

Research Article

Container Swap Trailer Transportation Routing Problem Based on Genetic Algorithm

Hua-wei Ma ^{1,2}, Lei Tao ^{1,2} and Xiao-xuan Hu ^{1,2}

¹School of Management, Hefei University of Technology, Anhui, Hefei 230009, China

²Key Laboratory of Process Optimization and Intelligent Decision-Making, Ministry of Education, Anhui, Hefei 230009, China

Correspondence should be addressed to Hua-wei Ma; colt_mhw@126.com and Lei Tao; 843051074@qq.com

Received 18 May 2017; Revised 22 December 2017; Accepted 23 January 2018; Published 20 February 2018

Academic Editor: Alessandro Arsie

Copyright © 2018 Hua-wei Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In swap trailer transportation routing problems, trucks and trailers conduct swap operations at special positions called trailer points. The parallelization of stevedoring and transportation can be achieved by means of these trailer points. This logistics organization mode can be more effective than the others. In this paper, an integer programming model with capacity and time-window constraints was established. A repairing strategy is embedded in the genetic algorithm (GA) to solve the model. The repairing strategy is executed after the crossover and mutation operation to eliminate the illegal routes. Furthermore, a parameter self-adaptive adjustment policy is designed to improve the convergence. Then numerical experiments are implemented based on the generated datasets; the performance and robustness of the algorithm parameter self-adaptive adjustment policy are discussed. Finally, the results show that the improved algorithm performs better than elementary GA.

1. Introduction

Swap trailer transportation is a logistics organization mode that uses the container as a carrier, assembling the goods in the container unit and using trucks pulling trailers with cargo containers to fulfill stevedoring and transportation functions. When compared to traditional transportation modes, container swap trailer transportation not only achieves the parallelization of stevedoring and transportation procedures, but also improves the efficiency of transportation, increases the utility rate of containers, and decreases transportation costs. Container transportation is widely used in large-scale harbor transportation, food logistics, and other bulk cargo logistics. The large harbor is an important hub that connects various transportation modes with container distribution centers. Cargo containers are transported to the container terminal by shipping from around the world. If the container cannot be evacuated in time, containers will accumulate at the terminal. This will not only affect the efficiency of the container terminal, but will also generate additional container-management fees and rental costs. Container swap trailer transportation could be a method to solve the vehicle-routing problem related to the container cargo evacuation issue.

Swap trailer transportation is a research branch of the truck and trailer routing problem (TTRP) that is receiving increasing attention. Drexel [1] considered the vehicle-routing problem with trailers and transshipments with multiple synchronization constraints, such as the time window, along with spacial and loading aspects. Scheuerer [2], Lin et al. [3], and Villegas et al. [4, 5] used different heuristic algorithms (i.e., simulated annealing, tabu search, greedy randomized adaptive search procedures); to solve the general truck and trailer routing problem, Cuda et al. [6] classified the VRP and TTRPs from a two-echelon routing problem standpoint.

With further research, more constraints (e.g., time window, capacity, and stochastic demands) have been introduced into the TTRP. Lin et al. [7] solved the truck and trailer routing problem with time windows by simulated annealing algorithm. Derigs et al. [8] considered the truck and trailer routing problem with a time window under different constraints, considering whether the load transfer exists or not. Mirmohammadsadeghi and Ahmed [9] used the memetic algorithm to solve the truck and trailer routing problem with time window constraints under the condition of stochastic demand. Batsyn and Ponomarenko [10] discussed a site-dependent heterogeneous fleet truck and trailer routing

problem, treating soft and hard time windows separately, based on the sequential greedy insertion algorithm. Torres et al. [11] proposed a model based on the soft computing method to solve the TTRP, under which a decision-maker may be permitted to violate some constraints. Some scholars have introduced the organization mode of container transportation to the TTRP problem. Nishimura et al. [12] proposed a dynamic routing method to solve the task assignment problem by single or multiple trailers in a maritime container terminal. Vidović et al. [13] studied a routing problem for trucks with containers of different capacities in the process of distribution and collection. The truck and trailer routing problem has recently been applied in real life. Gracia et al. [14] and Dooley et al. [15] used the truck and trailer routing problem to solve a biomass transportation problem and a problem of collection of different types of milk. Li et al. [16] introduced a two-echelon time-constrained vehicle-routing problem in line haul-delivery systems considering carbon dioxide emissions. Regnier-Coudert et al. [17] studied a vehicle-routing problem with time windows, heterogeneous vehicles, and multiple types of jobs and then solved the problem based on the real-world datasets.

To summarize, a review of the literature shows that the truck and trailer routing problem is attracting increasing attention. Except for the TTRP, little literature addresses the swap trailer problem. It is worth highlighting that trucks and trailers can separate and recombine in the swap trailer problem. We intend to use the transportation mode by which trucks pull a single trailer to solve the cargo evacuation, which occurs in container terminals, and to study the container swap trailer transportation routing problem with the goal of maximum profit while satisfying capacity and time-window constraints.

In some classical TTRPs, the solutions are always distinguished as three different types of routes, like Lin et al. [7] and Derigs et al. [8]. In the former's paper, an open TTRP with time windows was proposed, and a type of the vehicle customer nodes which were called parking places played the role of trailer points. In the latter's paper, truck and trailer routing problem with splitting demand was studied, a load transfer between the different types of customer nodes was considered, and the separation between trucks and trailers was prohibited.

However, in this paper the distribution network is represented as the set of Hamiltonian circuits. The trailer points are considered as a type of nodes which are independent of vehicle customer nodes and the demand is forbidden splitting. Moreover, the separation and recombination between the trucks and trailers are carried out at the trailer points. Therefore, the swap trailer problem can be considered as a different variant of the above-mentioned TTRP.

The remainder of this paper is organized as follows: Section 2 explicitly describes the problem and presents an integer programming formulation model. In Section 3, a genetic algorithm with a parameter self-adaptive adjustment policy is proposed to solve the model. And a repairing strategy was designed to correct the illegal routes. Three types of numerical experiments are implemented on our generated datasets in Section 4. In the next section, some discussions on

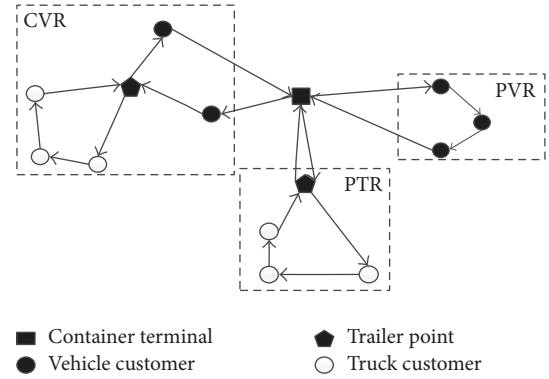


FIGURE 1: An illustration of a distribution network.

the performance, convergence, and robustness of algorithm based on the experimental results are given. Conclusions are drawn and future research directions are discussed in Section 6.

2. Problem Description and Mathematical Model

2.1. Problem Description. In this paper, we establish a distribution network composed of different nodes and joint arcs. There are three types of nodes: a container terminal, several trailer points, and customer nodes. The container terminal plays the role of the depot in the TTRP. The trailer points are defined as the specific locations where trucks and trailers can conduct swap operations (i.e., separation, recombination, and switch operations). Customer nodes are divided into two subsets. The first consists of the vehicle customer nodes that can be visited by a truck pulling a trailer. The second subset consists of the truck customer nodes that can only be visited by an individual truck.

Tours of vehicles, which are composed of many joint arcs, are classified under three types of routes in the distribution network: pure vehicle routes (PVRs), pure truck routes (PTRs), and complete vehicle routes (CVRs). A PVR represents a circuit that only includes a container terminal and vehicle customer nodes. A PTR denotes a circuit containing a container terminal, trailer points, and truck customer nodes. A CVR is a circuit consisting of all types of nodes. The distribution network is illustrated in Figure 1.

In this distribution network, some containers with goods are stored in the container terminal, and the goods should be transported to the corresponding customer nodes. Therefore, a simplified description of the container swap trailer transportation routing problem is as follows: a truck departs from the container terminal by pulling a container trailer, fulfills the demands of all customer nodes through choosing the appropriate routes, and finally returns to the original terminal. To improve the transportation efficiency, swap operations may be implemented at trailer points. The objective of the problem is to maximize profits by satisfying some constraints, such as those of time window and capacity. Here, it is assumed that

- (1) a truck is only allowed to pull one trailer;

- (2) both trucks and trailers have a certain capacity, but capacity exchanging between them is prohibited;
- (3) swap operations can only be carried out at the container terminal and trailer points.

2.2. The Model of the Problem. In this paper, we define a directed graph $V\langle N, E \rangle$ to represent the distribution network. $N = \{0\} \cup S \cup VC \cup TC$ is the node set, where node 0 represents the container terminal; $S = \{1, 2, 3, \dots, m\}$ is the set of trailer points; $VC = \{m+1, m+2, m+3, \dots, m+n_1\}$ is the set of vehicle customer nodes; and $TC = \{m+n_1+1, m+n_1+2, m+n_1+3, \dots, m+n_1+n_2\}$ is the set of truck customer nodes. $E = \{\langle i, j \rangle \mid \exists i, j \in N, i \neq j\}$ is the arc set.

(i) The Variables of Vehicles. Let $K = \{1, 2, 3, \dots, k\}$ represent the set of trucks and $L = \{1, 2, 3, \dots, l\}$ the set of trailers, where the maximum quantities of trucks and trailers are $|K|$ and $|L|$, respectively. Each truck $k \in K$ has its own capacity Q_k . Similarly, each trailer $l \in L$ has its own capacity Q_l . w_j is the reward of customer node j . c_{ijk} is the travel cost of truck k through node i to node j , and c_{ijl} is the travel cost of trailer l through node i to node j .

(ii) The Variables of Time-Window Constraints. Let t_{ij} be the travel time on the arc $\langle i, j \rangle$, at_j the arrival time of vehicles arriving at node j , and st_j the service time. Customer node j is associated with the time window $[e_j, r_j]$ that specifies that vehicles should visit the node j between time e_j and time r_j .

(iii) The Variables of Capacity Constraints. Let d_j^+ be the sum of cumulative demands of the routes before vehicles visit node j , and let d_j^- be the sum of cumulative demands of the routes after vehicles visit node j .

(iv) The Variables of Decision. x_{ijk} is used to indicate whether truck k visits node j after node i . If $x_{ijk} = 1$, truck k visits node j after node i ; otherwise, it does not. Similarly, if $y_{ijl} = 1$, trailer l visits node j after node i , and otherwise it does not. If $h_{kj} = 1$, truck k pulls trailer l , and otherwise it does not. z_j denotes whether the demand of node j is satisfied. If $z_j = 1$, the demand of node j is satisfied; otherwise, it is not satisfied.

2.3. Objective Function. The objective function consists of three parts. First is the sum of rewards after achieving all the demands of customer nodes. The second part is the total travel costs of all trucks, and the last part is the total travel costs of all trailers. The objective function is written as

$$\begin{aligned} & \sum_{j \in VC \cup TC} z_j w_j - \sum_{j \in TC \cup VC} \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ijk} x_{ijk} \\ & - \sum_{l \in L} \sum_{k \in K} \sum_{i \in (N-TC)} \sum_{j \in (N-TC)} c_{ijl} y_{ijl} h_{kl}. \end{aligned} \quad (1)$$

2.4. Constraints

(i) Depot Constraint. The depot constraint requires that vehicles start from the container terminal or trailer points and

return to the original terminal or corresponding trailer points after fulfilling their tasks. That is,

$$\begin{aligned} \sum_{k \in K} \sum_{j \in N} x_{ijk} &= \sum_{k \in K} \sum_{i \in N} x_{ijk} \leq |K|, \\ \sum_{l \in L} \sum_{j \in (N-TC)} y_{ijl} &= \sum_{l \in L} \sum_{i \in (N-TC)} y_{ijl} \leq |L|. \end{aligned} \quad (2)$$

(ii) Unique Accessibility Constraint. This requires that each customer node can be visited by a truck or a truck pulling a trailer only once at most:

$$\begin{aligned} \sum_{k \in K} \sum_{j \in VC \cup TC} x_{ijk} &\leq 1, \\ \sum_{l \in L} \sum_{j \in VC} y_{ijl} &\leq 1. \end{aligned} \quad (3)$$

(iii) Capacity Constraint. This constraint requires that the sum of cumulative demands in the routes including node j cannot exceed the sum of trucks and trailers before they visit the node. After vehicles visit node j , the sum of cumulative demands in the routes may be greater than the sum of the capacities of trucks and trailers. That is,

$$\begin{aligned} d_j^+ &\leq \sum_{i \in N} \sum_{j \in N} x_{ijk} Q_k + \sum_{i \in N} \sum_{j \in (N-TC)} \sum_{l \in L} y_{ijl} h_{kl} Q_l, \\ d_j^- &\geq \sum_{i \in N} \sum_{j \in N} x_{ijk} Q_k + \sum_{i \in N} \sum_{j \in (N-TC)} \sum_{l \in L} y_{ijl} h_{kl} Q_l. \end{aligned} \quad (4)$$

(iv) Time-Window Constraint. This constraint is necessary to ensure that the arrival time of vehicles cannot exceed the interval $[e_j, r_j]$. T is expressed as a large enough integer. That is,

$$\begin{aligned} e_j &\leq at_j \leq r_j, \\ t_{ij} + at_i + st_i - (1 - x_{ijk}) T &\leq at_j. \end{aligned} \quad (5)$$

(v) Secondary Circuit Constraint. We introduce this constraint to eliminate secondary circuits that may occur in the distribution network. P represents the set of nodes that have been visited by a truck or trailer:

$$\begin{aligned} \sum_{i \in P} \sum_{j \in P} x_{ijk} &\leq |P| - 1, \\ \sum_{i \in P} \sum_{j \in P} y_{ijl} &\leq |P| - 1. \end{aligned} \quad (6)$$

2.5. *The Model.* The mathematical model is as follows:

$$\begin{aligned}
& \sum_{j \in \text{VCUTC}} z_j w_j - \sum_{j \in \text{TCUVC}} \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ijk} x_{ijk} \\
& - \sum_{l \in L} \sum_{k \in K} \sum_{i \in (N-\text{TC})} \sum_{j \in (N-\text{TC})} c_{ijl} y_{ijl} h_{kl}, \\
\text{s.t. } & \sum_{k \in K} \sum_{j \in N} x_{ijk} = \sum_{k \in K} \sum_{i \in N} x_{ijk} \leq |K|, \\
& \sum_{l \in L} \sum_{j \in (N-\text{TC})} y_{ijl} = \sum_{l \in L} \sum_{i \in (N-\text{TC})} y_{ijl} \leq |L|, \\
& \sum_{k \in K} \sum_{j \in \text{VCUTC}} x_{ijk} \leq 1, \\
& \sum_{l \in L} \sum_{j \in \text{VC}} y_{ijl} \leq 1, \\
& d_j^+ \leq \sum_{i \in N} \sum_{j \in N} x_{ijk} Q_k + \sum_{i \in N} \sum_{j \in (N-\text{TC})} \sum_{l \in L} y_{ijl} h_{kl} Q_l, \\
& d_j^- \geq \sum_{i \in N} \sum_{j \in N} x_{ijk} Q_k + \sum_{i \in N} \sum_{j \in (N-\text{TC})} \sum_{l \in L} y_{ijl} h_{kl} Q_l, \\
& e_j \leq at_j \leq r_j, \\
& t_{ij} + at_i + st_i - (1 - x_{ijk}) T \leq at_j, \\
& \sum_{i \in P} \sum_{j \in P} x_{ijk} \leq |P| - 1, \\
& \sum_{i \in P} \sum_{j \in P} y_{ijl} \leq |P| - 1, \\
& x_{ijk}, y_{ijl}, h_{kl}, z_j \in \{0, 1\}.
\end{aligned} \tag{7}$$

3. The Genetic Algorithm

The genetic algorithm is a global optimization algorithm. It refers to the rule of ‘‘survival of the fittest,’’ from biological evolution, to solve the combinatorial optimization problem. McWilliams et al. [18] proposed a genetic algorithm to solve the schedule problem of an inbound trailer to unload at docks at central parcel consolidation terminals, and the results exceeded those of the random approach. Chen et al. [19] came up with a mix-cross operation to realize the loading and unloading operations simultaneously on quay cranes and solved the model of TTRP through a bilevel genetic algorithm. The results showed that the travel distance of yard trailers decreased to a large extent. Because of its parallel property, higher retrieval efficiency, and good robustness, we adopt the genetic algorithm to solve the container swap trailer transportation routing problem in this paper. To improve the convergence speed of the genetic algorithm, introducing a parameter self-adaptive policy is an effective method [20, 21]. Therefore, a parameter self-adaptive adjustment policy was introduced in Section 3.3.

3.1. *Chromosome Encoding.* In each chromosome, the serial number of the genes and the serial number of the trucks have

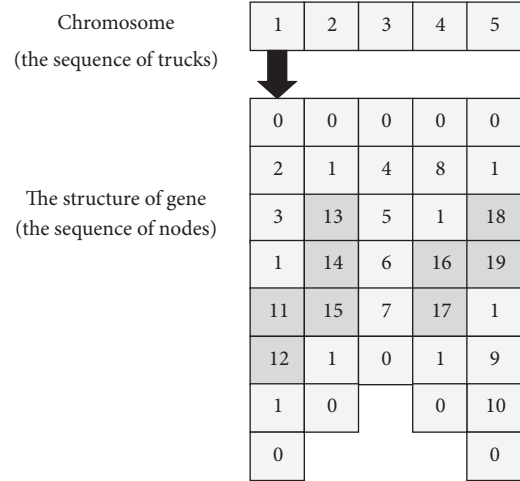


FIGURE 2: An illustration of the chromosome.

a one-to-one correspondence. We introduce structures to express the genes. The structure is composed of a sequence of nodes. The natural numbers are used to encode all the nodes in the network. Figure 2 is an illustration of the chromosome.

In the example, there are five trucks, and we assign a trailer to each truck. The chromosome is composed of five genes. The first gene is composed of the permutation of natural numbers (0, 2, 3, 1, 11, 12, 1, 0), where number 0 is the container terminal, number 1 is the trailer point, numbers 2 and 3 represent the truck customer nodes, and numbers 11 and 12 represent the vehicle customer nodes. Therefore, the route is in accord with the definition of CVR, which can be described as follows: The number one truck departs from the container terminal with a trailer. Then the truck drops off the trailer at the trailer point (number 1), and it will visit the truck customer nodes on its own. After visiting these customer nodes, the truck will return to the trailer point and hook up the trailer again; they will visit the rest of vehicle customer nodes together. Finally, they will return to the container terminal.

3.2. *The Generation Process of the Initial Solution.* Here we explain the main idea of the generation process. First, we randomly construct an initial route composed of all the customer nodes. Then, customer nodes are clustered to trucks, and trailer points are inserted if truck customer nodes exist in the routes. Finally, the container terminal is connected to the routes to complete the circuits. The explicit process is described as follows.

Step 1 (generate an initial route). A permutation of all customer nodes is performed randomly. The initial sequence is called *OriginRoute*.

Step 2 (assign the *OriginRoute* and cluster the customer nodes to trucks). Divide *OriginRoute* randomly into N subroutes and assign them to the N trucks. If there are only vehicle customer nodes in the subroutes, go to step 4 and conduct the PVR; otherwise, go to step 3.

Step 3 (insert the trailer points into the subroutes). If there are only truck customer nodes, we can choose a trailer point and insert it into the origin and end of the subroutes to get the PTR. If there is a mixture of vehicle and truck customer nodes, a trailer point will be selected stochastically. Regarding the sequence of successive truck customer nodes, insert the trailer point into the origin and the end of each sequence. As for individual truck customer nodes between two vehicle customer nodes, insert the same trailer point into the front and back of the truck customer node. In this way, the CVR can be conducted.

Step 4 (conduct the complete subroutes). Insert the container terminal into the origin and end of each respective subroute. Therefore, The N complete subroutes can be regarded as an initial solution of the problem.

3.3. Genetic Algorithm Procedure. In this paper, the initial solution may be a local optimum, and sometimes it may be an infeasible result because of the constraints. To improve the efficiency of the programming operation, we apply the crossover and mutation operations to the *OriginRoute* to exchange the structure of routes, helping us jump out of the local optimum and get a global optimal solution as often as possible. The steps of the genetic algorithm are as follows.

Step 1. Repeat execution of the generation process of the initial solution $POPSIZE$ times, and form the original population. Let $POP(t)$ represent the population and initialize $t = 1$.

Step 2. According to the objective of the model, we can calculate the function value of each chromosome in the t -generation population $POP(t)$.

Step 3. $fitness = F$ is the fitness function of the algorithm; this is used to calculate the fitness value of each chromosome.

Step 4 (selection operation). Based on the fitness value, the roulette method is used for searching $POPSIZE$ chromosomes from the t -generation population $POP(t)$ and generating the population composed of the chromosomes with high fitness values.

Step 5 (crossover operation). The two-point crossover operation is carried out in the *OriginRoute* of the population in pairs. First, select two intersection positions randomly in the two adjacent chromosomes that are about to be crossed. Second, exchange the middle part of chromosomes between the two intersection positions. The details of the process are as follows: Add the middle part of the second chromosome to the forepart of the first chromosome, and add the middle part of the first chromosome to the forepart of the second chromosome. Delete repeated nodes in the two chromosomes.

Step 6 (mutation operation). First, a mutation rate Pm ($0 < Pm < 1$) is given. Then, two customer nodes are randomly selected in each *OriginRoute*, and the positions of the two nodes are exchanged. Finally, the new *OriginRoute* of the population is generated after mutation.

Step 7 (the repairing strategy). Let the new *OriginRoute* carry out steps 3 and 4 of the generation process of the initial solution, and the offspring population is generated.

Step 8. Administer a constraint test to the chromosomes in the population $POP(t)$, mainly considering the time-window and capacity constraints. If a chromosome cannot pass the test, a penalty factor will be subtracted from the value of the objective function. This operation decreases the value of the fitness function so that chromosomes that cannot satisfy the constraints will be removed in the selection operation.

Step 9 (update operation). First, arrange the offspring population by the value of fitness function in descending order, and choose the front $SonNum$ chromosomes. Then, arrange the parent population in ascending order of the value of the fitness and choose the back $FatherNum$ chromosomes. Recombine these two sets of chromosomes as the new-generation population $POP(t + 1)$.

Step 10 (parameter self-adaptive adjustment policy). The adjustment formulas for crossover rate, mutation rate, and update number are shown as follows:

$$PCross = \begin{cases} \frac{f_{avg}}{f_{max}} & \text{if } f_{max} \leq MAX, \\ PCross_0 * \frac{f_{min}}{f_{max}} & \text{if } f_{max} > MAX, \end{cases}$$

PMutation

$$= \begin{cases} 1 - \frac{f_{avg}}{f_{max}} & \text{if } f_{max} \leq MAX, \\ PMutation_0 * \left(1 - \frac{f_{min}}{f_{max}}\right) & \text{if } f_{max} > MAX, \end{cases} \quad (8)$$

UpdateNumber

$$= \begin{cases} Update_0 & \text{if } f_{max} \leq MAX, \\ Update_0 - POPSIZE * \left(1 - \frac{f_{min}}{f_{max}}\right) & \text{if } f_{max} > MAX. \end{cases}$$

$PCross$, $PMutation$, and $UpdateNumber$ represent the crossover rate, mutation rate, and update number in the current iteration. $PCross_0$, $PMutation_0$, and $Update_0$ represent the original crossover rate and mutation rate, respectively. f_{avg} , f_{max} , and f_{min} are the average, maximum, and minimum of the objective function value until the current iteration. MAX is the maximum of the objective function values until the current iteration.

Step 11 (judgment of termination criteria). If the number of iterations has reached the maximum, or if the current solution has remained unchanged for M generations, output the current chromosome as the optimal solution. If not, let $t = t + 1$ and repeat steps 2–10.

4. Numerical Experiments and Computational Results

To verify the performance and effectiveness of the algorithm, we designed a series of numerical experiments based on

TABLE I: Parameters of genetic algorithm.

Population size	100
Maximum number of iterations	1,500
Crossover rate	0.9
Mutation rate	0.1
Update number	10

the Solomon Benchmark dataset. The genetic algorithm was coded in Matlab and the experiments were run on a computer with an Intel Xeon running at 3.2 GHz under Windows 7 Enterprise Edition (64 bits) with 4 GB of RAM.

4.1. Experimental Datasets. To test the algorithm, we made some adjustments to the Solomon Benchmark dataset. To generate the dataset, we let the first node represent the container terminal; the second to the fourth nodes represented the trailer points. The fifth to the 50th nodes denoted the vehicle customer nodes, and the rest of the nodes denoted the truck customer nodes. Also, we added a column of data to indicate the income of customer nodes in every set. We assumed that the income of each customer node was 1,000 and that the incomes of the other nodes were zero.

There are six types of datasets (C1, C2, R1, R2, RC1, and RC2) in the Solomon Benchmark dataset. The coordinates of nodes in sets C1 and C2 present the cluster distribution characteristics, while those of nodes in sets R1 and R2 are randomly distributed. The coordinates of nodes in sets RC1 and RC2 are a mixture of random and cluster distributions. In addition, sets R1, C1, and RC1 have a short scheduling horizon, while sets R2, C2, and RC2 have a long scheduling horizon.

Based on this, we introduced five trucks and five trailers to the distribution network. We defined the capacity of each truck and each trailer to be 200. Each truck had a speed of 60 km/h and a unit travel cost of 10. The unit travel time of each trailer was 5.

4.2. Numerical Experiments. In this section, three types of experiments were implemented. First, the validity of the basic genetic algorithm (GA) was tested on the above-mentioned datasets. Second, the sensitivity of parameters (e.g., crossover rate, mutation rate, and update number) was examined, the effect on the experimental results was analyzed. Finally, the effectiveness tests of the genetic algorithm with parameter self-adaptive policy (AGA) and the parameter sensitivity tests of the AGA were conducted on the combination of crossover rates and mutation rates. Based on this, the original parameters of the genetic algorithm were set in Table 1.

In the first type of experiments, the first group of each set was chosen. The genetic algorithm was carried out 20 times for each group of datasets. Figure 3 shows the changes of optimal results with the run of the genetic algorithm; it can be concluded that the optimal solution had reached convergence when the iteration was up to around 1,300. And the experimental results of the six types of datasets are shown in Table 2.

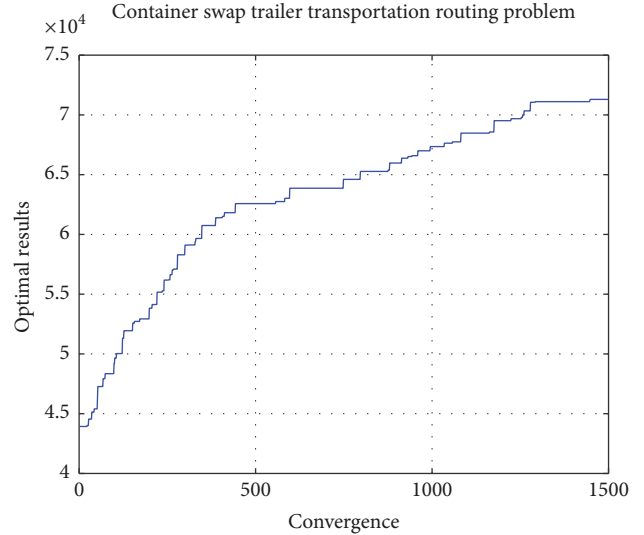


FIGURE 3: Changes of optimal results in genetic algorithm.

In Table 2, the gap shows the robustness of the optimal results and the convergence of different types of datasets. It can be seen that C2 has a narrower gap than C1. Conversely, dataset R1 has a narrower gap than dataset R2. Moreover, the robustness of RC2 is better than that of RC1. In conclusion, the genetic algorithm's robustness varied by the type of dataset.

In the second type of experiments, the crossover rate was set to 0.5, 0.6, 0.7, 0.8, and 0.9 in turn, and the mutation rate was 0.1, 0.2, 0.3, 0.4, and 0.5 when the update number was set to 10. We carried out 25 groups of experiments, one for each combination of the two parameters, and the experiments for each set were implemented 20 times. The results are as follows.

- (1) *Results of Dataset C1.* See Figure 4.
- (2) *Results of Dataset C2.* See Figure 5.
- (3) *Results of Dataset R1.* See Figure 6.
- (4) *Results of Dataset R2.* See Figure 7.
- (5) *The Results of Dataset RC1.* See Figure 8.
- (6) *Results of Dataset RC2.* See Figure 9.

From Figure 4, it can be concluded that the values of the optimal results increase with the crossover rate, and the effect of varied mutation rates is not evident in the optimal results. Regarding convergence, the effect of varied crossover rates is not evident, but the values represent a redundant trend with varied mutation rates. From Figure 5, the values of optimal results are increasing with increasing crossover rates. The effect of varied mutation rates is not evident in optimal results, but when the mutation rate was 0.1, the values were higher. As for convergence, the values represent an

TABLE 2: Experimental results of the six datasets.

Dataset Type	Optimal results				Convergence			
	Best	Avg	Worst	Gap	Best	Avg	Worst	Gap
C1	75,007.5	71,632.6	66,365.1	0.11	657	1,429.5	1,498	0.56
C2	71,115.8	67,283.1	64,588.1	0.09	1419	1,417.7	1,492	0.048
R1	67,390.6	64,601.1	60,400.2	0.10	551	1,162.0	1,393	0.60
R2	68,634.1	63,538.6	53,692.8	0.21	225	1,217.9	1,445	0.84
RC1	64,742.0	60,828.9	53,195.3	0.17	706	1,379.4	1,385	0.49
RC2	64,558.2	62,007.2	57,555.8	0.10	959	1,411.8	1,500	0.36

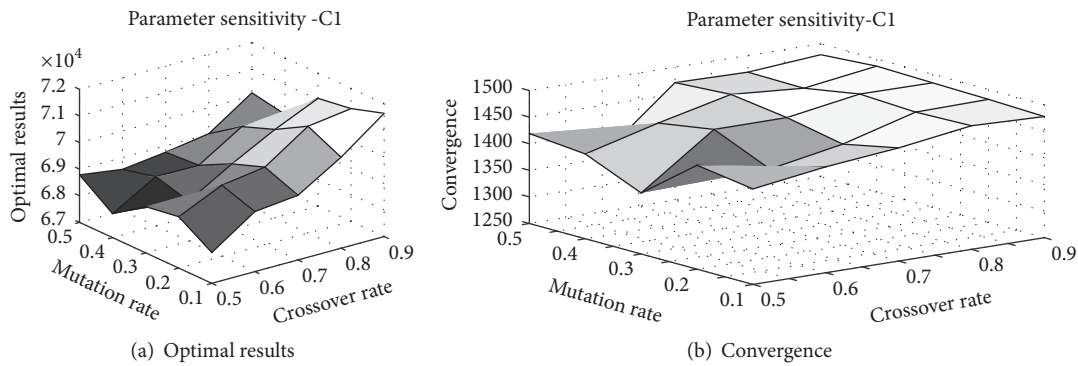


FIGURE 4: Results of dataset C1 under different crossover rates and mutation rates.

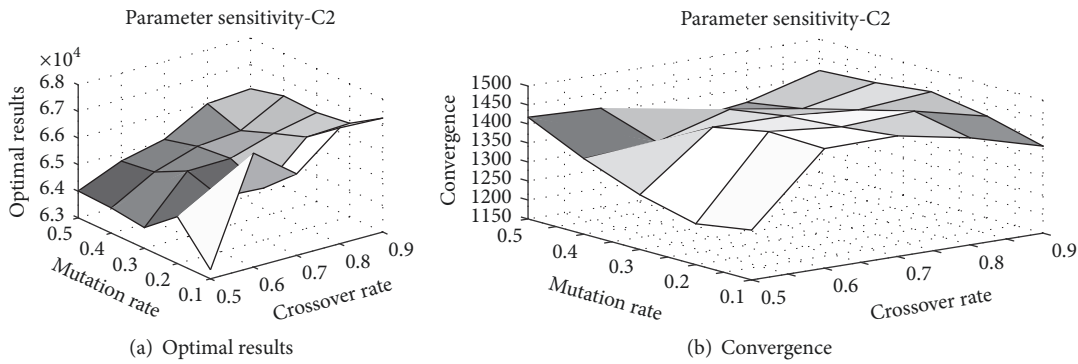


FIGURE 5: Results of dataset C2 under different crossover rates and mutation rates.

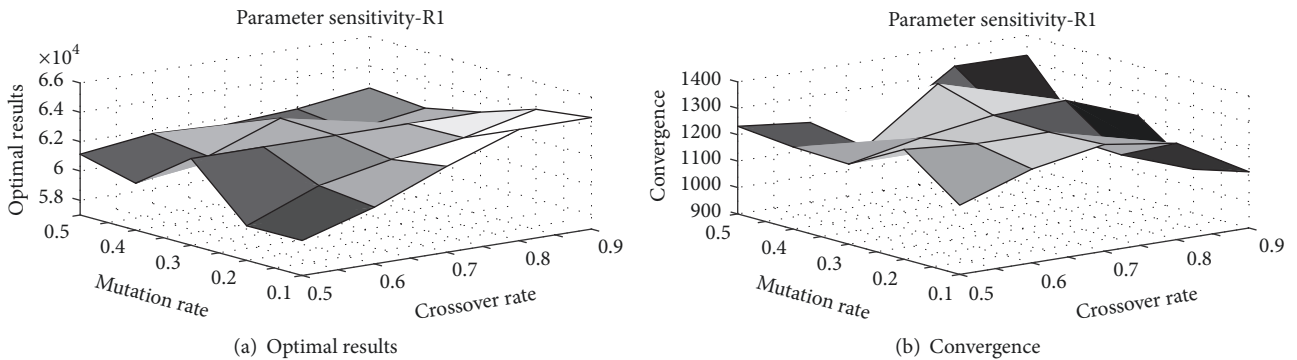


FIGURE 6: Results of dataset R1 under different crossover rates and mutation rates.

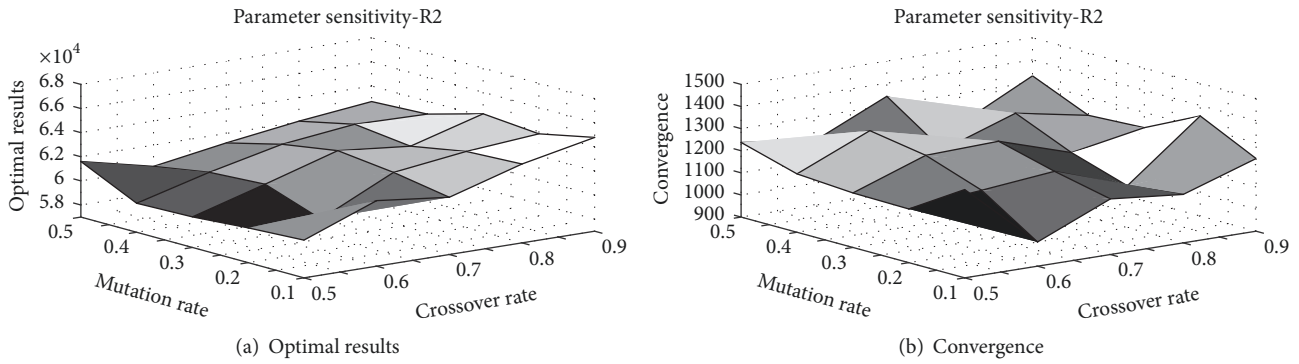


FIGURE 7: Results of dataset R2 under different crossover rates and mutation rates.

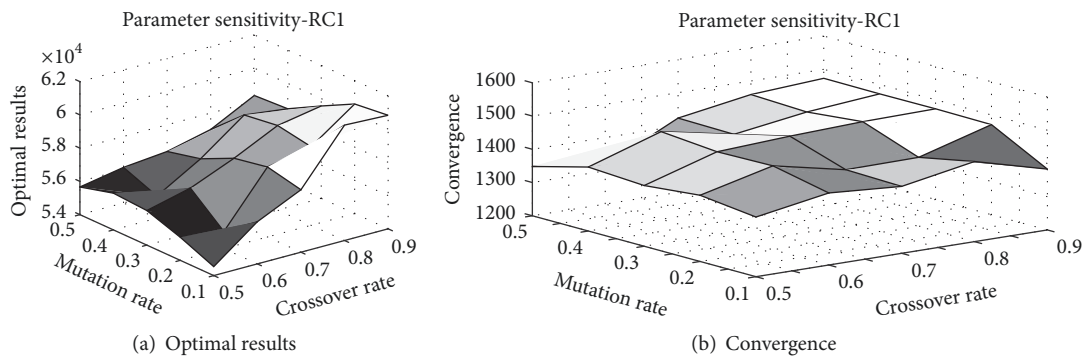


FIGURE 8: Results of dataset RC1 under different crossover rates and mutation rates.

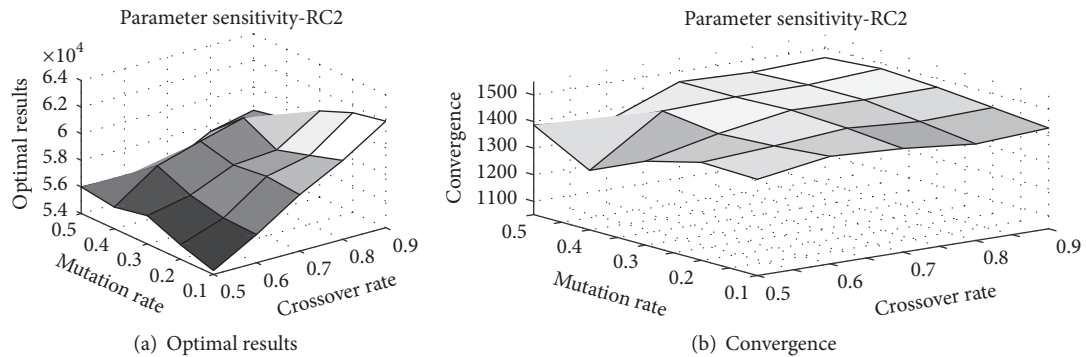


FIGURE 9: Results of dataset RC2 under different crossover rates and mutation rates.

increasing trend with varied crossover rates, but the changes of values with varied crossover rates are shocked. It can be concluded from Figure 6 that the value of the optimal results is increasing, while the convergence algebra first increases and then decreases with the increase of the crossover rate. Also, the variation of the mutation rate has no significant effect on the optimal results and convergence. Similarly, it can be seen that the variation of the mutation rate has no evident effect on the optimal results and convergence in Figure 7. As for the crossover rate, the values of the optimal results keep increasing; however, the values for convergence fluctuate.

For dataset RC1, it can be concluded from Figure 8 that the values of the optimal results increase with the increase of the crossover rate, while the change of the mutation rate shows a fluctuating trend. In addition, the influence of the crossover rate and mutation rate on convergence is not obvious. From Figure 9, the optimal results are increasing with the increase of the crossover rate, but the values for convergence do not change significantly. The variation of the mutation rate has no obvious effect on the optimal results and convergence.

In addition, to analyze the influence of the update number on the experimental results, the crossover and mutation rates

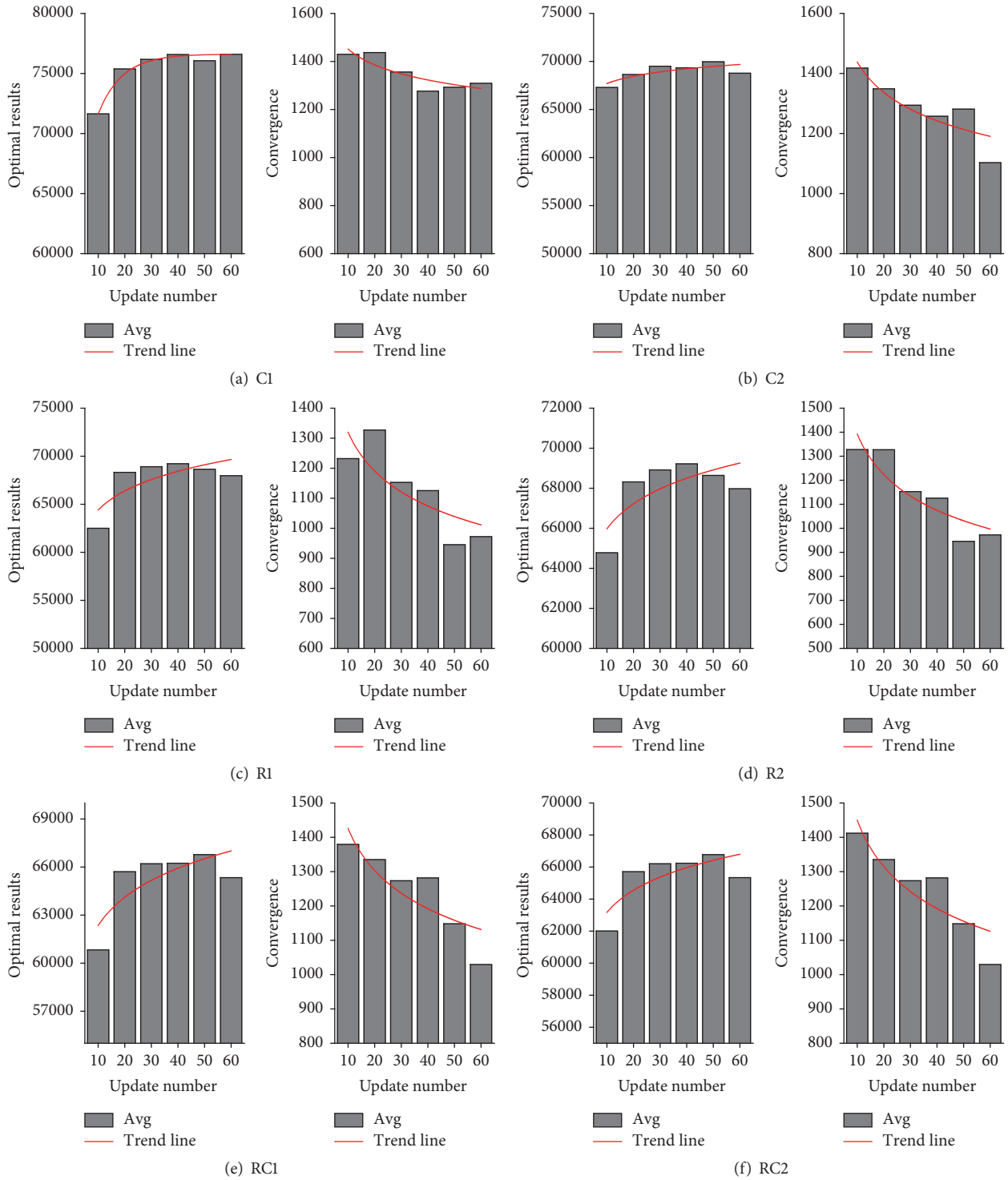


FIGURE 10: Results from six types of datasets under different update numbers.

were given to be 0.9 and 0.1, respectively, and the update number was set to 10, 20, 30, 40, 50, and 60. The algorithm was implemented 20 times on different datasets. The experimental results were shown as Figure 10.

From Figure 10, it can be concluded that the values of the optimal results increased with the increase of the update number for the six types of dataset, and the increasing trend is like a convex function. Conversely, with the increase of

TABLE 3: Experimental results of the six datasets.

Dataset Type	Optimal results				Convergence			
	Best	Avg	Worst	Gap	Best	Avg	Worst	Gap
C1	78712.7	75973.2	74063.7	0.034	713	1130.9	1473	0.515
C2	73740.8	70699.0	68710.6	0.041	1207	1113.9	1491	0.19
R1	71821.0	69357.2	68025.6	0.034	951	1180.6	1294	0.26
R2	71717.3	69399.2	65993.0	0.032	472	994	1050	0.55
RC1	78343.2	73995.8	69757.0	0.055	644	1079.1	1374	0.531
RC2	78062.6	74509.1	72038.7	0.045	520	1205.2	1427	0.635

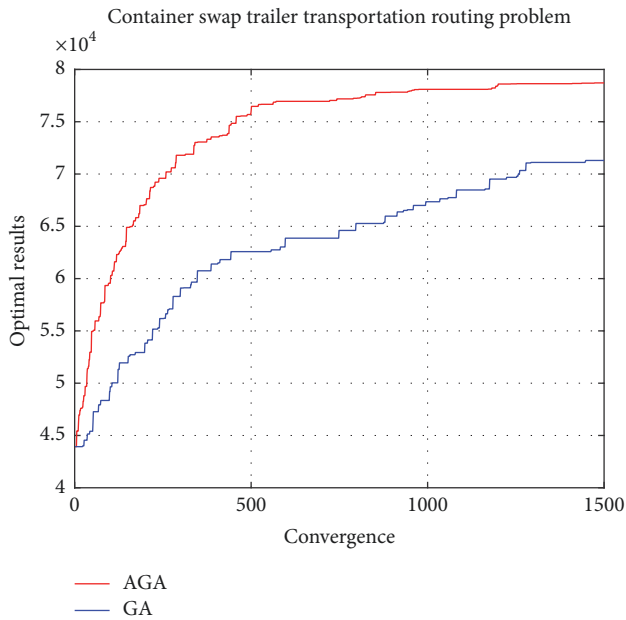


FIGURE 11: Changes of optimal results in GA and AGA.

the update number, the values of convergence decreased, and the decreasing trend was like a concave function. Therefore, there are some improvements in the optimal results and convergence of the genetic algorithm when increasing the value of the update number.

In the third type of experiments, the convergence curves of the genetic algorithm with parameter self-adaptive method (AGA) and basic genetic algorithm (GA) are given in Figure 11. It is obvious that the AGA had gradually converged when the algorithm reached 500 generations, but the GA had to iterate around 1300 generations. It shows that the AGA is superior to the GA in both the speed of optimization and convergence. And the concrete computational results on the six types of datasets are shown in Table 3.

Then, in order to test the effectiveness of the AGA in different original parameter settings, the $PCross_0$ was set to 0.5, 0.6, 0.7, 0.8, and 0.9, and the $PMutation_0$ was set to 0.1, 0.2, 0.3, 0.4, and 0.5 when the $Update_0$ was set to 50. The experiments for each combination of the two parameters

were carried out, and the experiments for each dataset were implemented 10 times. And the computational results were displayed as follows.

(1) Results of Dataset C1. See Figure 12.

(2) Results of Dataset C2. See Figure 13.

(3) Results of Dataset R1. See Figure 14.

(4) Results of Dataset R2. See Figure 15.

(5) Results of Dataset RC1. See Figure 16.

(6) Results of Dataset RC2. See Figure 17.

From the subgraphs (a) of Figures 12–17, it can be concluded that when the $PCross_0$ was set to 0.8 and the $PMutation_0$ was set to 0.4, the objective function value can converge faster than other parameter settings on the dataset C1 and the dataset C2. In the datasets R1, R2, RC1, and RC2, the objective function value performed better when the $PCross_0$ and the $PMutation_0$ were set to be 0.9 and 0.5. As for the computational results of convergence, the changing trend is similar under different parameter settings, although it sometimes shows little fluctuations.

5. Discussion

On the basis of the above experimental results, some discussions are given to further study the algorithm in this paper. From the results of the validity experiments, the model could be greatly solved by the genetic algorithm on the different types of datasets. It shows that our proposed algorithm has a certain generalization performance because our datasets are generated from a benchmark datasets. From the results of the parameter sensitivity experiments, the solving and converging speed of the basic genetic algorithm could obtain good effects by setting proper parameters (e.g., crossover rate, the mutation rate, the update number). And owing to the difference of datasets, the best parameters configuration may have divergence on the different types of datasets.

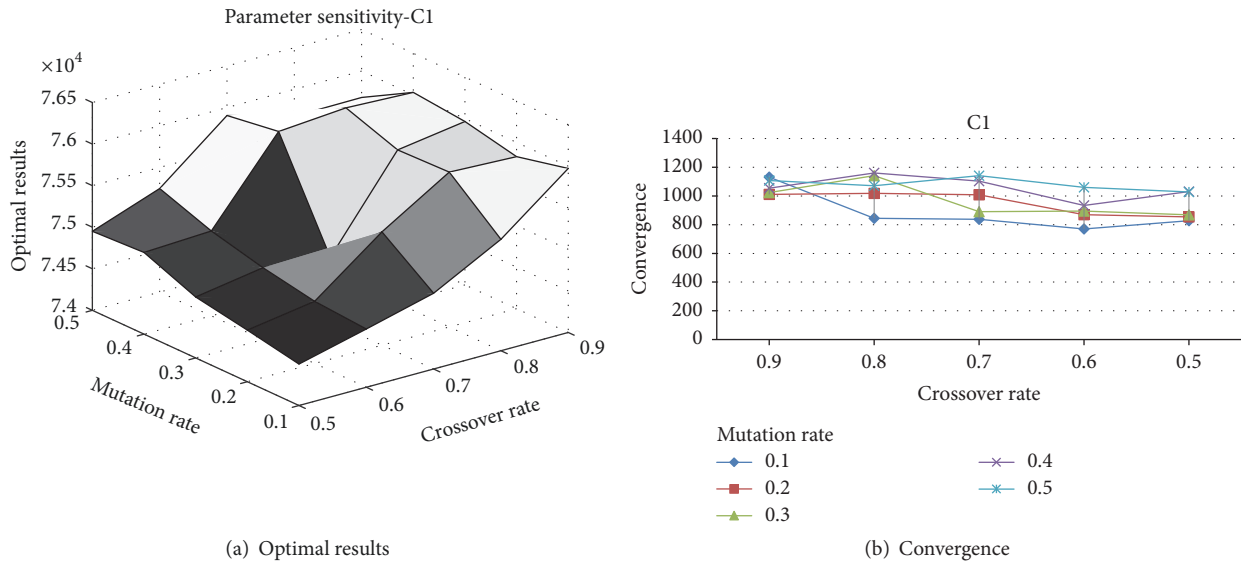


FIGURE 12: Results of dataset C1 under different crossover rates and mutation rates.

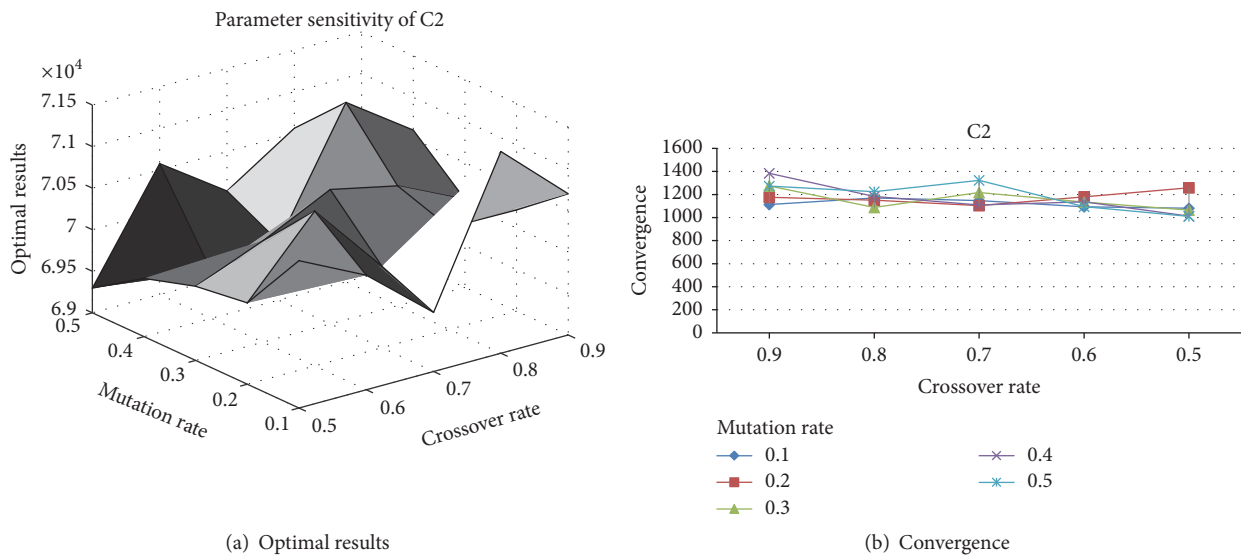
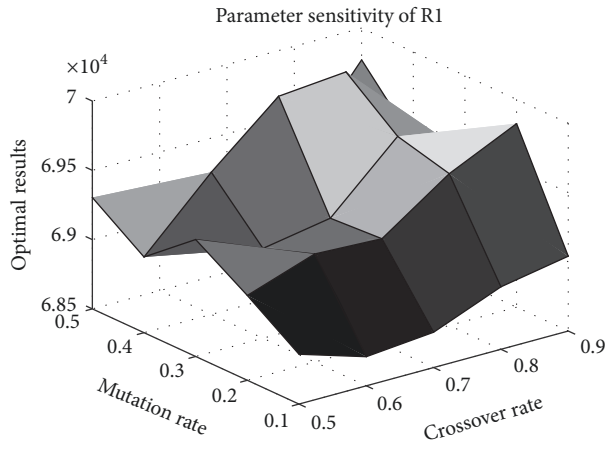


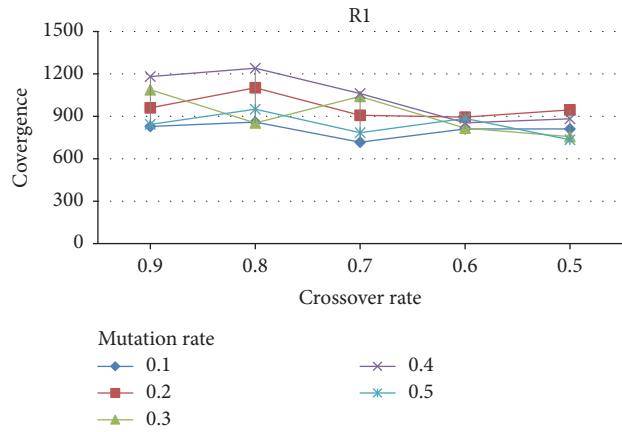
FIGURE 13: Results of dataset C2 under different crossover rates and mutation rates.

In order to pursue the better performance and convergence, a parameter self-adaptive adjustment policy was designed and embedded in the GA. Based on the above-mentioned best parameter configuration, the parameter self-adaptive adjustment policy is utilized to control the parameters to approach the best effect in the running process of algorithm. In the validity experiments, the AGA has shown its strong convergence and optimization speed. Besides, compared with the gap of optimal results between the AGA and GA in the Tables 2 and 3, it can be concluded that the gaps of optimal results are obviously reduced in the AGA. This indicates that the parameter self-adaptive policy can help us to improve the robustness of the genetic algorithm in this paper.

On the basis of the above, the improvement effect of the AGA on the six types of datasets are further discussed. The average values of optimal results solved by the AGA and the GA on six types of datasets were shown in Figure 18. It indicates that the optimal results and convergence on the six different types of datasets have been improved to some extent. Then, the improvement rates of optimal results and convergence were given in Figure 19. The improvement rates of optimal results and convergence on the datasets C1 and C2 are 8.35% and 7.56%, and 8.35% and 31.9%, respectively. On the datasets R1 and R2, the improvement rates of optimal results and convergence are 11.80% and 26.21%, and 12.46% and 22.11%, respectively. The corresponding values for datasets

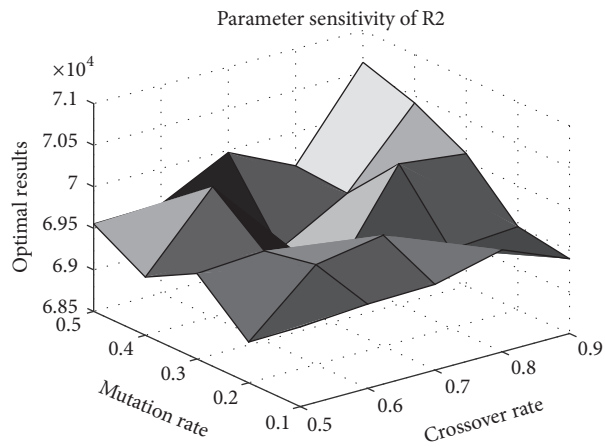


(a) Optimal results

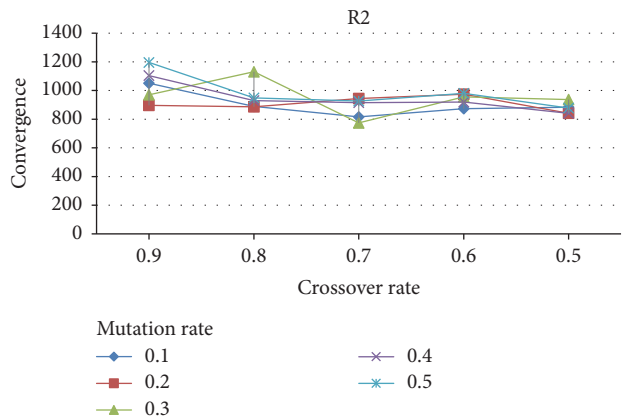


(b) Convergence

FIGURE 14: Results of dataset R1 under different crossover rates and mutation rates.



(a) Optimal results



(b) Convergence

FIGURE 15: Results of dataset R2 under different crossover rates and mutation rates.

RC1 and RC2 are 24.6% and 26.1%, and 26.9% and 28.7%, respectively. Thus, it shows that the improvement effect of AGA in datasets RC1 and RC2 are the best; however, the improvement effect in datasets C1 and C2 is not outstanding. The diverse improvement results may be caused by the difference in the nodes' position and time window among the datasets.

6. Conclusions

In this paper, we researched the swap trailer transportation routing problem to solve the container terminal cargo evacuation problem. We considered some constraints of

customer nodes, especially time-window and capacity constraints. The integer programming model used to solve the problem has been proven effective. The results of numerical experiments and parameter sensitivity experiments illustrate the feasibility of the genetic algorithm and the effects of different parameter configurations on different types of datasets. We conclude that it is difficult to take both the speed and quality of the solution into account under a single invariant parameter configuration. We summarized the best parameter configuration according to the analysis of the basic genetic algorithm. Based on this, we designed the genetic algorithm with parameter self-adaptive method (AGA) that can adjust the crossover rate, mutation rate, and update

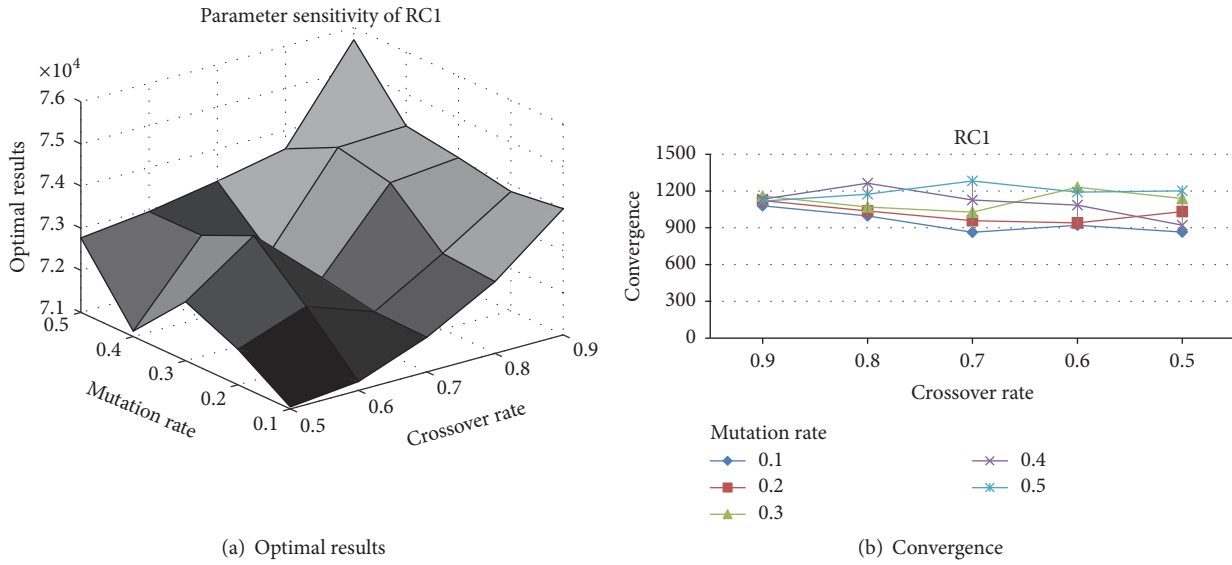


FIGURE 16: Results of dataset RC1 under different crossover rates and mutation rates.

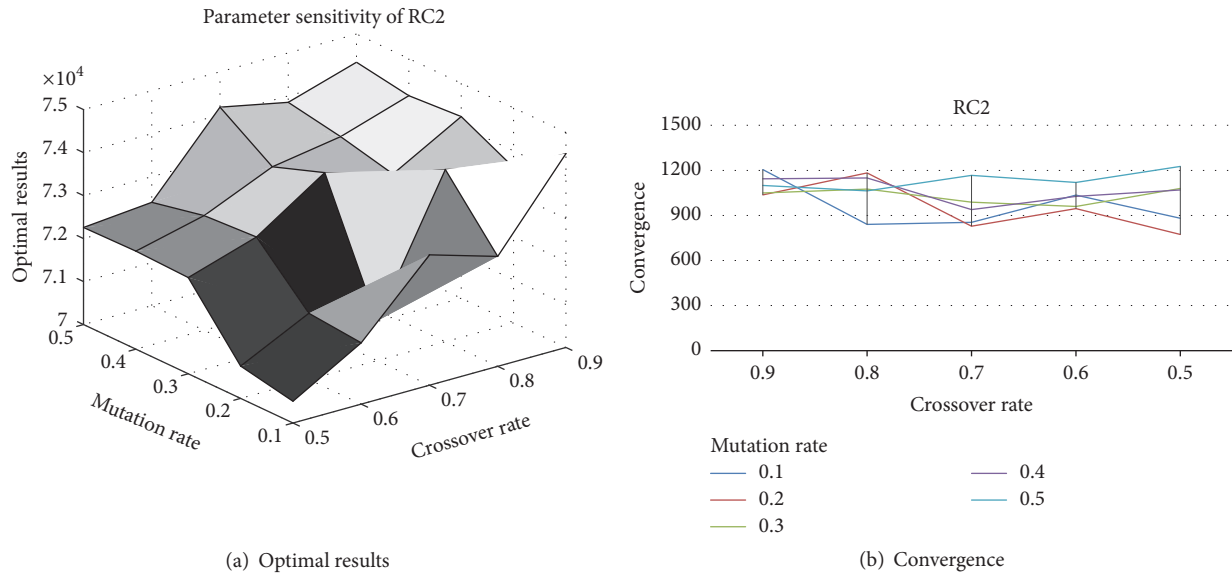


FIGURE 17: Results of dataset RC2 under different crossover rates and mutation rates.

number in the iteration process. In addition, the effectiveness of the AGA was tested with different types of datasets. The results showed that the genetic algorithm with parameter self-adaptive method could improve the quality and speed of solution with different types of datasets compared with the basic genetic algorithm, and the AGA has a better robustness.

This paper merely studied one aspect of the swap trailer transportation problem and may not be comprehensive or perfect. To facilitate the study of the container swap trailer transportation routing problem, we just considered the time-window and capacity constraints. There is a certain gap to bridge before achieving the solution of a more real and specific problem. More complicated problems and constraints must be discussed in real situations; for example, the demand

of the customer node may be uncertain, priority relations may exist between the customer nodes, and the congestion situation in the travel process may influence the travel time and the time-window constraints. In addition, there may be multiple routes in the swap trailer transportation routing problem, and it is necessary to consider routes according to the characteristics of different application scenarios. Therefore, our future work will consider more specific and practical problems, and we will expand the model we established in this paper to solve other feasible problems in different scenarios.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

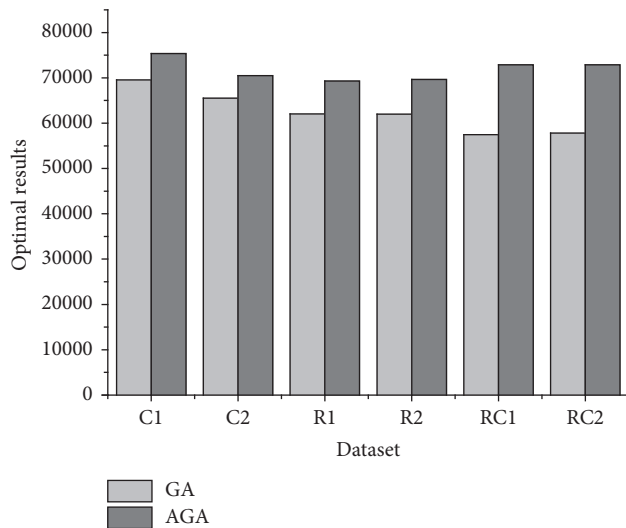


FIGURE 18: Optimal results of different datasets under different algorithms.

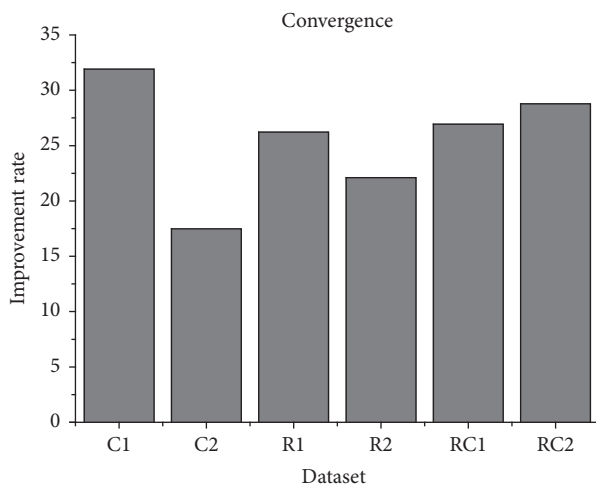


FIGURE 19: Improvement rates of different datasets.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (71472058, 71401048, 71671059, 71521001, 71690230, 71690235), the Anhui Provincial Natural Science Foundation (1508085MG140), and the Fundamental Research Funds for the Central Universities (JZ2016HG7B0726). The authors thank LetPub (<http://www.letpub.com>) for its linguistic assistance during the preparation of this manuscript.

References

- [1] M. Drexler, "Applications of the vehicle routing problem with trailers and transshipments," *European Journal of Operational Research*, vol. 227, no. 2, pp. 275–283, 2013.
- [2] S. Scheuerer, "A tabu search heuristic for the truck and trailer routing problem," *Computers & Operations Research*, vol. 33, no. 4, pp. 894–909, 2006.
- [3] S.-W. Lin, V. F. Yu, and S.-Y. Chou, "Solving the truck and trailer routing problem based on a simulated annealing heuristic," *Computers & Operations Research*, vol. 36, no. 5, pp. 1683–1692, 2009.
- [4] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco, "A GRASP with evolutionary path relinking for the truck and trailer routing problem," *Computers & Operations Research*, vol. 38, no. 9, pp. 1319–1334, 2011.
- [5] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco, "GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 780–794, 2010.
- [6] R. Cuda, G. Guastaroba, and M. G. Speranza, "A survey on two-echelon routing problems," *Computers & Operations Research*, vol. 55, pp. 185–199, 2015.
- [7] S.-W. Lin, V. F. Yu, and C.-C. Lu, "A simulated annealing heuristic for the truck and trailer routing problem with time windows," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15244–15252, 2011.
- [8] U. Derigs, M. Pullmann, and U. Vogel, "Truck and trailer routing—problems, heuristics and computational experience," *Computers & Operations Research*, vol. 40, no. 2, pp. 536–546, 2013.
- [9] S. Mirmohammadsadeghi and S. Ahmed, "Memetic heuristic approach for solving truck and trailer routing problems with stochastic demands and time windows," *Networks and Spatial Economics*, vol. 15, no. 4, pp. 1093–1115, 2015.
- [10] M. Batsyn and A. Ponomarenko, "Heuristic for a real-life truck and trailer routing problem ☆," *Procedia Computer Science*, vol. 31, pp. 778–792, 2014.
- [11] I. Torres, A. Rosete-Suárez, C. C. C. Alberto, and J. L. Verdegay, "Truck and trailer routing problem under fuzzy environment," in *Proceedings of the Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology*, 2015.
- [12] E. Nishimura, A. Imai, and S. Papadimitriou, "Yard trailer routing at a maritime container terminal," *Transportation Research Part E: Logistics and Transportation Review*, vol. 41, no. 1, pp. 53–76, 2005.
- [13] M. Vidović, G. Radivojević, and B. Raković, "Vehicle routing in containers pickup up and delivery processes," *Procedia-Social and Behavioral Sciences*, vol. 20, no. 20, pp. 335–343, 2011.
- [14] C. Gracia, B. Velazquez-Martí, and J. Estornell, "An application of the vehicle routing problem to biomass transportation," *Biosystems Engineering*, vol. 124, pp. 40–52, 2014.
- [15] A. E. Dooley, W. J. Parker, and H. T. Blair, "Modelling of transport costs and logistics for on-farm milk segregation in New Zealand dairying," *Computers and Electronics in Agriculture*, vol. 48, no. 2, pp. 75–91, 2005.
- [16] H. Li, L. Zhang, T. Lv, and X. Chang, "The two-echelon time-constrained vehicle routing problem in linehaul-delivery systems," *Transportation Research Part B: Methodological*, vol. 94, pp. 169–188, 2016.
- [17] O. Regnier-Coudert, J. McCall, M. Ayodele, and S. Anderson, "Truck and trailer scheduling in a real world, dynamic and heterogeneous context," *Transportation Research Part E: Logistics and Transportation Review*, vol. 93, pp. 389–408, 2016.
- [18] D. L. McWilliams, P. M. Stanfield, and C. D. Geiger, "Minimizing the completion time of the transfer operations in a central parcel consolidation terminal with unequal-batch-size inbound

- trailers," *Computers & Industrial Engineering*, vol. 54, no. 4, pp. 709–720, 2008.
- [19] C. Chen, Q. Zeng, and Z. Zhang, "An integrating scheduling model for mixed cross-operation in container terminals," *Transport*, vol. 27, no. 4, pp. 405–413, 2012.
- [20] L. M. San José-Revuelta, "A new adaptive genetic algorithm for fixed channel assignment," *Information Sciences*, vol. 177, no. 13, pp. 2655–2678, 2007.
- [21] Y. Xing, Z. Chen, J. Sun, and L. Hu, "An improved adaptive genetic algorithm for job-shop scheduling," in *Proceedings of the International Conference on Natural Computation IEEE Computer Society*, pp. 287–291, 2007.

