis_{ij}} \quad (26)$$

D. Pheromone Setting

1) *Pheromone Initialization*: The pheromone initialization of each colony is conducted according to

$$\tau_0^* = \begin{cases} (|Num| \times f_1(sol_{greedy}))^{-1}, & \text{if } * \text{ is } Qcolony \\ (|Num| \times f_2(sol_{greedy}))^{-1}, & \text{if } * \text{ is } Pcolony \\ (|Num| \times f_3(sol_{greedy}))^{-1}, & \text{if } * \text{ is } Tcolony \end{cases} \quad (27)$$

where *Num* refers to the number of both customers and the depot, and *sol_{greedy}* is an initial solution constructed through a greedy heuristic. In detail, *sol_{greedy}* is constructed by iteratively choosing the customer nearest to the current position of the vehicle as the next customer to be served. If no customers can be served by the vehicle being used, a new empty vehicle is put into use. Note that although the τ_0 of the three colonies are calculated based on the same solution (i.e., *sol_{greedy}*), they are different since different objective fitness values of *sol_{greedy}* are used.

2) *Pheromone Update*: Once an ant finishes the construction of a solution, the local update of the pheromone is carried out. For each edge traversed by the ant, the pheromone on the edge is updated according to

$$\tau^*(i, j) = (1 - \rho) \times \tau^*(i, j) + \rho \times \tau_0^* \quad (28)$$

where $*$ represents *Qcolony*, *Pcolony*, or *Tcolony*, which depends on the colony to which the ant belongs. Parameter ρ is the degree of update of the pheromone.

The global update is executed when all ants in three colonies complete the solution construction. Since the CCL model is multiobjective, a set of solutions (i.e., the Pareto solutions) can be regarded as the global best solutions due to the non-dominated relationship. Therefore, an archive with limited capacity, termed *Archive*, is introduced to store the Pareto solutions obtained by the three colonies during the search process. The update of *Archive* can be described as follows. In every generation, the solutions obtained by three colonies are added to *Archive*. Then, the solutions dominated by any other solutions are deleted from *Archive*. If the number of solutions in *Archive* still exceeds the maximum capacity of *Archive*, the crowding distance [24] of each solution is calculated. Then, solutions with the minimum crowding distance are removed from *Archive* until the number of solutions in *Archive* is reduced to the maximum capacity of *Archive*. Finally, the global update of pheromone is conducted on all solutions in *Archive*, which is shown as

$$\tau^*(i, j) = (1 - \varepsilon) \times \tau^*(i, j) + \varepsilon \times \begin{cases} (f_1(sol))^{-1}, & \text{if } * \text{ is } Qcolony \\ (f_2(sol))^{-1}, & \text{if } * \text{ is } Pcolony \\ (f_3(sol))^{-1}, & \text{if } * \text{ is } Tcolony \end{cases} \quad (29)$$

where *sol* is a solution in *Archive*. Parameter ε represents the pheromone enhancement degree.

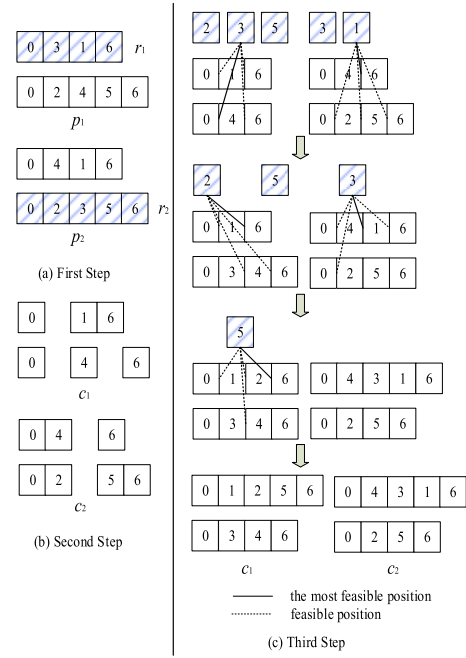


Fig. 3. Schematic diagram of the BCRC.

E. Ranking-Based Local Search Strategy

The RBLS strategy is applied to the solutions in *Archive*, and the neighbor operator adopted in the RBLS is the best cost route crossover (BCRC) proposed in [44]. The BCRC is a crossover operator originally designed for the genetic algorithm, which can also be applied in the MPMOACS. The main idea of the BCRC can be described as follows. First, two routes, r_1 and r_2 , are randomly selected from two different parent solutions, p_1 and p_2 , respectively. Second, customers served in r_1 are removed from p_2 , and customers served in r_2 are removed from p_1 . Finally, customers in r_1 are reinserted into the most feasible position of p_2 in random order. A similar operator is conducted on r_2 and p_1 . Note that a position is considered feasible for a customer as long as the customer can be served without violating any constraints.

A schematic diagram of the BCRC is shown in Fig. 3. In the first step, the first route of p_1 is selected as r_1 , and the second route of p_2 is selected as r_2 . Then, customer 3 and customer 1 are removed from p_2 ; and customers 2, 3, and 5 are removed from p_1 , resulting in c_1 and c_2 shown in Fig. 3(b). Finally, the customers of r_1 and r_2 are reinserted into the most feasible positions of c_2 and c_1 , respectively. If there is no feasible position for a customer, a new empty vehicle is added.

As a problem-specific operator for the VRPTW, the BCRC measures the most feasible position of a customer through the total costs of routes. However, the truly optimal position of a customer cannot be determined directly because there are three objectives in our model. To address the measurement of the feasible degree of a position, ranking-based fitness (*RF*) is introduced. The position with the maximum *RF* is regarded as the most feasible position for a customer to insert. The major steps of the calculation of *RF* are as follows.

First, the ranking vectors of p_1 and p_2 , which are denoted as rk_1 and rk_2 , respectively, are obtained. Specifically, both rk_1

Algorithm 2: Ranking-Based Local Search

Input: *Archive*
Output: *Archive*

```

1 Begin
2  $maxCnt \leftarrow 10, cnt \leftarrow 0;$ 
3 while  $cnt < maxCnt$  and  $|Archive| \geq 2$  do
4    $p_1 = RandomSelection(Archive);$ 
5    $p_2 = RandomSelection(Archive \setminus p_1);$ 
6   Calculation of  $w_1, w_2;$ 
7    $BCRC(p_1, p_2, c_1, c_2);$ 
8   if both  $c_1, c_2$  are feasible then
9     if  $RF(c_1) \leq 1$  and  $RF(c_2) \leq 1$  then
10       $cnt = cnt + 1;$ 
11   else
12      $cnt = cnt + 1;$ 
13   if  $c_1$  is feasible then
14     Update Archive using  $c_1;$ 
15   if  $c_2$  is feasible then
16     Update Archive using  $c_2;$ 
17 End

```

and rk_2 are vectors with three elements, corresponding to the three objectives of the CCL model, that is, $rk_1 = \{rk_{11}, rk_{12}, rk_{13}\}$ and $rk_2 = \{rk_{21}, rk_{22}, rk_{23}\}$. For objective o ($o = 1, 2, 3$) of p_i , rk_{io} represents the ranking of p_i among the solutions in *Archive* according to the ascending order of f_o , i.e., the fitness value of objective o . Then, for each newly generated solution c_i ($i = 1, 2$), the ranking-based fitness of c_i , denoted as $RF(c_i)$, is calculated as

$$RF(c_i) = \sum_{o=1}^3 w_{io} \times \frac{f_o(p_i)}{f_o(c_i)} \quad (30)$$

$$w_{io} = \frac{rk_{io}}{rk_{i1} + rk_{i2} + rk_{i3}} \quad (31)$$

where w_{io} is the weight depending on the ranking of objective o of p_i . A higher ranking indicates a larger weight. It is designed in this way because we think a higher ranking may imply more room for improvement. However, due to the conflict among objectives, the improvement of an objective may result in recessions of other objectives. Herein, c_i is regarded as better than p_i if $RF(c_i)$ is larger than 1. This is reasonable since at least one objective is improved when $RF(c_i)$ exceeds 1.

Finally, with the combination of the BCRC, the complete RBLS strategy is shown in **Algorithm 2**. To begin with, two different solutions p_1 and p_2 are selected randomly from the *Archive*, and then the BCRC operator is conducted on p_1 and p_2 , resulting in two new solutions (i.e., c_1 and c_2). As described in lines 8 to 12, a local search is considered successful only when both c_1 and c_2 are feasible and at least one of them is better than their parents (i.e., $RF(c_i) > 1$), and the variable cnt is used to record the number of failed local search operations. Once the number of failed local search operations reaches $maxCnt$ or the size of *Archive* is less than 2,

Algorithm 3: Rescheduling Strategy

Input: an old distribution scheme, *oldSol*
Output: the rescheduling distribution scheme, *newSol*

```

1 Begin
2  $rnum \leftarrow$  the number of routes in oldSol;
3 for  $i = 1 : rnum$  do
4    $r \leftarrow$  the  $i^{th}$  route in oldSol;
5    $\Phi \leftarrow$  the set of all undistributed orders in  $r$ ;
6   for  $j = 1 : |\Phi| - 1$  do
7     for  $k = j + 1 : |\Phi|$  do
8       /* for every two orders in  $\Phi$ ,
9        perform the 2-opt operator
10      */
11        $r' \leftarrow r$  with distribution sequence of  $\Phi(j)$ ,
12        $\Phi(k)$ , and all orders between them reversed;
13        $sol \leftarrow oldSol$  with the  $i^{th}$  route replaced by  $r'$ ;
14       if  $sol$  is better than oldSol then
15          $oldSol \leftarrow sol;$ 
16  $newSol \leftarrow oldSol;$ 
17 End

```

the RBLS is terminated. Besides, lines 13 to 16 indicate that c_1 and c_2 should be used to update the *Archive* through non-dominated sort if they are feasible.

F. Rescheduling Strategy

During the distribution process, the time-dependent driving speed in the real traffic network may be changed (e.g., reduced) unexpectedly due to disruption events like traffic accidents and/or bad weather. In this case, the distribution scheme obtained based on the normal situation without disruption events may become not as efficient as expected, rescheduling of the remaining undistributed orders is thus required. Therefore, a rescheduling strategy is proposed to deal with the rescheduling problem.

Algorithm 3 describes the procedure of the rescheduling strategy. The main idea is to reschedule the routes of an old solution (i.e., the old distribution scheme) one by one based on the 2-opt [22] operator. As shown in lines 5 to 8, when rescheduling an old route, the distribution sequence of any two undistributed orders and all orders between them are reversed, leading to a new route. For example, for a route $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$ with two selected undistributed orders as B and E, the resulting route is $A \rightarrow E \rightarrow D \rightarrow C \rightarrow B \rightarrow F$. In this way, every two undistributed orders in an old route can result in a new route, and a series of new routes for an old route are thus obtained. Then, the best new route among all the new routes is used to replace the old route if it is better than the old route. When all the old routes have been rescheduled, the resulting new solution is adopted as the rescheduling result.

IV. EXPERIMENTAL STUDIES**A. Dataset Generation**

Since the CCL model in this paper is oriented to the real transportation environment, a CCL dataset that considers

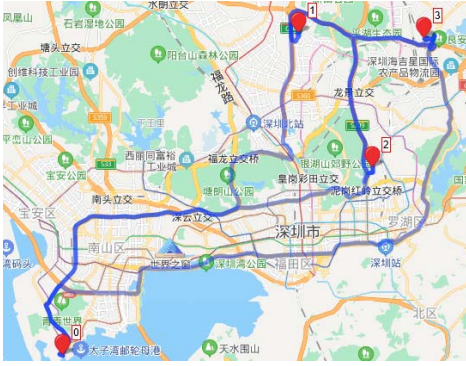


Fig. 4. Road network for the four selected places of Shenzhen, China.

the real transportation environment, which is not provided in previous work, is needed. Therefore, we establish a multi-source dataset based on real-captured transportation data of Guangzhou, China and Shenzhen, China.

First, to construct a test instance with n customers and 1 depot, $n + 1$ places are randomly selected to be the positions of customers and depot. Meanwhile, the road network that contains the driving paths between any two selected places is constructed. A schematic diagram of the road network based on the four selected places of Shenzhen, China is shown in Fig. 4. Second, the driving distance and time-dependent driving duration between any two places selected in the previous step are captured according to the road network. Note that the driving distance and driving duration are captured multiple times a day, and the capture operation is repeated for several days. Third, the average driving duration between any two places for each hour in a day can be obtained. Then, the average driving speed for each hour in a day that depends on both the driving distance and the average driving duration is calculated, resulting in the step function between the departure time and driving speed. In this case, the driving duration between any two places at any time can be calculated according to the step function.

Besides transportation data, order data and service data are indispensable for a complete test instance, which are generated randomly. To increase the diversity of instances, two distributions of time windows are adopted for each instance: a Gaussian distribution and a random distribution. In this way, 28 instances (14 for Guangzhou and 14 for Shenzhen) can be designed, and n is 30 and 40 for the instances of Guangzhou and Shenzhen, respectively. The 7 instances of Guangzhou with Gaussian distribution are denoted as GZ-G1, GZ-G2, ..., and GZ-G7, while the 7 instances of Guangzhou with random distribution are denoted as GZ-R1, GZ-R2, ..., and GZ-R7. Similarly, the 14 instances of Shenzhen are denoted as SZ-G1, SZ-G2, ..., SZ-G7, SZ-R1, SZ-R2, ..., and SZ-R7. The multi-source dataset can be downloaded from [45].

B. Experimental Design

1) *Experimental Purposes*: The experiments are conducted to achieve three purposes. The first purpose is to investigate the importance of real transportation environment consideration, and the second purpose is to evaluate the performance of the

MPMOACS on the multi-source data-associated multiobjective CCL model. The last purpose is to investigate the effectiveness of the RBLs strategy and the rescheduling strategy.

To achieve the first purpose, four modes of the CCL model are designed according to the adoption of different transportation data, and the MPMOACS approach is applied to each mode individually. The four modes correspond to four real transportation environments with different accuracies. The first mode is the CCL model with *linear distance* and *constant speed*, denoted as the LDCS mode. The linear distance between two places is similar to the Euclidean distance but considers the spherical property of the earth. The constant speed is determined as 60km/h. The second mode is the CCL model with *linear distance* and *real speed*, and the real speed is the time-dependent speed. Herein, the second mode is denoted as the LDRS mode. The third mode named RDCS is the CCL model with *real distance* and *constant speed*, and the real distance is the driving distance. The last mode is the CCL model with *real distance* and *real speed*, that is, with real driving distance and time-dependent speed, which is also the mode adopted in our model. Similarly, the last mode is denoted as the RDRS mode.

The experiments of the second purpose can be divided into two parts. The first part is experiments conducted on the multi-source dataset, and the MPMOACS is compared with the other six approaches, including the NSGA-II [24], MOEA/D [25], CCMO [21], TS-MOEA [22], the multi-objective local search (MOLS) approach [18], and the hybrid ant colony optimization (HACO) approach [20]. The NSGA-II and MOEA/D are the classic approaches for multiobjective optimization. CCMO is a coevolutionary framework developed recently and has been used for the optimization of the multiobjective VRP. The TS-MOEA is also a recently proposed optimization approach for the multiobjective VRP. The MOLS is proposed to solve the multiobjective VRP through the cooperation of multiple local search operations, and the HACO is proposed to solve the multiobjective VRP by integrating mutation operators into the ant colony optimization approach. Since the multi-source data-associated multiobjective CCL model is also developed from the VRP, all approaches mentioned above can be adapted to solve this model. Therefore, it is reasonable to compare the MPMOACS with these approaches. The second part is the comparison experiments of the above approaches on six benchmark instances [43] for the TDVRP. The benchmark instances for the TDVRP are adopted due to the time-dependent speed used in the instances, which is similar to the real driving speed in the multi-source data-associated CCL model. With the suggestion of [13], the working duration of the depot is evenly divided into 5 intervals, and the speed of each interval is set as [1.00, 1.60, 1.05, 1.60, 1.00] in sequence.

The third purpose is to investigate the effectiveness of the RBLs strategy and the rescheduling strategy. Herein, for the investigation of RBLs strategy, the compared approach named MPMOACS-noLS is the MPMOACS without the RBLs, which is designed by removing the RBLs from the MPMOACS.

2) *Experimental Setting*: The parameter configurations of all approaches are given as follows. Unless otherwise

specified, parameter configurations of all the compared approaches follow the suggestions of their original papers since these parameter configurations have been well-tuned in their original papers to solve the problems that are related to our proposed multiobjective CCL model. For the MPMOACS, the size of each colony is set as 10. The parameter q_0 for state transition is 0.9, and β is 2. The values of both ρ and ε are set as 0.1. The maximum size of *Archive* is 300. The population size of the NSGA-II is 300, the crossover rate p_c is 0.9, and the mutation rate p_m is $1.0/n$. As introduced in [25], the population size of the MOEA/D is 351 since there are three objectives. The size of the neighbor, T , is 20. The crossover rate p_c is 1.0, and the mutation rate p_m is $1.0/n$. Regarding CCMO, the population size is 50, p_c is 1.0, and p_m is also $1.0/n$. The TS-MOEA is a two-stage approach. The first stage is based on the EC-NSGA-II with a population of 100, and the second stage is developed from the MOEA/D with a population of 351 and T of 5. For MOLS, the population size is 300, the coefficient of tournament size is 0.5, and the population is updated in every iteration. In HACO, the population size is 35, the mutation rate is 0.6, q_0 is 0.6, ρ is 0.7, and the values of α , β , γ are 1, 4, 2 in order.

The execution time of all approaches on each instance in the multisource dataset is set as 200 seconds. For benchmark instances, the execution time is set as 300 seconds since the instance scales are larger. Considering the stochastic factors, each approach is conducted on each instance 10 times independently and the average results are compared. The experimental environment is a PC with an Intel i5-7400 CPU and 8.00 GB RAM.

3) *Performance Metrics*: Two performance metrics are employed for the comparisons among various modes or approaches.

The first metric is the hypervolume (HV) performance metric. The HV is a performance metric that can measure both the convergence and diversity of a mode/approach through the hypervolume constructed by the solutions obtained by the mode/approach and a reference point. A better mode/approach always has a larger HV. To calculate the HV, we first normalize the objective fitness values of all solutions obtained by all modes/approaches in all runs and set the reference point as [1.00, 1.00, 1.00]. Then, the HV is computed for each mode/approach in each run, and the average HV is adopted for comparison. Besides, the Wilcoxon rank-sum test at 0.05 significance level is used to compare the HV of RDRS mode/MPMOACS with the HV of the compared modes/approaches, and the comparison results are represented as '+', '-', or '≈', meaning that the performance of RDRS mode/MPMOACS is significantly superior, significantly inferior, or similar to that of the compared modes/approaches.

The second metric is $C(A, B)$ [46], which is used to compare the dominant relationship between solutions of the MPMOACS and other approaches. For approaches A and B , $C(A, B)$ is defined as

$$C(A, B) = \frac{| \{b \in B | \exists a \in A : a \text{ dominates or equals } b\} |}{|B|} \quad (32)$$

Approach A is regarded as better than approach B if $C(A, B)$ is greater than $C(B, A)$, meaning that more solutions of B are dominated by or equal to solutions of A . Herein, the $C(A, B)$ is calculated based on all non-dominated solutions found in all runs.

C. Importance of Real Transportation Environment Consideration

As introduced in Section IV-B, four modes of the CCL scheduling model, which are denoted as LDCS, LDRS, RDCS, and RDRS, are proposed to investigate the importance of considering the real transportation environment. Herein, the solutions obtained in the LDCS, LDRS, and RDCS are directly applied to the RDRS mode; and the average feasibility rates are shown in Table S.II in the supplementary material. The best results are shown in **boldface** for clarity.

It can be concluded from Table S.II in the supplementary material that the feasibility rates of all three modes are less than 40% and some of them are even 0, meaning that most solutions of traditional CCL models cannot be directly applied in the real transportation environment, which also indicates the importance of considering the real transportation environment. Besides, the LDRS dominates the other two modes on 19 of 28 instances, the RDCS surpasses the LDRS and LDCS on 8 instances, and the LDCS outperforms the other two modes on no instances. That is, both the LDRS and RDCS are better than the LDCS, and LDRS is even better than RDCS. This is because both LDRS and RDCS adopt more real information (i.e., real speed and real distance, respectively) than LDCS, leading to a more realistic transportation environment. In other words, the adoption of both real speed and real distance can improve the feasibility of scheduling results. Moreover, as LDRS is better than both LDCS and RDCS, the consideration of the real speed seems to be more important than that of the real distance.

Table S.III in the supplementary material presents the HV comparison results of the LDCS, LDRS, RDCS, and RDRS. It should be noted that only feasible solutions of the LDCS, LDRS, and RDCS are considered in the comparison. Besides, the HV of an instance with no feasible solutions is denoted as N/A, and the best results are shown in **boldface**. As can be observed, the HV of the RDRS is the greatest among the four modes on all instances in the multisource dataset. That is, the hypervolume constructed by solutions of the RDRS is larger than those of the other modes on the multisource dataset. Furthermore, the RDRS dominates the other three modes on all instances in terms of the statistical test, which shows the superiority of the RDRS. Thus, the necessity and significance of considering the real transportation environment for CCL scheduling are verified.

D. Investigation of the Performance of the MPMOACS

1) *Experimental Results on the Multisource Dataset*: In this subsection, the comparison results on two performance metrics (i.e., HV and $C(A, B)$) for the MPMOACS, NSGA-II, MOEA/D, CCMO, TS-MOEA, MOLS, and HACO are first presented and discussed. The best results are shown in

TABLE III
AVERAGE HV OF THE COMPARED APPROACHES AND THE MPMOACS ON THE MULTISOURCE DATASET

Instance	NSGA-II	MOEA/D	CCMO	TS-MOEA	MOLS	HACO	MPMOACS
GZ-G1	0.237 (+)	0.037 (+)	0.493 (+)	0.601 (+)	0.534 (+)	0.402 (+)	0.632
GZ-G2	0.335 (+)	0.046 (+)	0.541 (+)	0.586 (+)	0.452 (+)	0.406 (+)	0.633
GZ-G3	0.278 (+)	0.077 (+)	0.498 (+)	0.601 (+)	0.506 (+)	0.458 (+)	0.635
GZ-G4	0.257 (+)	0.033 (+)	0.596 (+)	0.688 (+)	0.556 (+)	0.466 (+)	0.702
GZ-G5	0.279 (+)	0.044 (+)	0.637 (+)	0.719 (+)	0.562 (+)	0.456 (+)	0.731
GZ-G6	0.160 (+)	0.018 (+)	0.559 (+)	0.622 (+)	0.509 (+)	0.333 (+)	0.677
GZ-G7	0.163 (+)	0.041 (+)	0.475 (+)	0.560 (≈)	0.426 (+)	0.302 (+)	0.575
GZ-R1	0.294 (+)	0.087 (+)	0.583 (+)	0.685 (+)	0.550 (+)	0.409 (+)	0.701
GZ-R2	0.241 (+)	0.060 (+)	0.525 (+)	0.586 (+)	0.419 (+)	0.420 (+)	0.611
GZ-R3	0.243 (+)	0.051 (+)	0.537 (+)	0.588 (+)	0.528 (+)	0.380 (+)	0.624
GZ-R4	0.245 (+)	0.063 (+)	0.474 (+)	0.545 (+)	0.536 (+)	0.313 (+)	0.592
GZ-R5	0.244 (+)	0.054 (+)	0.590 (+)	0.677 (≈)	0.595 (+)	0.381 (+)	0.681
GZ-R6	0.189 (+)	0.042 (+)	0.570 (+)	0.663 (+)	0.506 (+)	0.306 (+)	0.699
GZ-R7	0.214 (+)	0.025 (+)	0.514 (+)	0.574 (+)	0.474 (+)	0.386 (+)	0.614
SZ-G1	0.418 (+)	0.264 (+)	0.577 (+)	0.684 (+)	0.689 (+)	0.544 (+)	0.729
SZ-G2	0.380 (+)	0.120 (+)	0.514 (+)	0.609 (+)	0.564 (+)	0.459 (+)	0.679
SZ-G3	0.398 (+)	0.137 (+)	0.521 (+)	0.620 (+)	0.608 (+)	0.482 (+)	0.675
SZ-G4	0.427 (+)	0.122 (+)	0.537 (+)	0.606 (+)	0.620 (+)	0.453 (+)	0.676
SZ-G5	0.408 (+)	0.191 (+)	0.532 (+)	0.613 (+)	0.532 (+)	0.476 (+)	0.684
SZ-G6	0.388 (+)	0.098 (+)	0.514 (+)	0.609 (+)	0.541 (+)	0.347 (+)	0.667
SZ-G7	0.371 (+)	0.157 (+)	0.540 (+)	0.623 (+)	0.609 (+)	0.437 (+)	0.659
SZ-R1	0.365 (+)	0.209 (+)	0.531 (+)	0.635 (+)	0.651 (+)	0.504 (+)	0.718
SZ-R2	0.382 (+)	0.140 (+)	0.508 (+)	0.596 (+)	0.564 (+)	0.377 (+)	0.667
SZ-R3	0.382 (+)	0.146 (+)	0.512 (+)	0.596 (+)	0.561 (+)	0.443 (+)	0.657
SZ-R4	0.386 (+)	0.134 (+)	0.527 (+)	0.604 (+)	0.583 (+)	0.331 (+)	0.654
SZ-R5	0.343 (+)	0.155 (+)	0.516 (+)	0.600 (+)	0.587 (+)	0.474 (+)	0.645
SZ-R6	0.369 (+)	0.151 (+)	0.519 (+)	0.604 (+)	0.552 (+)	0.499 (+)	0.661
SZ-R7	0.379 (+)	0.133 (+)	0.499 (+)	0.599 (+)	0.574 (+)	0.425 (+)	0.656
num of +/≈/-	28/0/0	28/0/0	28/0/0	26/2/0	28/0/0	28/0/0	\

boldface. Then, the Pareto solutions of each approach on 2 instances are provided for a more intuitive comparison.

Table III provides the comparison results on the HV for all compared approaches and the MPMOACS. First, the HV of the MPMOACS is much better than that of both the NSGA-II and MOEA/D on all instances, which means that the MPMOACS is better than the NSGA-II and MOEA/D for multisource data-associated CCL model optimization on the multisource dataset. Moreover, the comparison results based on the statistical test also support the conclusion that the MPMOACS outperforms the NSGA-II and MOEA/D. This is reasonable since both the NSGA-II and MOEA/D are not problem-specific approaches designed for multisource data-associated CCL scheduling. Besides, the use of effective heuristic information also accounts for the better performance of the MPMOACS.

Comparison results of the CCMO, the TS-MOEA, and the MPMOACS on the multisource dataset also show the advantages of the MPMOACS. Specifically, the MPMOACS surpasses the CCMO and the TS-MOEA on all instances when the average HV is considered. Meanwhile, the last row of Table III shows that the MPMOACS significantly dominates the CCMO on all instances. Besides, the MPMOACS has significantly better performance on 26 instances when compared with the TS-MOEA while the TS-MOEA only obtains comparable results on 2 instances. Thus, it is reasonable to conclude that the MPMOACS has better performance on

CCL scheduling when compared with the CCMO and the TS-MOEA.

Meanwhile, Table III shows that MPMOACS outperforms MOLS and HACO on all instances in the multisource dataset in terms of both average HV and statistical test, while MOLS and HACO cannot surpass MPMOACS on any instance.

The $C(A, B)$ comparison results for all solutions obtained by each approach in 10 runs are shown in Table S.IV in the supplementary material. The table shows that the values of $C(A, B)$ are much greater than those of $C(B, A)$ on all instances when B is the NSGA-II or MOEA/D and A is the MPMOACS, meaning that the solutions of the MPMOACS dominate or equal most solutions of the NSGA-II and MOEA/D. The comparison results of the CCMO are similar to those of the NSGA-II and MOEA/D since $C(\text{MPMOACS}, \text{CCMO})$ also significantly exceeds $C(\text{CCMO}, \text{MPMOACS})$ on all instances. Besides, for cases where B is the NSGA-II, MOEA/D, or CCMO, the values of $C(B, A)$ are less than 0.100 on all instances while most values of $C(A, B)$ exceed 0.500, and some values of $C(A, B)$ even reach 1.000. As for the TS-MOEA, although $C(\text{TS-MOEA}, \text{MPMOACS})$ exceeds $C(\text{MPMOACS}, \text{TS-MOEA})$ on GZ-R5, $C(\text{MPMOACS}, \text{TS-MOEA})$ outperforms $C(\text{TS-MOEA}, \text{MPMOACS})$ on the other 27 instances. $C(\text{MPMOACS}, \text{MOLS})$ is greater than $C(\text{MOLS}, \text{MPMOACS})$ on 23 instances while the latter is better than the former on only 5 instances. The comparison results of $C(A, B)$ also show that HACO is almost completely dominated

TABLE IV
AVERAGE HV OF THE COMPARED APPROACHES AND THE MPMOACS ON THE BENCHMARK INSTANCES

Instance	NSGA-II	MOEA/D	CCMO	TS-MOEA	MOLS	HACO	MPMOACS
C101	0.257 (+)	0.080 (+)	0.189 (+)	0.331 (+)	0.528 (+)	0.209 (+)	0.585
C201	0.350 (+)	0.091 (+)	0.237 (+)	0.450 (+)	0.347 (+)	0.297 (+)	0.634
R101	N/A (+)	N/A (+)	0.196 (+)	0.299 (+)	0.471 (+)	0.122 (+)	0.603
R201	0.146 (+)	0.061 (+)	0.294 (+)	0.457 (+)	0.311 (+)	0.281 (+)	0.764
RC101	0.014 (+)	N/A (+)	0.187 (+)	0.291 (+)	0.427 (+)	0.091 (+)	0.670
RC201	0.233 (+)	0.057 (+)	0.255 (+)	0.416 (+)	0.339 (+)	0.309 (+)	0.729
num of +/~/−	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	\

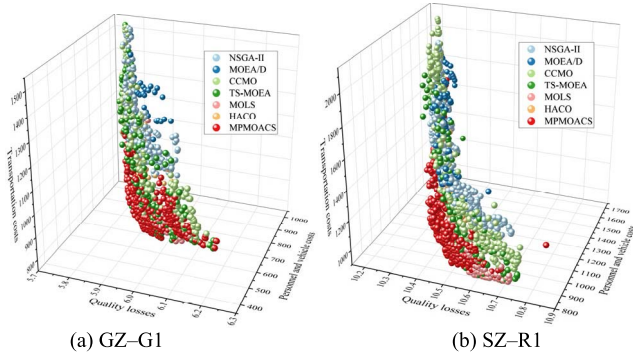


Fig. 5. Pareto solutions of the compared approaches and the MPMOACS on two instances in the multisource dataset.

by MPMOACS since all values of $C(\text{MPMOACS}, \text{HACO})$ are greater than 0.983, while all values of $C(\text{HACO}, \text{MPMOACS})$ is 0.000 except one value is 0.001. Thus, MPMOACS is considered better than MOLS and HACO in the measurement of $C(A, B)$.

Furthermore, the Pareto solutions of the MPMOACS and compared approaches on 2 instances are shown in Fig. 5, including one instance with Gaussian distribution and one instance with random distribution. The figure also shows that the performance of the MPMOACS is better.

In summary, the comparisons on two performance metrics (i.e., HV and $C(A, B)$) and the presentation of Pareto solutions show that MPMOACS has overall better performance on the multisource dataset when compared with the NSGA-II, MOEA/D, CCMO, TS-MOEA, MOLS, and HACO.

2) *Experimental Results on the Benchmark Instances:* Without loss of generality, experiments are conducted on six benchmark instances chosen from Solomon's benchmark [43], and the time-dependent speed was provided by Figliozzi [13]. The time-dependent speed of [13] is used because it is suggested by Figliozzi that the time-dependent speed can be directly applied to Solomon's benchmark, leading to a time-dependent benchmark of the VRPTW. The six instances chosen in our experiment are C101, C201, R101, R201, RC101, and RC201.

The experimental results on HV of the compared approaches and the MPMOACS are shown in Table IV, where N/A means that the approach cannot obtain any feasible solution within the maximum execution time. Besides, the best results are highlighted in **boldface**. The table shows that the performance of the MPMOACS is much better than those of the NSGA-II and MOEA/D. For example, the HV of NSGA-II and MOEA/D on RC201 are 0.233 and 0.057, respectively; while it is 0.729 for

the MPMOACS. Furthermore, the MOEA/D and the NSGA-II cannot obtain feasible solutions on R101 within the execution time provided.

Different from the NSGA-II and MOEA/D, both the CCMO and the TS-MOEA can obtain feasible solutions on all benchmark instances. However, the MPMOACS is superior to the CCMO on all instances, which is similar to the comparison results on the multisource dataset. The statistical test results also show that the MPMOACS has significantly better performance than the CCMO on all benchmark instances. Besides, the MPMOACS also dominates the TS-MOEA on all instances.

The average HV of MOLS and HACO are significantly inferior to that of MPMOACS, meaning that the performance of MPMOACS surpasses those of MOLS and HACO on benchmark instances.

Similar to the multisource dataset, the performance metric $C(A, B)$ is applied on all Pareto solutions of the MPMOACS and other approaches to measuring the dominant relationship between the MPMOACS and compared approaches. The comparison results are shown in Table S.V in the supplementary material, and the best results are in **boldface**. In general, it can be observed that the values of $C(A, B)$ are greater than $C(B, A)$ on all benchmark instances when A is MPMOACS and B is NSGA-II, MOEA/D, CCMO, TS-MOEA, or HACO. That is, MPMOACS performs better than NSGA-II, MOEA/D, CCMO, TS-MOEA, and HACO on all benchmark instances. Besides, MPMOACS is also superior to MOLS on 4 out of all the 6 benchmark instances.

The Pareto solutions of the MPMOACS and compared approaches on three randomly selected benchmark instances are plotted in Fig. S.1 in the supplementary material. The figure shows that the Pareto front of the MPMOACS is rather different from those of the compared approaches since the former shows better convergence and diversity. Thus, the comparison of the Pareto solution distribution of the MPMOACS and those of other approaches further shows that the MPMOACS has better performance on the optimization of benchmark instances.

E. Investigation of the RBLs Strategy

To investigate the effectiveness of the RBLs strategy, the MPMOACS-noLS approach is developed from the MPMOACS by removing the RBLs strategy. Comparison results of the MPMOACS-noLS and the MPMOACS are shown in Table S.VI in the supplementary material, with the best results presented in **boldface**. The MPMOACS is

regarded better than the MPMOACS-noLS since the HV of the MPMOACS is greater than that of the MPMOACS-noLS on all instances. For example, the HV of the MPMOACS on GZ-G3 is 0.679, while it is 0.427 for the MPMOACS-noLS. Thus, with the assistance of the RBLs strategy, the performance of the MPMOACS can be improved.

The box plot for HV of the MPMOACS-noLS and the MPMOACS is given as Fig. S.2 in the supplementary material. Box plot is a kind of graph that can illustrate the convergence and stability of an approach simultaneously. The convergence of an approach is reflected through the height of a box, and the stability is demonstrated by the thickness of a box. An approach with a thicker box is considered less stable. As presented in Fig. S.2, boxes of the MPMOACS are much higher than those of the MPMOACS-noLS, which indicates that the MPMOACS has an advantage in terms of convergence when compared with the MPMOACS-noLS. Moreover, it can be observed from Fig. S.2 that boxes of the MPMOACS-noLS are thicker than those of the MPMOACS on most instances, meaning that the MPMOACS is more stable than the MPMOACS-noLS.

F. Investigation of the Rescheduling Strategy

1) *Disruption Events Simulation*: To simulate the disruption events, two model parameters denoted as θ and sf are introduced.

Parameter θ is the number of edges that are affected by the disruption events. When the disruption events happen, θ untraveled edges from an old distribution scheme are selected randomly, and the driving speed of these edges is changed.

Parameter sf ($0 < sf < 1$) is the speed factor that indicates the reduction degree of driving speed. In detail, the new driving speed is calculated by multiplying sf with the original driving speed. Meanwhile, for a distribution scheme, the time that the disruption events happen is randomly generated, and the change of the driving speed is assumed to keep until the end of the distribution process.

2) *Objective Evaluation Criterion*: As the change of driving speed due to the disruption events may cause the violation of the hard time windows constraints of the CCL model, the rescheduling is expected to minimize the total late delivery duration at first. The total late delivery duration of a new solution named sol is denoted by $ld(sol)$, and is calculated as

$$ld(sol) = \sum_{k \in K} y_k \times \left(\sum_{j \in V} \sum_{i \in N} x_{jik} \times \max\{0, s_{ik} - e_i\} \right) \quad (33)$$

where s_{ik} is the time that vehicle k begins to serve customer i , and e_i is the upper bound of customer i 's time window. Besides, it is also preferred if the rescheduling strategy can obtain solutions with less objective fitness values (i.e., f_1 , f_2 , and f_3). Therefore, a two-level objective evaluation criterion is proposed to evaluate the new solutions obtained by the rescheduling strategy. Specifically, a new solution sol_1 is considered better than another new solution sol_2 if one of the following two conditions is satisfied: i) $ld(sol_1)$ is less than $ld(sol_2)$; ii) $ld(sol_1)$ equals $ld(sol_2)$ and the product of all the

objective fitness values (i.e., f_1 , f_2 , and f_3) of sol_1 is less than that of sol_2 .

3) *Results Discussion*: Without loss of generality, we simulate the disruption events on seven typical selected instances (i.e., GZ-G1, GZ-G2, GZ-R4, SZ-G4, SZ-G5, SZ-R6, and SZ-R7). Moreover, three old solutions obtained by MPMOACS to each of these instances are randomly selected to conduct the rescheduling. The disruption event is simulated on each of the $7 \times 3 = 21$ solutions, with the number of edges that are affected by the disruption event (i.e., θ) set as 5 and the speed factor sf set as 0.2, 0.4, 0.6, and 0.8, resulting in totally $21 \times 4 = 84$ scenarios.

The total late delivery duration and the product of all the objective fitness values for both old solutions and new solutions in all the 84 scenarios are presented in Table S.VII in the supplementary material. Note that the new solutions with higher qualities (smaller total late delivery duration or/and smaller product of all the objective fitness values) are shown in **boldface** for clarity. The results show that the rescheduling strategy can obtain new solutions that greatly reduce the total late delivery duration or at least reduce the objective values in 42 scenarios. Moreover, the results also show that the rescheduling strategy can find new solutions with better performance in more scenarios when sf decreases. This indicates that the rescheduling strategy has more advantages when disruption events become more serious. Thus, the proposed rescheduling strategy is regarded as effective for the rescheduling in the CCL model.

V. CONCLUSION

In this paper, we first establish the multisource data-associated multiobjective CCL model, which is a multiobjective optimization problem with the three objectives of minimizing the quality losses, personnel and vehicle costs, and transportation costs. By adopting real transportation data, our model can alleviate the problem that most solutions of traditional CCL models are inapplicable in reality. Then, the MPMOACS approach developed from the ACS and MPMO framework is proposed to optimize the model. Following the suggestion of MPMO, three ant colonies are employed in the MPMOACS. The first colony aims to optimize the first objective, and the second and third colonies are responsible for the optimization of the second and third objectives, respectively. Meanwhile, the RBLs strategy designed for multiobjective local search also improves the performance of MPMOACS. In addition, a rescheduling strategy is proposed to deal with the rescheduling caused by disruption events.

Comprehensive experiments on a multisource dataset and six benchmark instances are conducted to investigate the necessity of considering the real transportation environment and to evaluate the performance of the MPMOACS. As shown in the experimental results, the feasibility rates of modes without real transportation data are relatively low. Hence, the consideration of the real transportation environment in CCL scheduling is important. By comparing the MPMOACS with six state-of-the-art and very recent well-performing multiobjective approaches (i.e., NSGA-II, MOEA/D, CCMO,

TS-MOEA, MOLS, and HACO), it can be concluded that MPMOACS is an effective approach for multisource data-associated multiobjective CCL scheduling. Moreover, the employment of the RBLs strategy is shown to be helpful to the MPMOACS. The rescheduling strategy is seen as effective in dealing with the rescheduling problem.

REFERENCES

- [1] S. Mercier, S. Villeneuve, M. Mondor, and I. Uysal, "Time-temperature management along the food cold chain: A review of recent developments," *Comprehensive Rev. Food Sci. Food Saf.*, vol. 16, no. 4, pp. 647–667, Jul. 2017.
- [2] Q. Lin, "Cold chain transportation decision in the vaccine supply chain," *Eur. J. Oper. Res.*, vol. 283, pp. 182–195, Apr. 2020.
- [3] L.-J. Wu, L. Shi, Z.-H. Zhan, K.-K. Lai, and J. Zhang, "A buffer-based ant colony system approach for dynamic cold chain logistics scheduling," *IEEE Trans. Emerg. Topics Comput. Intell.*, early access, May 19, 2022, doi: [10.1109/TETCI.2022.3170520](https://doi.org/10.1109/TETCI.2022.3170520).
- [4] S. Kauf, "Smart logistics as a basis for the development of the smart city," *Transp. Res. Proc.*, vol. 39, pp. 143–149, Jan. 2019.
- [5] Y.-Y. Chen, Y.-J. Wang, and J.-K. Jan, "A novel deployment of smart cold chain system using 2G-RFID-Sys," *J. Food Eng.*, vol. 141, pp. 113–121, Nov. 2014.
- [6] Z.-G. Chen, Z.-H. Zhan, S. Kwong, and J. Zhang, "Evolutionary computation for intelligent transportation in smart cities: A survey," *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 83–102, May 2022.
- [7] S. Shashi, R. Cerchione, R. Singh, P. Centobelli, and A. Shabani, "Food cold chain management: From a structured literature review to a conceptual framework and research agenda," *Int. J. Logistics Manag.*, vol. 29, no. 3, pp. 792–821, Aug. 2018.
- [8] Y. Lemma, D. Kitaw, and G. Gatew, "Loss in perishable food supply chain: An optimization approach literature review," *Int. J. Sci. Eng. Res.*, vol. 5, no. 5, pp. 302–311, 2014.
- [9] L.-Y. Zhang, M.-L. Tseng, C.-H. Wang, C. Xiao, and T. Fei, "Low-carbon cold chain logistics using ribonucleic acid-ant colony optimization algorithm," *J. Cleaner Prod.*, vol. 233, pp. 169–180, Oct. 2019.
- [10] G. Qin, F. Tao, and L. Li, "A vehicle routing optimization problem for cold chain logistics considering customer satisfaction and carbon emissions," *Int. J. Environ. Res. Public Health*, vol. 16, no. 4, p. 576, 2019.
- [11] M. Gmira, M. Gendreau, A. Lodi, and J.-Y. Potvin, "Tabu search for the time-dependent vehicle routing problem with time Windows on a road network," *Eur. J. Oper. Res.*, vol. 288, no. 1, pp. 129–140, Jan. 2021.
- [12] Y. Deng, W. Zhu, H. Li, and Y. Zheng, "Multi-type ant system algorithm for the time dependent vehicle routing problem with time Windows," *J. Syst. Eng. Electron.*, vol. 29, no. 3, pp. 625–638, 2018.
- [13] M. A. Figliozzi, "The time dependent vehicle routing problem with time Windows: Benchmark problems, an efficient solution algorithm, and solution characteristics," *Transp. Res. E, Logistics Transp. Rev.*, vol. 48, no. 3, pp. 616–636, May 2012.
- [14] A. V. Donati, R. Montemanni, N. Casagrande, A. E. Rizzoli, and L. M. Gambardella, "Time dependent vehicle routing problem with a multi ant colony system," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1174–1191, 2008.
- [15] S. R. Balseiro, I. Loiseau, and J. Ramonet, "An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time Windows," *Comput. Oper. Res.*, vol. 38, no. 6, pp. 954–966, 2011.
- [16] C. Malandraki and M. S. Daskin, "Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms," *Transp. Sci.*, vol. 26, no. 3, pp. 185–200, 1992.
- [17] S. Ichoua, M. Gendreau, and J.-Y. Potvin, "Vehicle dispatching with time-dependent travel times," *Eur. J. Oper. Res.*, vol. 144, no. 2, pp. 379–396, Jan. 2003.
- [18] Z. Zhang, H. Qin, and Y. Li, "Multi-objective optimization for the vehicle routing problem with outsourcing and profit balancing," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1987–2001, May 2020.
- [19] S. F. Ghannadpour and F. Zandiyeh, "A new game-theoretical multi-objective evolutionary approach for cash-in-transit vehicle routing problem with time Windows (A real life case)," *Appl. Soft Comput.*, vol. 93, pp. 1–26, Aug. 2020.
- [20] H. Zhang, Q. Zhang, L. Ma, Z. Zhang, and Y. Liu, "A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time Windows," *Inf. Sci.*, vol. 490, pp. 166–190, Jul. 2019.
- [21] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 102–116, Feb. 2021.
- [22] J. Wang, T. Weng, and Q. Zhang, "A two-stage multiobjective evolutionary algorithm for multiobjective multidrop vehicle routing problem with time Windows," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2467–2478, Jul. 2019.
- [23] K. Deb, S. Chaudhuri, and K. Miettinen, "Towards estimating nadir objective vector using evolutionary approaches," in *Proc. 8th Annu. Conf. Genetic Evol. Comput. (GECCO)*, 2006, pp. 643–650.
- [24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Aug. 2002.
- [25] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [26] C. B. Kalayci and C. Kaya, "An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery," *Exp. Syst. Appl.*, vol. 66, pp. 163–175, Dec. 2016.
- [27] X. Xiang, Y. Tian, X. Zhang, J. Xiao, and Y. Jin, "A pairwise proximity learning-based ant colony algorithm for dynamic vehicle routing problems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5275–5286, Jun. 2022.
- [28] J. C. Molina, J. L. Salmeron, and I. Eguia, "An ACS-based memetic algorithm for the heterogeneous vehicle routing problem with time Windows," *Exp. Syst. Appl.*, vol. 157, Nov. 2020, Art. no. 113379.
- [29] B. Barán and M. Schaefer, "A multiobjective ant colony system for vehicle routing problem with time Windows," in *Proc. IASTED Int. Multi-Conf. Appl. Info.*, 2003, pp. 97–102.
- [30] X.-F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. L. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 113–128, Feb. 2016.
- [31] S.-Z. Zhou, Z.-H. Zhan, Z.-G. Chen, S. Kwong, and J. Zhang, "A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6784–6798, Nov. 2021.
- [32] D. Liang, Z.-H. Zhan, Y. Zhang, and J. Zhang, "An efficient ant colony system approach for new energy vehicle dispatch problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4784–4797, Nov. 2020.
- [33] Z.-G. Chen *et al.*, "Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2912–2926, Aug. 2018.
- [34] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.
- [35] X. Zhang, Z.-H. Zhan, W. Fang, P. Qian, and J. Zhang, "Multipopulation ant colony system with knowledge-based local searches for multiobjective supply chain configuration," *IEEE Trans. Evol. Comput.*, vol. 26, no. 3, pp. 512–526, Jun. 2022.
- [36] Z.-H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artif. Intell. Rev.*, vol. 55, no. 1, pp. 59–110, Jan. 2022.
- [37] J.-Y. Li *et al.*, "A multipopulation multiobjective ant colony system considering travel and prevention costs for vehicle routing in COVID-19-like epidemics," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 17, 2022, doi: [10.1109/TITS.2022.3180760](https://doi.org/10.1109/TITS.2022.3180760).
- [38] Z.-H. Zhan, J. Li, J. Cao, J. Zhang, H. S.-H. Chung, and Y.-H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.
- [39] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 587–602, Aug. 2019.
- [40] S.-C. Liu, Z.-G. Chen, Z.-H. Zhan, S.-W. Jeon, S. Kwong, and J. Zhang, "Many-objective job-shop scheduling: A multiple populations for multiple objectives-based genetic algorithm approach," *IEEE Trans. Cybern.*, early access, Sep. 13, 2022, doi: [10.1109/TCYB.2021.3102642](https://doi.org/10.1109/TCYB.2021.3102642).
- [41] L. Shi, Z.-H. Zhan, D. Liang, and J. Zhang, "Memory-based ant colony system approach for multi-source data associated dynamic electric vehicle dispatch optimization," *IEEE Trans. Intell. Transp. Syst.*, early access, Mar. 8, 2022, doi: [10.1109/TITS.2022.3150471](https://doi.org/10.1109/TITS.2022.3150471).

- [42] M. Gong and S. Liu, "Study on cold chain logistics with time Windows based on carbon emissions considering simultaneous pick up and distribution," in *Proc. 8th Int. Conf. Ind. Technol. Manag. (ICITM)*, Mar. 2019, pp. 301–305.
- [43] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time Window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, 1987.
- [44] B. Ombuki, B. J. Ross, and F. Hanshar, "Multi-objective genetic algorithms for vehicle routing problem with time Windows," *Appl. Intell.*, vol. 24, no. 1, pp. 17–30, 2006.
- [45] *MPMOACS_MultisourceData.zip*. Accessed: Mar. 1, 2022. [Online]. Available: <https://zhanapollo.github.io/zhanzh/resources/>
- [46] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.



Li-Jiao Wu (Student Member, IEEE) received the B.S. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2020, where she is currently pursuing the M.S. degree.

Her current research interests include evolutionary computation, ant colony systems, and their applications to real-world optimization problems.



Zong-Gan Chen (Member, IEEE) received the B.S. degree from the Sun Yat-sen University, Guangzhou, China, in 2016, and the Ph.D. degree in computer science and technology from the South China University of Technology, Guangzhou, China in 2021.

He is currently an Associate Professor with the School of Computer Science, South China Normal University, Guangzhou. His current research interests include evolutionary computation, swarm intelligence, and their applications in real-world optimization problems.



Chun-Hua Chen (Member, IEEE) received the bachelor's degree in computer science and the Ph.D. degree in information and communication engineering from the South China University of Technology, Guangzhou, China, in 2006 and 2012, respectively.

He is currently an Assistant Professor with the School of Software Engineering, South China University of Technology. His current research interests include evolutionary computation, knowledge graph, distributed systems, component-based software engineering, and their applications in industrial sector.



Yun Li (Fellow, IEEE) received the B.S. degree from Sichuan University, Chengdu, China, the M.E. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, and the Ph.D. degree from the University of Strathclyde, Glasgow, U.K.

In 1989, he was an Intelligent Systems Engineer with the U.K. National Engineering Laboratory, Glasgow. In 1990, he was a Post-Doctoral Research Engineer at Industrial Systems and Control Ltd., Glasgow. From 1991 to 2018, he was an Intelligent

Systems Lecturer, a Senior Lecturer, and a Professor at the University of Glasgow, Glasgow, and served as the Founding Director of the University of Glasgow, Singapore. He later served as the Founding Director of the Dongguan Industry 4.0 Artificial Intelligence Laboratory, Dongguan, China, and at i4AI Ltd., London, U.K. He is currently a Chartered Engineer with the U.K. He has published 280 papers, one of which since publication in the *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY* has been its most popular article every month. Since 1991, his research has been focused on computational artificial intelligence and its applications.

Prof. Li is an Associate Editor of *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS* and *IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE*.



Sang-Woon Jeon (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Yonsei University, Seoul, South Korea, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2011

From 2011 to 2013, he was a Post-Doctoral Associate at the School of Computer and Communication Sciences, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland.

From 2013 to 2017, he was an Assistant Professor at the Department of Information and Communication Engineering, Andong National University, Andong, South Korea. He has been an Associate Professor with the Department of Military Information Engineering (undergraduate school) and the Department of Electronics and Communication Engineering (graduate school), Hanyang University, Ansan, South Korea, since 2017. His research interests include network information theory, wireless communications, and machine learning. He was a recipient of the Haedong Young Scholar Award in 2017, which was sponsored by the Haedong Foundation and given by the Korea Institute of Communications and Information Science (KICS), the Best Paper Award of the IEEE International Conference on Communications in 2015, and the Best Thesis Award from the Department of Electrical Engineering, KAIST, in 2012.



Jun Zhang (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, Kowloon, Hong Kong, in 2002.

He is currently a Korea Brain Pool Fellow Professor with Hanyang University, South Korea. His current research interests include computational intelligence, cloud computing, high performance computing, operations research, and power electronic circuits.

Dr. Zhang was a recipient of the Changjiang Chair Professor from the Ministry of Education, China, in 2013, the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011, and the First-Grade Award in Natural Science Research from the Ministry of Education, in 2009. He is currently an Associate Editor of the *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* and the *IEEE TRANSACTIONS ON CYBERNETICS*.



Zhi-Hui Zhan (Senior Member, IEEE) received the bachelor's degree and the Ph.D. degree in computer science from the Sun Yat-sen University, Guangzhou, China, in 2007 and 2013, respectively.

He is currently the Changjiang Scholar Young Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. His current research interests include evolutionary computation, swarm intelligence, and their applications in real-world problems and in environments of smart cities, cloud computing, and big data.

Dr. Zhan was a recipient of the IEEE Computational Intelligence Society (CIS) Outstanding Early Career Award in 2021, the Outstanding Youth Science Foundation from National Natural Science Foundations of China (NSFC) in 2018, and the Wu Wen-Jun Artificial Intelligence Excellent Youth from the Chinese Association for Artificial Intelligence in 2017. His Doctoral Dissertation was awarded the IEEE CIS Outstanding Ph.D. Dissertation and the China Computer Federation Outstanding Ph.D. Dissertation. He is one of the World's Top 2% Scientists for both Career-Long Impact and Year Impact in Artificial Intelligence and one of the Highly Cited Chinese Researchers in computer science. He is currently the Chair of Membership Development Committee in IEEE Guangzhou Section and the Vice-Chair of IEEE CIS Guangzhou Chapter. He is currently an Associate Editor of the *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, the *Neurocomputing*, the *Memetic Computing*, and the *Machine Intelligence Research*.