

Research Article

Model and Algorithm for Container Allocation Problem with Random Freight Demands in Sychromodal Transportation

Yan Xu, Chengxuan Cao, Bin Jia, and Guangzhi Zang

State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Chengxuan Cao; cxcao@bjtu.edu.cn

Received 14 May 2015; Revised 28 September 2015; Accepted 4 October 2015

Academic Editor: Babak Shotorban

Copyright © 2015 Yan Xu 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.

This paper aims to investigate container allocation problem with random freight demands in sychromodal transportation network from container carriers' perspective. Firstly, the problem is formulated as a stochastic integer programming model where the overall objective is to determine a container capacity allocation plan at operational level, so that the expected total transportation profit is maximized. Furthermore, by integrating simulated annealing with genetic algorithm, a problem-oriented hybrid algorithm with a novel gene encode method is designed to solve the optimization model. Some numerical experiments are carried out to demonstrate the effectiveness and efficiency of the proposed model and algorithm.

1. Introduction

As stated in [1], sychromodal freight transportation involves a structured, efficient, and synchronized combination of two or more transportation modes. Through sychromodal transportation, the carriers or customers select independently at any time the best transportation mode based on the operational circumstances and/or customer requirements. While multimodal freight transportation is defined as the transportation of goods by a sequence of at least two different modes of transportation, intermodal freight transportation is defined as a particular type of multimodal transportation where the load is transported from an origin to a destination in one and the same intermodal transportation unit. There obviously exist some differences in sychromodal freight transportation, multimodal freight transportation, and intermodal freight transportation as stated above. Besides, resource management, which is on how and when to optimally utilize the limited available resources, is a typical problem at operational level in freight transportation network. Some examples of these resources are containers, vehicles (e.g., planes and trucks), trailers, rail cars, locomotives, equipment, crew, power, and so forth. Once a resource is allocated to an activity, it is no longer available for certain duration.

Containerization may improve safety, reduction of handling costs, standardization, and accessibility to multiple modes of transportation [2]. It plays a more and more important role in freight transportation network. While container central station is a large container handling station set up in accordance with the construction of modern logistics hub requirements, it is also a terminal station of loading and unloading containerized freight to maritime, hinterland, or air transportation. At present, many carriers have to determine the most profitable allocation of available capacity for utilizing limited container more efficiently. As pointed in [3], it is still a challenging task to optimize transportation profit by optimizing resource management. These reasons inspire us to research container capacity allocation problem. Since containerization transportation system is a considerable complex system, including container allocation, reposition, and retention, it is sometimes difficult to optimize carriers' transportation profit by determining the scheme of container capacity allocation for container carries in strategic stage [4]. Therefore, we mainly focus on container capacity allocation problem for carriers at operational level in this paper, that is, how to allocate the limited container to maximize transportation profit in sychromodal freight transportation network.

So far, many scholars have researched on capacity problem in transportation network. Bostel and Dejax [5] defined

the capacity allocation problem as allocating capacity to match each demand. Demirag and Swann [6] considered capacity allocation problem for central headquarters by formulating it as a mixed integer program, and some simple heuristic algorithms are present to solve it. Bell et al. [7] focused on container assignment problem by taking empty containers flow into consideration and introduced a frequency based model for a maritime network with port rotations services. Arnone et al. [8] extended Bell et al.'s frequency based approach to the multimodal network, by adding services capacity constraints, loading/unloading, and duty check times while empty containers repositioning was not treated. Besides, Cao et al. [9] formulated a stochastic integer programming model for the capacity allocation problem taking into account supply matching with random demands in rail container transportation network, excluding the transport by ship or truck (significantly reducing the complexity of the problem). In addition, they also proposed a large-scale 0-1 integer programming model, subjected to limited container capacities, to maximize the total transportation profit generated by all freight bookings would be accepted in a multiperiod planning horizon [10]. Similarly, Jacko [11] considered the optimal dynamic allocation problem on limited knapsack for perishable or nonperishable items and formulated it in the framework of Markov decision processes.

In this paper, we extend the container capacity allocation problem introduced by Cao et al. [9] to a synchromodal transportation network, including rail, river, and road transportation mode, while the empty containers repositioning problem is out of scope. Since the freight demands for container transportation are uncertain in synchromodal transportation network, this paper provides a stochastic integer programming model for the container capacity allocation problem with a single origin and multiple destinations. The aim of the model is to maximize the total transportation profit by allocating container capacity appropriately in planning horizon. Furthermore, a problem-oriented hybrid algorithm with a novel gene encoding method is proposed to address the large-scale stochastic integer programming formulation.

The rest of the paper is organized as follows. The container capacity allocation problem with random freight demands in synchromodal transportation network is formulated as a stochastic integer programming model in Section 2. According to the characteristics of the proposed model, a problem-oriented hybrid algorithm is designed to solve it in Section 3. And then, we perform some examples to indicate the efficiency of the proposed algorithm in Section 4. Section 5 concludes the paper with a summary of the capacity allocation problem with random demands.

2. Mathematical Formulation

This section aims to develop a mathematical model for container capacity allocation problem with random freight demands in synchromodal transportation network.

As mentioned above, the container capacity allocation decision is not only a tactical level decision, but also an operational level optimization decision when more freight

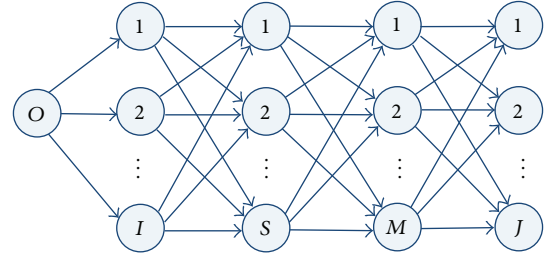


FIGURE 1: Network structure of the container capacity allocation problem.

demands or other pieces of information are available. The overall objective is to determine the limited container capacity allocation, to maximize the potential profit for container carriers. In this paper, we assume that the container capacity allocation plan has to be made with limited knowledge of future scenarios of parameters. Therefore, to incorporate the uncertainty in parameters, these parameters can be realized as one of S scenarios and the possible realizations probability of scenario s is denoted by P^s , as the scenario-based methods of dealing with random variables in [12].

The mathematical model is subjected to several technical constraints. The obvious constraints are that volume of freight should not exceed the total available capacity and the crew, truck, locomotive, and ship at the container central station are assumed sufficient. Additionally, there is at most one container transportation line to each destination for reducing complexity of the loading work. We also assume that the container transport lines of a certain transportation mode, which have not enough demands, would be abandoned.

In order to present the mathematical formulation of the container capacity allocation problem clearly, some notations are interpreted in Table 1.

As stated in Section 1, a container capacity allocation problem is mainly for the carriers who are in the container central station. Containers can be shipped to the place where freight demands occurred, but the available container capacity to each demand is certain after transportation planning is determined. Therefore, there is only one origin in order to simplify. While I freight types will be received at the origin to be shipped to J destinations during a planning horizon and M transportation modes, implying the means of transportation, are available between origin and each destination, besides, S future scenarios indicating the uncertain demands are under consideration. Figure 1 shows the network structure of the container capacity allocation with I freight types, S scenarios, M transportation modes, one origin, and J destinations.

Consequently, the container capacity allocation problem with random freight demands in synchromodal transportation network can be formulated as the following two-stage stochastic integer programming model (P1):

$$\text{Maximize } Z = \sum_{m \in \bar{M}} \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} r_{mij} \cdot x_{mij} - \sum_{s \in \bar{S}} P^s Q^s(x_{mij}) \quad (1)$$

$$\text{subject to } \sum_{m \in \bar{M}} \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} x_{mij} \leq H, \quad (2)$$

TABLE 1: List of notations.

Symbol	Description
(1) <i>Index sets</i>	
\bar{I}	Set of freight types $\{1, 2, \dots, i, \dots, I\}$.
\bar{J}	Set of destinations $\{1, 2, \dots, j, \dots, J\}$.
\bar{M}	Set of transportation modes $\{1, 2, \dots, m, \dots, M\}$.
\bar{S}	Set of scenarios $\{1, 2, \dots, s, \dots, S\}$.
\bar{K}_m	Set of freight transported by mode m .
(2) <i>Random data</i>	
D_{ij}^s	Demands for transportation of freight type i to destination j under scenario s .
α_{mij}^s	Penalty per unit shortage in supplied capacity for freight type i transported by mode m to destination j under scenario s .
β_{mij}^s	Penalty per unit overage in supplied capacity for freight type i transported by mode m to destination j under scenario s .
(3) <i>Deterministic parameters</i>	
r_{mij}	Profit for carrying a unit freight of type i to destination j by transportation mode m .
C_{mj}	Requested bottom supply of capacity for transportation mode m to destination j .
H	Amount of available container capacity at the origin O .
(4) <i>Decision variables</i>	
x_{mij}	Container capacity for transportation mode m of freight type i to destination j , which will be supplied for shipping cargoes in a planning horizon. In this paper, we assume that the unit of capacity is TEU (i.e., twenty-foot equivalent unit).
y_{mij}^s	Amount of shortage capacity for freight type i transported by mode m to destination j under scenario s .
z_{mij}^s	Amount of overage in capacity for freight type i transported by mode m to destination j under scenario s .
u_{mj}	Binary variable: that is, $u_{mj} = 1$, transportation mode m to destination j will be operated; $u_{mj} = 0$, otherwise.

$$\sum_{i \in \bar{I}} x_{mij} \geq C_{mj} u_{mj}, \quad \forall m \in \bar{M}, \quad \forall j \in \bar{J}, \quad (3) \quad u_{mj} \in \{0, 1\}, \quad \forall m \in \bar{M}, \quad \forall j \in \bar{J}, \quad (5)$$

$$\sum_{m \in \bar{M}} u_{mj} \leq 1, \quad \forall j \in \bar{J}, \quad (4) \quad x_{mij} \in \mathbb{Z}^+, \quad \forall (m, i, j) \in \bar{K}_m, \quad (6)$$

where, for all s ,

$$Q^s(x_{mij}) = \text{Minimize} \quad \sum_{m \in \bar{M}} \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} (\alpha_{mij}^s y_{mij}^s + r_{mij} z_{mij}^s + \beta_{mij}^s z_{mij}^s) \quad (7)$$

$$\text{subject to} \quad y_{mij}^s - z_{mij}^s = D_{ij}^s \cdot u_{mj} - x_{mij}, \quad \forall (i, j) \in \bar{K}, \quad \forall (m, i, j) \in \bar{K}_m, \quad s \in \bar{S}, \quad (8)$$

$$y_{mij}^s, z_{mij}^s \in \mathbb{Z}^+, \quad s \in \bar{S}, \quad \forall (m, i, j) \in \bar{K}_m, \quad (9)$$

where \mathbb{Z}^+ denotes the set of nonnegative integers. The objective function of first stage program (1) maximizes the total transportation profit by allocating capacity reasonably in operational planning horizon. The model is subjected to the following constraints. As the available capacity is limited, constraint (2) imposes that the aggregate supply of container capacity cannot exceed the available capacity at the origin O in the planning horizon. Constraint (3) assures the actual container capacity supplied by transportation mode m to destination j cannot be less than a certain bottom value, which may guarantee maximizing the utilization of limited resource. If less, cancel the transportation line of

mode m to destination j ; that is, $u_{mj} = 0$. Constraint (4) assures that there is at most one transportation mode to destination j during the planning horizon to reduce the complexity of loading work. Constraint (5) indicates that the decision variable u_{mj} is a binary variable, meaning that transportation mode m to destination j will be operated or not. The objective function of second stage objective function (7) minimizes the overall penalties for overage and shortage in capacity under scenario s , excepted item $r_{mij} z_{mij}^s$. Because z_{mij}^s indicates an overage in supplied capacity, it means that the container carriers lose the opportunities for making more transportation profit from the unutilized containers, that is,

$r_{mij}z_{mij}^s$. Thus, the total profit generated by carrying freight of type i to destination j by transportation mode m is $r_{mij}x_{mij} - r_{mij}z_{mij}^s$ under scenario s , whereas the profit deviation $r_{mij}z_{mij}^s$ is moved to (7). Constraint (8) implies the amount of overage or shortage in the supplied capacity under scenario s : for example, $u_{mj} = 1$; if $D_{ij}^s > x_{mij}$, it means the supplied capacity is in shortage, then $y_{mij}^s = D_{ij}^s - x_{mij}$, and $z_{mij}^s = 0$; if $D_{ij}^s < x_{mij}$, it means the supplied capacity is in overage, $z_{mij}^s = x_{mij} - D_{ij}^s$, and $y_{mij}^s = 0$; and if $D_{ij}^s = x_{mij}$, it means the supplied capacity is just right, $z_{mij}^s = y_{mij}^s = 0$. While $u_{mj} = 0$,

$$\begin{aligned}
\text{Maximize } Z &= \sum_{m \in \bar{M}} \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} r_{mij} \cdot x_{mij} - \sum_{s \in \bar{S}} P^s Q^s (x_{mij}) \\
&= \sum_{m \in \bar{M}} \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} r_{mij} \cdot x_{mij} - \sum_{s \in \bar{S}} P^s \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} \left(\alpha_{mij}^s (D_{ij}^s \cdot u_{mj} - x_{mij})^+ + (r_{mij} + \beta_{mij}^s) (x_{mij} - D_{ij}^s \cdot u_{mj})^+ \right) \quad (10) \\
&= \sum_{m \in \bar{M}} \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} r_{mij} \cdot x_{mij} - \sum_{s \in \bar{S}} P^s \sum_{m \in \bar{M}} \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} (\alpha_{mij}^s y_{mij}^s + r_{mij} z_{mij}^s + \beta_{mij}^s z_{mij}^s).
\end{aligned}$$

The constraints in program (P2) consist of constraint (2)–constraint (9), except the second stage objective function (7), for the objective functions of the two-stage objective functions (1) and (7) are integrated in the objective function (10).

To the best of our knowledge, there is no effective method to deal with the large-scale stochastic integer programming model (P2) even when there is only one scenario in the problem [12]. Moreover, the large-scale stochastic integer programming model (P2) is NP-hard problem [13]. It is very unlikely to devise a polynomial-time algorithm for it. Before presenting an effective heuristic algorithm, we analyze some special properties of the model (P2) below.

Proposition 1. *There is at least one feasible solution for the SIPP (P2).*

Proof. Actually, $u_{mj} = 0$, $x_{mij} = 0$, $y_{mij}^s = z_{mij}^s = 0$, $\forall (m, i, j) \in \bar{K}_m$, $s \in \bar{S}$ is a feasible solution for (P2), and $Z = 0$. \square

Proposition 2. *The large-scale SIPP (P2) has a finite optimal solution.*

Proof. Let $\bar{R} = \max_{(m,i,j) \in \bar{K}_m} r_{mij}$; for any feasible solution of large-scale SIPP (P2), we have

$$\begin{aligned}
Z &= \sum_{m \in \bar{M}} \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} r_{mij} \cdot x_{mij} \\
&\quad - \sum_{s \in \bar{S}} P^s \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} (\alpha_{mij}^s y_{mij}^s + r_{mij} z_{mij}^s + \beta_{mij}^s z_{mij}^s) \quad (11) \\
&\leq \sum_{m \in \bar{M}} \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} r_{mij} \cdot x_{mij} \leq \bar{R}H.
\end{aligned}$$

constraint (8) is meaningless. As the definition of shortage in capacity of y_{mij}^s and overage in capacity of z_{mij}^s in constraint (9), $y_{mij}^s = (D_{ij}^s \cdot u_{mj} - x_{mij})^+$, $z_{mij}^s = (x_{mij} - D_{ij}^s \cdot u_{mj})^+$, where $(a)^+$ denotes $\max(a, 0)$ for the real number a .

Since it is not easy to solve above two-stage stochastic integer program (P1), we reformulate the two-stage programming (P1) into one-stage stochastic integer programming problem (OSIPP) (P2) by integrating objective functions. The objective function of program (P2) is as the following:

So, the optimum of objective function of the large-scale SIPP (P2) has a finite upper bound $\bar{R}H$. \square

3. Simulated-Annealing-Based Genetic Algorithms (SAGA) for Capacity Allocation Problem

As stated above, for a NP-hard problem, it is widely accepted that there is no algorithm that can solve the problem in polynomial time. Generally, there are tens of thousands of integer decision variables in the real world capacity allocation problem with random demands for the container carrier and we cannot solve it by any known methods in reasonable time. So, we attempt to design a problem-oriented algorithm to solve the proposed mathematical formulation (P2).

The genetic algorithm (GA) was first introduced by Holland in 1975 [14]. It is a kind of global search heuristic used to find exact or high quality solutions for various optimization problems [15–17]. However, simulated annealing (SA), firstly introduced by Kirkpatrick et al. in 1983 to deal with combination optimization problem [18], is effective in local optimum search [19]. At present, the focus of algorithm research has changed from being rather algorithm-oriented to being more problem-oriented [20–23]. Therefore, according to the characteristic of the proposed model, a hybrid algorithm, called simulated-annealing-based genetic algorithms (SAGA) in this paper, is designed for it by uniting the respective advantages of GA and SA. The structure of SAGA is as illustrated in Figure 2.

As illustrated in Figure 2, the genetic operators are responsible for global research, mainly to produce traditional population A, while the simulated annealing operator is responsible for local research, dedicating to produce elite population B, where Pa is the splitting factor, which governs

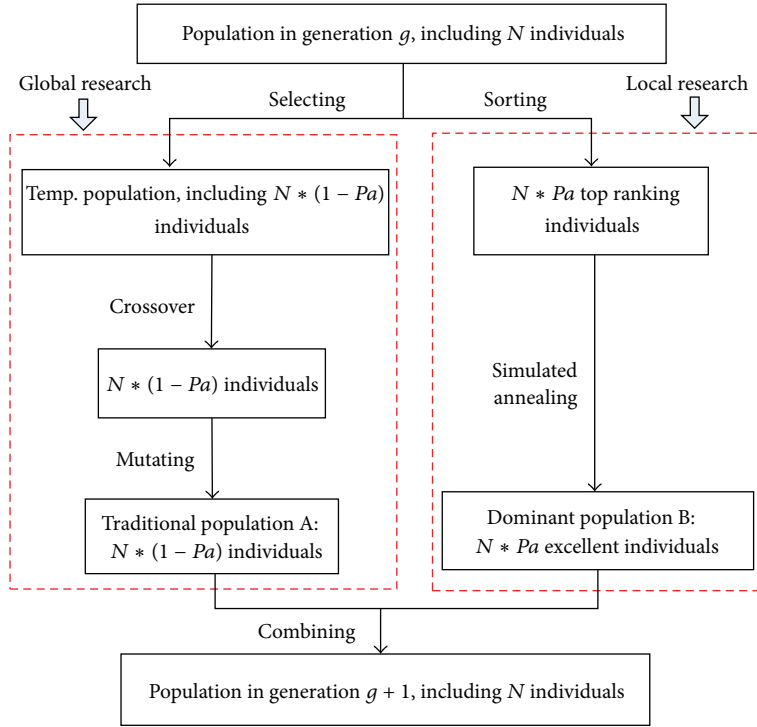


FIGURE 2: The structure of SAGA.

the size of populations A and B. In early evolution, populations mainly rely on genetic operators for global search. At this time, the size of population A would be larger with smaller Pa . In later period of evolution, populations primarily depend on the individual advantage to search solution near optimal solutions. In this case, the size of population B would be larger with larger Pa . In this paper, the regulation process is divided into two stages: that is, $P = 0.4Pa$ when $g \in (0, G_{\max}/2]$; $P = Pa$ when $g \in (G_{\max}/2, G_{\max}]$, where Pa is the maximum per centum of populations to dominant populations, G_{\max} is the maximum iterations, and g is the current iterations.

3.1. Coding and Decoding. Since most real life optimization problems are with multiple constraints, it is not easy to find a feasible solution. The methodologies discussed in previous literatures mostly relied on some extra penalty functions to revise the infeasible solutions. But, it is not easy to revise the infeasible solutions for most real world problem [13]. As the genes on chromosome are the deciding attributes for individual genetic trait, it is attractive to design a novel method of coding/encoding genes on chromosome without revision of infeasible solutions.

Specifically, the genes on the chromosomes are encoded by uniformly distributed random number between 0 and 1 in this paper. Then, the individuals directly participate in crossover, mutation, and simulated annealing operator. Before selection, the genes on each chromosome are decoded into a feasible solution according to the constraints listed in (P2). The decoding process is illustrated in Figure 3, where S_t denotes a certain chromosome t , R_u is the utilization

ratio of the total available capacity H , and X_t is the phenotype of chromosome S_t , indicating a feasible solution. The pseudocode of decoding process is presented in Algorithm 1, where S_t^l ($l \in \bar{L}$) denotes the l th gene on chromosome S_t , \bar{L} is a set of discrete locations $\{1, 2, \dots, l, \dots, L\}$ and $L = M \times J + 1 + M \times J \times I$, and M , I , and J represent the maximum number of transportation mode, freight type, and destination, respectively, as stated in Table 1. C^n ($n \in \bar{N}$) denotes the requested bottom supply of capacity in set \bar{C} , which is generated according to C_{mj} ; that is, $C_{11} = C^1$, $C_{12} = C^2, \dots, C_{MJ} = C^N$, where \bar{N} is a set $\{1, 2, \dots, n, \dots, N\}$, $N = M \times J$. X_t^l denotes decision variables u_{mj} or x_{mij} according to the specific location on the string of integer; that is, if $l < N$, X_t^l denotes decision variables u_{mj} ; else if $l > N + 1$, X_t^l denotes decision variables x_{mij} .

Algorithm 1. Input parameters: H , S_t , N , \bar{C} , M , I , and J .

Step 1 (handling constraint (2) and (3)). Set total utilization of container capacity $V_0 = \text{ceil}(H \cdot R_U)$, $a_1 = S_t^1 + S_t^2 + \dots + S_t^N$. For $l = 1 : N$, set $b_1^l = \sum_{i=1}^{i=l} S_t^i / a_1$, $AV_t^l = \text{ceil}(b_1^l \cdot V_0)$; then $x_t^l = AV_t^l - AV_t^{l-1}$; if $x_t^l \geq C^l$, set $X_t^l = 1$. Otherwise, $X_t^l = 0$, where x_t^l denotes the total allocating container capacity to a certain destination shipped by a certain transportation mode, corresponding to $x_t^l = \sum_{i \in \bar{I}} x_{mij}$; X_t^l corresponds to a certain decision variable u_{mj} .

Step 2 (handling constraint (4)). For $j = 1 : J$, set $U^j = S_t^j + S_t^{j+1} + \dots + S_t^{(M-1)j+1}$, if $U^j > 1$, meaning that there

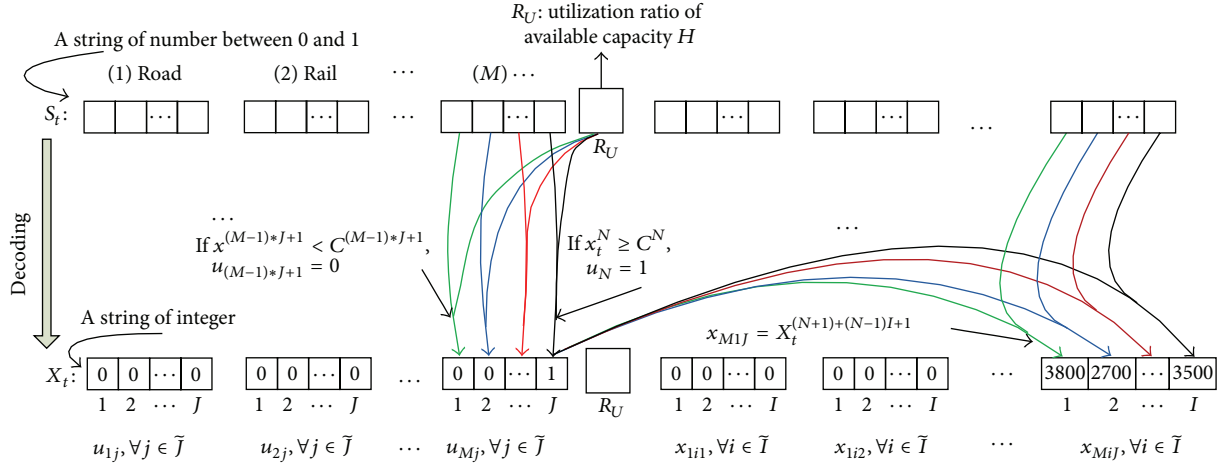


FIGURE 3: The mapping of decoding.

is more than one transportation mode to destination j , which violates constraint (4) in (P2). Thus, set $X_t^{l_0} = 0$, $l_0 \in \{j, J + j, \dots, (M - 1) \cdot J + j\}$. Find a correct location $l_1 = \text{Location_max}(S_t^j, S_t^{J+j}, \dots, S_t^{(M-1) \cdot J+j})$, which denotes the location of the maximum value in set $\{S_t^j, S_t^{J+j}, \dots, S_t^{(M-1) \cdot J+j}\}$ on chromosome S_t ; set $X_t^{l_1} = 1$.

Step 3 (allocating the container capacity)

Step 3.1. Set $l = 1, a_2 = S_t^1 \cdot X_t^1 + S_t^2 \cdot X_t^2 + \dots + S_t^N \cdot X_t^N$. For $n = 1 : N, b_2^n = \sum_{i=1}^{i=n} S_t^i \cdot X_t^i / a_2, AV_t^n = \text{ceil}(b_2^n \cdot V_0)$, the final container capacity allocated to l th location $V_0^n = AV_t^n - AV_t^{n-1}$.

Step 3.2. If $V_0^l > 0$, go to Step 3.3; else, go to Step 3.4.

Step 3.3. Allocate capacity to each item. Set $a_3 = S_t^{(N+1)+(l-1)I+1} + S_t^{(N+1)+(l-1)I+2} + \dots + S_t^{(N+1)+lI}$. For $k = 1 : I, b_3^k = \sum_{i=1}^{i=k} S_t^{(N+1)+(l-1)I+k} / a_3, AV_t^k = \text{ceil}(b_3^k \cdot V_0^l)$, then $X_t^{(N+1)+(l-1)I+k} = AV_t^k - AV_t^{k-1}$, where $X_t^{(N+1)+(l-1)I+k}$ corresponds to a certain decision variable x_{mij} .

Step 3.4. Set $l = l + 1$. If $l > N$, stop; output the results. Otherwise, return to Step 3.2, where $b_{1,2,3}^l$ is accumulated allocation proportion, $\text{ceil}(x)$ rounds x to the nearest integer greater than or equal to x , AV_t^l, AV_t^k represents, respectively, the accumulated allocation capacity to a certain destination shipped by a certain transportation mode and to a certain item, $AV_t^0 = 0, AV_t^0 = 0$.

According to Algorithm 1, all chromosomes can be decoded into feasible solutions satisfying all of the constraints without revision or elimination of invalid solution. For instance, for $x_t^{(M-1)J+1} < C^{(M-1)J+1}, u_{(M-1)J+1} = 0$; contrarily, for $x_t^N \geq C^N, u_N = 1$. Since the allocated container capacity to destination J by transportation mode M is not empty, that is, $u_N = 1$, it needs to further allocate the container capacity to each item; that is, $x_{M1J} = X_t^{(N+1)+(N-1)I+1} = 3800$,

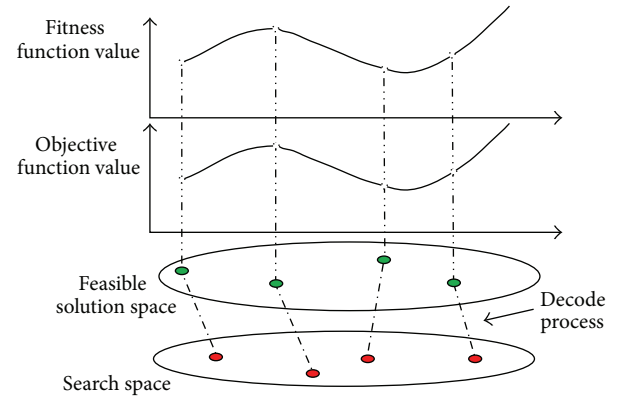


FIGURE 4: The relationship of search space, feasible solution space, the objective function value, and the fitness function value.

$x_{M2J} = 2700, \dots, x_{MIJ} = 3500$, as shown in Figure 3. Additionally, the encoding method can increase the flexibility in crossover, mutation, and simulated annealing operators since there is no concern about whether the potential solutions are feasible or not.

Furthermore, it seems quite natural to define its objective value as its fitness function value for maximum optimal problem; that is, $f(S) = Z(x)$. Thus, in this paper, the relationships of search space, solution space, the objective function value, and the value of fitness function are illustrated in Figure 4.

3.2. Operators

3.2.1. Reproduction and Selection. Up to now, various reproduction methods have been proposed and considered, for example, ranking selection, elitist ranking selection, expected value selection, elitist expected value selection, roulette wheel selection, and elitist roulette wheel selection [24]. In this paper, we further develop the elitism preserve strategy, called

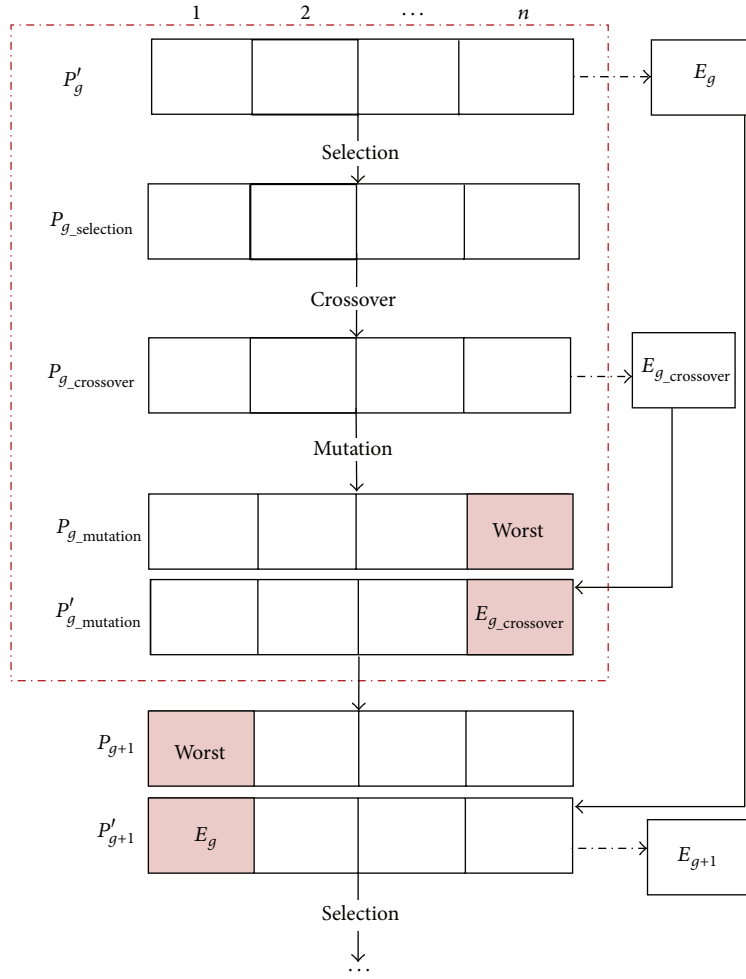


FIGURE 5: The process of DES.

developed elitism strategy (DES), and then adopt improved elitism roulette wheel selection to reproduce.

In the existing literature, elitism preserve strategy is preserving the elite and directly putting it into the population of following generation. Obviously, it ignores that elite may generate a more excellent individual in crossover operator which also disappears easily in mutation operator. Therefore, the new elitism strategy is improved by preserving the newborn elite $E_{g_crossover}$ which may be generated in crossover operator and then replacing the worst individual after mutation operator. The DES process is illustrated in Figure 5, where P'_g/P'_{g+1} , $P_{g_selection}$, $P_{g_crossover}$, and $P_{g_mutation}$ present the current g th/ $(g + 1)$ th populations, the current populations after selection, crossover, and mutation operator, respectively. $P'_{g_mutation}$ presents the current populations after replacing the worst individual by the newborn elite, P_{g+1} presents the $(g + 1)$ th populations after genetic operators, and E_g/E_{g+1} represents the elite in the g th/ $(g + 1)$ th populations. From Figure 5, it can be easily seen that the elites both in initial populations and in crossover populations can not only be preserved but also have chance to produce a preferable individual in DES.

3.2.2. Crossover. In traditional GA, the crossover position is mostly fixed. Employing the same crossover position may be not conducive to evolution, especially in later period of evolution that the crossover area should be larger gradually for individual similarity. For the advantage of code technique, the crossover position can easily be revised in this paper. Therefore, a nonuniform arithmetic crossover with adaptive crossover position is adopted. With two individuals S_1^g and S_2^g crossing, the new individuals are obtained as follows:

$$\begin{aligned} S_1^{g+1} &= \lambda S_2^g + (1 - \lambda) S_1^g, \\ S_2^{g+1} &= \lambda S_1^g + (1 - \lambda) S_2^g. \end{aligned} \tag{12}$$

In (12), $\lambda = \exp(-\lambda_0 \cdot G_{max}/g)$, where λ_0 is a constant, $\lambda_0 = 0.9$ in this paper, G_{max} is the maximum iterations, and g is the current iterations.

3.2.3. Mutation. For increasing the population diversity, a nonuniform mutation rate is adopted. The individual S_i^g

mutate into a new individual according to the following equation:

$$S_i^{g+1} = \begin{cases} 0.95S_i^g + 0.05\theta, & \text{if } c_1 < 0.5P_m, \\ 0.9S_i^g + 0.1\theta, & \text{if } c_1 < 0.9P_m, \\ \theta, & \text{if } c_1 < P_m, \end{cases} \quad (13)$$

where θ is $1 \times n$ dimensional matrix formed by uniformly distributed random number between 0 and 1, n is the number of genes on each chromosome, P_m is fixed mutation rate, and c_1 is a uniformly distributed random number between 0 and 1. Equation (13) implies that the mutation rate is variable according to the uniformly distributed random number c_1 .

3.2.4. Simulated Annealing Operator (SAO). Simulated annealing is set as an operator in simulated-annealing-based genetic algorithms (SAGA). According to the characteristic of simulated annealing, the out layer loop temperature of simulated annealing operator in each generation is gradually decreased with evolution forward; that is, $T_g \leftarrow T_{g-1} \cdot \beta$, where β is the decay scale and is 0.9 in this paper. Then, set T_g as initial temperature T in simulated annealing operator of current generation g .

However, the temperature is lowering by $t \leftarrow t \cdot \alpha^{I_t}$ in simulated annealing operator, where t is the current temperature and I_t is the current iteration. It is obvious that the cooling process is related not only to the decay scale α , $\alpha \in [0, 1]$, but also to the current iteration I_t . Furthermore, a diverse method of perturbation is adopted for the stability is not the same at different temperatures. The detailed procedure of simulated annealing operator is described in Algorithm 2.

Algorithm 2. Input the out layer loop temperature T_g , S_i , L , and I_{\min} .

Step 1 (initialization). Set $T = T_g$, $t = T_g$, $I_t = 0$, and $S = S_i$. Compute fitness $f(S)$ and $\text{PreBestS} \leftarrow S$, $\text{PreBestZ} \leftarrow f(S)$, $\text{BestS} \leftarrow \text{PreBestS}$, $\text{BestZ} \leftarrow \text{PreBestZ}$, where PreBestS is the preoptimal solution, PreBestZ is the preoptimal value of object function, BestS is the current optimal solution, and BestZ is the current value of object function.

Step 2. For $l = 1 : L$, where L is the Markov length.

Step 2.1 (movements). Perturb the placement through a defined movement and the specific perturbation model is as follows:

$$S' = \begin{cases} 0.7S + 0.3\theta & t \in \left(\frac{T}{2}, T\right], \\ 0.85S + 0.15\theta & t \in \left(\frac{T}{10}, \frac{T}{2}\right], \\ 0.95S + 0.05\theta & t \in \left[0, \frac{T}{10}\right), \end{cases} \quad (14)$$

where θ is defined as in Section 3.2.3, S is the previous best solution, and S' is a newborn solution after perturbing; t is the current temperature in the simulated annealing operator.

Step 2.2 (calculate score). Calculate the change in the score for perturbation. Calculate the increment of objectives by $\Delta f = f(S') - f(S)$, where $f(S')$ is the fitness of the newborn solution S' .

Step 2.3 (Metropolis Criterion). If $\Delta f > 0$, accept the newborn solution S' , $S \leftarrow S'$. Otherwise, accept the newborn S' with a certain probability, $S \leftarrow S'$, where $P = \exp(\Delta f/(kt))$, where k is the Boltzmann constant, $k = 0.85$ in this paper.

Step 2.4 (estimate S). If $f(S) > \text{BestZ}$, set $\text{PreBestS} \leftarrow \text{BestS}$, $\text{PreBestZ} \leftarrow \text{BestZ}$, $\text{BestS} \leftarrow S$, $\text{BestZ} \leftarrow f(S)$.

Step 3 (termination conditions). If $(\text{BestZ} - \text{PreBestZ})/\text{PreBestZ} < e$ (parameter tolerance $e = 10^{-4}$) or $I_t > I_{\min}$, where I_{\min} is the least iterations, stop and output the result. Otherwise, go to Step 4.

Step 4 (lowering temperature). Set $I_t \leftarrow I_t + 1$, $t \leftarrow t \cdot \alpha^{I_t}$; go to Step 2.

3.3. Termination Conditions. When applying the proposed hybrid algorithms to solve the capacity allocation problem (P2), we must obtain an approximate solution of desirable precision within a proper time. For this reason, two parameters G_{\max} and G_{\min} are introduced, which denote the maximum and the minimum iterative generation, respectively. Then the following termination conditions are imposed:

- (i) If $G_t > G_{\min}$ and $(f_{\max} - f_{\text{mean}})/f_{\max} < \varepsilon$, stop.
- (ii) If $G_t > G_{\max}$, stop. Otherwise, return to iterative process.

ε is a tolerance parameter ($\varepsilon = 0.0005$ in this paper), G_t is the current generation, and f_{\max} , f_{mean} are the maximum fitness and the mean fitness in the current populations, respectively.

4. Numerical Experiments

Some numerical experiments are carried out to demonstrate the applications of the mathematical formulation and efficiency of the proposed algorithm. The algorithm has been coded in C++ and run under Microsoft Windows 7 Professional (Intel Core 2 Duo CPU 2.26 GHz and 2.0 GB of RAM). CPU times were obtained through the C++ function clock ().

Example 1. As shown in Table 2, we firstly suppose a small-scale instance of the optimization capacity allocation problem for container carrier in synchromodal transportation only with one origin: Shanghai and 3 destinations: Dalian, Zhengzhou, and Chongqing, which are container central stations. In the case, there are 3 transportation modes: River, Rail, and Road, 4 freight types: Electronic products, Mechanical equipment, Pharmaceuticals, and Textiles, and 2 scenarios, that is, $M = 3$, $I = 4$, $J = 3$, and $S = 2$ with 227 variables, 86 constraints. Since shortage in container capacity means that carriers lose the chance to make transportation profit, the penalty coefficient of shortage

TABLE 2: Basic information.

The number of scenarios in a planning horizon: 2 The container central station of origin: Shanghai		
Freight type	Probability of each scenario	Destination
(1) Electronic products	(1) 1/2	(1) Dalian
(2) Mechanical equipment	(2) 1/2	(2) Zhengzhou
(3) Pharmaceuticals		(3) Chongqing
(4) Textiles		

is much more than that of overage, almost equaling unit delivery profit. On the other side, the overage is relatively less as more penalty is included in $r_{mij}z_{mij}^s$ which indicates that the container carriers lose the opportunities for making more transportation profit from the unutilized containers z_{mij}^s . These penalty coefficients are consistent with [9]. The relevant experimental data are tabulated in Tables 3–5, including capacity and demands information, capacity and demands information, profit and penalties information.

Before verifying the effectiveness and efficiency of the proposed SAGA, the actual optimal solution of the above numerical example is obtained through the branch and bound method (B-and-B) in GLPK 3.2.2 (see the website of free software GLPK, 2002).

$u_{22} = 1, u_{31} = 1, u_{33} = 1, x_{212} = 3800, x_{222} = 4500, x_{132} = 3200, x_{142} = 1900, x_{311} = 2400, x_{321} = 3200, x_{331} = 2500, x_{341} = 4400, x_{313} = 3800, x_{323} = 2700, x_{333} = 3400,$ and $x_{343} = 3500$. Other decision variables equal zero and the actual optimal objective value $opt_obj = 19428550$.

Firstly, we test the sensitivity and feasibility of the proposed algorithm with respect to different parameters settings. Randomly generate 10 sets of parameters in SAGA, and for each set of experiments, the above small-scale instances were performed 20 times. The computational results are listed in Table 6, where $P_c, P_m, P_{size}, G_{min},$ and G_{max} represent the population size, crossover probability, mutation probability, and maximum and minimum iterative generation in genetic operators, respectively; $T_0, \alpha, L,$ and I_{min} represent the initial temperature, the factor of step length, Markov length, and the minimum iteration in simulated annealing operator, respectively. Besides, Best, Average, and Time represent the approximate optimal objectives, the average objectives, and the average processing time in 20 times, respectively. The relative gap between best approximate optimal objectives in each set of experiments and exact optimal objectives is denoted by δ and calculated by the following formulation:

$$\delta = \frac{opt_obj - Best}{opt_obj} \times 100\%. \quad (15)$$

It is easy to see from these results that the total average objectives and variance are 18901431 and 2.71%, respectively, which illustrates the robustness and steadiness of the proposed solution methodology. Additionally, it is interesting to note that the processing time is sensitive to the setting of parameters and varies in the numerical experiments for different parameters. For instance, the average processing

time in seventh sets is just 39.1s since relatively smaller parameters are adopted. In comparison, the tenth sets cost the longest processing time due to their much larger parameter settings. Actually, most of processing time in each experiment is wasted and useless. With the parameters in sets 7, the obtained best objective in this set of experiments is 19119450 and the relative gap is only 1.59%. Therefore, with consideration of processing time and objectives precision, the parameters in sets 7 are adopted in the rest of experiments; that is, $P_c = 0.1, P_m = 0.8, P_{size} = 50, G_{min} = 50, G_{max} = 70, Pa = 0.12, T_0 = 3000, \alpha = 0.9, L = 120,$ and $I_{min} = 10$.

Example 2. In this example, some experiments are implemented to demonstrate the efficiency of SAGA by comparing the results with these obtained by the traditional genetic algorithm (GA) and simple simulated annealing algorithm (SA). Besides, the parameter values used in SAGA were set as the same as mentioned in Example 1. In consideration of fairness, $P_c, P_m, T_0,$ and α are set as the same in SAGA, while P_{size}, G_{max} in GA and L, I_{min} in SA are adjusted to try to guarantee that the processing time of the three algorithms is more or less the same. In general, these parameters were adopted in GA: $P_c = 0.1, P_m = 0.8, P_{size} = 500,$ and $G = 500,$ while $T_0 = 3000, \alpha = 0.9, L = 600,$ and $I_{min} = 300$ in SA. Additionally, the code/decode techniques and other relative parameters are the same as SAGA.

For the small-problem in Example 1, 20 times were performed through SAGA, GA, and SA, respectively. The results and processing time of SAGA, GA, and SA are shown in Figure 6. One of the evolutionary performances of these algorithms is illustrated in Figure 7.

It is easily seen from Figure 6 that the results obtained through GA are relatively stable, while there are big fluctuations both in objectives and in processing time through SA. Moreover, in SA, the longer processing time does not imply more outstanding results. Obviously, SA has the initial solution-dependence disease during the evolutionary procession. It means that if the initial optimum has high quality, it can be quickly close to global optimal solutions. From Figure 7(a), we can find that GA can quickly find a relatively optimal solution in about 20 iterations. Unfortunately, it falls into the local optimal after 220 iterations. However, from Figure 7(b), it is easily seen that the simulated annealing procession can hardly produce a better solution after 45 accept points, which implies that SA is limited to local optimal research and incapable to break through local scope. It is worthwhile to note that it just needs 5 iterations to find a relatively optimal solution through SAGA, which is even better than that obtained through GA in its whole procession, as shown in Figure 7(c). Meanwhile, after 20 iterations, the objective is better than 1.9×10^7 . By contrasting Figures 7(a)–7(c), we can draw a conclusion that SAGA has significant advantages, such as higher efficiency and shorter processing time since it takes full advantages of the remarkable characteristic of GA in global search and that of SA in local search. The comparative results are shown in Table 7.

For the small-scale instance, it can be seen from Table 7 that the relative gap and the average processing time through

TABLE 3: Capacity and demands information.

Freight type	Total available capacity (TEU) in the planning horizon: 40000					
	Dalian		Zhengzhou		Chongqing	
	Scenario 1	Scenario 2	Scenario 1	Scenario 2	Scenario 1	Scenario 2
1	2400	2200	3800	3500	3800	3200
2	3200	3100	4500	4100	2700	2300
3	2500	2300	3200	2900	3400	3100
4	4400	4200	1900	1500	3500	3000
Transportation mode	The requested bottom supply of capacity for each transportation mode to each destination					
Road	3000		4000		3000	
Rail	4000		5000		4000	
River	3000		3000		4000	

TABLE 4: Profit (CNY¥) information.

Freight type	The unit delivery profit for each freight type shipped to each destination by each transportation mode											
	Dalian				Zhengzhou				Chongqing			
	Road	Rail	River	Average	Road	Rail	River	Average	Road	Rail	River	Average
1	500	490	580	523	440	510	460	470	460	490	480	477
2	550	540	600	563	450	520	440	470	490	470	490	483
3	530	490	570	530	470	530	430	477	480	500	510	497
4	440	480	560	493	450	470	450	457	500	450	460	470

TABLE 5: Penalties (CNY¥) for per overage and shortage in supplied capacity under each scenario.

Transportation mode	Scenario	Dalian				Zhengzhou				Chongqing			
		1	2	3	4	1	2	3	4	1	2	3	4
Penalty per overage in supplied capacity for delivery to each destination by each mode under each scenario													
Road	1	86	74	78	74	87	87	76	67	75	54	66	86
	2	76	87	70	61	86	59	87	80	55	72	83	79
Rail	1	87	68	85	83	72	81	57	59	82	61	65	70
	2	82	90	85	56	81	60	65	78	54	82	71	67
River	1	89	78	68	88	83	73	80	75	66	73	82	66
	2	81	80	88	87	77	81	59	64	67	67	74	73
Penalty per shortage in supplied capacity for delivery to each destination by each mode under each scenario													
Road	1	496	449	600	523	578	587	487	494	511	592	599	466
	2	562	584	577	600	442	526	553	587	514	597	507	463
Rail	1	590	504	482	542	519	520	592	520	571	506	465	505
	2	501	469	450	583	454	469	444	534	576	556	438	477
River	1	571	515	438	594	480	484	590	585	583	485	502	507
	2	528	498	450	459	571	550	472	489	496	595	597	468

SAGA are much smaller than through GA and SA. Although the minimum relative gap obtained through SA is 0.07%, which is less than 1.59% obtained through SAGA, the average processing time is much more than SAGA. Moreover, the worst objective is much smaller than SAGA. Referring to Figure 6, it is easily seen that SA is quite unsteady. However, the quantity of results obtained through GA is in between SA and SAGA, but it may not be unsatisfactory. Meanwhile, compared with B-and-B, the average processing time is

smaller, and the relative gap is within the allowed relative gap range.

Example 3. In this example, we consider a larger-scale instance of the optimization capacity allocation problem.

In the medium-scale instance: $M = 3$, $I = 8$, $J = 5$, and $S = 9$, with 2338 variables and 1105 constraints. It is difficult to input real data manually to the model with tens of thousands of parameters and variables. Therefore, we

TABLE 6: The computational results.

Sets	P_c	P_m	P_{size}	G_{min}	G_{max}	Pa	T_0	α	L	I_{min}	Time ($\times 10^1$)	Best ($\times 10^1$)	Average ($\times 10^7$)	δ
1	0.3	0.7	120	60	80	0.09	5000	0.7	90	12	7.64	1.9200	1.9100	1.18%
2	0.1	0.9	50	30	50	0.13	2000	0.9	100	10	2.36	1.9088	1.8837	1.75%
3	0.2	0.8	110	60	70	0.12	5000	0.9	110	12	11.07	1.9138	1.9031	1.50%
4	0.3	0.7	60	30	50	0.14	5000	0.9	80	13	2.63	1.9020	1.8655	2.10%
5	0.2	0.7	120	60	80	0.11	5000	0.8	110	10	8.96	1.9127	1.9082	1.55%
6	0.1	0.8	80	40	60	0.14	3000	0.9	100	14	6.58	1.9133	1.8772	1.52%
7	0.1	0.8	50	50	70	0.12	3000	0.9	120	10	3.91	1.9119	1.8991	1.59%
8	0.1	0.8	110	30	50	0.07	4000	0.8	120	12	3.59	1.9092	1.8718	1.73%
9	0.4	0.9	80	60	80	0.14	5000	0.7	80	15	8.31	1.9160	1.8923	1.38%
10	0.4	0.8	80	60	80	0.14	4000	0.8	110	15	11.31	1.9164	1.9010	1.36%
Average												1.9124	1.8901	

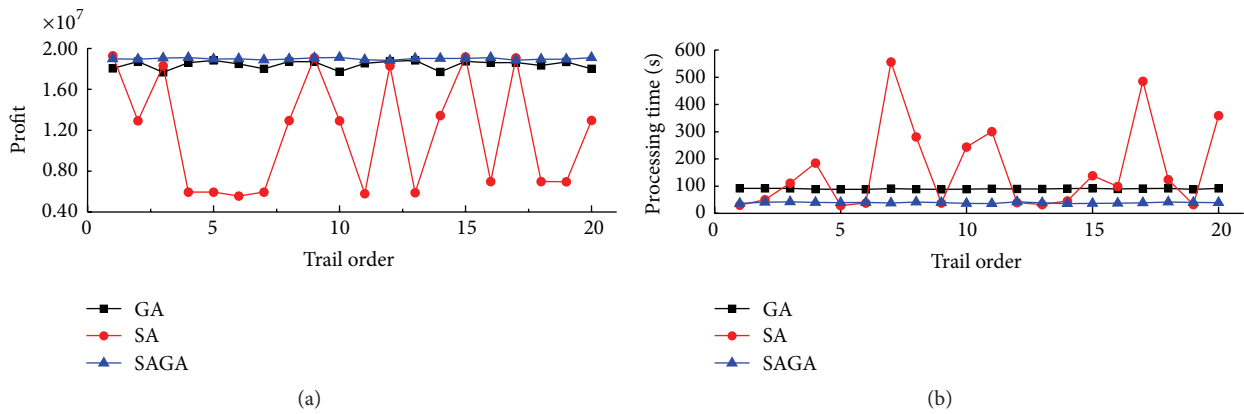


FIGURE 6: The results and processing time of SAGA, GA, and SA.

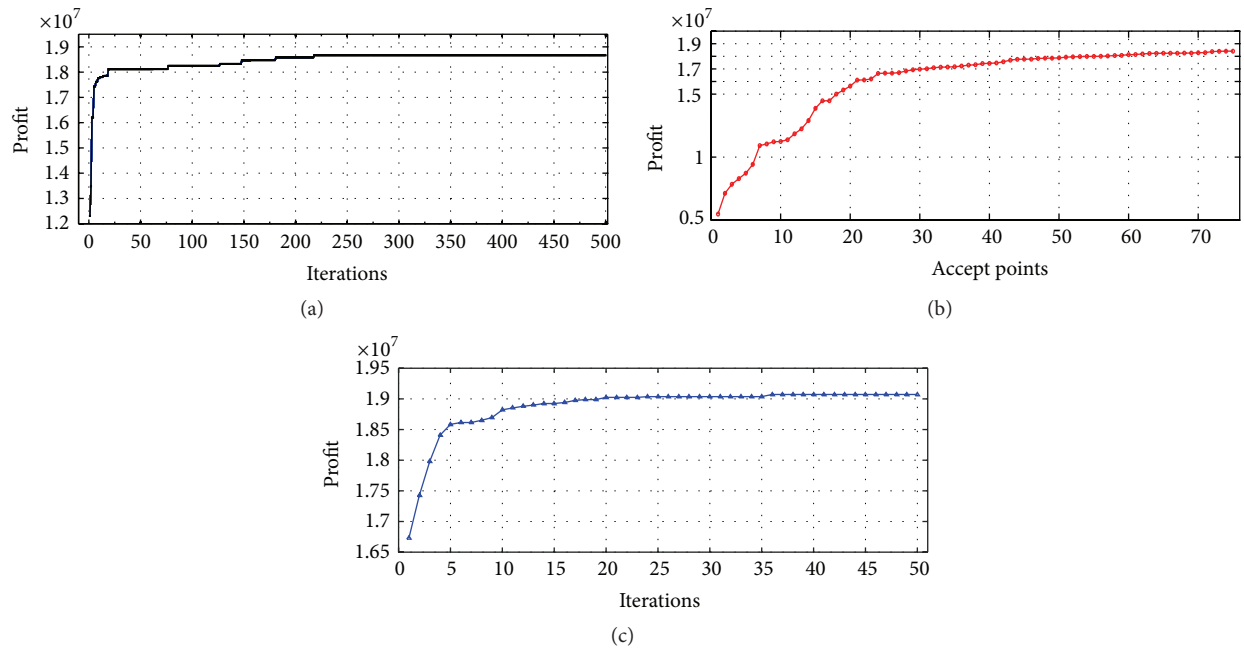


FIGURE 7: Evolutionary performances of these algorithms, where (a) is the optimal solution of each generation through GA; (b) is the optimal solution of each accepted point through SA; (c) is the optimal solution of each generation through SAGA.

TABLE 7: Experimental results for the small problem: 227 variables and 86 constraints (20 times).

Methods	Best ($\times 10^7$)	Average ($\times 10^7$)	Worst ($\times 10^7$)	Time ($\times 10^1$ s)	δ
GA	1.8831	1.8417	1.7672	9.07	3.08%
SA	1.9293	1.1727	0.5576	16.88	0.07%
SAGA	1.9119	1.8991	1.8842	3.91	1.59%
B-and-B	1.9429			13.58	

TABLE 8: Experimental results for large-scale problem: 2338 variables and 1105 constraints (20 times).

Methods	Best ($\times 10^7$)	Average ($\times 10^7$)	Worst ($\times 10^7$)	Time ($\times 10^1$ s)	δ
GA	5.1493	5.0848	4.9471	13.42	—
SA	5.3107	3.7473	1.4473	20.93	—
SAGA	5.4416	5.4158	5.3831	7.58	—
B-and-B	#			#	

represents the exact optimal solution which cannot be obtained through the B-and-B method.

randomly generated an instance from uniform distribution to test our algorithm for solution quality and computational time. Let $[x]$ denote the largest integer that is less than or equal to x . In the section, D_{ij}^s , H , and C_{mj} are uniformly generated in the range $[1800, 4500]$, $[1800 \times I \times J, 4500 \times I \times J]$, $[1800 \times I, 4500 \times I]$, respectively. The parameters r_{mij} , α_{mij}^s , and β_{mij}^s are randomly generated from uniform distribution in the range $[450, 650]$, $[400, 550]$, $[60, 90]$, respectively. And then, SAGA, GA, SA, and B-and-B for the instance are performed 20 times, respectively. The results are tabulated in Table 8, where “#” represents the exact optimal solution which cannot be obtained through the B-and-B method. Thus, there is no relative gap between the exact optimal solution and approximate optimal solutions obtained through SAGA, GA, and SA, indicated by “—” in the table.

In Table 8, it proves once more that SA is very unsteady, regardless of the results or processing time. However, the results through GA are relatively stable but unsatisfactory. Contrarily, SAGA has obvious advantages in average objectives and processing time, showing promising computational results.

From what has been discussed in Examples 1–3, we can conclude that the numerical solutions of the proposed SAGA are very close to the exact solution, and the relative gaps are all in the acceptable relative gap range. The main reason is that the proposed oriented-problem algorithm combined the GA with SA, so that it is good not only at local search but also in global search. The excellent individuals may have more opportunity to produce better individuals by simulated annealing operator. Meanwhile, it can overcome the unstable characteristic of SA by genetic operators. To sum up, the proposed algorithm is an effective algorithm for the model (P2) of the container capacity allocation problem with random demands for the container carrier in synchronodal transportation.

5. Conclusions

On the basis of analyzing the research status of capacity allocation in synchronodal transportation, this paper presents a stochastic integer programming model for the container capacity allocation problem with random freight demands in synchronodal transportation network from container carriers’ perspective. In this paper, the purpose of the proposed model is to maximize the expected total transportation profit over optimal decisions. Since there is no effective polynomial-time algorithm to solve the proposed formulation in the real world, a problem-oriented intelligence algorithm is designed for the proposed formulation while some numerical examples are carried out to demonstrate the effectiveness and efficiency of the hybrid algorithm.

As the above analysis in Section 4, four types of algorithms, including GA, SA, SAGA, and B-and-B, are tested on two instances, respectively. The proposed SAGA, by integrating simulated annealing method with genetic algorithm, shows promising computational results. The SAGA with a novel encoding method and a developed elitism strategy, comparing to simple GA and SA, has obvious advantage in processing time and results. To sum up, with the problem-oriented algorithm, the proposed programming formulation has important practical significance for the container carrier.

For the container capacity allocation problem in this research without considering the repositioning of empty containers, one can consider the capacity allocation problem by taking the repositioning of empty containers into account to minimizing the operational cost in future research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported in part by the National Basic Research Program of China (Grant no. 2012CB725400), the National Natural Science Foundation of China (Grant nos. U1434209 and 71131001-1), the Research Foundation of State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University (Grant nos. RCS2014ZT19 and RCS2015ZZ003), and the Fundamental Research Funds for the Central Universities (nos. 2015YJS088), Beijing Jiaotong University.

References

- [1] M. Steadieseifi, N. P. Dellaert, W. Nuijten, T. Van Woensel, and R. Raoufi, “Multimodal freight transportation planning: a literature review,” *European Journal of Operational Research*, vol. 233, no. 1, pp. 1–15, 2014.
- [2] T. G. Crainic, “Long-haul freight transportation,” in *Handbook of Transportation Science*, vol. 56 of *International Series in Operations Research & Management Science*, pp. 451–516, Springer, New York, NY, USA, 2003.

- [3] T. G. Crainic and K. H. Kim, "Intermodal transportation," *Transportation*, vol. 14, pp. 467–537, 2006.
- [4] A. M. Newman and C. A. Yano, "Scheduling direct and indirect trains and containers in an intermodal setting," *Transportation Science*, vol. 34, no. 3, pp. 256–270, 2000.
- [5] N. Bostel and P. Dejax, "Models and algorithms for container allocation problems on trains in a rapid transshipment shunting yard," *Transportation Science*, vol. 32, no. 4, pp. 370–379, 1998.
- [6] O. C. Demirag and J. L. Swann, "Capacity allocation to sales agents in a decentralized logistics network," *Naval Research Logistics*, vol. 54, no. 7, pp. 796–810, 2007.
- [7] M. G. H. Bell, X. Liu, P. Angeloudis, A. Fonzone, and S. H. Hosseinloo, "A frequency-based maritime container assignment model," *Transportation Research Part B: Methodological*, vol. 45, no. 8, pp. 1152–1161, 2011.
- [8] M. Arnone, S. Mancini, and A. Rosa, "Formulating a mathematical model for container assignment optimization on an intermodal network," *Procedia—Social and Behavioral Sciences*, vol. 111, pp. 1063–1072, 2014.
- [9] C. Cao, Z. Gao, and K. Li, "Capacity allocation problem with random demands for the rail container carrier," *European Journal of Operational Research*, vol. 217, no. 1, pp. 214–221, 2012.
- [10] C. Cao, Z. Gao, and K. Li, "Optimal rail container shipment planning problem in multimodal transportation," *Engineering Optimization*, vol. 44, no. 9, pp. 1057–1071, 2012.
- [11] P. Jacko, "Resource capacity allocation to stochastic dynamic competitors: knapsack problem for perishable items and index-knapsack heuristic," *Annals of Operations Research*, pp. 1–25, 2013.
- [12] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer Science & Business Media, 2011.
- [13] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman Publishers, San Francisco, Calif, USA, 1979.
- [14] J. Holland, *Adaption in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [15] L. Yang, Z. Gao, and K. Li, "Railway freight transportation planning with mixed uncertainty of randomness and fuzziness," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 778–792, 2011.
- [16] S. Siva Sathya and S. Kuppuswami, "Gene silencing—a genetic operator for constrained optimization," *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 5801–5808, 2011.
- [17] K. Deep, K. P. Singh, M. L. Kansal, and C. Mohan, "A real coded genetic algorithm for solving integer and mixed integer optimization problems," *Applied Mathematics and Computation*, vol. 212, no. 2, pp. 505–518, 2009.
- [18] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [19] C. Koulamas, S. Antony, and R. Jaen, "A survey of simulated annealing applications to operations research problems," *Omega*, vol. 22, no. 1, pp. 41–56, 1994.
- [20] S. Y. Zhang, Z. H. Cai, and Z. G. Zhan, "Solving 0-1 knapsack problem based on genetic algorithm with improved simulated annealing," *Microelectronics & Computer*, vol. 2, no. 15, 2011.
- [21] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: a survey," *Applied Soft Computing Journal*, vol. 11, no. 6, pp. 4135–4151, 2011.
- [22] J.-W. Chung, S.-M. Oh, and I.-C. Choi, "A hybrid genetic algorithm for train sequencing in the Korean railway," *Omega*, vol. 37, no. 3, pp. 555–565, 2009.
- [23] S. C. H. Leung, D. Zhang, C. Zhou, and T. Wu, "A hybrid simulated annealing metaheuristic algorithm for the two-dimensional knapsack packing problem," *Computers & Operations Research*, vol. 39, no. 1, pp. 64–73, 2012.
- [24] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, 2000.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

