

Research Article

Modelling and Metaheuristic for Gantry Crane Scheduling and Storage Space Allocation Problem in Railway Container Terminals

Ming Zeng,¹ Wenming Cheng,² and Peng Guo²

¹School of Mechatronic Engineering, Southwest Petroleum University, Chengdu 610500, China

²School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China

Correspondence should be addressed to Ming Zeng; zm_1209@126.com

Received 12 April 2017; Accepted 23 October 2017; Published 20 November 2017

Academic Editor: J. R. Torregrosa

Copyright © 2017 Ming Zeng 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.

The gantry crane scheduling and storage space allocation problem in the main containers yard of railway container terminal is studied. A mixed integer programming model which comprehensively considers the handling procedures, noncrossing constraints, the safety margin and traveling time of gantry cranes, and the storage modes in the main area is formulated. A metaheuristic named backtracking search algorithm (BSA) is then improved to solve this intractable problem. A series of computational experiments are carried out to evaluate the performance of the proposed algorithm under some randomly generated cases based on the practical operation conditions. The results show that the proposed algorithm can gain the near-optimal solutions within a reasonable computation time.

1. Introduction

With the promotion of OBOR (One Belt and One Road) strategy, the trade contacts between China and the other countries have been improved significantly. As one of the most convenient and efficient modes of transportation, container transport has been popularized and developed all over the world. For some developed countries, the containerization ratio of general cargo has reached 80%, and the proportion of container cargo also has achieved 20%~40% in all the amount of railway freight [1]. Although the railway container transportation of China starts relatively late, the trend of development is very well. For example, in 2015, the year-on-year growth of the railway container freight volume in China is up to 20.2% [2]. In terms of the mid-long term railway network plan and the eleventh five-year plan of China Railway, a railway container transportation support system which takes 18 railway container terminals as the hub will be established. These terminals are located in the main economic centers and the significant ports, so the transportation resources are extremely rich and the developing potential is tremendous.

Multimodal and door to door transports are the main modes of modern railway container freight organization. The logistics service functions and levels determine the survivability and competitive capacity of the railway container terminals in the entire networked logistics system. As the crucial operation processes, the container handling and storage play a vital role at the aspects of accelerating cargo movement and saving circulation costs in the railway container terminals. Currently, the railway container terminals of China especially the sea port, land port, and metropolis located container freight transport hubs already had comparatively comprehensive handling, transferring, and storage system. However, some problems still exist in the operation systems of railway container terminals, such as low efficiency, high expenses, and weak response to the demand changes; thus the needs of modern logistics cannot be satisfied and even the competitiveness of the whole railway container transportation system is influenced. The reasonable management and control to the processes of railway container terminal have great importance to rationalize the equipment utilization, improve the operation ability and service quality, and enhance the safety and convenience of freight delivery.

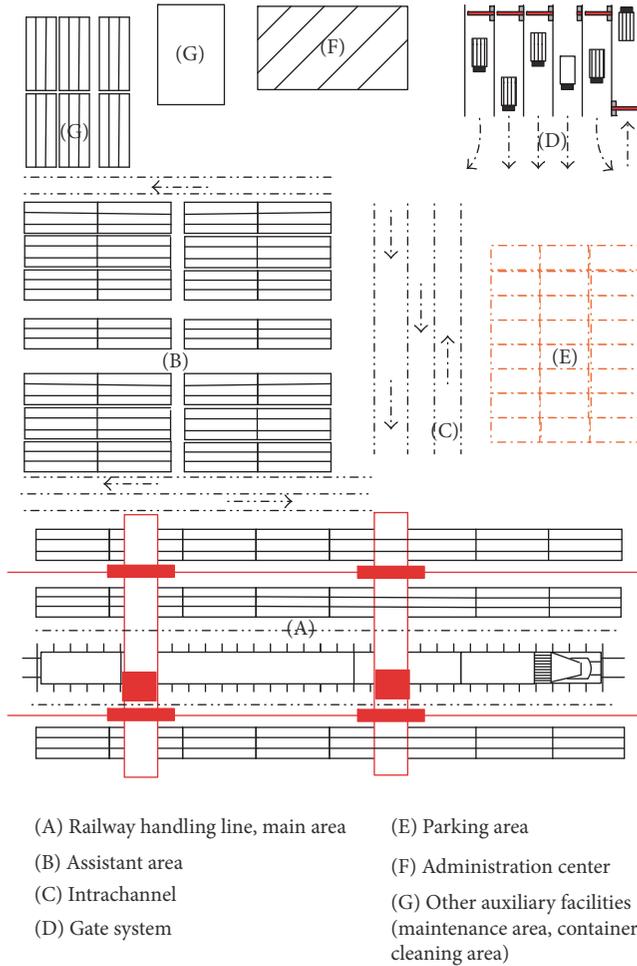


FIGURE 1: Sketch map of railway container terminal.

Compared with the traditional railway freight station, the railway container terminal is quite different in respect of the function and the equipment. As the product of modern logistics, railway container terminal is a complete logistics system to carry out many service tasks. In accordance with the modern logistics service functions of railway container terminal, there are certain functional zones established such as railway handling line, main area, assistant area, intrachannel, gate system, parking area, administration center, and other auxiliary facilities. The specific layout and functional zone division are presented in Figure 1.

In a typical railway container terminal, there are several operation regions established along with the rail line, and a number of gantry cranes are distributed in these regions to implement the handling tasks of container trains. When a railway container train arrived, the suitable stop position must be determined promptly. Normally, the container trains are assigned to the stop position near to the storage spaces in the main area, namely, the expected position. Whereas if the real position that the container train stopped is far away from the expected position, the unloading time of containers to the storage spaces will increase greatly, and the operation efficiency would be adverse influenced accordingly.

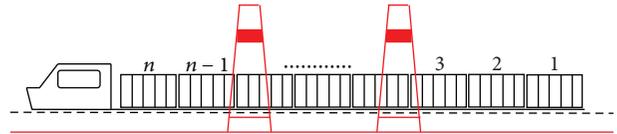


FIGURE 2: Task division of a container train.

After fixing the stop position, that is, the railway handling line in main area, then accordingly the unloading operations will be carried out. Some containers are unloaded to the outer container trucks through the gantry cranes and then delivered directly to the consignee. The rest of containers are unloaded to the available storage space of the main area through the gantry cranes, or to the inner container trucks, and then transferred to the assistant area. However, during the practical operation process, the number of containers unloaded to the assistant area is comparatively less and the storage position is relatively fixed in the actual unloading operation process; therefore the research of this paper is mainly focus on the gantry crane scheduling and storage space allocation problem in the main area.

For the purpose of facilitating the analysis and mathematical modelling, the containers that need to be unloaded are ranked from the tail to the head of a train, and the serial numbers are applied to indicate the task numbers of the unloading operations. Figure 2 illustrates the task division of a container train. Gantry crane is the main equipment placed over two railway tracks and have the cantilever to conduct the parallel processing of container transferring in the railway container terminal. For the sake of safety, all the gantry cranes which share the same tracks are not allowed to cross each other, and adequate distance between the adjacent gantry cranes must be kept as well, which will be the interference constraints in the mathematical model.

In this paper, the gantry crane scheduling and storage space allocation problem are integrated to be investigated which has more practical significance comparatively. The remainder of this paper is structured as follows: Section 2 surveys the literatures with respect to the gantry crane scheduling and storage space allocation problems in both railway container terminals and port terminals. Section 3 formulates a mixed integer programming model, and subsequently in Section 4, an improved backtracking search algorithm is proposed to obtain the solution. Then the computational experiments are conducted in Section 5. And finally, Section 6 concludes the article with an outline of future study.

2. Literature Review

Because the effect factors of the gantry crane scheduling and storage space allocation in railway container terminal are complex and the constraints are numerous, the simulation methods based on software are adopted to present the practical situation in most of the researches. Guo et al. [3] and Benna and Gronalt [4] established the simulation model according to the operation process in railway container

terminal separately, and a series of experiments were conducted to evaluate the performance under various configuration. Nonetheless, the mathematical model is required for providing a precise description to the problem. Jeong and Kim [5] integrated the gantry crane scheduling, in and out container flow, and inner truck stop position decision problems as the railway freight station scheduling problem and proposed the relevant genetic and heuristic algorithm. Wang and Zhu [6] constructed a optimization model to minimize the idle time of gantry crane, and an ant colony algorithm was presented correspondingly. They also formulated the storage space allocation optimization models and introduced the rolling horizon method to solve this problem [7]. Boysen et al. [8] carried out a train scheduling model in railway container yard and acquired the solution by using the dynamic programming method together with the beam search heuristic algorithm. The above-mentioned researches have begun to study the gantry crane scheduling and storage space allocation problem in railway container terminals, whereas these models are relatively simple and without considering the interference constraints.

Unlike the small number of studies in railway container terminals, the researches of quay gantry crane scheduling and berth allocation problems in port terminals which can provide some crucial inspiration for our research are comparatively rich. The quay gantry crane scheduling problem was first concerned by Daganzo [9] and then this problem has been studied constantly and intensively. As the objectives of operation, containers are gathered by different scopes by the researchers. Lee et al. [10] and Boysen et al. [11] carried on the research of quay gantry crane scheduling problem with bays. The former adopted a mixed integer programming method to establish the mathematical model and took the noncrossing constraints into consideration. The latter subdivided the bay area and regarded the cranes as two different sets. Meanwhile, the relevant optimization model was developed and two solution procedures (exact and heuristic) were proposed. However, the quay gantry crane scheduling problem with bays is proven to be NP-complete when the noncrossing and safety distance constraints were included [12]. So currently, the tasks are often aggregated by container groups [13–15]. Chen et al. [16] took a further study for the quay crane scheduling problem in the indented berth. Some properties of this problem were carefully analyzed and a mixed integer programming model was set up. Unsal and Oguz [17] synthetically considered some practical constraints and offered a programming model for this quay crane scheduling problem as well as the integrated scheduling problem. Expósito-Izquierdo et al. [18] illustrated the importance of the quay crane scheduling problem and designed a hybrid estimation of distribution algorithm along with the local search to obtain the best solution. Beyond that, an approximate evaluation is adopted to improve the computation.

The gantry cranes are assumed to travel freely and without any interference constraints in the original researches [19, 20]; however it does not conform to the actual operation situations of the gantry cranes in port terminals. As the research continues, the interference constraints of the gantry cranes have been widely concerned [21–23]. Although the

introduction of interference constraints is more realistic, the complexity of the problem increases as well. In some related optimization field, the evolutionary or heuristic methods are put forward. Chlumecký et al. [24] adopted the genetic algorithm to optimize a rainfall-runoff model and provided a new concept of a random number generator. Su et al. [25] proposed a comprehensive learning artificial bee colony algorithm which used a pool of optimal foraging strategies to balance the exploration and exploitation tradeoff to solve the dynamic environments problems. Cui et al. [26] provided a ranking-based adaptive ABC algorithm, the parent food sources are chosen based on their rankings, and the selection probability of the food source is on the basis of the corresponding ranking. Particle swarm optimization is also widely used in solving the optimization problems. However, the premature convergence and stagnation which are caused by the subtraction-based solution update mechanism are the prominent problems. Kiran [27] presented a distribution-based update rule to overcome these issues and investigated it on 13 benchmark functions together with 3 engineering optimization problems. Meanwhile, some researchers employ the hybrid approach to improve the single algorithm [28, 29]. Li et al. [30] combined the local search phase in PSO with two global search phases in ABC in order to avoid the local optimal problem of PSO when solving the high-dimensional optimization problems and the slow convergence speed of ABC. Kiran and Gündüz [31] recombined the global best solutions gained by the PSO and ABC and took the recombination solution as the global best of PSO and the neighbor food source of ABC. For the gantry crane scheduling problem, Diabat and Theodorou [32] presented a genetic algorithm to solve the quay crane assignment and scheduling problem; meanwhile they proved the solution of genetic algorithm was better than that of exact methods from a series of computational experiments. Monaco and Sammarra [33] established a mixed integer model for the quay crane scheduling problem and proposed a tabu search algorithm to obtain the feasible solutions. Other heuristic approaches such as variable neighborhood search [34] and simulated annealing [35] are also used quite commonly.

To achieve a more realistic and reliable programming, the quay crane assignment problem, the quay crane scheduling problem, and the berth allocation problem are integrated to study at present. The implementation of integration mainly includes two approaches: (1) take the solution of the first problem as the input of the other problem. (2) Settle the two problems alternately. Raa et al. [36] combined the factors of vessel priorities, operation time, and preferred berthing locations to form a MILP model for the quay crane assignment and berth allocation problem; besides a hybrid heuristic approach was provided to get the solution. Elwany et al. [37] simultaneously handled the quay crane assignment and berth allocation problem. They also used a heuristics-based method to solve this NP-hard problem and acquired the satisfactory solutions. Different with the above two papers which focus on the number of cranes distributed to the vessel, Zeng et al. [38] pay more attention to the specific crane assigned to serve the vessel. In comparison, the researches of integrated quay crane scheduling and berth allocation

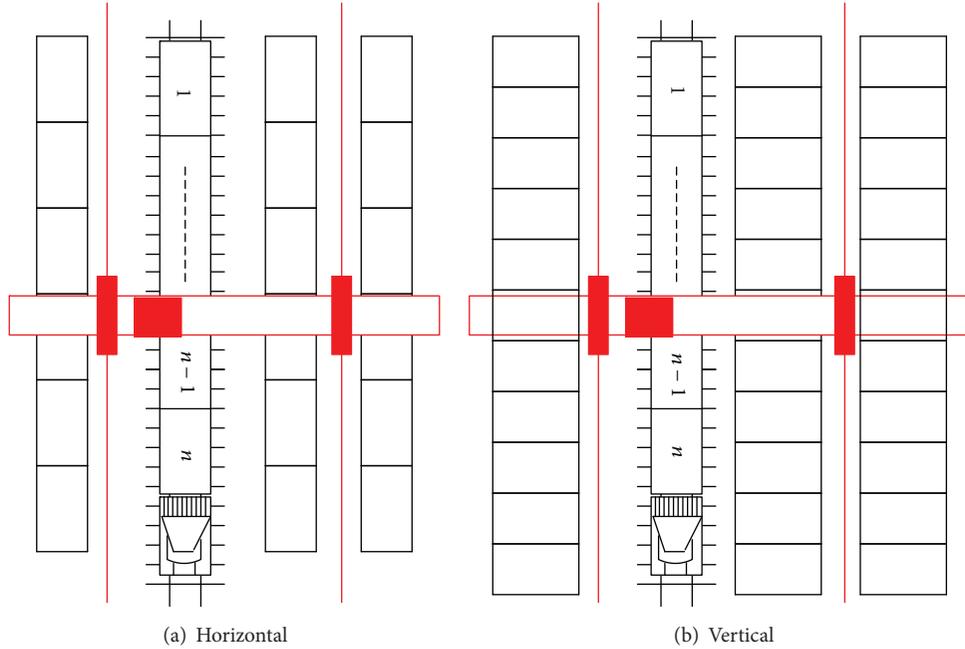


FIGURE 3: Scheme of the storage direction in the main area of railway container terminal.

problem are seldom. Song et al. [39] made use of the bilevel programming approach to solve this integrated problem and applied an improved genetic algorithm together with the branch-and-bound method to gain the solution. Lee and Wang [40] proposed a mixed integer programming model with the genetic algorithm to figure out the discrete quay crane scheduling and berth allocation problem. Nonetheless, some significant affecting factors as berthing position are not concerned adequately in these researches.

The storage space allocation problem in port terminals is first studied by Zhang et al. [41]. And then Chen and Lu [42] deal with the problem of storage location assignment for outbound containers. They not only manage the storage locations in bays but also determine the specific storage locations for every container. The mixed integer programming model and the hybrid sequence stacking algorithm were utilized to settle this problem. To minimize the realignment and optimize the operations of the terminal, Rodriguez-Molins et al. [43] provided a domain-dependent heuristic planner which also can be applied to acquire the optimal allocation for each container. Silva et al. [44] proposed a GRASP and tabu search based hybrid algorithm to solve the dynamic space allocation problem which is verified to be a NP-hard problem. With the purpose of decreasing the expected waiting time of outer trucks, Yu and Qi [45] researched both the block space allocation problem and the correlative remarkshaling problem. Three models were presented for distinct storing conditions, and a heuristic algorithm was designed accordingly. However, because the facilities scheduling factors are rarely incorporated in these models, there is still a certain limitation.

The gantry crane scheduling and storage space allocation are two closely connected problems; the proper storage space allocation is important in improving the performance of

gantry cranes and reducing the operation time. Thus, an integrated frame of the two problems will be more efficient for the practical operation. However, there are abundant researches giving attentions to the quay crane scheduling, berth allocation, storage space allocation, or the integrated problems in port terminals; there is no research that concerns the integrated problem of gantry crane scheduling and storage space allocation in railway container terminals. Meanwhile, various appropriate solution approaches are lacked in solving this problem.

3. Mathematical Formulation

In this paper, the objective of the gantry crane scheduling and storage space allocation problem is to minimize the handling time of the container train. The handling procedures of gantry cranes to transfer the containers to the specific locations are as follows: firstly, the crane and trolley move to the current position of the unloading task. Then shift to the empty storage space or the transshipment truck with the heavy load and finish the drop operation. Finally, loosen the lock and lift with the empty load. There are two storage modes existing in the main area of railway container terminal: one is along with the railway track as shown in Figure 3(a), and another is vertical to the railway track which is displayed in Figure 3(b). Thus, the different model parameters ought to be input in the light of the storage direction of the containers.

Before establishing the gantry crane scheduling and storage space allocation model, the following assumptions are required:

- (1) The operation efficiencies of all the gantry cranes are the same.

- (2) All the gantry cranes have the unique traveling speed.
- (3) The gantry cranes in the same rail cannot cross each other when conducting the handling tasks of a train and a safety distance must be always kept between the adjacent gantry cranes.
- (4) The processing task of a gantry crane should be finished before another task proceeded.

On the basis of the assumptions and the objectives, the mathematical model can be established according to the practical operation processes of the main area in the railway container terminal. The notations are defined as below:

N : the set of containers in need of operating; the number of tasks $n \in N$

G : the set of gantry cranes, $g \in G$

M : the set of storage spaces in main area; the storage space index $m \in M$

t_g^0 : the earliest available time of gantry crane g

l_g^0 : the initial position of g (denoted by the task number)

l_i : operation position of the i th task (denoted by the task number)

s^0 : the expected stop position of container train

s : the real stop position of container train

δ : the safety distance required between the adjacent gantry cranes

C_i : the completion time of the i th task

C_{\max} : the maximum completion time of all the unloading tasks of the container train

t^0 : the traveling time of crane between the adjacent tasks

t_{ij} : the required traveling time of crane from the i th task to the j th task

d_{im} : the distance between the central axis of railway track and the m th storage space

d_c : the drop distance of the gantry crane to the main area with the heavy load

v_t : the traveling speed of the trolley of gantry crane

v_d : the drop speed of gantry crane with heavy load

t_{im} : the sum of the average lift time of gantry crane with heavy load, the average rise and fall time of gantry crane with empty load, and the alignment locking and unlocking time when the i th task is arranged to the m th storage space in main area.

t_h : the horizontal traveling time of the trolley of gantry crane to the inner truck.

t_{ki} : the sum of the average lift and drop time of gantry crane with heavy load, the average rise and fall time of gantry crane with empty load, and the

alignment locking and unlocking time when the i th task is transferred by the inner truck.

O_i^0 : the original operation time of the i th task.

O_i : the actual operation time of the i th task.

Γ : a large enough positive number

$x_{ig} = \{1, \text{the } i\text{th task is assigned to gantry crane } g; 0, \text{otherwise}\}$

$y_{im} = \{1, \text{the } i\text{th task is arranged to the } m\text{th storage space in main area; } 0, \text{otherwise}\}$.

$k_i = \{1, \text{the } i\text{th task is transferred to the assistant area by the inner truck; } 0, \text{otherwise}\}$

$z_{ij} = \{1, \text{the } i\text{th task is completed no later than the start time of the } j\text{th task; } 0, \text{otherwise}\}$.

Objective:

$$\min C_{\max} \quad (1)$$

Subject to:

$$C_{\max} \geq C_i \quad \forall i \in N, \quad (2)$$

$$\sum_{i=1}^n y_{im} \leq M \quad \forall m \in M, \quad (3)$$

$$\sum_{g=1}^G x_{ig} = 1 \quad \forall i \in N, \quad (4)$$

$$\sum_{m=1}^M y_{im} \leq 1 \quad \forall i \in N, \quad (5)$$

$$\sum_{m \in M} \sum_{i=1}^N y_{im} + \sum_{i=1}^n k_i = N, \quad (6)$$

$$y_{im} + k_i \leq 1 \quad \forall i \in N, \quad m \in M, \quad (7)$$

$$t_{ij} = t^0 \cdot |l_i - l_j| \quad \forall 1 \leq i < j \leq n, \quad (8)$$

O_i^0

$$= y_{im} \cdot \left(\max \left(t_{ij}, \frac{d_{im}}{v_t} \right) + \frac{d_c}{v_d} + t_{im} \right) + k_i \quad (9)$$

$$\cdot \left(\max (t_{ij}, t_h) + t_{ki} \right) \quad m \in M, \quad \forall 1 \leq i < j \leq n,$$

$$O_i = O_i^0 \cdot (1 + \alpha |s - s^0|) \quad \forall i \in N, \quad (10)$$

$$\sum_{g=1}^G x_{ig} \cdot t_g^0 + O_i \leq C_i \quad \forall i \in N, \quad (11)$$

$$C_i - (C_j - O_j) + z_{ij} \cdot \Gamma \geq 0 \quad \forall 1 \leq i < j \leq n, \quad (12)$$

$$C_i - (C_j - O_j) + (1 - z_{ij}) \cdot \Gamma \leq 0 \quad \forall 1 \leq i < j \leq n, \quad (13)$$

$$\Gamma \cdot (z_{ij} + z_{ji}) \geq \sum_{g=1}^G g \cdot x_{ig} - \sum_{h=1}^G h \cdot x_{jh} + (i - j) \quad (14)$$

$$\forall 1 \leq i < j \leq n,$$

$$i + \delta + 1 - j \leq \Gamma \cdot (z_{ij} + z_{ji}) \quad \forall 1 \leq i < j \leq n, \quad (15)$$

$$x_{ig}, y_{im}, k_i \in \{0, 1\}, \quad \forall 1 \leq i, j \leq N; 1 \leq g \leq G. \quad (16)$$

The objective function (1) is to minimize the maximum completion time of all the operations, and the maximum completion time is determined by constraint (2). Constraint (3) ensures the number of containers unloaded to the main area is not more than the total number of storage spaces in the main area. Constraint (4) guarantees a task can only be handled by one gantry crane. Constraint (5) determines that a container can just occupy one storage space of the main area. Constraint (6) and Constraint (7) emphasize that the containers only can be unloaded to the main area or the assistant area, and the total number of containers to these two areas equals the number of containers in the container train. Constraint (8) defines the required moving time of crane from the i th task to the j th task. Constraint (9) means the operation time of the i th task is the sum of the moving time, rise and fall time, alignment and locking time, and alignment and unlocking time of the gantry crane when the container is unloaded to the main area or to the inner truck and transferred to the assistant area. In practice, the traveling of the crane and trolley often occurred at the same time, so the moving time is the larger value between the traveling times of crane and trolley. Constraint (10) redefines the operation time of the i th task on considering the stop position of the container train. Constraint (11) undertakes that the start time of a task is not earlier than the soonest available time of the gantry crane g which is utilized to complete this task. Constraints (12) and (13) provide a definition to the decision variable z_{ij} . The variable z_{ij} equals 1 if and only if the i th task is completed before the j th task and equals 0 otherwise. Constraint (14) ensures the gantry cranes in the same track do not cross each other. The numbers of tasks and gantry cranes are in an ascending sort order from the tail of the container train. When $Z_{ij} + Z_{ji} = 0$ (i.e., the i th task and the j th task are conducted simultaneously, and the index of task i is less than j), if the i th task is operated by gantry crane g and the j th task is handled by gantry crane h , then $g + 1 \leq h$. Similar to Constraint (14), Constraint (15) ensures that the adjacent gantry cranes always keep a safety distance when carrying out the simultaneous operation. Constraint (16) furnishes the value ranges of all the decision variables.

4. The Backtracking Search Algorithm for the Problem under Study

4.1. The Backtracking Search Algorithm. The backtracking search algorithm (BSA) is a relatively novel evolutionary algorithm at present; it was first introduced by Civicioglu [46] in 2013. Compared with the other evolutionary algorithms such as artificial bee colony (ABC) and particle swarm optimization (PSO) [46], the backtracking search algorithm has a better performance in model fitness, solution speed, and solving precision for solving the nonlinear, nondifferentiable and multimode complicated optimization problem. The reason why the backtracking search algorithm has such favorable property is the unique experimental population creation strategy, the control modes of search direction matrix and search boundary, and the operation methods of crossover and mutation. Even more worth mentioning, the search direction matrix of this algorithm is generated from the memorized previous generation of population and random selecting; therefore, the search efficiency and accuracy of the optimal solution are enhanced.

The backtracking search algorithm also adopts three genetic operators of selection, mutation, and crossover to produce the experimental population; the fundamental framework is as follows:

Step 1 (initialization): (repeat)

Step 2 (selection-I): (experimental population creation)

Step 3 (mutation)

Step 4 (crossover): (end of the experimental population creation)

Step 5 (selection-II): (end of the repeat when the stop conditions are satisfied).

(1) *Initialization.* The uniform distribution expression (17) is introduced in the backtracking search algorithm to create the initial population P , in which P_i is the individual of the population P , N is the population size, and D is the dimension of the optimization problem, that is, the design parameter of the problem, $i = 1, 2, 3, \dots, N$, $j = 1, 2, 3, \dots, D$.

$$P_{ij} \sim U(\text{low}_j, \text{up}_j). \quad (17)$$

(2) *Selection-I.* As mentioned before, the search direction matrix is created by randomly choosing from the historical population which is memorized in the backtracking search algorithm. In selection-I, the historical population is initialized through formula (18), and then whether the historical population should be updated in accordance with the criterion (19) at the beginning of iteration is decided. Accordingly, the calculation efficiency is improved by means of the historical population recording and search direction navigating methods. Moreover, for the purpose of increasing the diversity of the individuals and avoiding premature

convergence, the random permutations are made for some individuals while the historical population is determined.

$$\text{old}P_{ij} \sim U(\text{low}_j, \text{up}_j), \quad (18)$$

$$\text{if } a < b \quad (19)$$

$$\text{then } \text{old}P := P \mid a, b \sim U(0, 1).$$

(3) *Mutation*. During the producing of each generation of experimental population, (20) is utilized to mutate the population and develop the preliminary form of experiment population. F in (20) is the coefficient of amplitude control to command the range of search direction matrix information acquisition. Obviously, the experimental population created in this step acquires parts of the favorable information from the historical population. Therefore, it plays a significant role in increasing the search efficiency of optimal solution.

$$\text{Mutant} = P + F \cdot (\text{old}P - P). \quad (20)$$

(4) *Crossover*. After the step of mutation, the experimental population undertook a crossover to get the ultimate form T of experimental population of this generation. Different with the traditional methods of crossover, a rather complicated nonuniform method is employed in this algorithm; the concrete operation is as follows: firstly, assign the mutated experimental population *Mutant* to the population T . Then generate a $N \times D$ binary integer matrix A . There are two preset ways randomly selected for the matrix generation, one is making use of the mix rate M , as presented in (21), and another is choosing an element stochastically to carry out the operation at one time, as (22). Eventually, conduct the crossover of experimental population according to the value of the elements in the binary integer matrix A . When the element in the binary integer matrix A is equal to 1, the elements in experimental population T are in line with the corresponding elements in population P . The variation rate of elements in population T and the individual value after crossover are controlled separately by the mix rate parameter M and boundary control mechanism.

$$\begin{aligned} A_{i,u(1:[M\text{-rnd}\cdot D])} &= 0 \mid u \\ &= \text{permuting}(\langle 1, 2, 3, \dots, D \rangle), \end{aligned} \quad (21)$$

$$A_{i,\text{randi}(D)} = 0. \quad (22)$$

(5) *Selection-II*. The greedy choice strategy is introduced in this step to compare the fitness value of all the individuals in the experimental population T and the population P ; then the individuals T_i with the larger fitness value in the experimental population T are assigned to P_i to accomplish the update. Subsequently, decide whether the best value of fitness P_{best} in population P exceeds the global optimum value at present. If so, P_{best} is taken as the new global optimum value. Repeat the steps mentioned above until the stop conditions are met.

To sum up, the specific implementation procedures of backtracking search algorithm are induced as below.

Step 1 (initialization). Determine the input and output parameters, define the algorithmic function, randomly create the original population P and $\text{old}P$, and calculate the initial fitness value of population P .

Step 2. Repeat the following procedures.

Step 3 (selection-I). Develop the historical population.

$$\text{if } a < b \mid a, b \sim U(0, 1) \text{ then } \text{old}P := P.$$

Step 4 (mutation). Produce the preliminary experimental population.

$$\text{Mutant} = P + F \cdot (\text{old}P - P).$$

Step 5 (crossover). (a) Generate the binary integer matrix A .

$$\text{if } c < d \mid c, d \sim U(0, 1) \text{ then}$$

$$\begin{aligned} A_{i,u(1:[M\text{-rnd}\cdot D])} &= 0 \mid u = \\ &\text{permuting}(\langle 1, 2, 3, \dots, D \rangle) \end{aligned}$$

else

$$A_{i,\text{randi}(D)} = 0$$

end if

(b) Develop the experimental population T in accordance with the value of elements in matrix A .

$$\text{if } A_{ij} = 1 \text{ then } T_{ij} := P_{ij}$$

Step 6. When T_{ij} overflows the bound, adopt the boundary control mechanism to reassign.

$$T_{ij} = \text{rnd} \cdot (\text{up}_j - \text{low}_j) + \text{low}_j$$

Step 7 (selection-II). Greedily select the individuals with larger fitness value to form the population P .

Step 8. When the best value of fitness P_{best} in population P exceeds the current global optimum value; consider the P_{best} as the new global optimum value. Output the optimal result.

Step 9. End the repeat while the stop conditions are satisfied.

4.2. *The Improved Backtracking Search Algorithm*. The original backtracking search algorithm can memorize the historical population and the search direction to enhance the calculation efficiency. Nonetheless, the search effect depends on the ability of acquiring the favorable information from the historical population to find a better solution. In consequence, with the purpose of further improving the search ability, enlightened by the particle swarm optimization algorithm, the selection step of backtracking search algorithm is modified. [47] Taking advantage of global optimum value information, the historical population developing formula (18) is replaced by the following equation:

$$\begin{aligned}
\text{old}P_{ij} &= P_{ij} + \phi_{ij} (P_{ij} - P_{kj}) \\
&+ c_1 \text{rand}(0, 1) (P_j^{\text{gbest}} - P_{ij}) \\
&+ c_2 \text{rand}(0, 1) (P_j^{\text{best}} - P_{ij}),
\end{aligned} \quad (23)$$

where ϕ_{ij} is the random real number ranging from -1 to 1 and follows the uniform distribution. P_j^{gbest} is the j th element of global optimum value found at present. P_j^{best} is the j th element of optimum value obtained by the current iteration. c_1 and c_2 are two positive constants.

Furthermore, to avoid falling into the local optimal solution and increasing the diversity of population, when the global optimum value cannot proceed to be optimized and the stop conditions are not satisfied, randomly create the population P according to (24) and then mutate the i th solution of this population in accordance with (25) or (26) by using the global optimum value information. Finally, compare the fitness value of the i th solution in randomly created population, that is, P_i^{rand} and the fitness value of the i th solution in mutated population, that is, P_i^{mutate} ; choose the one with larger fitness value as the new population.

$$P_{ij}^{\text{rand}} = P_j^{\text{min}} + \text{rand}(0, 1) (P_j^{\text{max}} - P_j^{\text{min}}). \quad (24)$$

When $\text{rand}(0, 1) \leq 0.5$,

$$\begin{aligned}
P_{ij}^{\text{mutate}} &= P_{ij} + \text{rand}(0, 1) \times \left(1 - \frac{\text{iter}}{\text{iter}_{\text{max}}}\right)^\theta \\
&\times (P_j^{\text{gbest}} - P_{ij}).
\end{aligned} \quad (25)$$

When $\text{rand}(0, 1) > 0.5$,

$$\begin{aligned}
P_{ij}^{\text{mutate}} &= P_{ij} + \text{rand}(0, 1) \times \left(1 - \frac{\text{iter}}{\text{iter}_{\text{max}}}\right)^\theta \\
&\times (P_j^{\text{best}} - P_{ij}),
\end{aligned} \quad (26)$$

where θ is the scale parameter and the value is the positive integer commonly ranging from 2 to 5.

Significantly, the backtracking search algorithm is the continuous intelligent optimization algorithm. It cannot be directly applied to the discrete gantry crane scheduling and storage space allocation problem in railway container terminal. Therefore, the permutation sequence approach is adopted in this paper to settle this problem. Through the permutation sequence approach, the structure of solution is recoded to denote the sequence of unloading tasks in container train. For example, Figure 4 demonstrates the encoding approach for 6 tasks. Correspondingly for decoding, the unloading tasks are assigned to the earliest available gantry cranes and the appropriate storage spaces on the basis of the obtained sequence; the procedures are presented in Algorithm 1.

The individual of population is denoted by the task numbers 1–6.

The decoding procedures of above-mentioned algorithm are present as in Algorithm 1.

Sequence	①	②	③	④	⑤	⑥
Population	2	1	6	4	3	5
Operation time	10	20	15	30	10	25

FIGURE 4: Interpretation of the encoding scheme.

5. Computational Experiments

The established mathematical model and the improved backtracking search algorithm are tested in this section by using three different cases which intend to settle the small size, the medium size, and the large size problems. Meanwhile, as the classical evolutionary and metaheuristic methods, the genetic algorithm (GA) and the artificial bee colony (ABC) are adopted to make the comparison. Subsequently, the results are analyzed comprehensively so as to obtain some significant conclusions. Each of the algorithms is run ten times for every case, and the reported results are the mean value of the ten calculations. The crossover and mutation probabilities of GA are ordered to be 0.4 and 0.1. All the computational procedures are implemented in MATLAB by the personal computer equipped Intel Core 2.4 GHz Processor and 2.0 GB RAM.

5.1. Instances Generation. For all the three cases, the storage mode in the main area is along with the railway track. The ratio between the containers unloaded to the main area and the containers transferred to the assistant area by the inner trucks is set as 8 : 2. The initial position of the gantry cranes is randomly selected in $[1, n]$. The traveling time of crane between the adjacent tasks t^0 is set to 7.2 s. The safety distance required between the adjacent gantry cranes δ is set to 1. Let the stop position factor α be 0.2. The traveling speed of the trolley of gantry crane v_t is 0.5 m/s; the drop speed of gantry crane with heavy load v_d is 0.2 m/s; the horizontal traveling time of the trolley of gantry crane to the inner truck t_h is 24 s, and the t_{im} and t_{ki} are 97 s and 137 s, respectively. By convention, three layers of each storage space are arranged in the main area, and the drop distances of the gantry crane with the heavy load (d_c) to the first, second, and third layers are 8.8 m, 6.3 m, and 3.8 m separately. Assume there are 52 storage spaces available at present, and the distance between the central axis of railway track and the m th storage space (d_{im}) is revealed in Table 1. Besides, the constants c_1 and c_2 in the backtracking search algorithm are all 1.5, and θ is set as 4.

Furthermore, the small size tests are conducted with the number of tasks ranging from 6 to 12, and the number of gantry cranes is 2 to 3. Similarly, as the medium size cases, all the combinations of $n = \{15, 16, 17, 18, 19, 20\}$ and $g = \{2, 3, 4\}$ are tested. For the large size problems, the number of tasks is chosen from $\{26, 27, 28, 29, 30, 40, 50\}$, and the number of gantry cranes is selected from $\{2, 3, 4\}$.

5.2. Results and Discussion. The computational results of the small size cases are provided in Table 2. As predicted,

```

Procedure: transform the tasks to the gantry crane schedule and storage space allocation
Inputs:  $n, g, M, t_g^0, t_g^o, \delta$  //  $M$  is the total amount of storage spaces in main area,  $t_g^0$  is the
earliest available time of gantry crane  $g$ ,  $t_g^o$  is the initial position of  $g$  (denoted by the task
number),  $t^o$  is the traveling time of crane between the adjacent tasks,  $\delta$  is the safety
distance required between the adjacent gantry cranes//
Output:  $C_{\max}$  //  $C_{\max}$  is the maximum completion time of all the unloading tasks of the container train//
begin
  calculate cmax0 and sol0 //find a good initial solution//
  let pop  $\leftarrow$  (popsize, dim, low, up) //dim =  $n$ //
  let fitness  $\leftarrow$  (pop,  $n, g, \text{bay}, O_i, \text{QCR}, \text{QCL}, t_{ij}, \text{QC-TS}, \text{pct}$ )
  set fitness(index) = cmax0, pop(index, :) = sol0 //using the initial solution to replace the worst
  solution in the population//
  set historical_pop = pop //historical_pop is swarm-memory of BSA//
  while non_iter  $\leq$  nonmax && epk  $\leq$  epoch do //epoch is the maximum iteration counter//
    if rand < rand,
      historical_pop = pop;
    end if
    calculate historical_pop,  $F$  and map; //  $F$  is the scale factor//
    if rand < rand,
      for  $i = 1:\text{popsize}$ ,
         $u = \text{randperm}(\text{dim})$ ;
        map( $i, u(1:\text{ceil}(\text{DIM\_RATE} * \text{rand} * \text{dim}))$ ) = 1;
      end for
    else
      for  $i = 1:\text{popsize}$ ,
        map( $i, \text{randi}(\text{dim})$ ) = 1
      end for
    end if //implement the SELECTION-I process//
    offsprings = pop + (map.* $F$ ).*(historical_pop-pop)
    offsprings = BoundaryControl(offsprings, low, up) //implement the MUTATION and
    CROSSOVER process//
    calculate fitnessoffsprings;
    ind = fitnessoffsprings < fitnesspop;
    fitnesspop(ind) = fitnessoffsprings(ind);
    pop(ind, :) = offsprings(ind, :);
    [best, ind] = min(fitnesspop);
    if best < globalminimum
      globalminimum = best
      globalminimizer = pop(ind, :);
      non_iter = 0;
    else
      non_iter = non_iter + 1;
    end if //implement the SELECTON-II process//
    best_iter = [best_iter globalminimum];
    epk = epk + 1;
  end while
end

```

ALGORITHM 1: The backtracking search algorithm calculation procedure.

TABLE 1: The distance between the central axis of railway track and the storage space.

Number of the locations	Distance (m)
1	16.27
2	19.00
3	25.10
4	27.82
5	6.47
6	19.00
7	8.91
8	16.27
9	37.92
10	19.00
11	25.10
12	16.27
13	6.47
14	8.91
15	35.18
16	19.00
17	25.10
18	37.92
19	27.82
20	6.47
21	16.27
22	35.18
23	6.47
24	19.00
25	27.82
26	8.91
27	25.10
28	6.47
29	16.27
30	27.82
31	8.91
32	19.00
33	8.91
34	16.27
35	37.92
36	25.10
37	27.82
38	37.92
39	25.10
40	19.00
41	27.82
42	8.91
43	16.27
44	25.10
45	6.47
46	8.91
47	19.00
48	37.92
49	27.82
50	16.27
51	37.92
52	6.47

the runtime presents an increasing trend. However, all the optimal solutions of these cases can be acquired within 2 seconds by using the improved backtracking search algorithm. Comparatively, the runtime of GA and ABC is less than BSA, but some of the cases cannot be solved optimally by these two algorithms.

It can be seen from the outcomes in Table 3 that the runtime of these three algorithms grows a lot when solving the medium size cases. Obviously, the performance of BSA is superior to GA and ABC for the medium size problems; even the runtime of BSA is 6.99 seconds which is a little longer than GA and ABC. In addition, when the number of gantry cranes increased, the performance of BSA is influenced to a certain extent. The reason is that, with the increasing of gantry cranes, the effect of interference related constraints becomes more prominent when the algorithm searches for the optimal solution. Therefore, to avert the local optimum, the runtime of BSA gets longer.

Table 4 gives the optimal solutions and the corresponding runtime of GA, ABC, and BSA for the large size cases. For tests 13 to 21, the runtime of all the algorithms has a sharp rise. Particularly, with regard to the case with $n = 50$ and $g = 4$, the runtime of BSA is up to 204.10 seconds. Nevertheless, the effectiveness of BSA is much more visible when working out the large size problems within an acceptable runtime horizon. The average value of BSA is 548.24 while GA is 569.24 and ABC is 555.71.

From small size, medium size, and large size tests in Tables 2, 3, and 4, the proposed BSA is proved to be a prominent algorithm to solve the gantry crane scheduling and storage space allocation problem in railway container terminal. In the meantime, some features of BSA also can be obtained during the computation: (1) owing to the characteristics of the problem, the number of gantry cranes has some influence on the performance of BSA. (2) The historical population developing method together with the population creation and mutation strategy is redesigned; thus, the devised BSA takes more time to implement the calculation compared with GA and ABC.

6. Conclusion

This paper contributes to research the gantry crane scheduling and storage space allocation problem in railway container terminal through providing a mathematical model. The stop position of railway train, the handling procedures, noncrossing requirement, the safety distance of gantry cranes, and the storage modes in the main area were comprehensively considered in the established model. Moreover, to be more in line with the practice, the completion time of a task contains both the operation time and the traveling time of gantry crane. Then a novel metaheuristic backtracking search algorithm (BSA) was proposed to find the optimal solution. To achieve the goals of improving the search ability, avoiding falling into the local optimal solution and increasing the diversity of population, the historical population developing method, the population creation technique, and the mutation strategy

TABLE 2: Results of the random small size cases.

Number of the cases	Size ($n \times g$)	GA		ABC		BSA	
		Value	Run time (s)	Value	Run time (s)	Value	Run time (s)
1	6×2	129	0.15	128	0.23	128	0.33
2	6×3	106	0.19	106	0.30	106	0.46
3	7×2	159	0.20	159	0.31	157	0.47
4	7×3	121	0.26	121	0.46	121	0.51
5	8×2	193	0.26	193	0.41	193	0.49
6	8×3	143	0.29	143	0.62	143	0.74
7	9×2	221	0.28	216	0.54	214	0.69
8	9×3	163	0.33	162	0.80	162	0.97
9	10×2	236	0.31	236	0.71	234	0.78
10	10×3	178	0.47	178	1.09	178	1.26
11	11×2	255	0.45	255	0.96	255	1.03
12	11×3	192	0.58	192	1.37	190	1.58
13	12×2	279	0.49	278	1.16	278	1.27
14	12×3	233	0.70	233	1.85	233	1.95
Average		186.29	0.35	185.71	0.77	185.14	0.90

TABLE 3: Results of the random medium size cases.

Number of the cases	Size ($n \times g$)	GA		ABC		BSA	
		Value	Run time (s)	Value	Run time (s)	Value	Run time (s)
1	15×2	350	2.25	347	2.83	343	4.47
2	15×3	241	2.59	238	3.64	234	5.26
3	15×4	247	2.93	236	4.41	229	6.13
4	16×2	372	2.41	367	2.97	364	4.95
5	16×3	265	2.78	263	4.12	263	5.61
6	16×4	244	3.26	239	5.65	230	7.02
7	17×2	381	2.57	381	3.36	381	5.19
8	17×3	289	3.04	280	4.98	274	6.04
9	17×4	243	3.63	236	6.34	232	7.78
10	18×2	411	2.76	404	4.02	401	5.50
11	18×3	298	3.37	291	5.96	285	7.06
12	18×4	247	4.01	243	7.82	239	8.93
13	19×2	439	2.98	428	5.03	426	6.47
14	19×3	305	3.69	297	6.74	293	7.91
15	19×4	254	4.47	248	9.71	243	10.42
16	20×2	483	3.26	472	5.83	467	7.03
17	20×3	336	4.05	329	7.85	326	8.86
18	20×4	257	4.88	253	10.17	249	11.20
Average		314.56	3.27	308.44	5.64	304.39	6.99

of BSA were improved. Eventually, the computational experiment was conducted to assess the performance of the devised algorithm. It turned out that the proposed BSA is effective in solving the gantry crane scheduling and storage space allocation model and can acquire the approximate optimal solution within an acceptable time. Meanwhile, compared

with GA and ABC, the BSA algorithm is more competitive when solving the medium and large size problems.

Furthermore, there are some related works that can be done for the following research. For example, take the railway container terminal as a whole, combine the front operation region to the back yard, and integrate truck scheduling into

TABLE 4: Results of the random large size cases.

Number of the cases	Size ($n \times g$)	GA		ABC		BSA	
		Value	Run time (s)	Value	Run time (s)	Value	Run time (s)
1	26 × 2	598	5.04	594	16.20	585	18.35
2	26 × 3	427	6.32	416	18.34	409	22.67
3	26 × 4	320	7.18	307	26.56	298	32.86
4	27 × 2	635	5.47	629	17.19	626	20.28
5	27 × 3	431	7.95	428	19.46	417	25.61
6	27 × 4	350	8.60	335	29.37	322	38.01
7	28 × 2	672	5.71	661	18.01	655	22.34
8	28 × 3	464	9.69	449	20.82	445	28.02
9	28 × 4	355	10.94	342	31.49	334	40.37
10	29 × 2	684	6.02	673	18.93	668	23.73
11	29 × 3	459	11.22	450	25.51	450	30.76
12	29 × 4	366	13.33	347	32.64	342	41.93
13	30 × 2	712	6.37	706	19.98	705	26.31
14	30 × 3	501	12.36	482	26.15	478	32.74
15	30 × 4	378	15.40	371	34.88	362	44.23
16	40 × 2	921	21.76	903	52.75	895	63.64
17	40 × 3	637	34.68	614	60.34	602	78.85
18	40 × 4	423	42.15	412	87.23	406	91.67
19	50 × 2	1206	36.47	1165	113.24	1146	102.88
20	50 × 3	797	63.19	784	129.95	773	151.36
21	50 × 4	618	81.23	602	162.67	595	204.10
Average		569.24	19.58	555.71	45.80	548.24	54.32

the gantry crane scheduling and storage space allocation problem.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

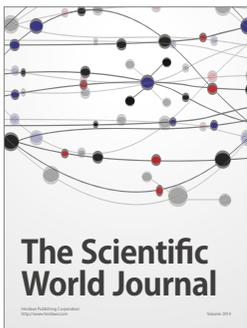
Acknowledgments

This research is partially supported by the Young Scholar Development Fund of SWPU (no. 201599010077) and the National Natural Science Foundation of China (no. 51405403 and no. 51205328).

References

- [1] L. M. Xu, L. Yang, X. Yang, and P. B. Zheng, "Study on technical development trend of railway container transport organization in China," *Railway Transport and Economy*, vol. 34, no. 12, pp. 18–25, 2012.
- [2] Railway Statistical Bulletin, Railway Administration of the People's Republic of China, http://www.nra.gov.cn/xwzx/zlzx/hytj/201603/t20160303_21466.shtml.
- [3] P. Guo, W. Cheng, and M. Zhang, "Simulation on the operation influence by external factors for railway container logistic center," in *Proceedings of the International Conference of Logistics Engineering and Management: Logistics for Sustained Economic Development—Infrastructure, Information, Integration (ICLEM '10)*, pp. 2915–2922, Chengdu, China, October 2010.
- [4] T. Benna and M. Gronalt, "Generic Simulation for rail-road container Terminals," in *Proceedings of the Winter Simulation Conference*, pp. 2656–2660, Miami, Fla, USA, December 2008.
- [5] B. J. Jeong and K. H. Kim, "Scheduling operations of a rail crane and container deliveries between rail and port terminals," *Engineering Optimization*, vol. 43, no. 6, pp. 597–613, 2011.
- [6] L. Wang and X. Zhu, "Rail mounted gantry crane scheduling optimization in railway container terminal based on hybrid handling mode," *Computational Intelligence and Neuroscience*, vol. 2014, Article ID 682486, 8 pages, 2014.
- [7] L. Wang, X. Zhu, and Z. Xie, "Storage space allocation of inbound container in railway container terminal," *Mathematical Problems in Engineering*, vol. 2014, Article ID 956536, 10 pages, 2014.
- [8] N. Boysen, F. Jaehn, and E. Pesch, "Scheduling freight trains in rail-rail transshipment yards," *Transportation Science*, vol. 45, no. 2, pp. 199–211, 2011.
- [9] C. F. Daganzo, "The crane scheduling problem," *Transportation Research Part B: Methodological*, vol. 23, no. 3, pp. 159–175, 1989.
- [10] D.-H. Lee, J. H. Chen, and J. X. Cao, "Quay crane scheduling for an indented berth," *Engineering Optimization*, vol. 43, no. 9, pp. 985–998, 2011.
- [11] N. Boysen, S. Emde, and M. Flidner, "Determining crane areas for balancing workload among interfering and noninterfering cranes," *Naval Research Logistics (NRL)*, vol. 59, no. 8, pp. 656–662, 2012.
- [12] Y. Zhu and A. Lim, "Crane scheduling with non-crossing constraint," *Journal of the Operational Research Society*, vol. 57, no. 12, pp. 1464–1471, 2006.
- [13] Y. Wang and K. H. Kim, "A quay crane scheduling algorithm considering the workload of yard cranes in a container yard,"

- Journal of Intelligent Manufacturing*, vol. 22, no. 3, pp. 459–470, 2011.
- [14] S. Nguyen, M. Zhang, M. Johnston, and K. Tan, “Hybrid evolutionary computation methods for quay crane scheduling problems,” *Computers & Operations Research*, vol. 40, no. 8, pp. 2083–2093, 2013.
- [15] P. Legato and R. Trunfio, “A local branching-based algorithm for the quay crane scheduling problem under unidirectional schedules,” *4OR*, vol. 12, no. 2, pp. 123–156, 2014.
- [16] J. H. Chen, D.-H. Lee, and J. X. Cao, “Heuristics for quay crane scheduling at indented berth,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 6, pp. 1005–1020, 2011.
- [17] O. Unsal and C. Oguz, “Constraint programming approach to quay crane scheduling problem,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 59, pp. 108–122, 2013.
- [18] C. Expósito-Izquierdo, J. L. González-Velarde, B. Melián-Batista, and J. M. Moreno-Vega, “Hybrid estimation of distribution algorithm for the quay crane scheduling problem,” *Applied Soft Computing*, vol. 13, no. 10, pp. 4063–4076, 2013.
- [19] R. I. Peterkofsky and C. F. Daganzo, “A branch and bound solution method for the crane scheduling problem,” *Transportation Research Part B: Methodological*, vol. 24, no. 3, pp. 159–172, 1990.
- [20] A. Lim, B. Rodrigues, F. Xiao, and Y. Zhu, “Crane scheduling with spatial constraints,” *Naval Research Logistics (NRL)*, vol. 51, no. 3, pp. 386–406, 2004.
- [21] D. Lee, H. Q. Wang, and L. Miao, “Quay crane scheduling with non-interference constraints in port container terminals,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 1, pp. 124–135, 2008.
- [22] F. Meisel and C. Bierwirth, “A unified approach for the evaluation of quay crane scheduling models and algorithms,” *Computers & Operations Research*, vol. 38, no. 3, pp. 683–693, 2011.
- [23] M. H. Hakam, W. D. Solvang, and T. Hammervoll, “A genetic algorithm approach for quay crane scheduling with non-interference constraints at Narvik container terminal,” *International Journal of Logistics Research and Applications*, vol. 15, no. 4, pp. 269–281, 2012.
- [24] M. Chlumecký, J. Buchtele, and K. Richta, “Application of random number generators in genetic algorithms to improve rainfall-runoff modelling,” *Journal of Hydrology*, vol. 553, pp. 350–355, 2017.
- [25] W. Su, H. Chen, F. Liu et al., “A novel comprehensive learning artificial bee colony optimizer for dynamic optimization biological problems,” *Saudi Journal of Biological Sciences*, vol. 24, no. 3, pp. 695–702, 2017.
- [26] L. Cui, G. Li, X. Wang et al., “A ranking-based adaptive artificial bee colony algorithm for global numerical optimization,” *Information Sciences*, vol. 417, pp. 169–185, 2017.
- [27] M. S. Kiran, “Particle swarm optimization with a new update mechanism,” *Applied Soft Computing*, vol. 60, pp. 670–678, 2017.
- [28] F. Jiang, H. Xia, Q. Anh Tran, Q. Minh Ha, N. Quang Tran, and J. Hu, “A new binary hybrid particle swarm optimization with wavelet mutation,” *Knowledge-Based Systems*, vol. 130, pp. 90–101, 2017.
- [29] P. Bipin and P. Rao, “Linearization of high power amplifier using modified artificial bee colony and particle swarm optimization algorithm,” *Procedia Technology*, vol. 25, pp. 28–35, 2016.
- [30] Z. Li, W. Wang, Y. Yan, and Z. Li, “PS-ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems,” *Expert Systems with Applications*, vol. 42, no. 22, pp. 8881–8895, 2015.
- [31] M. S. Kiran and M. Gündüz, “A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems,” *Applied Soft Computing*, vol. 13, no. 4, pp. 2188–2203, 2013.
- [32] A. Diabat and E. Theodorou, “An integrated quay crane assignment and scheduling problem,” *Computers & Industrial Engineering*, vol. 73, pp. 115–123, 2014.
- [33] M. F. Monaco and M. Sammarra, “Quay crane scheduling with time windows, one-way and spatial constraints,” *International Journal of Shipping and Transport Logistics*, vol. 3, no. 4, pp. 454–474, 2011.
- [34] C. Expósito-Izquierdo, B. Melián-Batista, and J. M. Moreno-Vega, “Variable neighbourhood search for the quay crane scheduling problem,” in *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA '11)*, pp. 463–468, November 2011.
- [35] P. Legato, R. M. Mazza, and R. Trunfio, “Simulation-based optimization for discharge/loading operations at a maritime container terminal,” *OR Spectrum*, vol. 32, no. 3, pp. 543–567, 2010.
- [36] B. Raa, W. Dullaert, and R. V. Schaeren, “An enriched model for the integrated berth allocation and quay crane assignment problem,” *Expert Systems with Applications*, vol. 38, no. 11, pp. 14136–14147, 2011.
- [37] M. H. Elwany, I. Ali, and Y. Abouelseoud, “A heuristics-based solution to the continuous berth allocation and crane assignment problem,” *Alexandria Engineering Journal*, vol. 52, no. 4, pp. 671–677, 2013.
- [38] Q. Zeng, Z. Yang, and X. Hu, “Disruption recovery model for berth and quay crane scheduling in container terminals,” *Engineering Optimization*, vol. 43, no. 9, pp. 967–983, 2011.
- [39] L. Song, T. Cherrett, and W. Guan, “Study on berth planning problem in a container seaport: Using an integrated programming approach,” *Computers & Industrial Engineering*, vol. 62, no. 1, pp. 119–128, 2012.
- [40] D.-H. Lee and H. Q. Wang, “Integrated discrete berth allocation and quay crane scheduling in port container terminals,” *Engineering Optimization*, vol. 42, no. 8, pp. 747–761, 2010.
- [41] C. Zhang, J. Liu, Y. Wan, K. G. Murty, and R. J. Linn, “Storage space allocation in container terminals,” *Transportation Research Part B: Methodological*, vol. 37, no. 10, pp. 883–903, 2003.
- [42] L. Chen and Z. Lu, “The storage location assignment problem for outbound containers in a maritime terminal,” *International Journal of Production Economics*, vol. 135, no. 1, pp. 73–80, 2012.
- [43] M. Rodriguez-Molins, M. A. Salido, and F. Barber, “Intelligent planning for allocating containers in maritime terminals,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 978–989, 2012.
- [44] G. C. Silva, L. Bahiense, L. SatoruOchi, and P. O. Boaventura-Netto, “The dynamic space allocation problem: applying hybrid GRASP and Tabu search metaheuristics,” *Computers & Operations Research*, vol. 39, no. 3, pp. 671–677, 2012.
- [45] M. Yu and X. Qi, “Storage space allocation models for inbound containers in an automatic container terminal,” *European Journal of Operational Research*, vol. 226, no. 1, pp. 32–45, 2013.
- [46] P. Civicioglu, “Backtracking search optimization algorithm for numerical optimization problems,” *Applied Mathematics and Computation*, vol. 219, no. 15, pp. 8121–8144, 2013.
- [47] P. Guo, W. Cheng, Z. Zhang, M. Zhang, and J. Liang, “Gantry crane scheduling with interference constraints in railway container terminals,” *International Journal of Computational Intelligence Systems*, vol. 6, no. 2, pp. 244–260, 2013.



Hindawi

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

