



Research article

A genetic algorithm with two-step rank-based encoding for closed-loop supply chain network design

Bowen Ding¹, Zhaobin Ma¹, Shuoyan Ren¹, Yi Gu¹, Pengjiang Qian¹ and Xin Zhang^{1,2,*}

¹ School of Artificial Intelligence and Computer Science, and Jiangsu Key Laboratory of Media Design and Software Technology, Jiangnan University, Wuxi 214122, China

² Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

* **Correspondence:** Email: zhangxin@jiangnan.edu.cn.

Abstract: The closed-loop supply chain (CLSC) plays an important role in sustainable development and can help to increase the economic benefits of enterprises. The optimization for the CLSC network is a complicated problem, since it often has a large problem scale and involves multiple constraints. This paper proposes a general CLSC model to maximize the profits of enterprises by determining the transportation route and delivery volume. Due to the complexity of the multi-constrained and large-scale model, a genetic algorithm with two-step rank-based encoding (GA-TRE) is developed to solve the problem. Firstly, a two-step rank-based encoding is designed to handle the constraints and increase the algorithm efficiency, and the encoding scheme is also used to improve the genetic operators, including crossover and mutation. The first step of encoding is to plan the routes and predict their feasibility according to relevant constraints, and the second step is to set the delivery volume based on the feasible routes using a rank-based method to achieve greedy solutions. Besides, a new mutation operator and an adaptive population disturbance mechanism are designed to increase the diversity of the population. To validate the efficiency of the proposed algorithm, six heuristic algorithms are compared with GA-TRE by using different instances with three problem scales. The results show that GA-TRE can obtain better solutions than the competitors, especially on large-scale instances.

Keywords: closed-loop supply chain network design; genetic algorithm; constrained optimization; large-scale optimization

1. Introduction

Global competition and the demand for integration among enterprises have been pushing the development of the supply chain network (SCN). SCN has been applied in many fields in reality, which is generally composed of different members and materials flowing between the members, such as electronics [1] and the food industry [2]. The common members of SCN include suppliers, warehouses, distributors and so on [3,4]. The main purpose of supply chain network design is to maximize the profit of the whole network and to meet the demands of different facilities at the same time.

Recently, the researchers have paid more attention to the closed-loop supply chain (CLSC) under the background of the deepening environmental awareness and the development of remanufacturing industry [5]. A CLSC network consists of the forward and reverse networks that consider both production and recycling [6]. The forward supply chain concentrates on delivering products for customers effectively. In contrast, the reverse supply chain aims to decrease the cost of customers and lower pollution by helping the collection and distribution of the recycling products to the appropriate facilities. Therefore, the CLSC model has great practical application foreground. Mosallanezhad et al. [7] proposed a shrimp CLSC model to meet the growing market demand for fresh food and to minimize the total cost, which also considers the balance for demands of poultry and livestock food market at supplying the shrimp products. Salehi-Amiri et al. [8] designed a sustainable CLSC network for the walnut industry, which considers meeting the demand of different markets and provides the channel of returned products for the second use. Cheraghalipour et al. [9] proposed a bi-objective CLSC model for citrus fruits, which aims to minimize the total cost of the whole CLSC network and satisfy the customers' demands as much as possible. The uncertainty generally exists in CLSC due to various factors, such as the economy or environment [10,11]. Chouhan et al. [12] developed a CLSC network with uncertain demands. The line balancing strategy is addressed to handle uncertainty. Fathollahi-Fard et al. [13] proposed a dual-channel CLSC network under uncertainty for the tire industry, and a fuzzy approach called Jimenez's method is applied to tackle the uncertain parameters.

The network design of CLSC is proved to be an NP-hard problem [14], and therefore it is hard for most traditional approaches to get the satisfied solutions for this problem. Moreover, commercial software such as CPLEX and GAMS usually require a lot of time or even are unsolvable when they solve the large-scale and complex mathematical CLSC models. As intelligent algorithms have a good global searching ability, many intelligent algorithms are used for solving this kind of problem and specific methods can be designed according to the specific problem [15–18].

Genetic algorithm (GA) as a popular and efficient intelligent algorithm has been widely applied in CLSC problems. HJ et al. [19] proposed a GA-based heuristic algorithm that was based on the dynamic integrated distribution network and simplex transshipment algorithm, and this method was used to optimize the integrated network for third-party logistics providers. Hamed et al. [20] proposed a new hybrid algorithm that used the operators of particle swarm optimization (PSO) to improve the performance of GA for solving the large-scale CLSC problems. Abir et al. [21] constructed a multi-objective CLSC model to minimize both carbon emission and the total cost, and the multi-objective GA and the weight sum method were used to solve the proposed problem. Besides, other heuristic algorithms are also applied to the CLSC problems, such as PSO [22–24], ant colony optimization (ACO) [25], red deer algorithm [26], and hybrid heuristic algorithms [27,28].

For the optimization of CLSC network design, the solution encoding is an important part of a competitive algorithm. However, few researches focus on it. Gen et al. [29] proposed an improved GA

to solve the supply chain problem, and developed an improved priority-based encoding method for chromosomes. A freight matrix is used to calculate the transportation route's priority, and the gene information is composed of corresponding priority. The advantage is that the chromosomes can contain more information with fewer genes. However, if there are many constraints to satisfy, encoding and decoding operations will be time-consuming due to the constraint verification. In addition, the CLSC problem often has multiple complex constraints, and the complexity of the problem would increase significantly with large scale networks [30]. Therefore, how to deal with multiple constraints and obtain high-quality solutions is also a key problem for an intelligent algorithm to solve the CLSC problem.

This paper focuses on the proposed model of the CLSC and develops a genetic algorithm with two-step rank-based encoding (GA-TRE) to solve the multi-constrained and large-scale CLSC problem. According to the characteristics of the CLSC problem and GA operators, a new encoding method is proposed, called two-step rank-based encoding. Since, the main problems of the proposed model are the decision of route planning and delivery volume, this encoding scheme decomposes the encoding process into two steps, and different encoding steps are used to solve different subproblems. The first step encoding is used to determine the alternative transportation routes, and the second step encoding is to determine the specific route and the delivery volume among the CLSC network. Moreover, a rank-based method is used to achieve greedy solutions in the second step encoding. The two-step encoding mechanism can help to relieve the multi-constrained challenges, since the feasibility of the transportation routes will be predicted according to the relevant constraints after the first step of encoding. In this way, the constraints can be verified separately according to different subproblems, which can reduce the difficulty of solving the problem and find illegal solutions as soon as possible.

Besides, for the canonical GA, it is difficult to obtain legal solutions after genetic operators and time-consuming to handle large-scale constrained problems. Therefore, in this paper, the genetic operators are only used for the chromosome of the route for reinforcing the exploring ability of GA to tackle this kind of problem. Since the chromosome of the transportation route can be represented by binary encoding, it involves fewer constraints than the complete problem. This method will help to reduce the probability of generating illegal solutions. To reduce the chance of the population being trapped into the local optimum and explore the solution space, a new mutation operator which is called stage-mutation, and an adaptive population disturbance mechanism are designed. Stage-mutation can increase the diversity of chromosomes based on the characteristics of the CLSC problem, and the adaptive population disturbance mechanism can effectively explore more search space by adding new individuals in different periods of the algorithm. Comprehensive validation of the proposed algorithm is conducted by comparing it with other algorithms in the different scales of instances. The experimental results show that GA-TRE has a better performance on the CLSC problem.

In conclusion, the main contributions of the proposed algorithm are as follows:

- 1) A two-step rank-based encoding scheme for the CLSC problems is proposed to handle the constraints and increase the algorithm efficiency, which is also used for improving the crossover and mutation operators;
- 2) The constraints handling and greedy solutions achieving are considered on different encoding steps;
- 3) A new mutation operator based on CLSC problems and an adaptive population disturbance mechanism are designed to improve the population diversity of the GA-TRE algorithm.

The rest of this paper is organized as follows. Section 2 outlines the problem description and the detailed model formulation. Section 3 describes the proposed encoding method and algorithm,

respectively. Section 4 shows the exhaustive experiment which is conducted to validate the performance of the GA-TRE, and the conclusion follows in Section 5.

2. CLSC problem definition

2.1. Problem description

Nowadays, environmental protection has attracted increasing attention. A variety of closed-loop supply chains have been designed for recycling, such as the recycling of plastic products, tires [31], and leather products [32]. However, the design of the CLSC network for different industries and different scenarios will bring burden on manpower and material resources, especially for small and medium-sized enterprises. Hence, this paper aims to propose a CLSC model to match more application scenarios. The proposed CLSC network contains seven kinds of members: suppliers (S), manufacturers (M), retailers (R), customer regions (CR), collection points (CP), recycling centers (RC), and waste disposal plants (WDP). The proposed CLSC network structure is shown in Figure 1.

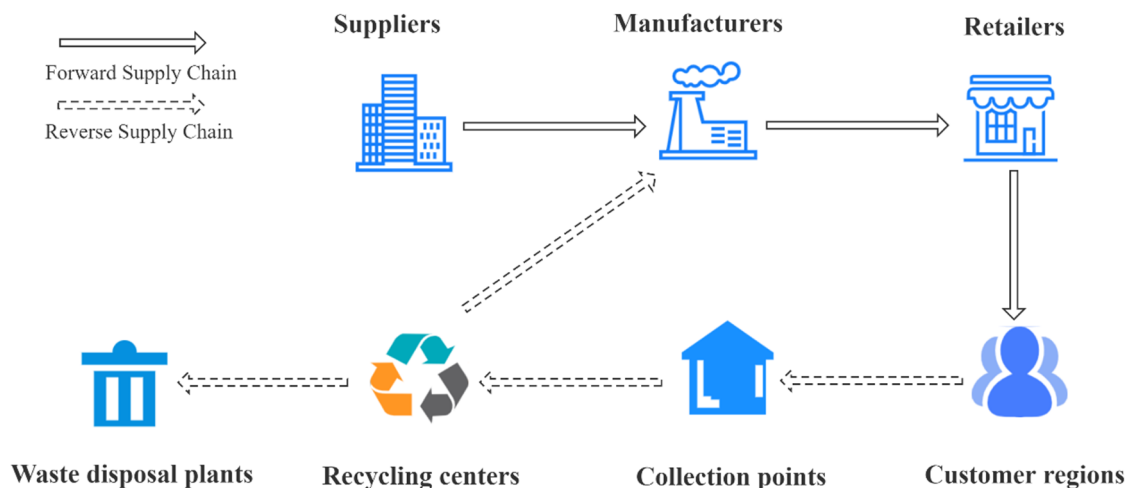


Figure 1. Schematic of the CLSC network design.

In this model, certain facilities have their capacity/demand ($c_{\square}^{\square}/d_{\square}$), opening fixed cost (f_{\square}^{\square}), and variable cost (x_{\square}). The transportation cost (m_{jk} , cr_{vl} , etc.) exists when a certain quantity of raw materials or products are transported among different facilities (q_{\square}^{\square}). Moreover, to be closer to reality, the customers in the model do not refer to a single buyer but represent a not fixed customer region, and the total customer demands in this model must be met. In the process of customers buying products from retailers, the specific number of customers, positions, and quantity of commodities are difficult to determine. Therefore, the concept of customer region is used in this model, and the route from retailers to customer regions is not planned. The demand of retailers comes from the demand of the customer, and therefore the sum of all the retailer demand must be met by the manufacturers.

Because the products exceed the service cycle or are damaged, customers will discard them. At present, the used product collection points have been set up in most cities in China. Each point will receive the used products and transport them to the recycling center after centralized sorting (S). For

customers, they will get a certain amount of subsidy (p_2). Besides, because the government supports the environmental protection industry, the price of raw materials from the recycling center is cheaper than from manufacturers, which contributes to the smooth flow of the reverse supply chain to a certain extent. The minimal recycling rate of customer regions (μ_v) is preset in this model. The used products are transported from the collection points to the recycling centers for recycling and processing. During the whole process, most parts of products can be recycled as raw materials and be transported to the manufacturers. Some parts which cannot be recycled will be disposed of by the waste disposal plant for burning or landfills. The maximal disposal rate (φ) is also preset in this model.

Some assumptions are applied in this model.

- 1) Recycled materials from the recycling centers have the same quality as the materials from suppliers.
- 2) There are no products or materials to flow among the members from the same level.
- 3) The maximum manufacturing capacity and storage capacity of each member are fixed.
- 4) In the first time, the supply chain flows after the customer purchases the product, and the recycling center receives the used product. In this way, the manufacturer can receive raw materials from both suppliers and recycling centers.

2.2. Model formulation

The notations of indices, parameters, and decision variables about this model are shown as follows:

Indices

i	index of suppliers, $i \in \{1, 2, \dots, I\}$
j	index of manufacturers, $j \in \{1, 2, \dots, J\}$
k	index of retailers, $k \in \{1, 2, \dots, K\}$
v	index of customer regions, $v \in \{1, 2, \dots, V\}$
l	index of collection points, $l \in \{1, 2, \dots, L\}$
m	index of recycling centers, $m \in \{1, 2, \dots, M\}$
w	index of waste disposal plants, $w \in \{1, 2, \dots, W\}$

Parameters

C_i^s	capacity of supplier i
C_j^m	capacity of manufacturer j
C_l^{cp}	capacity of collection point l
C_m^{rc}	capacity of recycling center m
d_k	demand of retailer k
d_v	demand of customers in customer region v
S_{ij}	unit cost of transportation from supplier i to manufacturer j
m_{jk}	unit cost of transportation from manufacturer j to retailer k
cr_{vl}	unit cost of transportation from customer region v to collection point l
cp_{lm}	unit cost of transportation from collection point l to recycling center m
rc_{mj}	unit cost of transportation from recycling center m to manufacturer j
rc_{mw}	unit cost of transportation from recycling center m to waste disposal plant w
f_j^m	fixed cost of manufacturer j
f_l^{cp}	fixed cost of collection point l
f_m^{rc}	fixed cost of recycling center m
s	unit cost of sorting out the used products

x_0	unit cost of producing new products
x_1	unit cost of purchasing raw materials from suppliers
x_2	unit cost of purchasing recycled materials from recycling center
x_3	unit cost of disposing the material which cannot be recycled
x_4	unit cost of decomposing the used products in recycling centers
p_1	unit price of new products
p_2	unit cost that collection point pay to customers for the used products
μ_v	the minimal recycling rate of customer region v
φ	the maximal disposal rates

Decision variables

q_{ij}^s	number of raw materials shipped from supplier i to manufacturer j
q_{jk}^m	number of products from manufacturer j to retailer k
q_{vl}^{cr}	number of used products from customers region v to collection point l
q_{lm}^{cp}	number of used products from collection point l to recycling center m
q_{mj}^{rc}	number of recycled materials from recycling center m to manufacturer j
q_{mw}^{rc}	number of waste materials from recycling center m to waste disposal plant w
q_j^{new}	number of new produced products in manufacturer j

$$\alpha_j \begin{cases} 1 & \text{if products are produced at manufacturer } j \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_l \begin{cases} 1 & \text{if collection point } l \text{ is open} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_m \begin{cases} 1 & \text{if recycling center } m \text{ is open} \\ 0 & \text{otherwise} \end{cases}$$

Objective function

The objective function of the proposed closed-loop supply chain problem is to maximize the total profit which is equal to the total income minus total expenditure as shown in Eq (1).

$$\text{Maximize } TP = TI - TE \quad (1)$$

The total income is the sum of the cost of all new products, which is displayed in Eq (2).

$$TI = p_1 \sum_k d_k \quad (2)$$

The total expenditure consists of the total transportation cost, the total fixed cost, and total processing cost as represented by Eq (3).

$$TE = TC + FC + PC \quad (3)$$

Equation (4) shows the total transportation cost.

$$TC = \sum_i \sum_j s_{ij} q_{ij}^s + \sum_j \sum_k m_{jk} q_{jk}^m + \sum_v \sum_l cr_{vl} q_{vl}^{cr} + \sum_v \sum_m cp_{lm} q_{lm}^{cp} + \sum_m \sum_j rc_{mj} q_{mj}^{rc} + \sum_m \sum_w rc_{mw} q_{mw}^r \quad (4)$$

Equation (5) shows the total fixed costs of the manufacturers, collection points, and recycling centers.

$$FC = \sum_j f_j^m \alpha_j + \sum_l f_l^{cp} \beta_l + \sum_m f_m^{rc} \delta_m \quad (5)$$

Maintaining the operation of the closed-loop supply chain also needs some necessary costs. Eq (6) shows the total processing cost which is equal to the sum of seven parts, including the cost of obtaining used products to pay customers in collection points, the cost of sorting out and classifying used products in collection points, the cost of recycling used products in the recycling centers, the cost of dealing with non-recyclable waste, the cost of producing new products, and the cost of purchasing materials in manufacturers.

$$PC = p_2 \sum_v d_v \mu_v + s \sum_v d_v \mu_v + x_4 \sum_l \sum_m q_{lm}^{cp} + x_3 \sum_m \sum_w q_{mw}^{rc} + x_0 \sum_j q_j^{new} + x_1 \sum_i \sum_j q_{ij}^s + x_2 \sum_m \sum_j q_{mj}^{rc} \quad (6)$$

Subject to

$$\sum_j q_{ij}^s \leq c_i^s \quad \forall i \in I \quad (7)$$

$$\sum_i q_{ij}^s \leq c_j^m \alpha_j \quad \forall j \in J \quad (8)$$

Constraints (7) and (8) make sure the quantity of raw materials between manufacturers and suppliers should not exceed the capacity of each supplier and manufacturer.

$$\sum_k q_{jk}^m \leq c_j^m \alpha_j \quad \forall j \in J \quad (9)$$

$$\sum_j q_{jk}^m = d_k \quad \forall k \in K \quad (10)$$

$$\sum_k d_k = \sum_v d_v \quad (11)$$

Constraint (9) shows the quantity of products supplied by each manufacturer to retailers should not exceed the capacity of the manufacturer. Constraints (10) and (11) restrict that the demand of retailers and customers must be satisfied.

$$\sum_l q_{vl}^{cr} \geq \mu_v d_v \quad \forall v \in V \quad (12)$$

$$\sum_v q_{vl}^{cr} \leq c_l^{cp} \beta_l \quad \forall l \in L \quad (13)$$

Constraint (12) guarantees the number of used products provided by each customer region must be more than minimum collection rates. Constraint (13) shows the capacity limitation of collection points.

$$\sum_m q_{lm}^{cp} \leq c_l^{cp} \beta_l \quad \forall l \in L \quad (14)$$

$$\sum_l q_{lm}^{cp} \leq c_m^{rc} \delta_m \quad \forall m \in M \quad (15)$$

Constraints (14) and (15) formulate the quantity of used products between collection points and recycling centers should not violate the capacity limitation of collection points and recycling centers, respectively.

$$\sum_j q_{mj}^{rc} \leq c_m^{rc} \delta_m \quad \forall m \in M \quad (16)$$

$$\sum_m q_{mj}^{rc} + \sum_i q_{ij}^s = \sum_k q_{jk}^m \quad \forall j \in J \quad (17)$$

Constraint (16) shows the capacity limitation between manufacturers and recycling centers. Constraint (17) restricts suppliers and recycling centers providing raw materials for each manufacturer should be equal to the quantity of transporting to retailers.

$$\sum_v q_{vl}^{cr} \geq \sum_m q_{lm}^{cp} \quad \forall l \in L \quad (18)$$

$$\sum_j q_{mj}^{rc} \leq \sum_l q_{lm}^{cp} \quad \forall m \in M \quad (19)$$

Constraints (18) and (19) restrict the inflow and outflow quantity of collection points, recycling centers, and manufacturers.

$$\sum_j q_j^{new} = \sum_j \sum_k q_{jk}^m \quad (20)$$

Constraint (20) restricts the total quantity of new products should be equal to the total quantity of products transported from manufacturers to retailers.

$$\sum_w q_{mw}^{rc} + \sum_j q_{mj}^{rc} \leq \sum_l q_{lm}^{cp} \quad \forall m \in M \quad (21)$$

$$\sum_w q_{mw}^{rc} \leq \phi \sum_l q_{lm}^{cp} \quad \forall m \in M \quad (22)$$

Constraint (21) restricts the quantity of recycled materials and the waste materials of each recycling center should be less than the quantity of used products from collection points to the recycling centers. Constraint (22) shows the quantity of waste materials from recycling centers to waste disposal plants should be less than the maximum value preset.

$$\alpha_j, \beta_l, \delta_m \in \{0, 1\} \quad \forall j \in J, \forall l \in L, \forall m \in M \quad (23)$$

$$q_{ij}^s, q_{jk}^m, q_{vl}^{cr}, q_{lm}^{cp}, q_{mj}^{rc}, q_{mw}^{rc}, q_j^{new} \in N \quad \forall i \in I, \forall j \in J, \forall k \in K, \forall v \in V, \forall l \in L, \forall m \in M, \forall w \in W \quad (24)$$

Constraints (23) and (24) show the value ranges of decision variables.

3. The proposed GA

Compared with the traditional optimization method based on mathematical analysis, heuristic algorithms have the advantages of optimizing black-box problems, have a good global search capability, and are simple to implement and use. Therefore, heuristic algorithms are widely applied for various challenging optimization problems, such as scheduling [33,34], vehicle routing [35], online learning [36], multi-objective optimization [37,38], and feature selection [39,40]. In this paper, an improved GA was designed to optimize the proposed CLSC problems.

3.1. Two-step rank-based encoding

Due to the complexity of the CLSC optimization problem proposed in this paper, it is important to propose a new encoding method in GA. Firstly, according to the definition of the problem model, the chromosome should contain the information of the transportation routes, the quantity of materials or products among the facilities, and the open state of the facilities. Secondly, the encoding should be helpful to generate feasible solutions. Thirdly, the design of chromosomes needs to be convenient for genetic operators. This paper proposes a two-step rank-based encoding method to solve the above problems. Figure 2 shows the chromosome in the encoding of each step with the numerical example of $I = 3, J = 4, K = 5, V = 3, L = 4, M = 2, W = 1$.

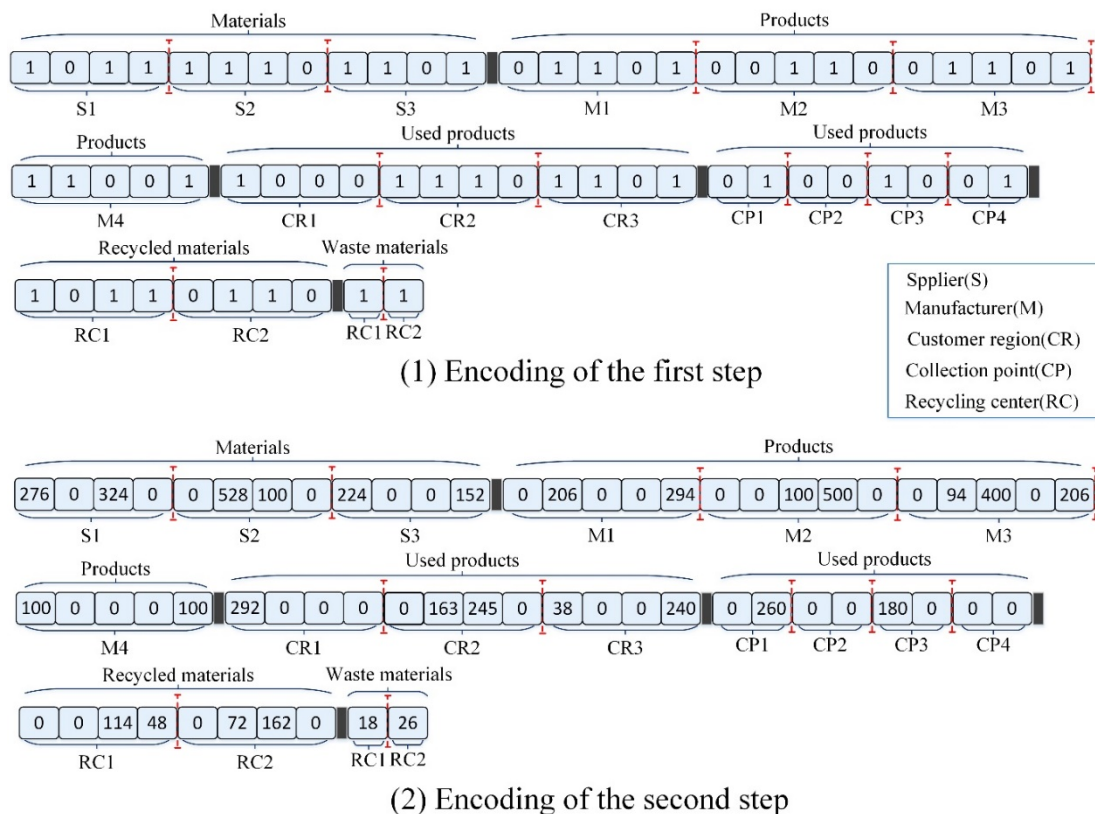


Figure 2. The chromosome after each step encoding.

The first step of encoding is only to determine the transportation route. The chromosome can use

binary encoding to express the information of the route and each gene is generated randomly. For the binary number, 1 means this route can be used, and 0 means not. As shown in Figure 2(1), the chromosome has six sections corresponding to different stages of CLSC. In the first section, 12 (3 × 4) genes represent the route information of suppliers supplying raw materials to manufacturers. For example, the first four genes (1, 0, 1, 1) represent that supplier 1 can supply raw materials to manufacturers 1, 3 and 4, but not supply materials to manufacturer 2. In the second section, 20 (4 × 5) genes represent the route information of the manufacturer supplying new products to the retailers. For example, the first five genes (0, 1, 1, 0, 1) represent that manufacturer 1 can supply products to retailers 2, 3 and 5, but not provide products to retailers 1 and 4 and so on. The length of a chromosome can be calculated as $I \times J + J \times k + V \times L + L \times M + M \times J + M \times W$. It should be noted that the route of retailers to customer regions is not considered as mentioned before in the problem description.

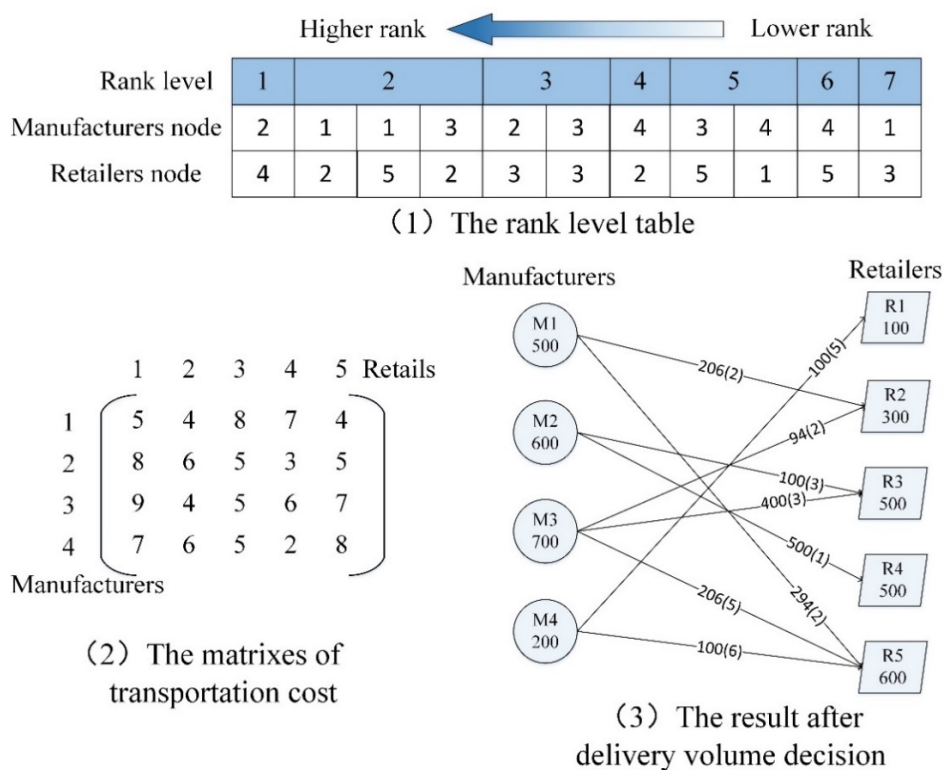


Figure 3. An instance for delivery volume decision.

After the first step, a chromosome representing the transportation route has been established. The second step encoding is to decide the quantity of transportation materials or products according to the route, which is called delivery volume decision. As shown in Figure 2(2), the chromosome of the second encoding remains the same structure as the first step encoding, but the content has been changed. For example, the first four genes (276, 0, 324, 0) in the first section represent that supplier 1 provides 276 and 324 units of raw materials to manufacturers 1 and 3, respectively but provides nothing to manufacturers 2 and 4. In order to explain the encoding process of the second step clearly, Figure 3 shows the details of delivery volume decision in the stage of manufacturers to retailers, which uses the numerical example of Figure 2. Figure 3(1) shows the rank level table, and the lowest value means the highest rank. The rank level of the routes is built according to the transportation cost, and

the lower cost corresponds to the higher rank. The transportation route will be allocated from the higher rank to the lower rank. Figure 3(2) shows the matrixes of transportation cost. Figure 3(3) shows the result after the delivery volume decision, and the numbers in parentheses are the rank level of the route. The numbers in the circles and parallelograms are the capacity of the manufacturer and demand of the retailer, respectively. The detailed methods of second step encoding are as follows:

- Step 1:* Build a rank table by using the matrixes of transportation costs.
- Step 2:* Set the routes of the top rank level as the current allocation routes. For example, in Figure 3(3), the route from manufacturer 2 (M2) to retailer 4 (R4) is the first to allocate.
- Step 3:* Check whether the current allocation routes only have one route. If yes, go to Step 4a. Otherwise, go to Step 4b.
- Step 4a:* Compare the provider's capacity and receiver's demand of the route, and the delivery volume can be set as the smaller one. Update the capacity and demand value of the corresponding facilities. For example, in Figure 3(3), the capacity of manufacturer 2 (M2) is 600 units and the demand of retailer 4 (R4) is 500 units, and therefore the route's delivery volume will be 500 units. The remaining capacity of manufacturer 2 is updated as 100, and the demand of retailer 4 is updated as 0.
- Step 4b:* Multiple routes with the same rank level will be classified into three situations.
- Situation 1:** If the route's provider and receiver facility are all different from other routes in this level, allocate the delivery volume as step 4a until all routes in this situation have been allocated.
- Situation 2:** If the route's provider facility is the same as other routes in this level, then let N denote the number of these routes. Randomly generate a number x_i ($i = 1, 2 \dots N - 1$) that ranges from 0 to the value of provider's capacity as provider's 'temporary capacity' for the random route and allocate the delivery volume as step 4a. Repeat this step for all the first ($N-1$) routes, and let x_N be the provider's remaining capacity and allocate the delivery volume as Step 4a for the last route.
- For example, in rank level 2 in Figure 3(3), the routes manufacturer 1 (M1) to retailers 2 and 5 (R2 and R5) have the same provider facility and rank level, and therefore randomly generate a number 206 as the 'temporary capacity' of the manufacturer 1 and randomly select one route which is the route of M1 to R2. The remaining capacity of M1 is updated as 294 ($500 - 206$) units and the remaining demand of R2 as 94 ($300 - 206$) units after allocated. Then, M1 allocates 294 units to R5 as the last route in this situation. The remaining capacity of M1 is updated as zero units and the demand of R5 as 306 ($600 - 294$) units.
- Situation 3:** If the route's receiver facility is the same as the other route in this level, the allocation method is the same as situation 2.
- Step 5:* Update the rank level table, and then repeat Steps 2 to 5 until the encoding of *this stage* of CLSC ends.

The customer demand must be satisfied in this model. Therefore, the second step encoding starts from manufacturer to retailer and ends from supplier and recycling center to manufacturer, and the detailed encoding sequence is shown in Figure 4. It should be noted that the procedure of the encoding is simultaneously in the section of supplier and recycling center to the manufacturer. Therefore, the

routes of suppliers to manufacturers and recycling centers to manufacturers need to be ranked together.

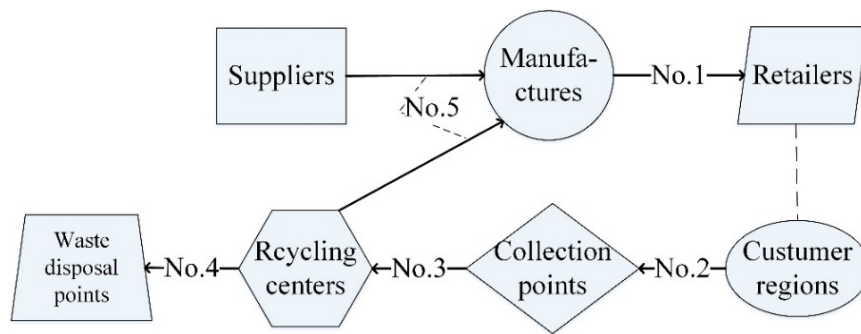


Figure 4. The sequence of second step encoding.

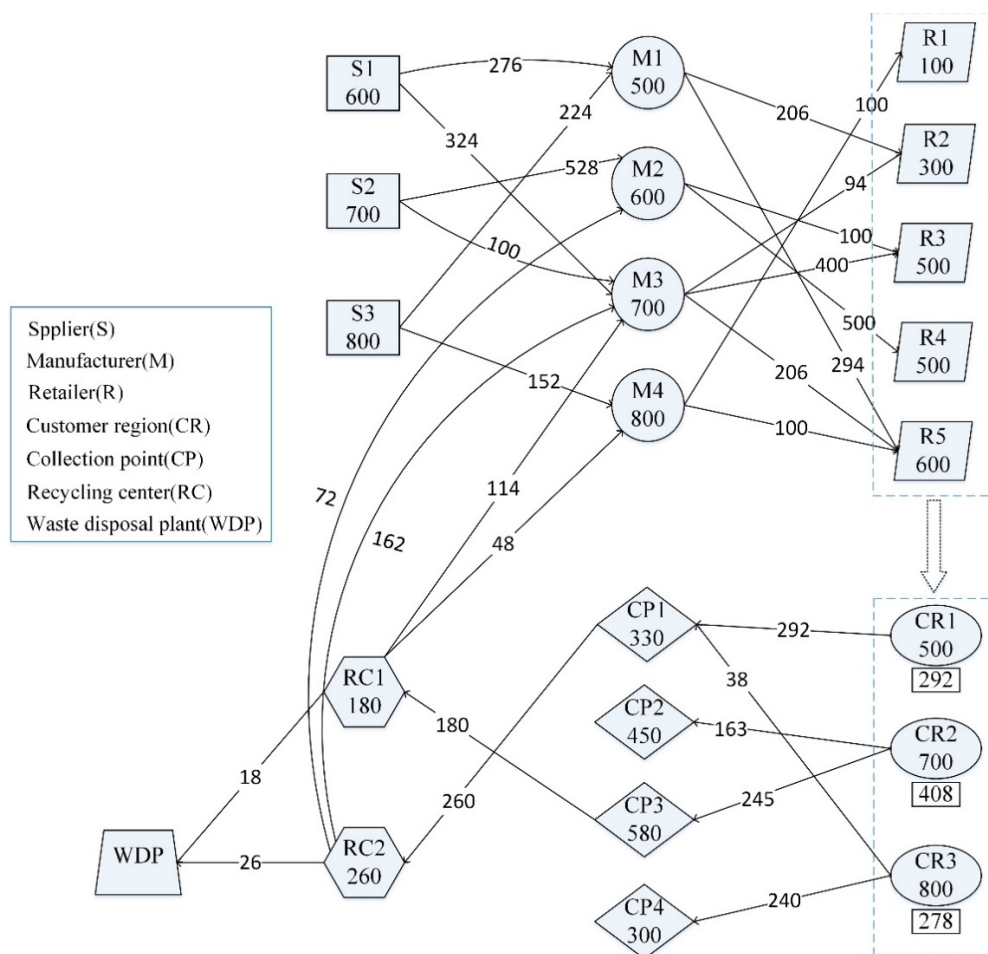


Figure 5. The whole encodes for the problem model corresponding to the example in Figure 2(2).

The diagram of the final encodes of the proposed problem model is shown in Figure 5, which corresponds to Figure 2(2). The numbers below the abbreviation of CLSC network members represent the capacity. The numbers in the routes represent the delivery volume. The numbers in the rectangular box under the customer regions represent the final delivery volume to the collection points.

3.2. Constraint check

There are many constraints in the proposed model, and therefore the constraints checking step is necessary. Constraints check is not only used for the legitimacy examining of newly generated chromosomes, but also for the offspring after the operators of GA such as crossover and mutation. In this paper, constraints check includes the capacity check of routes and delivery volume check.

The purpose of the delivery volume check is to test whether the chromosome satisfies the constraints (7) to (22) after the second step encoding. Capacity check of routes aims to check whether binary chromosome satisfies the capacity constraints after the first step encoding. If the chromosome fails to pass the capacity check, it means that no matter how the delivery volume is allocated, it will violate the constraints finally. Therefore, the route that violates the constraint will be regenerated. The details of the capacity check are shown below.

- 1) The total capacity of active manufacturers must be able to satisfy the total demand of all retailers.
- 2) The total capacity of active recycling centers and suppliers must be able to satisfy the total demand of all customers.
- 3) The quantity of products of each customer to recycling centers cannot be zero simultaneously, which means the customer must return the used products.
- 4) If the used products are transported to an active recycling center, then the delivery volume of the recycling center to waste disposal plants should not be zero simultaneously.

3.3. GA operators

3.3.1. The overall procedure of the proposed GA

In this paper, for preserving high-quality solutions, an elite strategy is used. The best E individuals of the parent are saved as the elites. If the best E individuals of the next generation are changed, the E elites will replace the worst individuals of the current population to avoid losing high-quality solutions, and E can be set according to the problem scale. The flowchart of the proposed algorithm shows in Figure 6.

3.3.2. Initialization and selection

An initial population includes a certain number of chromosomes generated based on two-step rank-based encoding and verified by constraint check. In this way, there are all feasible solutions in the initial population, since it is difficult for GA without the initialization based on the two-step rank-based encoding to generate feasible solutions, which is verified in the experiment in Sections 4.2 and 4.4. In the iteration of GA-TRE, the roulette selection is used as the selection operator [41].

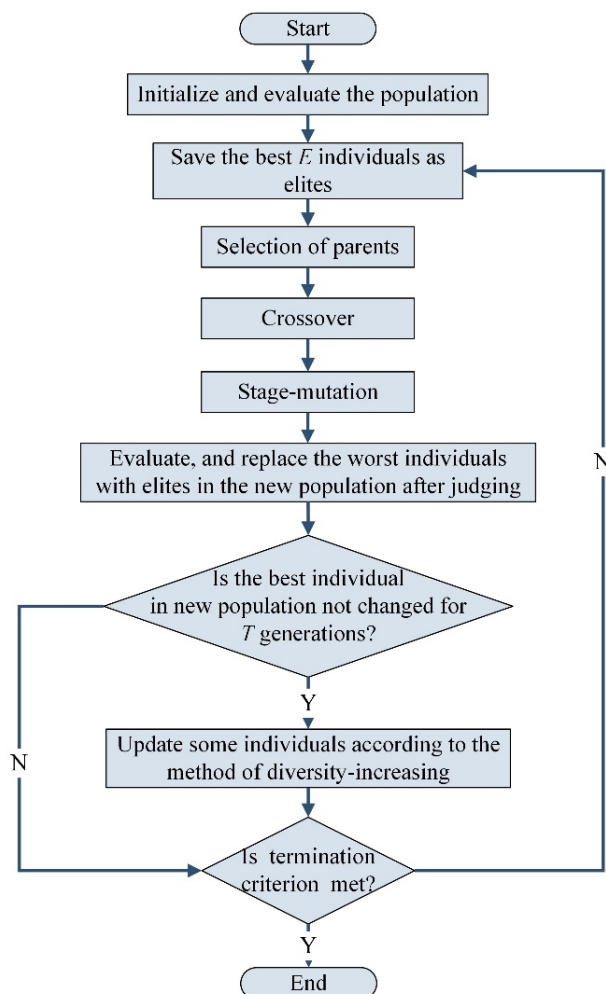


Figure 6. The flow chart of the proposed algorithm.

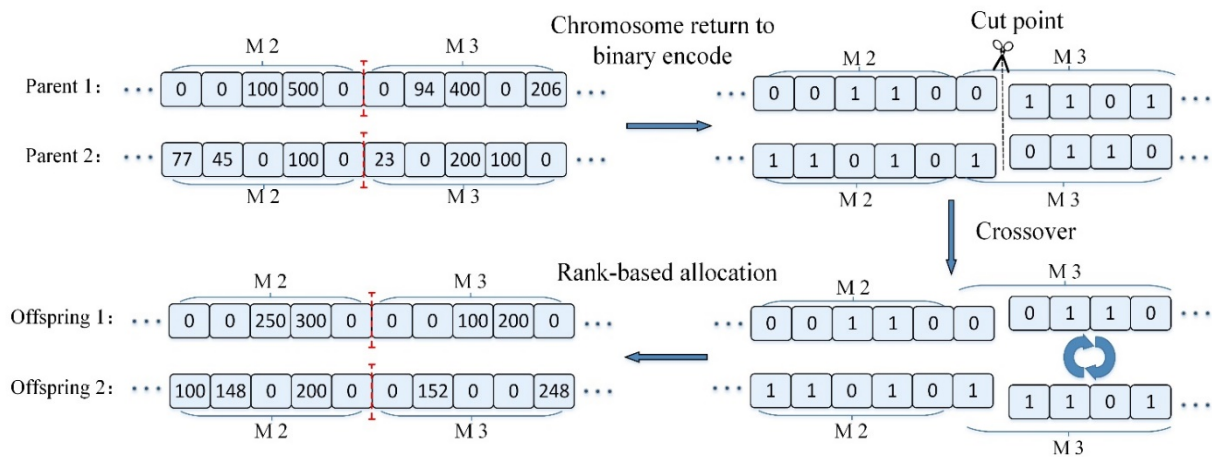
3.3.3. Crossover

For the proposed encoding method in this paper, the value of genes and their positions in the chromosome represent different information. It is difficult for the new individuals after crossover to satisfy the constraints. Hence, dramatic changes in the parent chromosomes are unfavorable for the preservation of the promising part of solutions. In this paper, the genes in the parent chromosome will degrade to binary encoding, and the single point crossover operation is implemented to minimize the change of the parent chromosomes. The position of the crossover point is random. The rank-based method of the second step encoding is used to reallocate delivery volume for child chromosomes. The offspring needs to be checked for constraints after crossover, and the crossover operator will be executed again if constraints are not met. An example of the crossover process and the pseudo-code of the crossover are shown in Figures 7 and 8, respectively.

Algorithm 1: Crossover

Data: Population P , crossover rate C_r
Result: Population P

- 1 $Children \leftarrow \emptyset$;
- 2 $Matingpool \leftarrow$ Select individuals from P according to C_r and roulette wheel function;
- 3 **while** $|Children| < |P|$ **do**
- 4 Choose a pair of parents from $Matingpool$;
- 5 Parents degrade to binary representation then crossover. Denote the offspring as c_1, c_2 ;
- 6 **if** c_1 or c_2 pass the capacity check **then**
- 7 rank-based allocation for c_1, c_2 ;
- 8 **if** c_1 or c_2 pass the constraints check **then**
- 9 $Children \leftarrow Children \cup \{c_1, c_2\}$;
- 10 **else**
- 11 restore parents and repeat the crossover and check operations;
- 12 **else**
- 13 restore parents and repeat the crossover and check operations;

Figure 7. The pseudo code of crossover.**Figure 8.** Crossover process.**3.3.4. Stage-mutation**

Stage-mutation is designed by combining with the characteristics of the problem in this paper, which aims to reduce the probability of population trapped into the local optimum. In the stage-mutation, the chromosomes degrade to binary and one section of the chromosome is randomly selected which corresponds to a kind of member of the CLSC, such as manufacturers to retailers. The binary

encoding of the selected section is randomly generated. If the new chromosome with binary representation passes the capacity check, the rank-based method of the second step encoding is used for reallocation. Moreover, the constraints check is used for the new chromosome. The mutation operator will be executed again if constraints are not met. The pseudo-code of stage-mutation and an example of the process are shown in Figures 9 and 10, respectively.

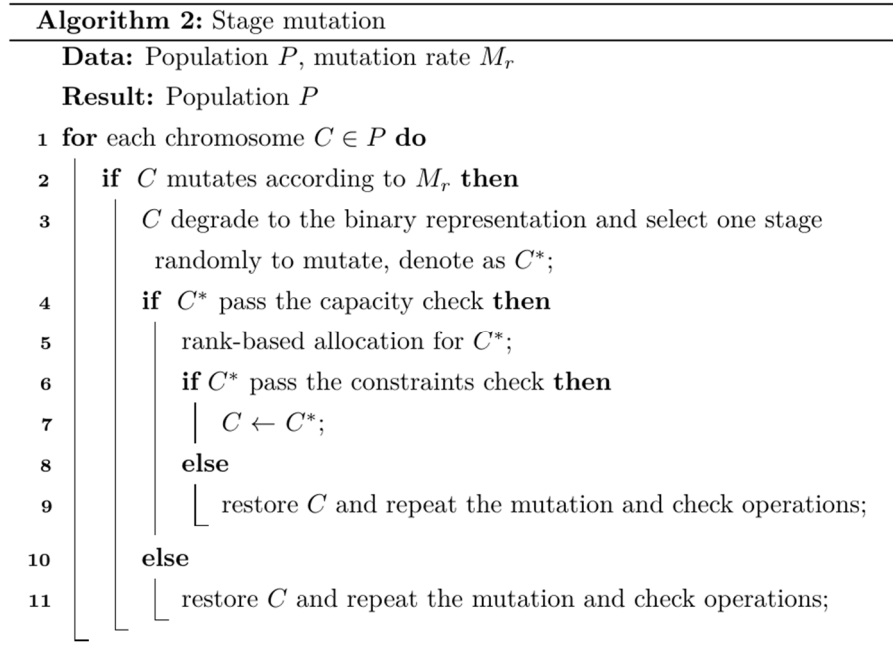


Figure 9. The pseudo code of stage-mutation.

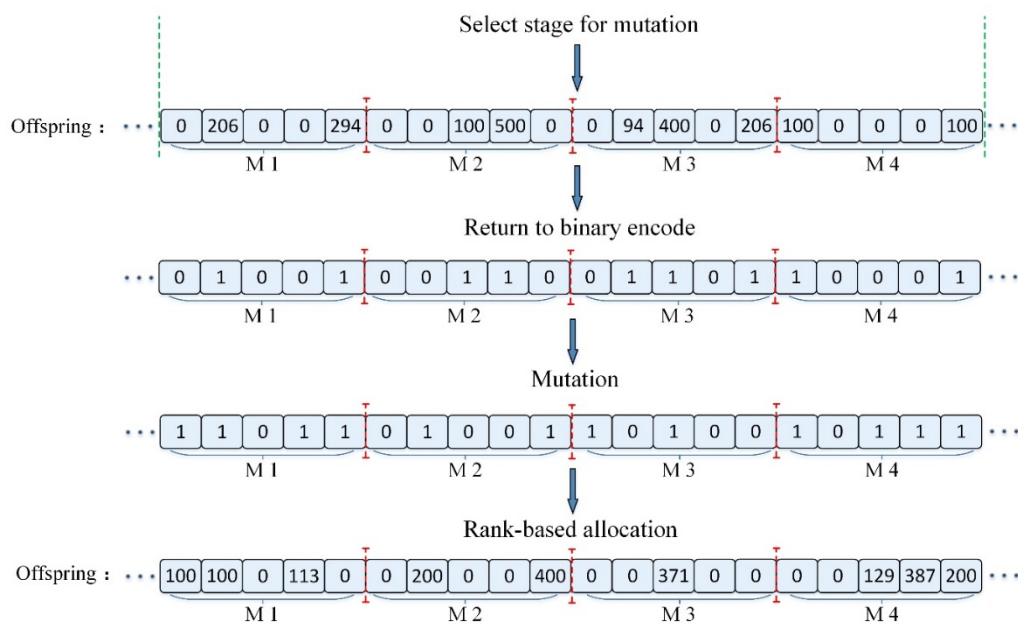


Figure 10. Stage-mutation process.

3.3.5. The diversity-increasing method

To enhance the diversity of individuals in the population and explore more solution space, if the best X individuals remain unchanged after T iterations, X new solutions will be generated to replace the worst X solutions in the population. T can be set according to the problem scale, and X can be changed adaptively according to different periods of population iteration. In detail, at the initial stage of the iteration, the population aims to explore, and therefore it is better to have a smaller X . In the middle of the iteration, the population will gradually converge. To prevent premature convergence and search for better solutions, X needs to be larger. At the later stage of iteration, the algorithm needs fast convergence. Therefore, X is not suitable for too large. To obtain different values of X in different periods of the iteration, an adaptive method of setting X is designed:

$$Jump_rate = \frac{(1 - 4 \cdot \beta) \cdot (Iter^2 - Max_iteration \cdot Iter)}{Max_iteration^2} + \alpha \quad (25)$$

$$X = N_pop \cdot Jump_rate \quad (26)$$

where $Jump_rate$ is the proportion of the population, X is the number of individuals updated in the population, $Iter$ is the current number of iterations, $Max_iteration$ is the maximum number of iterations, α is the minimum value of $Jump_rate$, and β is the maximum value of $Jump_rate$, and N_pop is the number of individuals in the population.

4. Experimental results and analysis

4.1. Experimental setting

To evaluate the performance of the proposed algorithms for solving the CLSC problem, three different scales are used. Each scale includes three instances. There are $3(\text{scales}) \times 3(\text{instances}) = 9$ instances in total. Moreover, the dimension of the problem-scale III is up to 292, and the range of each dimension is zero to hundreds, which is large-scale for an integer programming problem. The instance data are generated randomly according to the reality [29]. The details of each problem scale are reported in Tables 1 and 2.

Table 1. Scale configurations and execution time of the instance sets.

Scale	Size (S, M, Rt, Cu, Co, Rc, WDP)	Dimension (D)	Execution time (Seconds)
I	(3, 2, 3, 2, 2, 1)	21	30
II	(6, 4, 5, 3, 4, 2, 1)	74	180
III	(12, 8, 10, 6, 8, 4, 1)	292	360

Six algorithms were compared and implemented, including the classical algorithm GA, DE (differential evolution) [42], CSO (competitive swarm optimizer, which is designed for the large-scale optimization) [43], the current relatively new algorithms MPA (marine predators algorithm) [44], RDA (red deer algorithm) and SLPSO (social learning particle swarm optimization) [45,46]. For a fair

comparison, the solution structure is the same as GA-TRE for the compared algorithms as Figure 2 (2), and the variables are generated randomly from zero to the corresponding boundary value. The penalty function method is used on infeasible solutions in these compared algorithms for meeting the constraints. GA uses single-point crossover and mutation, and DE uses the binomial crossover in Reference [42]. The specific parameters of the algorithms are shown in Table 3.

Table 2. Optimal parameter settings of algorithms.

Algorithm	Tuned Parameters
GA-TRE	$N_pop = 100, Crossover_rate = 0.9, Mutation_rate = 0.2, \alpha = 0.25, \beta = 0.6, T = 10$
GA	$N_pop = 100, Crossover_rate = 0.9, Mutation_rate = 0.1$
DE	$N_pop = 100, Crossover_rate = 0.6, F = 0.4$
MPA	$N_pop = 100, FADS = 0.2, \beta = 1.5$
RDA	$N_pop = 500, Nmale = 75, \alpha = 0.9, \beta = 0.4, \gamma = 0.7$
SLPSO	$N_pop = 100, M = 100, \alpha = 0.5, \beta = 0.02$
CSO	$N_pop = 100$

Table 3. The Parameters and their surfaces for test instances.

Parameters	Surfaces
c_i^s, c_j^m	Rand ~ [100, 1000]
$c_l^{cp}, c_m^{rc}, d_k, d_v$	Rand ~ [100, 800]
$s_{ij}, m_{jk}, cr_{vl}, cp_{lm}, rc_{mj}, rc_{mw}$	Rand ~ [1, 30]
μ_v	Rand ~ {0.4, 0.45, 0.5, 0.55}
φ	Rand ~ {0.1, 0.13, 0.15, 0.18}

Each instance was tested with 30 replications, and the best, the worst, the average, and the standard deviation (std) values were recorded. All the algorithms were implemented in python 3.8 and run on a PC with the Intel Core i7-11700 and 16.0-GB memory.

4.2. Experimental comparisons with other algorithms

The experimental results of the algorithms for different scales and instances are shown in Tables 4–6. The results in *italic* type mean that the global optimal solution is illegal and the results in **bold** type are the best solutions among all the algorithms.

As it is seen in Table 4, GA-TRE obtains the maximum profit results on all the tested instances, and its mean result is 2000 higher than other algorithms at least. On account of the problem scale being small, most algorithms can obtain feasible solutions. SLPSO and MPA have similar results on instance 1, and SLPSO performs better than other algorithms on the rest of the instances. The results of DE, MPA, CSO, and SLPSO do not have a dramatic gap in general. In addition, RDA and GA cannot obtain feasible solutions on all the instances. In Table 5, the results show that GA-TRE yields the maximum profit results on all the tested instances with an obvious advantage. With the problem-scale increasing, GA, DE, CSO, and RDA cannot obtain feasible solutions in all the tested instances. MPA and SLCSO sometimes can obtain feasible solutions in some instances. From the observation of Table 6, the performance of GA-TRE does not deteriorate in the largest problem-scale, and all results are still the

best on all the tested instances. Besides, all the compared algorithms cannot obtain feasible solutions in this problem scale.

Table 4. The results of scale I.

Instances	Algorithms	Results			
		Worst	Best	Average	Std
1	GA-TRE	26125	26501	26282.43	74
	GA	$-3.16E + 19$	$-3.02E + 18$	$-7.13E + 18$	1.14E + 19
	DE	20769	22068	21322.1	486
	MPA	22575	23449	23208.67	460
	CSO	18751	22251	21506.14	991
	SLPSO	22032	23507	22935.5	360
	RDA	$-4.614E + 19$	$-1.13E + 19$	$-2.71E + 19$	1.48E + 19
2	GA-TRE	29164	29240	29206.7	127
	GA	$-4.05E + 19$	$-2.36E + 18$	$-1.46E + 19$	1.36E + 19
	DE	25128	26731	26133.9	617
	MPA	24452	26292	25533.6	576
	CSO	22597	26312	24466.2	1223
	SLPSO	26173	26472	26297.3	259
	RDA	$-7.13E + 19$	$-2.53E + 18$	$-2.61E + 19$	2.37E + 19
3	GA-TRE	27650	27981	27831.6	251
	GA	$-5.48E + 19$	$-2.33E + 17$	$-1.42E + 19$	1.75E + 19
	DE	20484	22149	21295	992
	MPA	19802	23188	21996.7	896
	CSO	17972	20198	18753.9	1099
	SLPSO	23098	24987	24017.9	561
	RDA	$-4.18E + 20$	$-1.46E + 19$	$-5.93E + 19$	1.94E + 20

From the perspective of solution quality, GA-TRE obtains the best results in different problem scales, and therefore it shows an absolute advantage over the compared algorithms. DE and the three recent popular algorithms—MPA, CSO and SLPSO can also obtain promising results on a small problem scale. However, with the problem scale increasing, these algorithms cannot obtain feasible solutions. It indicates that these algorithms are ineffective to handle multi-constraints when facing large-scale problems. Besides, GA and RDA cannot obtain feasible solutions on all the problem scales. The reason may be that the randomly generated initial population of GA and RDA almost do not have feasible solutions, and it is difficult to improve the feasibility of the infeasible solutions with the classical genetic operators, such as crossover and mating commander.

The standard deviation values were recorded to evaluate the reliability of the algorithms, as shown in Tables 4–6. The results show that GA-TRE achieves the minimal value in different instances with three problem scales, which means it has a better robust ability. SLPSO, MPA and DE have near results in the small problem scale and outperform the rest algorithms. As for the algorithms achieving the

unfeasible solutions, the standard deviation value seems to have little effect on the evaluation of the performance.

Table 5. The results of scale II.

Instances	Algorithms	Results			
		Worst	Best	Average	Std
1	GA-TRE	50729	51877	51368.1	373
	GA	$-1.74E + 20$	$-2.07E + 18$	$-1.12E + 20$	4.34E + 19
	DE	$-1.87E + 19$	$-2.10E + 8$	$-4.67E + 15$	2.63E + 18
	MPA	37890	40257	38819.8	798
	CSO	$-1.03E + 21$	$-5.43E + 19$	$-8.20E + 20$	3.96E + 20
	SLPSO	$-5.45E + 12$	41719	$-1.82E + 11$	1.67E + 12
	RDA	$-5.35E + 22$	$-8.70E + 21$	$-1.08E + 22$	2.01E + 22
2	GA-TRE	60115	63498	61997.9	588
	GA	$-1.97E + 20$	$-1.31E + 19$	$-4.94E + 19$	7.65E + 19
	DE	$-5.09E + 16$	$-1.98E + 8$	$-1.33E + 14$	5.82E + 15
	MPA	-41537852	42711	-1364431.2	18461524
	CSO	$-5.61E + 17$	$-7.29E + 14$	$-1.26E + 16$	3.36E + 16
	SLPSO	$-9.39E + 12$	46832	$-1.72E + 12$	2.98E + 12
	RDA	$-4.72E + 22$	$-2.71E + 20$	$-2.34E + 21$	3.91E + 21
3	GA-TRE	44009	46921	45566.7	674
	GA	$-1.23E + 20$	$-1.28E + 19$	$-4.91E + 19$	5.36E + 19
	DE	$-1.73E + 19$	$-1.10E + 8$	$-7.87E + 17$	5.82E + 18
	MPA	11912	30792	26758.9	8903
	CSO	$-3.76E + 19$	$-5.59E + 16$	$-6.44E + 18$	1.29E + 18
	SLPSO	$-3.14E + 14$	36818	$-1.24E + 13$	2.98E + 13
	RDA	$-5.56E + 23$	$-7.37E + 20$	$-9.31E + 22$	1.20E + 23

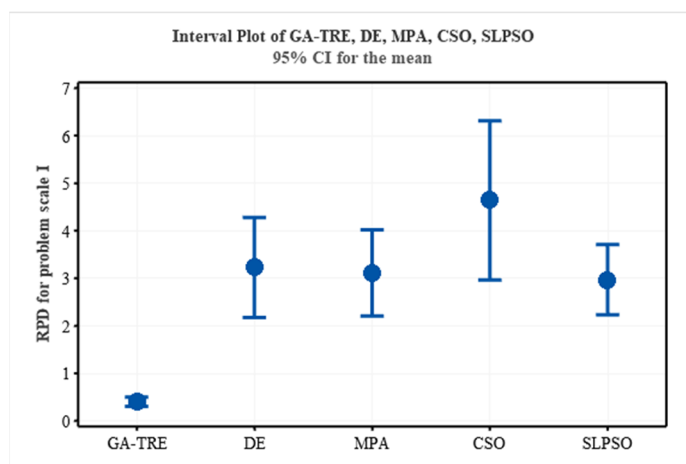
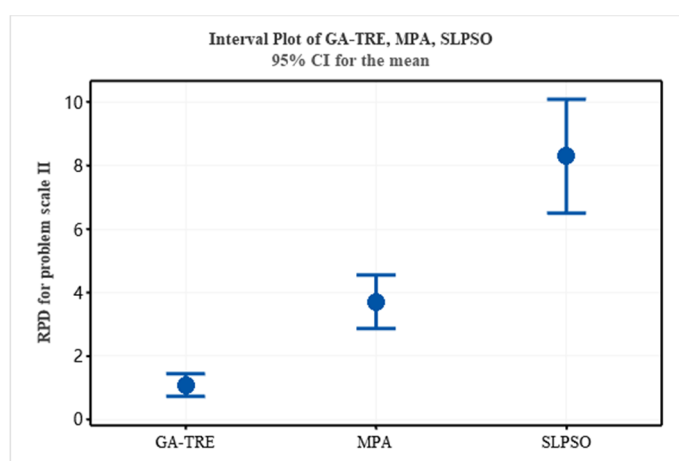


Figure 11. Interval plot for the scale I problems based on the algorithms which achieve feasible solutions (at 95% confidence interval)

Table 6. The results of scale III.

Instances	Algorithms	Results			
		Worst	Best	Average	Std
1	GA-TRE	95176	99738	97662.3	760
	GA	$-1.51E + 22$	$-2.79E + 20$	$-7.96E + 20$	$5.87E + 21$
	DE	$-1.66E + 23$	$-9.98E + 22$	$-1.32E + 23$	$5.59E + 22$
	MPA	$-6.91E + 17$	$-1.77E + 15$	$-4.09E + 16$	$2.31E + 17$
	CSO	$-2.21E + 23$	$-1.18E + 23$	$-1.34E + 23$	$7.08E + 22$
	SLPSO	$-9.64E + 22$	$-1.76E + 22$	$-4.81E + 22$	$3.11E + 22$
	RDA	$-2.11E + 23$	$-2.39E + 22$	$-3.43E + 22$	$6.38E + 22$
2	GA-TRE	99834	122280	109987.4	1321
	GA	$-7.12E + 20$	$-8.99E + 19$	$-1.97E + 20$	$3.01E + 20$
	DE	$-1.27E + 23$	$-8.93E + 22$	$-1.07E + 23$	$5.31E + 22$
	MPA	$-1.91E + 15$	$-3.38E + 13$	$-4.26E + 14$	$7.28E + 14$
	CSO	$-7.62E + 23$	$-2.43E + 22$	$-2.81E + 23$	$2.16E + 23$
	SLPSO	$-9.17E + 22$	$-2.92E + 22$	$-5.48E + 22$	$4.11E + 22$
	RDA	$-4.11E + 23$	$-5.39E + 21$	$-7.10E + 22$	$1.31E + 23$
3	GA-TRE	110175	129657	121147.5	1766
	GA	$-4.01E + 20$	$-1.23E + 20$	$-2.36E + 20$	$2.12E + 20$
	DE	$-6.51E + 23$	$-1.15E + 23$	$-1.68E + 23$	$1.93E + 23$
	MPA	$-5.29E + 16$	$-1.51E + 13$	$-7.49E + 14$	$1.72E + 15$
	CSO	$-4.62E + 23$	$-1.88E + 22$	$-6.33E + 22$	$1.17E + 23$
	SLPSO	$-7.29E + 23$	$-2.67E + 19$	$-2.88E + 21$	$2.40E + 23$
	RDA	$-6.11E + 22$	$-8.11E + 21$	$-2.46E + 22$	$2.17E + 22$

**Figure 12.** Interval plot for the scale II problems based on the algorithms which achieve feasible solutions (at 95% confidence interval).

Furthermore, the relative percentage deviation (RPD) is used to evaluate and compare the performance of the algorithms. For the maximization problem, RPD can be defined as follows:

$$RPD = \frac{Max_{sol} - Alg_{sol}}{Max_{sol}} \times 100 \quad (27)$$

where Alg_{sol} is the value of the objective function in individual trials and Max_{sol} depicts the best solution among trials. To intuitively observe the difference of algorithms' performance based on the RPD, the interval plots for different problem scales at 95% confidence limit are shown in Figures 11 and 12. It should be noticed that this analysis only includes the algorithms which can achieve feasible solutions, and therefore, GA-TRE, DE, MPA, CSO, and SLPSO for the problem scale I are drawn in Figure 11, and GA-TRE, MPA, and SLPSO for the problem scale II are drawn in Figure 12. As shown in Figure 11 for the problem scale I, GA-TRE outperforms all other algorithms, and there is a minor difference among DE, MPA, and SLPSO. Figure 12 demonstrates that GA-TRE still maintains the best performance among the algorithms with the problem scale increasing.

Finally, the one-way analysis of variance (ANOVA) is applied to verify the statistical validity of the results (based on standard deviation). The three problem scales have been all included in this analysis. The crucial measure parameter in ANOVA analysis is the p value, if it is lower than 0.05 that means the algorithms exist different. The results were calculated by the Minitab, and the obtained p value is equal to zero. Therefore, it proves that there are obvious differences between the performances of algorithms.

4.3. Analysis of convergence speed

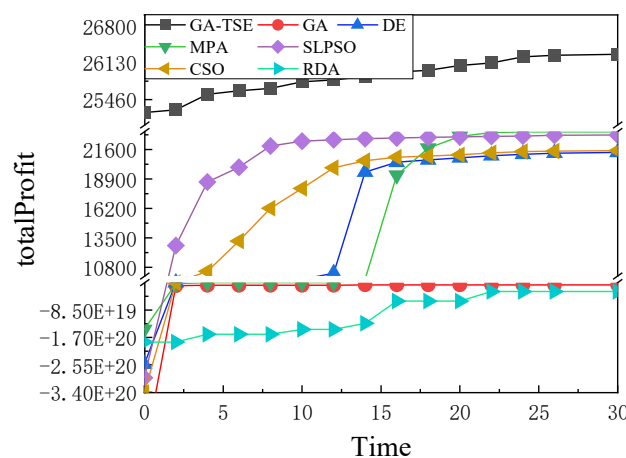


Figure 13. Convergence curves of all algorithms (scale I).

To analyze the convergence speed of algorithms, the convergence curves are drawn for different scales, as shown in Figures 13–15. From the observation of figures, the convergence curves of GA-TRE are rising gradually under different scales. Other algorithms will stagnate after several generations of iteration. Therefore, GA-TRE can obtain better solutions than other algorithms. The convergence speed of MPA is the slowest on all scales, and GA is the fastest on the contrary. That is because MPA has three different phases for exploration and exploitation, but the first two phases for exploration consume too much time for the complex constraints, and GA can firstly obtain its best solutions. SLPSO, CSO, and DE have similar convergence speed in scales I and III, and RDA and GA

have similar convergence speeds in scales II and III. In addition, with the problem scale increasing, the first iteration time of the GA-TRE is different from other algorithms as shown in Figure 15. The reason is that the individuals in the initial population are all feasible solutions, but it will take more time to initialize the population.

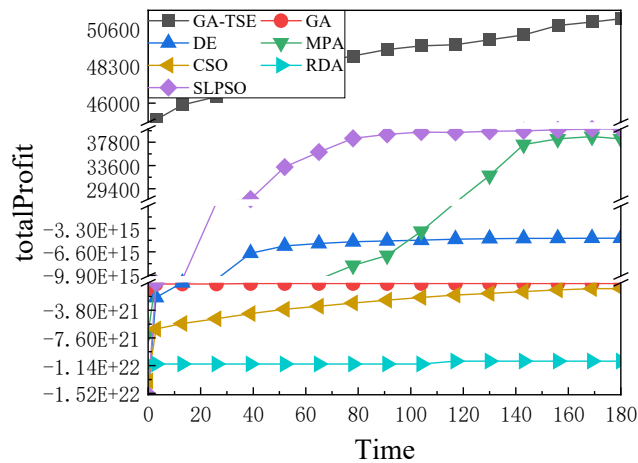


Figure 14. Convergence curves of all algorithms (scale II).

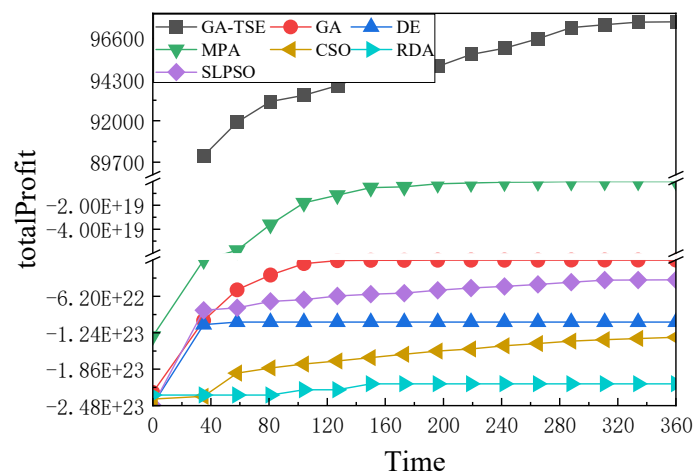


Figure 15. Convergence curves of all algorithms (scale III).

4.4. Effectiveness of two-step rank-based encoding

To validate the effectiveness of the two-step rank-based encoding, this encoding method is combined with other algorithms and tested on different problem scales. Since the proposed encoding is designed for the GA, only GA and DE (which has similar evolutionary operators to GA) are combined with the proposed encoding. It should be noticed that GA still uses single-point crossover and mutation. The experimental results are shown in Table 7.

Table 7. Results of algorithms with and without the two-step rank-based encoding.

Scales	Algorithms	Results		
		Worst	Best	Average
I	GA	$-4.23E + 19$	$-1.87E + 18$	$-1.23E + 19$
	GA-e	26271	26913	26578.9
	DE	22127	23649	22917
	DE-e	25100	25828	25455.7
	GA-TRE	27646	27907	27774
II	GA	$-1.65E + 20$	$-9.23E + 18$	$-7.02E + 19$
	GA-e	46004	49956	47599.8
	DE	$-1.20E + 19$	$-1.73E + 8$	$-2.64E + 17$
	DE-e	44595	47476	45955.8
	GA-TRE	51618	54099	52977
III	GA	$-5.40E + 21$	$-1.64E + 20$	$-4.10E + 20$
	GA-e	94273	99065	96136.1
	DE	$-3.15E + 23$	$-1.01E + 23$	$-1.36E + 23$
	DE-e	93181	96126	94243.8
	GA-TRE	101728	117225	109599.1

From the comparison of Table 7, it can be seen that GA-e (which is GA with two-step rank-based encoding) can obtain promising feasible solutions on all problem scales, and DE-e (which is DE with two-step rank-based encoding) also can obtain feasible solutions on scales II and III. The important reason why GA-e and DE-e can obtain feasible solutions at different scales is that the initial population and new individuals of these algorithms are generated based on the two-step rank-based encoding and pass the constraint checking. The advantage of the initialization is that all solutions are legal after the initialization. However, it will consume more time to initialize the population and generate new individuals with the increase of the problem scale. The comparison between GA-e and DE-e in Table 7 shows that GA-e is better than DE-e on different scales and instances. It indicates that the vector difference strategy of DE is not suitable for the proposed encoding method. Moreover, from the comparison between GA-TRE and GA-e in Table 7, it can be seen that GA-TRE obtains better results on different scales and instances. Therefore, it also proves the proposed operators including stage-mutation, elite strategy, and the adaptive population disturbance mechanism in this paper are effective for the proposed CLSC problems.

4.5. Analysis of Parameters

There are five key parameters of the GA-TRE algorithm to be investigated, including the crossover rate, mutation rate, parameter T of the diversity-increasing method and parameters α and β in (25). In this part, these parameters will be taken by a series of different values for observing the impact of the results, which aims to find a better combination of parameters. Specifically, the crossover rate is set by ten values from 0.1 to 1 (interval for 0.1), the mutation rate is set by six values from 0.1 to 0.6 (interval for 0.1), α is set from 0.1 to 0.35 (interval for 0.05), β is set from 0.45 to 0.7 (interval

for 0.05), and T is set from 3 to 17 (interval for 2). Also, the instances of problem scale II are selected for the test problem, and the results are presented in Figures 16–18.

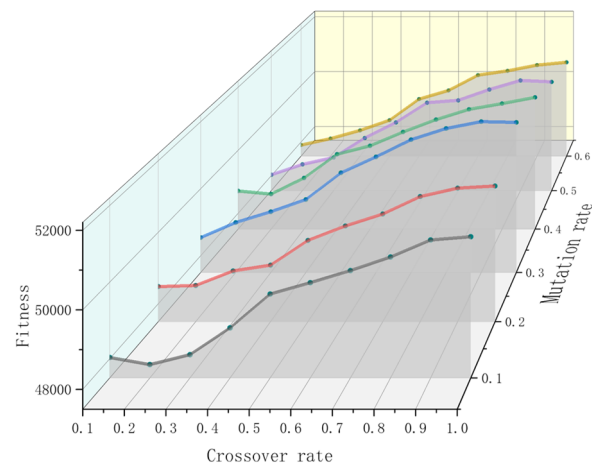


Figure 16. Fitness behavior for analysis of crossover and mutation rate.

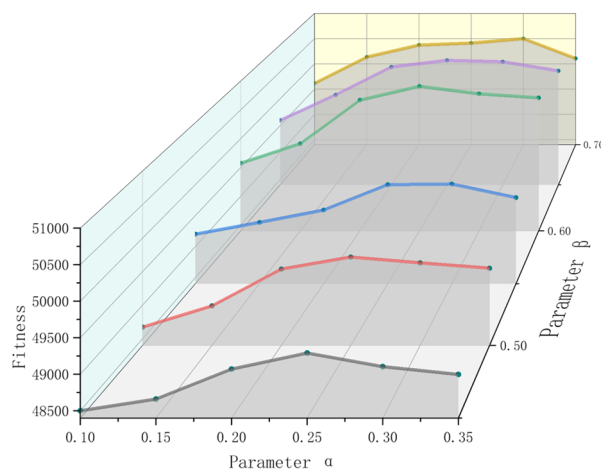


Figure 17. Fitness behavior for analysis of parameters α and β .

It can be observed from Figure 16, the results become better with the increase of crossover rate under different mutation rates and get relatively stable when the crossover rate is about 0.9. It indicates that the frequent crossover operations can promote sharing of excellent solutions in the population. The change of mutation rate seems to have a relatively smaller effect on the fitness, and the fitness value shows an upward to downward trend with the increase of mutation rate. Because mutation operation brings better diversity for the population, it also needs to take the algorithm consuming more time. As seen in Figure 17, the increase of parameters α will lead to near parabola shape changes for the fitness curve. The reason may be that if parameter α is too big, it will bring relatively large population changes in the early stage of the algorithm, which affects the exploitation ability. Also, the fitness value shows an increasing trend with the increase of parameter β , but it will get worse results

when β is higher than about 0.6, and the reason is similar to the former. Besides, parameter T is not suitable for setting too large from the observation in Figure 18.

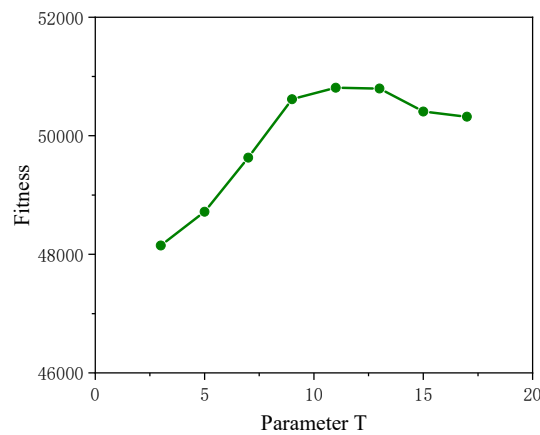


Figure 18. Fitness behavior for analysis of parameters T.

From the above comparisons and analysis, the five parameters crossover rate, mutation rate, parameter α , β and T are set as 0.9, 0.2, 0.25, 0.6 and 10 for the GA-TRE algorithm under problem scale II, respectively. In addition, the same analysis method is also applied to the other problem scales, the results of parameters combination for scale I and scale III are set as (0.9, 0.1, 0.25, 0.6 and 6) and (0.9, 0.25, 0.3, 0.65 and 10), respectively.

4.6. Sensitivity analysis

In this section, sensitivity analysis for the key parameters of the proposed CLSC model is conducted. The problem scale II experimental instance 1 was selected as a test problem and GA-TRE was selected as the optimizer. There are three scenarios created for sensitivity analysis to investigate the behavior of the objective function under the change of different CLSC members' capacity/demand, and recycling/disposal rate.

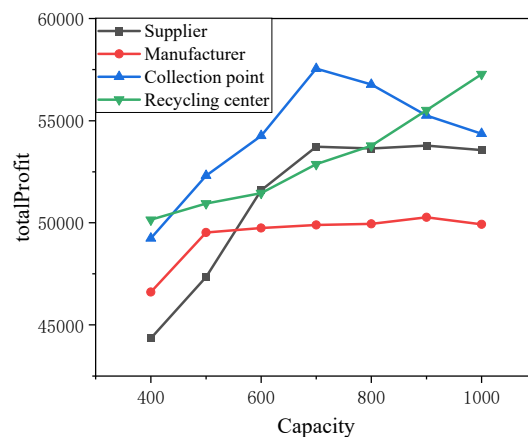


Figure 19. Behavior of the objective function for sensitivity analysis (first scenario).

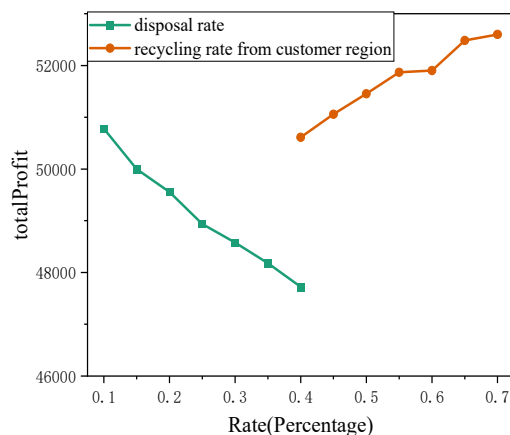


Figure 20. Behavior of the objective function for sensitivity analysis (second scenario).

The first scenario discusses the changes in the capacity of supplier (c_i^s), manufacturer (c_j^m), collection point (c_l^{cp}), and recycling center (c_m^{rc}), and each parameter varies between 400 to 1000 units. The second scenario explores the effect of changes in the minimal recycling rate of customer region (μ_v), and the maximal disposal rate (φ). The last scenario investigates the influence of demand of customer (d_v) and retailer (d_k). Moreover, other parameters remain unchanged when one parameter is tested. The performance of the objective function for the different scenarios is shown in Figures 19–21.

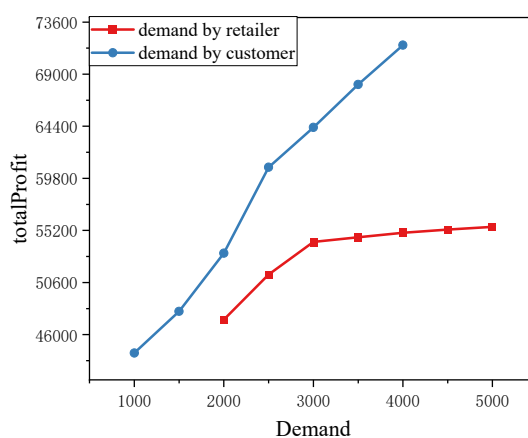


Figure 21. Behavior of the objective function for sensitivity analysis (third scenario).

As seen from the results in Figure 19, the capacity of the considered CLSC members is positively associated with an increase in the objective function. But notably, the capacity of the collection center exceeds a certain threshold, it will have a negative impact on the objective function. The reason may be that excessive recycled products lead to an increase in consumer subsidies, but the operation capacity of the recycling center is limited. Therefore, the supply chain decision-makers should consider carefully to avoid decreasing profit in the process of recycling products. According to Figure 20, the value of the objective function decreases with the increase of the disposal rate. It indicates that a bigger disposal rate in the recycling center brings more cost to deal with waste and affects the quantity of product recycling to a certain extent. In contrast, a higher recycling rate will provide cheaper recycled

materials for manufacturers to increase the total profit. As shown in Figure 21, with the increase in the demand of retailers and customers, there is a significant increase in the objective function. However, the demand of customers has a greater impact on the supply chain network than the demand of retailers. The above analysis of the important parameters of the model can provide a basic reference for decision-makers who are willing to use the proposed model in this paper.

5. Managerial insights

In this paper, a CLSC network is designed to maximize the total profits of the supply chain. The goal of the proposed model is to match more application scenarios under considering the development of sustainability. It can significantly help in providing the guidelines on how to take strategic decisions for the industries which have products with recyclable value, such as electronics and plastic.

The important managerial insights are generated by the sensitivity analyses of the model as given in Figures 19–21. The decision-makers should carefully plan the capacity of recycling centers, and each supplier and manufacturer can also avoid waste of production capacity (shown in Figure 19). In the reverse supply chain, it is crucial to balance the recycling rate from customers and disposal rate in the recycling center by under considering both increasing profits and environmental protection (shown in Figure 20). According to Figure 21, the demand of retailers and customers are positively associated with profits. In addition, the customer does not refer to a single buyer but represent a not fixed customer region for more closing the reality. The relationship between the demand of customer regions and retailers needs more detailed investigation for decision-makers.

As validated in the experiment, the proposed CLSC problem can be well solved by the GA-TRE algorithm, and the validation is presented in section 4. It can provide decision-makers to solve their similar or specific network design problems. Moreover, the key parameters tuning of GA-TRE refers to Figures 16–18.

6. Conclusions

Increasing environmental, legislative, and social concerns are forcing companies to take attention to green development, and therefore the CLSC model has become more and more popular. This paper solves an important practical issue considering the design and optimization of a general CLSC model. Due to the complexity of the problem, a novel genetic algorithm named GA-TRE is proposed. For handling the complex constraints and facilitating genetic operators, two-step rank-based encoding is proposed. Besides, for increasing the diversity, a new mutation operator and an adaptive population disturbance mechanism are designed. To validate the proposed algorithm, nine test problems are produced in three scales, and the performance and reliability of the proposed algorithm are evaluated in comparison with other heuristic algorithms. The experimental results demonstrate that GA-TRE outperforms all the compared algorithms on all the instances for the proposed CLSC problem.

For future studies, the CLSC model can extend as the multi-objective by considering sustainability and service quality to improve the versatility [3], and the proposed algorithms can be improved further to solve multi-objective problems. Moreover, the model with uncertain factors could be designed [10–13,22]. Finally, the evolutionary multitasking algorithm as a new search paradigm can also be used to solve supply chain problems as a promising direction for future research [47,48].

Acknowledgments

The work described in this paper was substantially supported in part by the National Natural Science Foundation of China under Grant No. 62106088, in part by the High-level personnel project of Jiangsu Province (JSSCBS20210852), and in part by the Fundamental Research Funds for the Central Universities, Jiangnan University (JUSRP121070), and Jilin University (93K172021K15).

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. M. C. Chen, Y. H. Hsiao, H. Y. Huang, Semiconductor supply chain planning with decisions of decoupling point and VMI scenario, *IEEE Trans. Syst. Man. Cybern. Syst.*, **47** (2017), 856–868. <https://doi.org/10.1109/tsmc.2016.2521740>
2. K. Govindan, A. Jafarian, R. Khodaverdi, K. Devika, Two-echelon multiple-vehicle location-routing problem with time windows for optimization of sustainable supply chain network of perishable food, *Int. J. Prod. Econ.*, **152** (2014), 9–28. <https://doi.org/10.1016/j.ijpe.2013.12.028>
3. T. Trisna, M. Marimin, Y. Arkeman, T. C. Sunarti, Multi-objective optimization for supply chain management problem: A literature review, *Decis. Sci. Lett.*, **5** (2016), 283–316. <https://doi.org/10.5267/j.dsl.2015.10.003>
4. S. K. De, K. Bhattacharya, B. Roy, Solution of a pollution sensitive supply chain model under fuzzy approximate reasoning, *Int. J. Intell. Syst.*, **36** (2021), 5530–5572. <https://doi.org/10.1002/int.22522>
5. S. K. Srivastava, Green supply-chain management: A state-of-the-art literature review, *Int. J. Manag. Rev.*, **9** (2007), 53–80. <https://doi.org/10.1111/j.1468-2370.2007.00202.x>
6. S. H. Amin, G. Q. Zhang, M. N. Eldali, A review of closed-loop supply chain models, *J. Data Inf. Manage.*, **2** (2020), 279–307. <https://doi.org/10.1007/s42488-020-00034-y>
7. B. Mosallanezhad, M. Hajiaghaei-Keshteli, C. Triki, Shrimp closed-loop supply chain network design, *Soft Comput.*, **25** (2021), 7399–7422. <https://doi.org/10.1007/s00500-021-05698-1>
8. A. Salehi-Amiri, A. Zahedi, N. Akbapour, M. Hajiaghaei-Keshteli, Designing a sustainable closed-loop supply chain network for walnut industry, *Renew Sustainable Energy Rev.*, **141** (2021). <https://doi.org/10.1016/j.rser.2021.110821>
9. A. Cheraghalipour, M. M. Paydar, M. Hajiaghaei-Keshteli, A bi-objective optimization for citrus closed-loop supply chain using Pareto-based algorithms, *Appl. Soft Comput.*, **69** (2018), 33–59. <https://doi.org/10.1016/j.asoc.2018.04.022>
10. A. M. Fathollahi-Fard, A. Ahmadi, S. M. J. M. Al-e-Hashem, Sustainable closed-loop supply chain network for an integrated water supply and wastewater collection system under uncertainty, *J. Environ. Manag.*, **275** (2020). <https://doi.org/10.1016/j.jenvman.2020.111277>
11. A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, S. Mirjalili, Multi-objective stochastic closed-loop supply chain network design with social considerations, *Appl. Soft Comput.*, **71** (2018), 505–525. <https://doi.org/10.1016/j.asoc.2018.07.025>

12. V. K. Chouhan, S. H. Khan, M. Hajiaghaei-Keshteli, S. Subramanian, Multi-facility-based improved closed-loop supply chain network for handling uncertain demands, *Soft Comput.*, **24** (2020), 7125–7147, <https://doi.org/10.1007/s00500-020-04868-x>
13. A. M. Fathollahi-Fard, M. A. Dulebenets, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, M. Safaeian, H. Mirzahasseinian, Two hybrid meta-heuristic algorithms for a dual-channel closed-loop supply chain network design problem in the tire industry under uncertainty, *Adv. Eng. Inf.*, **50** (2021). <https://doi.org/10.1016/j.aei.2021.101418>
14. E. Lesnaia, I. Vasilescu, S. C. Graves, The complexity of safety stock placement in general-network supply chains, in *Innovation in Manufacturing Systems and Technology (IMST)*, **1** (2005). <http://hdl.handle.net/1721.1/7537>
15. A. Niccolai, L. Bettini, R. Zich, Optimization of electric vehicles charging station deployment by means of evolutionary algorithms, *Int. J. Intell. Syst.*, **36** (2021), 5359–5383. <https://doi.org/10.1002/int.22515>
16. H. B. Ammar, W. B. Yahia, O. Ayadi, F. Masmoudi, Design of efficient multiobjective binary PSO algorithms for solving multi-item capacitated lot-sizing problem, *Int. J. Intell. Syst.*, **37** (2021), 1–28. <https://doi.org/10.1002/int.22693>
17. M. Mojtahedi, A. M. Fathollahi-Fard, R. Tavakkoli-Moghaddam, S. Newton, Sustainable vehicle routing problem for coordinated solid waste management, *J. Ind. Inf. Integr.*, **23** (2021). <https://doi.org/10.1016/j.jii.2021.100220>
18. A. M. Fathollahi-Fard, A. Ahmadi, B. Karimi, Sustainable and robust home healthcare logistics: A response to the Covid-19 pandemic, *Symmetry*, **14** (2), 193. <https://doi.org/10.3390/sym14020193>
19. H. J. Ko, G. W. Evans, A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs, *Comput. Oper. Res.*, **34** (2007), 346–366. <https://doi.org/10.1016/j.cor.2005.03.004>
20. S. Hamed, K. Govindan, A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks, *Appl. Math. Model.*, **39** (2015), 3990–4012. <https://doi.org/10.1016/j.apm.2014.12.016>
21. A. S. Abir, I. A. Bhuiyan, M. Arani, M. M. Billal, Multi-objective optimization for sustainable closed-loop supply chain network under demand uncertainty: A genetic algorithm, in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, (2020), 1–5. <https://doi.org/10.1109/ICDABI51230.2020.9325648>
22. X. Zhang, K. J. Du, Z. H. Zhan, S. Kwong, T. L. Gu, J. Zhang, Cooperative coevolutionary barebones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties, *IEEE Trans. Cybern.*, **50** (2020), 4454–4468. <https://doi.org/10.1109/TCYB.2019.2937565>
23. W. C. Yeh, T. L. W, C. M. L, Y. C. Lee, Y. Y. Chung, J. S. Lin, Application of simplified swarm optimization algorithm in deteriorate supply chain network problem, in *2016 IEEE Congress on Evolutionary Computation (CEC)*, (2016), 2695–2700. <https://doi.org/10.1109/CEC.2016.7744127>
24. K. Patne, N. Shukla, S. Kiridena, M. K. Tiwari, Solving closed-loop supply chain problems using game theoretic particle swarm optimisation, *Int. J. Prod. Res.*, **56** (2018), 5836–5853. <https://doi.org/10.1080/00207543.2018.1478149>

25. V. M. Esteves, M. C. Joao, C. A. Silva, A. P. Póvoa, M. I. Gomes, SCant-design: Closed loop supply chain design using ant colony optimization, in *2012 IEEE Congress on Evolutionary Computation*, (2012), 1–8. <https://doi.org/10.1109/CEC.2012.6252944>
26. A. Samadi, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, Solving a discounted closed-loop supply chain network design problem by recent metaheuristics, in *Fuzzy Information and Engineering-2019*, Springer Singapore, (2020), 3–24. https://doi.org/10.1007/978-981-15-2419-2_1
27. A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, S. Mirjalili, Hybrid optimizers to solve a tri-level programming model for a tire closed-loop supply chain network design problem, *Appl. Soft Comput.*, **70** (2018), 701–722. <https://doi.org/10.1016/j.asoc.2018.06.021>
28. M. Hajiaghaei-Keshteli, A. M. Fathollahi-Fard, Sustainable closed-loop supply chain network design with discount supposition, *Neural Comput. Appl.*, **31** (2019), 5343–5377. <https://doi.org/10.1007/s00521-018-3369-5>
29. M. Gen, F. Altiparmak, L. Lin, A genetic algorithm for two-stage transportation problem using priority-based encoding, *OR Spectrum*, **28** (2006), 337–354. <https://doi.org/10.1007/s00291-005-0029-9>
30. B. Fahimnia, H. Davarzani, A. Eshragh, Planning of complex supply chains: A performance comparison of three meta-heuristic algorithms, *Comput. Oper. Res.*, **89** (2018), 241–252. <https://doi.org/10.1016/j.cor.2015.10.008>
31. N. Sahebjamnia, A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, Sustainable tire closed-loop supply chain network design: Hybrid metaheuristic algorithms for large-scale networks, *J. Clean Prod.*, **196** (2018), 273–296. <https://doi.org/10.1016/j.jclepro.2018.05.245>
32. S. S. Theagarajan, H. L. Manohar, Lean management practices to improve supply chain performance of leather footwear industry, in *2015 International Conference on Industrial Engineering and Operations Management (IEOM)*, (2015), 1–5. <https://doi.org/10.1109/IEOM.2015.7093717>
33. M. A. Dulebenets, An adaptive polyploid memetic algorithm for scheduling trucks at a cross-docking terminal, *Inf. Sci.*, **565** (2021), 390–421. <https://doi.org/10.1016/j.ins.2021.02.039>
34. M. A. Dulebenets, A comprehensive multi-objective optimization model for the vessel scheduling problem in liner shipping, *Int. J. Prod. Econ.*, **196** (2018), 293–318. <https://doi.org/10.1016/j.ijpe.2017.10.027>
35. J. Pasha, M. A. Dulebenets, M. Kavousi, O. F. Abloye, H. Wang, W. Guo, An optimization model and solution algorithms for the vehicle routing problem with a “factory-in-a-box”, *IEEE Access*, **8** (2020), 134743–134763. <https://doi.org/10.1109/ACCESS.2020.3010176>
36. H. Zhao, C. Zhang, An online-learning-based evolutionary many-objective algorithm, *Inf. Sci.*, **509** (2020), 1–21. <https://doi.org/10.1016/j.ins.2019.08.069>
37. Y. Tian, T. Zhang, J. Xiao, X. Zhang, Y. Jin, A coevolutionary framework for constrained multi-objective optimization problems, *IEEE Trans. Evol. Comput.*, **25** (2021), 102–116. <https://doi.org/10.1109/TEVC.2020.3004012>
38. K. Li, R. Chen, G. Fu, X. Yao, Two-archive evolutionary algorithm for constrained multiobjective optimization, *IEEE Trans. Evol. Comput.*, **23** (2019), 303–315. <https://doi.org/10.1109/TEVC.2018.2855411>

39. P. Wang, B. Xue, M. Zhang, J. Liang, A grid-dominance based multi-objective algorithm for feature selection in classification, in *2021 IEEE Congress on Evolutionary Computation (CEC)*, (2021), 2053–2060. [https://doi.org/ 10.1109/CEC45853.2021.9504832](https://doi.org/10.1109/CEC45853.2021.9504832)
40. A. Li, B. Xue, M. Zhang, A forward search inspired particle swarm optimization algorithm for feature selection in classification, in *2021 IEEE Congress on Evolutionary Computation (CEC)*, (2021), 786–793. [https://doi.org/ 10.1109/CEC45853.2021.9504949](https://doi.org/10.1109/CEC45853.2021.9504949)
41. A. Lipowski, D. Lipowska, Roulette-wheel selection via stochastic acceptance, *Phys. A*, **391** (2012), 2193–2196. <https://doi.org/doi.org/10.1016/j.physa.2011.12.004>
42. S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE Trans. Evol. Comput.*, **15** (2011), 4–31. <https://doi.org/10.1109/tevc.2010.2059031>
43. R. Cheng, Y. C. Jin, A competitive swarm optimizer for large scale optimization, *IEEE Trans. Cybern.*, **45** (2015), 191–204. <https://doi.org/10.1109/TCYB.2014.2322602>
44. A. Faramarzi, M. Heidarinejad, S. Mirjalili, A. H. Gandomi, Marine Predators Algorithm: A nature-inspired metaheuristic, *Expert Syst. Appl.*, **152** (2020), 1–28. <https://doi.org/10.1016/j.eswa.2020.113377>
45. A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, Red deer algorithm (RDA): a new nature-inspired meta-heuristic, *Soft Comput.*, **24** (2020), 14637–14665. <https://doi.org/10.1007/s00500-020-04812-z>
46. R. Cheng, Y. C. Jin, A social learning particle swarm optimization algorithm for scalable optimization, *Inf. Sci.*, **291** (2015), 43–60. <https://doi.org/10.1016/j.ins.2014.08.039>
47. L. Feng, Y. Huang, L. Zhou, J. Zhong, A. Gupta, K. Tang, et al., Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem, *IEEE Trans. Evol. Comput.*, **51** (2021), 3143–3156. [https://doi.org/ 10.1109/TCYB.2019.2962865](https://doi.org/10.1109/TCYB.2019.2962865)
48. L. Feng, L. Zhou, A. Gupta, J. Zhong, Z. Zhu, K. C. Tan, et al., Solving generalized vehicle routing problem with occasional drivers via evolutionary multitasking, *IEEE Trans. Cybern.*, **51** (2021), 3171–3184. [https://doi.org/ 10.1109/TCYB.2019.2955599](https://doi.org/10.1109/TCYB.2019.2955599)



AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)