Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 7641670, 14 pages
https://doi.org/10.1155/2019/7641670

Research Article

Adaptive Autotuning Mathematical Approaches for Integrated Optimization of Automated Container Terminal

Meisu Zhong,[1] Yongsheng Yang,[1] Yamin Zhou,[1] and Octavian Postolache[2]

[1] *Institute of Logistics Science & Engineering, Shanghai Maritime University, Shanghai 201306, China*
[2] *Instituto de Telecomunicacões/ISCTE-IUL, Lisboa, Portugal*

Correspondence should be addressed to Yongsheng Yang; yangys_smu@126.com

Received 18 June 2019; Revised 16 September 2019; Accepted 16 October 2019; Published 27 November 2019

Academic Editor: Marco Mussetta

With the development of automated container terminals (ACTs), reducing the loading and unloading time of operation and improving the working efficiency and service level have become the key point. Taking into account the actual operation mode of loading and unloading in ACTs, a mixed integer programming model is adopted in this study to minimize the loading and unloading time of ships, which can optimize the integrated scheduling of the gantry cranes (QCs), automated guided vehicles (AGVs), and automated rail-mounted gantries (ARMGs) in automated terminals. Various basic metaheuristic and improved hybrid algorithms were developed to optimize the model, proving the effectiveness of the model to obtain an optimized scheduling scheme by numerical experiments and comparing the different performances of algorithms. The results show that the hybrid GA-PSO algorithm with adaptive autotuning approaches by fuzzy control is superior to other algorithms in terms of solution time and quality, which can effectively solve the problem of integrated scheduling of automated container terminals to improve efficiency.

1. Introduction

As the global economic growth accelerates, the demand of container transportation expands progressively. The automated terminal plays an important role in the global supply chain. However, energy consumption and carbon emission increase sharply, and how to reduce the energy and costs and how to improve the efficiency have been the goal of ports [1, 2]. In the fierce competition between ports, the automated operation mode can not only reduce labor costs but also improve the service level, which will attract more customers, to meet the port requirements of large scale and high efficiency. So the automated container terminals devote themselves to shortening the working time and advancing the economic efficiency, which have become the key to sustainable development of ports. However, due to the high cost of equipment in automated terminals, it is hard to increase the number of common used equipment to improve the efficiency [3], such as quay cranes (QCs), automatic guided vehicles (AGVs), and yard cranes (YCs). Therefore, the reasonable scheme of integrated scheduling of three kinds of equipment for loading and unloading container operations has become the key to improving efficiency of automated terminals.

Over the past decade, automated terminals have become the development trend of ports in China; for example, Xiamen Port has been gradually developed in practice operation, Shanghai Yangshan Deep Water Port and Qingdao Port have been finished and put into operation at present, and many ports are under transformation or construction of automated terminals. And specifically, in Xiamen Port, QCs use double trolley to replace single trolley for operation, AGVs have been used instead of trucks for horizontal transportation, and automated rail-mounted gantries (ARMGs) have almost displaced tyre cranes in yards. The reasonable scheduling scheme can make full use of resources, reduce the berth time of ships and the waiting time of equipment, and improve the efficiency of loading and unloading in ports. Therefore, it is of great significance to study the integrated scheduling of QCs, AGVs, and ARMGs to improve the efficiency and help to save energy in ports.

Our contribution is two-fold. First, most of the research about automated terminals is about single or two parts of equipment. This paper considers the actual situation of loading and unloading operation modes, putting forward a method to realize the integrated scheduling of QCs, AGVs, and ARMGs, with the objective to minimize overall operation time. This well-thought-out model is more reliable and realistic and has better schedule optimization.

The second is that most of the research studies of optimization algorithms in solving this kind of problem use basic metaheuristic algorithms. We perform extensive numerical experiments to clearly compare the accurate algorithm (branch and bound), the basic metaheuristic algorithms (genetic algorithm, particle swarm optimization, and bat algorithm), and the improved metaheuristic algorithm (hybrid GA-PSO algorithm and hybrid BAT-GA) to better illustrate this problem, obtaining optimum solutions in short time. And this result shows that the hybrid GA-PSO algorithm with a fuzzy logic controller for adaptive autotuning is effective and shows significant improvement compared with other methods.

This paper is organized as follows. Section 2 reviews the relevant literature. In Section 3, the integrated scheduling problem is formulated as a mixed integer programming (MIP) model. Several algorithms are proposed in Section 4. Numerical experiments based on simulation optimization are conducted to examine effectiveness of the proposed method and the performance of different algorithms in Section 5. Finally, some conclusions of this research are drawn in Section 6.

2. Literature Review

Automated terminal system is very complex, involving multiple subsystems. There are numerous studies on the scheduling and optimization of automated terminals such as the scheduling of a single equipment separately [4–7], the collaborative scheduling between two handling equipment [8–12], and the integrated scheduling between three handling equipment [13–16]. Because automated terminals are a complex integrated system, the research of single- or two-equipment scheduling is not significant for the improvement of operation efficiency of automated terminals, which also does not accord with the actual situation of automated terminals.

2.1. Accurate Algorithms of Automated Terminals. For accurate algorithms of container terminals, most of the research studies focused on branch and bound algorithm (B&B) and data envelopment analysis (DEA). For example, Pjevčević et al. [17] studied the collaborative scheduling of QCs and AGVs in automated terminals, used the DEA method to analyze the operation efficiency of unloading containers, and finally obtained the reasonable scheduling scheme. Jiang and Jin [18] researched the integrated scheduling problem of QCs, AGVs, and YCs, with the goal of minimizing cost of QCs, by the B&B method to solve the built mixed integer programming model, through numerical experiments showing that this method can improve the efficiency of terminals. There are also some comparisons of accurate algorithms and other algorithms. Alsoufi et al. [4], based on the scheduling problem of QCs, used the CPLEX to evaluate and compare the B&B method with the genetic algorithm, from small- to large-scale examples, but the results showed that the genetic algorithm had better performance.

As mentioned above, the accurate algorithm can get the exact solution, but when the scale is relatively large, it shows lack of feasibility, due to the influence of computation time and complex logic relation in program [19–21]. So more works chose to use metaheuristic algorithms.

2.2. Basic Metaheuristic Algorithm of Automated Terminals. For basic metaheuristic algorithms of container terminals, most of the literature mainly used genetic algorithm (GA), particle swarm optimization (PSO), and simulated annealing (SA), which are usually based on the perception or experience to construct algorithms. For example, Moghaddam et al. [22] proposed a mixed integer programming model of QC scheduling, and it was hard to obtain the optimal solution through optimization software in reasonable time, but the GA solved this problem and showed the certain superiority. Lu and Le [23] solved the collaborative scheduling model of AGVs and YCs by using PSO algorithm, and the results showed that this method could obtain the optimized solution. Combining berth allocation with the QC scheduling problem, Salhi et al. [24] compared the results of GA with those of the exact CPLEX and showed the rationality of using the GA to find the optimal solution. However, in recent years, there has been much research on the comparison of different algorithms of automated terminals. Homayouni et al. [14] researched the integrated scheduling of container terminals, adopting the GA and comparing it with the SA, and the result indicated that with the increase in the number of containers, the solving efficiency of the GA is obviously better than the SA. To minimize the delay time and energy consumption of ships, He et al. [25] used the GA and PSO algorithm to effectively solve this problem and discussed the solving ability of different algorithms.

Although the basic metaheuristic algorithms may not obtain the exact solution, it can obtain the suboptimal solution and the complexity of computing time is relatively low, and comparatively, its applicability and stability are better than those of the accurate algorithm [26–28]. However, taking more factors into consideration in automated terminals, the model is more complex and the basic metaheuristic algorithm has shown up the disadvantages such as longer computing time and slow convergence speed [29, 30].

2.3. Improved Metaheuristic Algorithm of Automated Terminals. For the application of improved metaheuristic algorithms to container terminals, many attempts have been made to improve different algorithms. For example, to solve the ship scheduling problem, Wang et al. [31] used the improved discrete chaotic PSO algorithm, and the simulation optimization results showed that this method was suitable for this problem. Kaveshgar et al. [32] proposed that the extended GA designed to solve the scheduling problem of QCs can find the optimal solution of a large-scale example in a shorter time. According to the working efficiency of QC scheduling, preparation time, and makespan, Legato et al. [33] established the mixed integer programming model, with the modified B&B method, through numerical experiments obtaining the high quality of scheduling solution. A genetic coupling heuristic algorithm by Gu et al. [34] was designed to solve the problem of cooperative scheduling of yards in terminals, and the performance of the model and proposed algorithm was confirmed with reference to numerous cases. Shu et al. [35] studied the scheduling of automatic stacker cranes in automated terminals, established a mathematical model, and proposed an improved multiobjective genetic algorithm, which satisfied the requirements of time sensitivity in decision, improving the intelligence degree of container terminals. Li et al. [36], with the minimum shipping distance and time as a target, has set up a cooperative scheduling model of berths and QCs, based on the improved PSO algorithm to test the performance of the model by numerical experiments. In order to improve the efficiency of the berth and horizontal transportation, Dkhil et al. [13] combined the multiobjective optimization model with Pareto promotion, using the modified adaptive tabu search algorithm to find the efficiency index. De et al. [37], considering the scheduling and path planning in terminals, proposed a mixed integer nonlinear programming model by using an effective search algorithm and, comparing it with PSO and genetic algorithms, illustrated the superiority of the proposed algorithm.

On the whole, improved metaheuristic algorithms are obviously better than metaheuristic algorithms in convergence speed and accuracy [38–41], which is mainly problem-oriented and shows improvement according to the structure and characteristics of the problem. So a majority of studies still selected the improved metaheuristic algorithms to solve this kind of problem in recent years.

3. Model Formulation

Automated container terminals have many parts, and this paper mainly studies the seaside operation area, the horizontal transport, and the storage yard, which are correlated and constrained mutually. The seaside operation area is equipped with the QCs for loading and unloading containers; AGVs are used in horizontal transportation, which can move containers from the QC to the yard; the storage yard is equipped with ARMGs. The yard has two ARMGs and AGV-mates in front of each block. Figure 1 shows the layout of a typical automated container terminal.

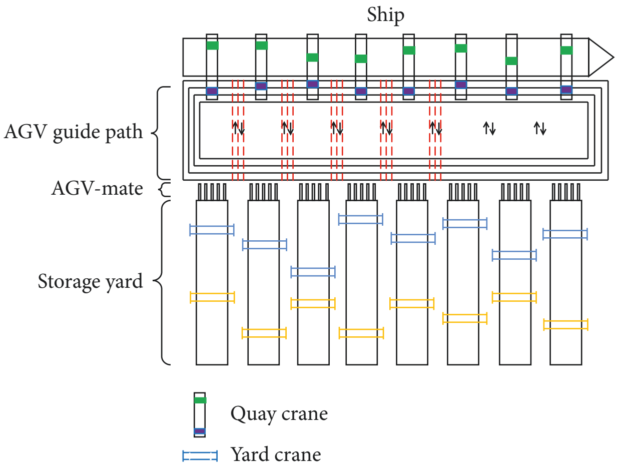


Figure 1: Layout of automated container terminal.

The main operational process of the loading and unloading of this problem is as follows: (1) The AGV received the unloading order (loading order) to transport the container from the QC to the assigned block (to transport the container from the AGV-mate to the assigned QC). Firstly, the main trolley of QC unloads the container from the ship to the transfer platform of QC and then the portal trolley unloads the containers from the transfer platform to the AGV, as shown in Figure 2. (2) The AGV transports the container to the AGV-mate in front of the specified block, which obeys the operation scheduling rules; the AGV can provide service to any QCs and blocks, and the path of transportation has been set. (3) The front ARMG puts the container from AGV-mate to the staging area in the block, and the back ARMG puts the container to the specified location of yard or the external truck, and then, the AGV will do the next task. The loading process is the opposite.

3.1. Assumptions

3.1. Assumptions. This paper assumes the following conditions for this scheduling problem of container terminals:

(1) The main trolley of QC loads the container to the transfer platform or unloads the container from the transfer platform to the ship, and its running time is related to the position of the container in ships, which is subjected to uniform distribution.

(2) The portal trolley loads the container to AGV or unloads the container from the transfer platform to AGV and its running time is fixed, without regard to the capacity of transfer platform and the number of AGV-mates.

(3) The QC and the ARMG can only load and unload one container at a time. AGVs can only transport one container at a time.

(4) The time for a crane (including both QCs and ARMGs) to release/pick up a container on/from the AGV/block is negligible, the time for the transfer of container from the AGV-mate to the storage yard is a fixed value, and the time for the transfer of containers from the storage yard to the specified location of yard or the external truck satisfies uniform distribution.

(5) In order to reduce the empty-loading ratio and improve the utilization of AGVs, the AGV completes a loading task and then to perform an unloading task or after performing an unloading task to complete a loading task.

3.2. Model Parameter

(1) Parameter set

U: set of import containers
L: set of export containers

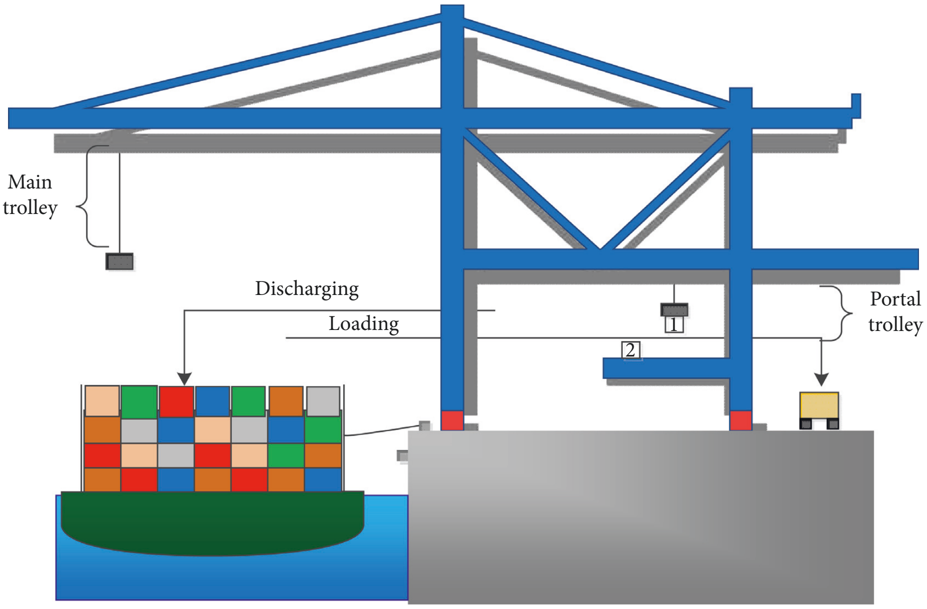


Figure 2: Operating modes of loading and unloading.

N: set of containers, $N = U \cup L$
V: set of AGVs
Q: set of QCs
B: set of blocks
S: set of dummy starting point of QCs
F: set of dummy finishing point of QCs
$O_s : O_s = Q \cup S$
$O : O = O_s \cup O_F$
$O_F : O_F = Q \cup F$

(2) Symbolic parameters

T_1 is the time of unloading the container by the portal trolley to the apron under the QC. It is also known as the time of discharge of the container in the apron to the transfer platform.
T_2 is the time recorded when the first ARMG puts the container on the storage area.
M is a very large positive number.
d_{ik} is the time when the QC k uses the main trolley to handle the ith container from the ship to the transfer platform. It is also known as the time when the QC k puts the ith container from the transfer platform to the ship.
t_i is the time taken by AGV to transport the ith container.
z_{ik} is the time when the second ARMG puts the ith container from the storage area to the end position after the QC k handling of the ith container.

(3) Decision variables

Not 0-1 variables:

f_{ik} is the time when the QC k has finished the ith container
g_{ik} is the time when the QC k starts to handle the ith container, $i \in N_k$
p_{ik} is the time when the QC k uses the portal trolley to get the ith container from AGV or the ith container to be put on the AGV
z_{ik} is the time when the QC k unloads the ith container on AGV-mate or loads the ith container from AGV-mate

0-1 variables:

x_{ikjl} is the AGV, which just handling the ith container of the QC k is scheduled to handle the jth container of QC l, $i \in U, j \in L$ or $i \in L, j \in U$
α_{ikc} is the AGV c to handle the ith container of QC k, $\alpha_{ikc} = 1$, otherwise $\alpha_{ikc} = 0$
θ_{ikb} is the ith container of QC k located in block b, $\theta_{ikb} = 1$, otherwise $\theta_{ikb} = 0$

3.3. Model. The mathematical programming model is set up by the above parameters, to minimize the loading and unloading time of the ship; this integrated problem can be formulated as follows:

$$\min : \max \left(F_k - S_k\right). \quad (1)$$

The objective of this model is to minimize the time difference between finish of the last task and start of the first task, which represents the loading and unloading time of the ship.

Subject to constraints,

$$F_k = \max_{k \in O_F} \{f_{ik}\}, \quad \forall i \in N_k, \quad (2)$$

$$S_k = \min_{k \in O_S} \{u_{ik}\}, \quad \forall i \in N_k. \quad (3)$$

Constraint (2) means that it selects the last time from the set of all containers task, as the the completing time of the last task. The constraint expressed in (3) ensures that the earliest time is considered from a time set of all containers as the starting time of the first task:

$$\sum_{l\in O_F}\sum_{j\in L} x_{ikjl} = 1, \quad \forall i \in U, k \in O_S, \tag{4}$$

$$\sum_{k\in O_S}\sum_{i\in U} x_{ikjl} = 1, \quad \forall j \in L, l \in O_F, \tag{5}$$

$$\sum_{c\in V} \alpha_{ikc} = 1, \quad \forall i \in N_k, k \in O_S, \tag{6}$$

$$\sum_{i\in N_k} \alpha_{ikc} = 1, \quad \forall c \in V, k \in O_F, \tag{7}$$

$$\sum_{b\in B} \theta_{ikb} = 1, \quad \forall i \in N_k, k \in O_S, \tag{8}$$

$$\sum_{i\in N_k} \theta_{ikb} = 1, \quad \forall b \in B, k \in O_S. \tag{9}$$

Constraint expressed in (4) ensures that the same AGV is used to complete the loading task of the QC, after it has finished the unloading task. Constraint (5) implies that after the QC finishes a loading task, the same AGV is used to complete the unloading task of the QC. Constraint (4) and Constraint (5) confirm the completion of the loading and unloading processes. Constraint (6) indicates that each loading and unloading task can only be performed by one AGV. Constraint (7) implies that every AGV only performs one task at a time. Constraint (8) means that the ARMG transports a container to be stacked in the assigned block of the storage yard. Constraint (9) expresses that each ARMG in block can only load and unload one container at a time:

$$g_{ik} + d_{ik} + T_1 \le p_{ik}, \quad \forall i \in U, k \in O, \tag{10}$$

$$p_{ik} + \sum_{c\in V} t_i \alpha_{ikc} \le z_{ik}, \quad \forall i \in U, k \in O, \tag{11}$$

$$z_{ik} + T_2 + \sum_{b\in B} h_{ik}\theta_{ikb} \le f_{ik}, \quad \forall i \in U, k \in O. \tag{12}$$

Constraints (10) to (12) represent the corresponding relationship of time between a ship starting and finishing a loading task. Constraint (10) represents the time when the AGV begins to transport the container from the QC, based on the time that the portal trolley of the QC takes to handle the container. Constraint (11) means the relationship of the time when the AGV starts to transport the container to the AGV-mate. Constraint (12) signifies the relationship between the time the container transportation is finished by the ARMG and the time the container is transported to the AGV-mate:

$$z_{ik} + \sum_{c\in V} t_i \alpha_{ikc} \le z_{jl} + \mathrm{M}\left(1 - x_{ikjl}\right), \quad \forall i \in U,\ j \in L,\ k \in O_S,\ l \in O_F. \tag{13}$$

Constraint (13) gives the relationship between the time when the same AGV finishes an unloading task and then starts the next loading task:

$$g_{jl} + \sum_{b\in B} h_{jl}\theta_{jlb} + T_2 \le z_{jl}, \quad \forall j \in L,\ l \in O, \tag{14}$$

$$z_{jl} + \sum_{c\in V} t_j \alpha_{jlc} \le p_{jl}, \quad \forall j \in L,\ l \in O, \tag{15}$$

$$p_{jl} + T_1 + d_{jl} \le f_{jl}, \quad \forall j \in L,\ l \in O_F. \tag{16}$$

Constraints (14) to (16) express the relationship between the time when the container is initially loaded from the block and the completion of loading at each time of shipment. Constraint (14) represents the time when the ARMG in the back of block obtains the container, which is less than the time needed for the AGV to obtain the container from the AGV-mate. Constraint (15) means that the starting time of AGV from the block does not exceed the time of the AGV arriving at the QC. Constraint (16) implies the time taken by the portal trolley in QC to obtain the container from the AGV which is no more than the time when the loading task is finished:

$$p_{jl} + \sum_{c\in V} t_j \alpha_{jlc} \le p_{ik} + \mathrm{M}\left(1 - x_{jlik}\right), \quad \forall i \in U,\ j \in L,\ k \in O_S,\ l \in O_F. \tag{17}$$

Constraint (17) represents the time relation between the completion of a loading task by the AGV and start of the next unloading task:

$$g_{(i+1)k} - g_{ik} = d_{ik} + d_{(i+1)k}, \quad \forall i \in U, k \in O, \tag{18}$$

$$g_{(i+1)k} - g_{ik} = h_{ik} + h_{(i+1)k}, \quad \forall i \in L, k \in O, \tag{19}$$

$$d_{ik} > 0,\ g_{ik} \ge 0,\ \mathrm{h}_{ik} > 0,\ p_{ik} > 0,\ f_{ik} > 0,\ z_{ik} > 0, \quad \forall i \in N_k,\ k \in O, \tag{20}$$

$$t_i \ge 0, \quad \forall i \in N_k. \tag{21}$$

Constraint (18) explains the time relation of the main trolley of QC, when it starts to unload two consecutive containers. Constraint (19) represents the time relationship of the ARMG in the back of block, when it loads two consecutive containers. Constraint (20) represents the range of time parameters. Constraint (21) expresses the range of time parameters about the AGVs' transportation.

4. Proposed Algorithm

4.1. Hybrid GA-PSO (HGA-PSO) Algorithm with Fuzzy Logic Controller to Adaptive Autotuning. The genetic algorithm (GA) and particle swarm optimization (PSO) are two well-known metaheuristic methods of optimization [42, 43]. The GA takes all the individuals in the population as the research object, which is a method to search the optimal solution by simulating the biological evolution process in nature. The solving problem of a population is according to the evolution by natural selection (probability optimization), each subsequent generation evolved better approximation. With the aid of crossover and mutation of genetic operators, generating the new population of solution and then to select individual by the fitness value, the selected individual will be more adaptable to the environment. The best individual after decoding can be used as the approximate optimal solution of the problem.

PSO is a parallel algorithm which is inspired by biological population characteristics and has strong robustness, simulating the random searching process of birds for hunting and extending it to multidimensional space. All particles by evaluating their fitness function to determine the current location, with the speed of particles to adjust the distance and direction of their flights, and every particle has the memory function and can remember the best searching site.

The flight speed of the particle can be dynamically adjusted by the flight experience of the particle and its companions; PSO can flexibly update the position of the particle in real-time by using equations (22) and (23), which shows its superiority [44]. Compared with the GA, which is easier to implement, the PSO does not require alteration of many parameters. The fitness of particle can be calculated as

$$v_k(t+1) = v_k(t) + b_1 r_1\left(h_{\mathrm{best}_k} - x_k(t)\right) + b_2 r_2\left(g_{\mathrm{best}} - x_k(t)\right), \tag{22}$$

$$x_k(t+1) = x_k(t) + v_k(t+1), \tag{23}$$

where $v_k(t)$ is the velocity of particle at the tth iteration; b_1 and b_2 are the acceleration constants; and r_1 and $r_2 \in [0, 1]$, which follows the uniform random distribution.

GA is an algorithm showing stability and applicability, and its remarkable performance has been proved in many research studies [45]. However, the metaheuristic GA cannot guarantee the optimal solution in all cases, due to the uncertain parameters. Yun and Gen [46] researched the adaptive autotuning strategy by changing the average fitness of GA, taking advantage of the mathematical optimization method, which can adaptively update crossover and mutation rates during the genetic search processes, to achieve the fuzzy logic control in the parent and their offspring. If this approach is applied in this problem, the average fitness at generation t can be set as follows:

$$\Delta f_{\mathrm{avg}}(t) = \overline{f_{P_s}(t)} - \overline{f_{O_s}(t)} = \frac{1}{P_s}\sum_{k=1}^{P_s} f_k(t) - \frac{1}{O_s}\sum_{k=1}^{O_s} f_k(t), \tag{24}$$

where P_s and O_s are the population size and offspring size that satisfy the constraints, respectively, and $f(t)$ is the adaptability function, representing the individual fitness.

We proposed that this hybrid GA-PSO is an improved metaheuristic algorithm. It not only combines the PSO parameters with GA operators but also uses the fuzzy logic controller with the initialization of parameters and particles, which through the adaptive auto tuning strategy can constantly adjust in the iteration process of population. The overall procedure of this method is illustrated in Figure 3 to obtain a better solution to this problem. The specific reasons of using this improved hybrid GA-PSO algorithm are as follows: (a) it has more adaptability and compatibility, and it accords with the characteristics and structure of this model in container terminals and is suited for dealing with many parameters and relative complex logic relations; (b) it makes use of advantages of different metaheuristic algorithms (PSO and GA) and has better global searching ability, especially in solving process; (c) it can improve the diversity of the population and the ability of continuous optimization. On the whole, it can promise a better performance in solving the actual problem compared with the other algorithms [47–49].

As discussed above, we consider the integrated scheduling of QCs, AGVs, and ARMGs in container terminals; the task encoding procedure of this is as follows: (1) applying the smallest position value (SPV) rule [48, 50]; (2) assigning the tasks' codes to the related particles; (3) identifying the operation sequence in each task. This paper uses integer encoding, assuming that there are 2 QCs, each QC having 4 loading containers and 4 unloading containers, and 4 AGVs to transport containers. The QCs and blocks have been matched, but the chromosomes are relatively long and multiple-point crossover and swapping mutation are used, as shown in Figure 4. We select multiple crossing points on chromosomes randomly and exchange the subsequence between the parent and the offspring. After the crossover, to effectively improve the chromosomes, the genes in chromosomes are needed to be checked and repaired by the swapping mutation.

The final encoded solution should estimate the objective function, which minimizes the loading and unloading time of the ship as given in equations (2) to (21). Equation (1) is used to evaluate the total fitness values of this problem.

4.2. Hybrid BAT-GA (HBAT-GA).

Bat algorithm (BAT) is used to simulate the random search of bats using the sonar to detect prey and avoid obstacle in nature, the process of optimization and searching is also the movement and prey of the individual bat in populations. Taking advantage of echolocation technology of bats, using different pulse frequencies and loudness, the BAT solved the problem by evaluating the value of fitness function to judge the location. It has the advantage of distributed and fast convergence but is still a new swarm intelligence algorithm, due to the instability of convergence and imprecise calculation in solving problems [51, 52]. The hybrid BAT-GA is a method which utilizes the crossover, mutation, and selection of GA, to increase the diversity of population and the ability of global search. Inspired from diverse metaheuristic algorithms, various improved kinds of BAT were designed and used to solve many optimization problems successfully [53].

In BAT, each bat is defined by its position X_i, velocity V_i, pulse frequency F_i, pulse loudness A_i, and pulse emission rate R_i to search the space; each bat updates the following equations:

```
procedure: hybrid GA-PSO for the integrated scheduling in ACTs
input: problem data, PSO parameters ($f(x)$, $b_1$, $b_2$, maxIter), GA parameters
($P_s$, $M_g$, $P_m$, $P_c$) and {$P_c(t-1)$, $P_m(t-1)$, $\Delta f_{avg}(t-1)$, $\Delta f_{avg}(t)$, $\varepsilon$, $\gamma$}
output: the best solution
begin
  $t \longleftarrow$ o, iteration number;
  intialize ($v_k$, $x_k$) for each particle $k$;// $P(t) = [x_k(t)]$: population
  evaluate $x_k(t)$ by decoding routine and keep the best solution;
  while $t \leq$ maxIter do
    for each particle $x_k$ in swarm do
      update velocity $v_k(t+1)$ and position $x_k(t+1)$ by (22) and (23);
      find the local best position of the particle $h_{best_k}$;
      evaluate $x_k(t+1)$ by encoding routine;
      find the global best position for the particle $g_{best_k}$;
      if $f(x_k(t+1)) < f(h_{best_k})$ then
        update $h_{best_k} = x_k(t+1)$
    end
    $g_{best} = \text{argmin}\{f(h_{best_k}, f(g_{best}))\}$
    creat offspring $C(t)$ from $x_k(t+1)$ by multi-point crossover routine;
    creat offspring $C(t)$ from $x_k(t+1)$ by insertion mutation routine;
    check and repair all offspring $C(t)$ for feasible solution;
    improve offspring $C(t)$ by hill-climb method routine;
    evaluate $C(t)$ by decoding routine and update the best solution by
    comparing with $g_{best}$;
    select $P(t+1)$ from $P(t)$ and $C(t)$ by routine wheel selection routine;
    tune parameter by the fuzzy logic controller to adaptive auto tuning by (24);
      if $\varepsilon \leq \Delta f_{avg}(t-1) \leq \gamma$ and $\varepsilon \leq \Delta f_{avg}(t) \leq \gamma$
        then increase $P_m$ and $P_c$ for next generation;
      if $-\gamma \leq \Delta f_{avg}(t-1) \leq -\varepsilon$ and $-\gamma \leq \Delta f_{avg}(t) \leq -\varepsilon$
        then decrease $P_m$ and $P_c$ for next generation;
      if $-\varepsilon \leq \Delta f_{avg}(t-1) \leq \varepsilon$ and $-\varepsilon \leq \Delta f_{avg}(t) \leq \varepsilon$
        then rapidly increase $P_m$ and $P_c$ for next generation;
    $t \longleftarrow t+1$;
  end
  output: the best solution
end;
```

Figure 3: Procedure for hybrid GA-PSO and the fuzzy logic controller for adaptive autotuning.

$$\begin{aligned}
F_i &= F_{\min} + \left(F_{\max} - F_{\min}\right)\beta, \quad \beta \in [0,1],\\
V_i^{t+1} &= V_i^t + \left(X_i^t - X_i\right)F_i,\\
X_i^{t+1} &= X_i^t + V_i^{t+1},\\
X_i^{\text{new}} &= X_i^{\text{old}} + \varepsilon A_i, \quad \varepsilon \in [-1,1],\\
A_i^{t+1} &= \alpha A_i^t, \quad 0 < \alpha < 1,\\
R_i^{t+1} &= R_i^0 - \exp(-\sigma t), \quad \sigma > 0.
\end{aligned} \tag{25}$$

Figure 5 shows the whole pseudo code of hybrid BAT-GA. The process should be repeated until reaching the maximum population. The gene with global optimum will be returned as the best fitness once meeting the optimal stopping criteria.

5. Computational Experiments and Discussion

To achieve the integrated scheduling of QCs, AGVs, and ARMG, we used the commercial software AIMMS 3.11 for small-sized problems, which obtained branch and bound (B&B) algorithm in CPLEX. Due to the increase in the number of containers, it was difficult to obtain optimal solutions. Therefore, we adopted the metaheuristic algorithm to obtain approximate optimal solution for large-sized problems. We also provided the comparison results between the metaheuristic algorithm and AIMMS to verify the effectiveness of the metaheuristic algorithm in small-sized problems. We implemented the various heuristic and improved heuristic algorithms in MATLAB 2018a on a computer with an Intel (R) Core (Tm) CPU@3.40 i7-6700 GHz and 4 GB RAM running the Windows 10 operation system. In order to reduce the deviation caused by the randomness of the heuristic algorithm, every problem was solved 20 times, where the average computation time (CPU time) and best fitness value (BFV) were used as the final results.

5.1. Initial Setting

(1) The number of loading and unloading containers varied from 1 to 1000, where 4~100 was considered as the small-sized problem and 100~1000 as the large-sized problem. We also considered the number of blocks and ARMGs in the ranges 2~8 and 4~16, while the number of considered AGVs varied from 8 to 24.

(2) In this work, we obtained the port operation values from the Xiamen Automated Container terminal. The processing time of the main trolley that placed the container onto the transfer platform followed the uniform distribution U (20, 40) s. The processing time of the portal trolley that placed the container onto the AGV from the transfer platform was fixed at 20 s, and the processing time of the ARMG localized in the front of the yard, which obtained the container from the AGV-mate and then unloaded it in the storage area, was fixed at 25 s. The processing time of ARMG localized in the back of yard that moved the container from the storage area to the assigned truck followed the uniform distribution U (20, 30) s. The path of horizontal transportation is set in advance. All of these values correspond with a real-time situation of the terminal.

(3) The GA parameters were set based on preliminary tests, including a crossover rate (P_c) of 0.9, mutation rate (P_m) of 0.1, population size (P_s) of 50, and maximum generation (M_g) of 500. The PSO parameters included alternate rate (P_a) [0.8, 2, 2] and precious rate (P_r) 0.05. The BAT parameters included α of 0.95, σ of 0.9, pulse loudness $A_i \in [0, 1]$, and pulse emission rate $R_i \in [0, 0.5]$.

5.2. Results for Small-Sized Problems. Thirteen small-sized experiments were performed, where the number of containers varied from 4 to 100. Table 1 shows that for the small-sized problems, the GA achieves approximately BFV compared to the B&B in terms of speed, where the CPU time of the former algorithm ranged from 3.46 to 12.66 s and that of B&B ranges from 12.37 s to 11534.82 s. The B&B cannot be applied for experiments with more than 40 containers, which increases the computation time exponentially. The results also confirm that B&B cannot solve the large-sized problems within a reasonable time frame. In addition, we observe that the difference of BFV between the GA and the B&B is small, where the maximum gap rate of BFV is 4.20%.

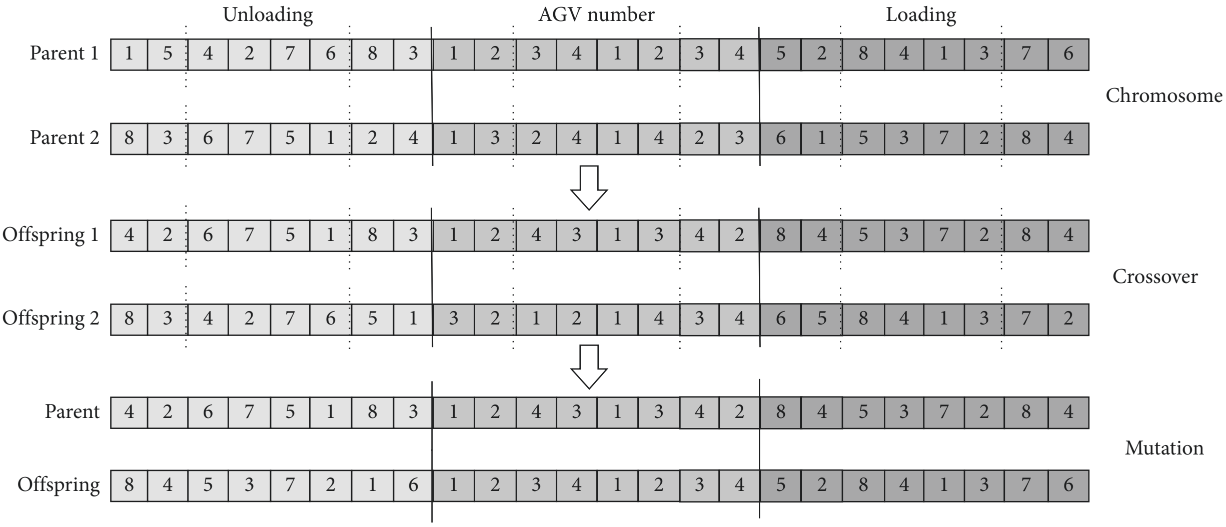

Unloading
AGV number
Loading
Parent 1 1 5 4 2 7 6 8 3 1 2 3 4 1 2 3 4 5 2 8 4 1 3 7 6
Parent 2 8 3 6 7 5 1 2 4 1 3 2 4 1 4 2 3 6 1 5 3 7 2 8 4
Chromosome
Offspring 1 4 2 6 7 5 1 8 3 1 2 4 3 1 3 4 2 8 4 5 3 7 2 8 4
Offspring 2 8 3 4 2 7 6 5 1 3 2 1 2 1 4 3 4 6 5 8 4 1 3 7 2
Crossover
Parent 4 2 6 7 5 1 8 3 1 2 4 3 1 3 4 2 8 4 5 3 7 2 8 4
Offspring 8 4 5 3 7 2 1 6 1 2 3 4 1 2 3 4 5 2 8 4 1 3 7 6
Mutation

Figure 4: Example of crossover and mutation operations.

```
procedure: Hybrid BAT-GA for the integrated scheduling in ACTs
input: Problem data, BAT parameters [position $X(i)$, velocity $V(i)$, pluse rates $R(i)$, pluse loudness
$A(i)$, emission rate $R(i)$] and GA parameters ($P_s$, $M_g$, $P_m$, $P_c$)
output: the best solution (fitness value)$f$
begin
  $t \longleftarrow o$,
  initialize [$X(i)$, $V(i)$] for each bat i; //$f$: population
  define the pulse frequency $F(i)$ at $X(i)$;
  calculate the fitness of each initial position $F(i)$ at $X(i)$ by (25);
  while ($i <$ maximum number of iterations) and (optimal solution not found) do
    for each bat $X(i)$ in population do
      generate the new solution by adjusting the $F(i)$ and $V(i)$ and adapting velocity $V(i)_r$ by (26)
      and (27);
      find and evaiuate a new soluton $f^*$;
      if (rand > $R(i)$) then
        generate a local solution around the selected best solution
      end if
      $i \longleftarrow i + 1$;
    end for
    send the local best to genetic population;
    for each genetic in population do
      generate a global best solution $f^{**}$ by crossover rate $P_c$ and mutation rate $P_m$;
      if (boundary $< f^{**}$) then
        generate a random solution with in the boundary
      end if;
      rank the chromosome to determine the best solution;
    end for
      $i \longleftarrow i + 1$
  end while
  evaluate fitness and met the criteria rule;
  output: the best solution
end;
```

Figure 5: Procedure for hybrid BAT-GA.

The GA can obtain the optimal fitness value for the first two cases. The BFV exhibits the same growth trend as the number of containers increases, as shown in Table 1.

5.3. Results for Large-Sized Problems. As the CPU time increases, it becomes more difficult to use the B&B method to solve the large-sized optimization problems. GA can generate an optimal solution in small-sized problems within a reasonable time frame, so we used some metaheuristic algorithms (PSO,BAT) to solve the large-sized problems about the automated terminals. We considered a number of containers in an interval of 150–1000 with 8–24 AGVs, 2–8 QCs, and 2–8 blocks to perform simulations. Table 2

Table 1: Result of computational experiments in small sizes.

No.	Containers	AGVs/QCs/block	B&B		GA		BFV gap rate (%)
			CPU (s)	BFV (s)	CPU (s)	BFV (s)	
1	4	2-2-2	12.37	157	3.46	157	0
2	8	2-2-2	21.56	338	2.74	338	0
3	10	3-2-2	18.90	280	1.88	281	0.35
4	15	4-2-3	24.67	384	4.59	386	0.52
5	20	4-2-3	244.89	433	4.36	438	1.15
6	30	5-3-4	573.78	394	8.85	405	2.79
7	40	6-3-3	11534.82	405	10.76	422	4.20
8	50	7-3-4	—	—	15.95	517	—
9	60	8-3-5	—	—	21.17	594	—
10	70	8-3-3	—	—	10.83	643	—
11	80	9-4-4	—	—	14.64	897	—
12	90	9-4-5	—	—	18.98	942	—
13	100	10-4-5	—	—	12.66	816	—

Table 2: Result of large-sized problems by heuristic algorithms.

No.	Container	AGV/QC/block	PSO		GA		BAT	
			CPU (s)	BFV	CPU (s)	BFV (s)	CPU(s)	BFV (s)
14	150	8/4/3	25.39	1556	23.51	1359	22.64	1402
15	200	9/4/4	35.20	1709	29.08	1408	31.29	1469
16	250	10/4/5	56.52	1673	42.86	1426	44.57	1644
17	300	11/4/6	79.67	1526	63.56	1759	68.15	1509
18	350	12/5/3	106.77	1782	89.22	1643	88.43	1688
19	400	13/5/4	143.29	1653	120.47	1726	134.98	2093
20	450	14/5/5	184.88	1864	157.80	1831	168.49	1849
21	500	15/5/6	233.96	1926	197.31	2150	219.38	2433
22	550	16/6/7	276.49	2364	246.80	2110	267.60	2042
23	600	17/6/5	319.76	2399	305.47	2331	311.97	2483
24	650	18/6/6	348.01	2697	359.71	2509	367.49	2608
25	700	19/6/7	449.15	2907	427.06	2776	422.61	2394
26	750	20/7/7	580.67	3028	500.72	2507	508.26	3077
27	800	21/7/7	690.77	3597	584.69	3144	645.78	3384
28	850	22/7/8	830.16	3960	657.24	3820	792.48	4022
29	900	23/7/8	1197.12	4838	749.50	3643	958.64	4282
30	950	24/8/8	1438.16	4697	857.14	3926	1197.65	4308
31	1000	24/8/8	1722.52	5213	1009.26	4418	1528.06	4836

presents the simulation results of PSO, GA, and BAT. Viewed generally, we can find that the GA performs better than others; the BFV of GA compared with PSO and BAT shows more optimization, with less time of CPU.

However, with the increase of the number of containers, the computation time has been increased sharply; the time was still long that we used the metaheuristic algorithm to solve this problem, which may not meet the needs of scheduling in current container terminals, so we tried to take advantage of the improved heuristic algorithms to solve this optimization problems faster and more stably. So the hybrid GA-PSO algorithm (HGA-PSO) and hybrid BAT-GA (HBAT-GA) were used to solve the large-sized problems, and the results are given in Table 3.

From our experiments, uniting Tables 2 and 3, we conclude that (1) our proposed hybrid GA-PSO algorithm performs more stably to obtain near-optimal solutions to large-sized problems (examples 15, 25, 30, and 31 in Table 3); from example 31, the number of containers is 1000 and the CPU time is less than 8 minutes (452.08 s), which comply with the requirements of scheduling operation in terminals; (2) the BFV increases with the number of containers, as well as it takes longer time to obtain the solution with the increased number of QCs, AGVs, and blocks; (3) the appropriate integrated scheduling scheme is very important; with the sharp increase of containers, it always choose to add the equipment number of QCs, AGVs, and blocks, but it may not truly reduce the completion time and improve overall operational efficiency (examples 16, 17, and 18 in Table 2; examples 17, 18, and 19 in Table 3).

Figures 6–8 show the performance comparisons for different algorithms. Figure 6 shows the simulation of 650 containers, 18 AGVs, 6 QCs, and 6 blocks; GA and PSO reach the convergence at the 350 and 420 generations, but HGA-PSO achieves the convergence before 130 generations and has a better optimal fitness value. The HGA-PSO combines the stability of GA and the velocity of PSO, which

Table 3: Result of large-sized problems by improved heuristic algorithm.

No.	Container	AGV/QC/block	HGA-PSO		HBAT-GA	
			CPU (s)	BFV (s)	CPU (s)	BFV (s)
14	150	8/4/3	14.38	1355	21.67	1286
15	200	9/4/4	17.26	1285	23.58	1312
16	250	10/4/5	21.64	1310	26.11	1461
17	300	11/4/6	26.55	1309	32.09	1606
18	350	12/5/3	32.49	1391	38.77	1477
19	400	13/5/4	38.86	1328	49.62	1521
20	450	14/5/5	44.91	1696	73.81	1397
21	500	15/5/6	51.13	1548	101.22	1653
22	550	16/6/7	58.76	1613	154.34	1797
23	600	17/6/5	70.41	1683	213.07	1768
24	650	18/6/6	82.37	1856	246.08	1932
25	700	19/6/7	106.55	1838	317.64	2288
26	750	20/7/7	138.84	2027	382.69	2711
27	800	21/7/7	182.43	2068	430.55	2486
28	850	22/7/8	236.11	2262	499.70	2864
29	900	23/7/8	340.05	2737	562.78	2688
30	950	24/8/8	396.71	3094	659.07	3808
31	1000	24/8/8	452.08	3275	766.04	3989

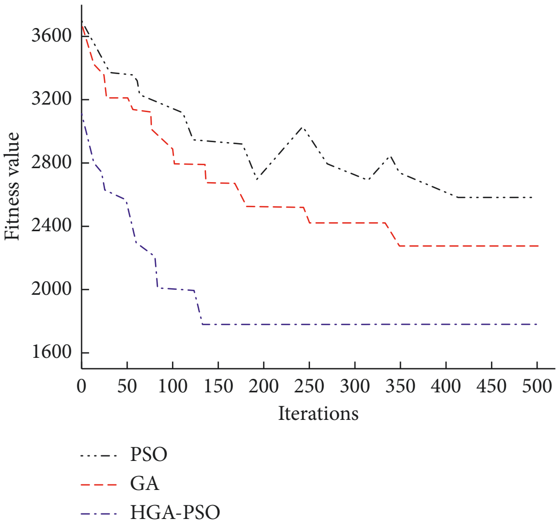


Figure 6: Typical convergence of PSO, GA, and HGA-PSO algorithms for example with 650 containers, 18 AGVs, 6 QCs, and 6 blocks.

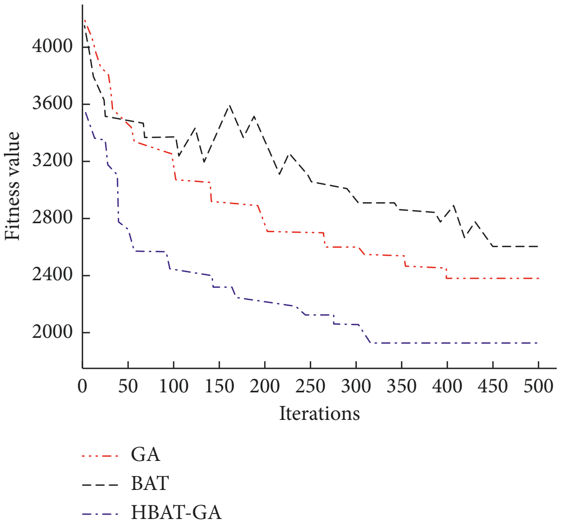


Figure 7: Typical convergence of GA, BAT, and HBAT-GA for example with 700 containers, 19 AGVs, 6 QCs, and 7 blocks.

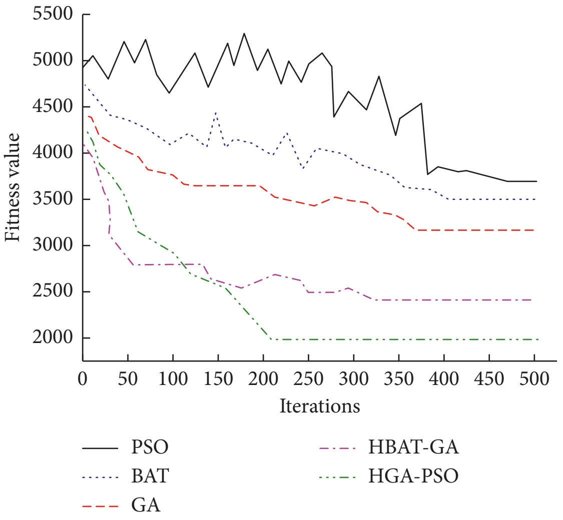


Figure 8: Performance of the five algorithms for 800 containers, 21 AGVs, 7 QCs, and 7 blocks.

performs surpassingly in terms of both solution time and quality. Figure 7 shows our simulation of 700 containers, 19 AGVs, 6 QCs, and 7 blocks; the HPSO-GA reaches the convergence at 320 generations, which has been compared with the GA and BAT, and shows better optimization in solving time and quality. But due to the instability of BAT, it has limited influence on the time of obtaining solutions. Figure 8 illustrates the performance of five algorithms. Comparison of heuristic algorithms such as GA, PSO and BAT shows that the GA is more stable and reliable, the convergence speed of the PSO is the slowest and unstable, and the BAT has certain fluctuations and slow speed in the process of getting solutions. The improved metaheuristic algorithms of HGA-PSO and HBAT-GA, compared with the basic metaheuristic algorithm (GA, BAT, and PSO), have a faster convergence speed. The HBAT-GA reaches the convergence at 320 generations, but HGA-PSO realizes the convergence almost at 210 generations and has a better optimal fitness value. The HGA-PSO outperforms the similar HBAT-GA in terms of both solution time and quality.

In order to test the stability of each algorithm in dealing with the fitness value in terms of iterations, the CPU time is reported and compared in Figure 9. It shows that the HGA-PSO outperforms the other algorithms in terms of execution time, which ensures to yield quality solution in reasonable runtime. The performance trend of 31 examples with the five algorithms in Tables 1–3 is depicted in Figure 10. There are some gaps between the fitness values of GA and HGA-PSO;

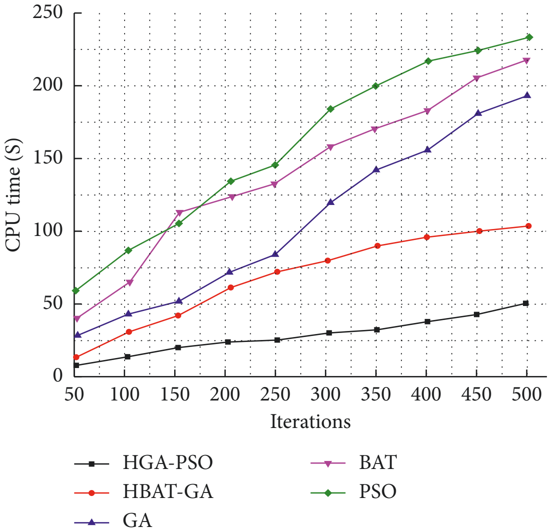


FIGURE 9: Evaluation of CPU time in terms of the number of iterations in five algorithms.

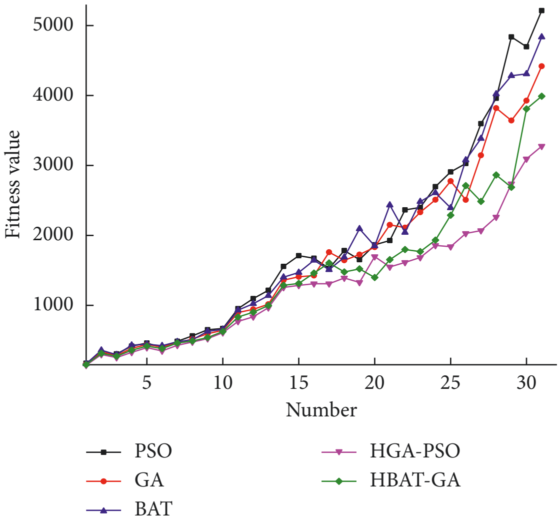


FIGURE 10: Performance trends of five algorithms with 31 examples.

the gap is less in the small-sized problems. However, the increase of gap as the number of containers increases in the large-sized problems has become a significant problem. In other words, the HGA-PSO is markedly superior to GA, PSO, BAT, and HBAT-GA in solving these large-scale calculation problems. These promising results are due mainly to the hybrid that balances disadvantages of the algorithm and enhances the ability and diversification in HGA-PSO.

To clarify the efficiency and robustness of the improved algorithms, taking example 31 as a case of statistical analysis data in Table 4, according to mean optimal fitness and CPU time, we compared the results of different algorithms with the actual operation (in Xiamen Port) and performed the Pareto optimization for the percent of deviation and improvement, that weighs the optimization in terms of solution time and quality. The result shows that the HGA-PSO has the least value in the percent of deviation and improvement compared with the actual operation. Although the advantages of HGA-PSO are reduced through the comprehensive balance in solving time and quality, the value of Pareto optimization of HGA-PSO is 18.98, which is smaller than the others and closer to the actual operation. We fully demonstrate that the proposed HGA-PSO has better robustness and effectiveness compared with other algorithms in solving this problem.

Taken together, our simulation experiments indicate that the proposed HGA-PSO is reliable in solving this problem of varying scales and can be readily applied to the integrated scheduling of QCs, AGVs, and ARMGs in current automated terminals.

6. Conclusion

An integrated scheduling of QCs, AGVs, and ARMGs was proposed in this paper for improving the working efficiency and service level of automated terminals. We established an

Table 4: Test results of example 31 of the optimization algorithm.

Algorithm	Mean optimal fitness	Percent of deviation D (%)	Mean CPU time	Percent of improvement I (%)	Pareto optimization $\partial D + \phi I$ (%)
Actual operation	3607	0	593.81	0	0
B&B	3382	6.24	Exceed capacity	—	—
PSO	5213	44.52	1722.52	190.07	141.55
GA	4418	22.48	1009.26	69.96	54.13
BAT	4836	34.07	1528.06	157.33	116.24
HGA-PSO	3275	9.20	452.08	23.87	18.98
HBAT-GA	3989	10.59	766.04	29.00	22.86

MIP model with the goal of minimizing the loading and unloading time of ships and then using the accurate algorithm (B&B), the basic metaheuristic algorithm (PSO, GA, and BAT), and the improved metaheuristic algorithm (HGA-PSO and HBAT-GA) to check the validity of the model.

We improved the HGA-PSO with adaptive autotuning approaches by fuzzy control and compared with the other algorithms (PSO, GA, BAT, and HBAT-GA) to validate the favorable convergence speed of HGA-PSO. Through a series of numerical experiments, we show that the proposed HGA-PSO has better robustness and effectiveness compared with other algorithms in terms of solution time and quality of this problem. When the number of containers was 1000, the best fitness value was 3275 s, the computation time was 452.08 s, and the value of pareto optimization was the smallest with 18.98. It illustrated this method can effectively reduce the working time and advance the operating efficiency in automated terminals.

As future work, the computation time of HGA-PSO is still relatively lengthy in solving large-sized experiments, rendering it inapplicable to dynamic real-time scheduling problems. We can try using GPU parallel computing to reduce the computation time. We also believe that applying other new improved metaheuristic algorithms, such as artificial intelligence or machine learning, may yield even better results than this HGA-PSO. Besides, the following research of automated terminals on the balance of throughput and energy, the AGVs path planing and effiency optimizationwill be more meaningful.

Data Availability

The automated container terminals data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Shanghai Science and Technology Commission (nos. 19595810700 and 18295801100) and Yunnan Science and Technology Commission (no. 2018IB022).

References

[1] G. Parise, L. Parise, L. Martirano, P. Ben Chavdarian, and A. Ferrante, "Wise port and business energy management: port facilities, electrical power distribution," *IEEE Transactions on Industry Applications*, vol. 52, no. 1, pp. 18–24, 2016.

[2] Y. Yang, M. Zhong, H. Yao, F. Yu, X. Yu, and O. Postolache, "Internet of things for smart ports: technologies and challenges," *IEEE Instrumentation & Measurement Magazine*, vol. 21, no. 1, pp. 34–43, 2018.

[3] Y. Yang, M. Zhong, and Y. Dessouky, "An integrated scheduling method for AGV routing in automated container terminals," *Computers & Industrial Engineering*, vol. 126, pp. 482–493, 2018.

[4] G. Alsoufi, X. Yang, and A. Salhi, "Combined quay crane assignment and quay crane scheduling with crane inter-vessel movement and non-interference constraints," *Journal of the Operational Research Society*, no. 4, pp. 1–12, 2017.

[5] P. Angeloudis and M. G. H. Bell, "An uncertainty-aware AGV assignment algorithm for automated container terminals," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 3, pp. 354–366, 2010.

[6] H. Rashidi and E. P. K. Tsang, "A complete and an incomplete algorithm for automated guided vehicle scheduling in container terminals," *Computers & Mathematics with Applications*, vol. 61, no. 3, pp. 630–641, 2011.

[7] H. Zhang and K. H. Kim, "Maximizing the number of dual-cycle operations of quay cranes in container terminals," *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 979–992, 2009.

[8] L. Chen, A. Langevin, and Z. Lu, "Integrated scheduling of crane handling and truck transportation in a maritime container terminal," *European Journal of Operational Research*, vol. 225, no. 1, pp. 142–152, 2013.

[9] S. Hartmann, "Scheduling reefer mechanics at container terminals," *Transportation Research Part E: Logistics and Transportation Review*, vol. 51, no. 51, pp. 17–27, 2013.

[10] Z.-H. Hu, J.-B. Sheu, and J. X. Luo, "Sequencing twin automated stacking cranes in a block at automated container terminal," *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 208–227, 2016.

[11] N. Kaveshgar and N. Huynh, "Integrated quay crane and yard truck scheduling for unloading inbound containers," *International Journal of Production Economics*, vol. 159, no. 3, pp. 168–177, 2015.

[12] L. Tang, J. Zhao, and J. Liu, "Modeling and solution of the joint quay crane and truck scheduling problem," *European Journal of Operational Research*, vol. 236, no. 3, pp. 978–990, 2014.

[13] H. Dkhil, A. Yassine, and H. Chabchoub, "Multi-objective optimization of the integrated problem of location assignment and straddle carrier scheduling in maritime container terminal at import," *Journal of the Operational Research Society*, vol. 2017, pp. 1–23, 2017.

[14] S. M. Homayouni, S. H. Tang, and O. Motlagh, "A genetic algorithm for optimization of integrated scheduling of cranes,

vehicles, and storage platforms at automated container terminals," *Journal of Computational and Applied Mathematics*, vol. 270, pp. 545–556, 2014.

[15] J. Luo and Y. Wu, "Modelling of dual-cycle strategy for container storage and vehicle scheduling problems at automated container terminals," *Transportation Research Part E: Logistics and Transportation Review*, vol. 79, pp. 49–64, 2015.

[16] X. Zhang, Q. Zeng, and Z. Yang, "Modeling the mixed storage strategy for quay crane double cycling in container terminals," *Transportation Research Part E: Logistics and Transportation Review*, vol. 94, pp. 171–187, 2016.

[17] D. Pjevčević, I. Vladisavljević, K. Vukadinović et al., "Application of DEA to the analysis of AGV fleet operations in a port container terminal," *Procedia-Social and Behavioral Sciences*, vol. 20, no. 6, pp. 816–825, 2011.

[18] X. J. Jiang and J. G. Jin, "A branch-and-price method for integrated yard crane deployment and container allocation in transshipment yards," *Transportation Research Part B: Methodological*, vol. 98, pp. 62–75, 2017.

[19] N. Al-Dhaheri and A. Diabat, "A Lagrangian relaxation-based heuristic for the multi-ship quay crane scheduling problem with ship stability constraints," *Annals of Operations Research*, vol. 248, no. 1-2, pp. 1–24, 2016.

[20] S. Emde and N. Boysen, "Berth allocation in container terminals that service feeder ships and deep-sea vessels," *Journal of the Operational Research Society*, vol. 67, no. 4, pp. 551–563, 2016.

[21] A. Santini, H. A. Friberg, and S. Ropke, "A note on a model for quay crane scheduling with non-crossing constraints," *Engineering Optimization*, vol. 47, no. 6, pp. 860–865, 2015.

[22] R. T. Moghaddam, A. Makui, A. Salahi et al., "An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports," *Computers & Industrial Engineering*, vol. 56, pp. 241–248, 2009.

[23] Y. Lu and M. Le, "The integrated optimization of container terminal scheduling with uncertain factors," *Computers & Industrial Engineering*, vol. 75, no. 1, pp. 209–216, 2014.

[24] A. Salhi, G. Alsoufi, and X. Yang, "An evolutionary approach to a combined mixed integer programming model of seaside operations as arise in container ports," *Annals of Operations Research*, vol. 4, pp. 1–30, 2017.

[25] J. He, Y. Huang, W. Yan, and S. Wang, "Integrated internal truck, yard crane and quay crane scheduling in a container terminal considering energy consumption," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2464–2487, 2015.

[26] A. Wang, M. Ranjbar, and N. Jamili, "Scheduling of loading and unloading operations in a multi stations transshipment terminal with release date and inventory constraints," *Computers & Industrial Engineering*, vol. 106, pp. 20–31, 2017.

[27] 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.

[28] O. Mustapha, A. E. H. Alaoui, and J. Boukachour, "An efficient genetic algorithm to solve the intermodal terminal location problem," *Endocrinology*, vol. 75, no. 5, pp. 586–591, 2015.

[29] P. Guo, W. Cheng, and Y. Wang, "Parallel machine scheduling with step-deteriorating jobs and setup times by a hybrid discrete cuckoo search algorithm," *Engineering Optimization*, vol. 47, no. 11, pp. 1564–1585, 2015.

[30] M. Wang, B. Li, G. Zhang et al., "Population evolvability: dynamic fitness landscape analysis for population-based metaheuristic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 99, p. 1, 2017.

[31] S. Wang, K. Zheng, J. Zheng et al., "An improved discrete particle swarm optimization for tugboat scheduling problem in container terminal," *Medical Education*, vol. 34, no. 7, pp. 573–579, 2010.

[32] N. Kaveshgar, H. Nathan, R. Saeed et al., "An efficient genetic algorithm for solving the quay crane scheduling problem," *Expert Systems with Applications*, no. 39, pp. 13108–13117, 2012.

[33] P. Legato, R. Trunfio, and F. Meisel, "Modeling and solving rich quay crane scheduling problems," *Computers & Operations Research*, vol. 39, no. 9, pp. 2063–2078, 2012.

[34] T. Gu, T. Gu, B. Lu et al., "Genetic mechanism-based coupling algorithm for solving coordinated scheduling problems of yard systems in container terminals," *Computers & Industrial Engineering*, vol. 89, pp. 34–42, 2015.

[35] F. Shu, W. Mi, X. Li, N. Zhao, C. Mi, and X. Yang, "A double-population genetic algorithm for ASC loading sequence optimization in automated container terminals," *Journal of Coastal Research*, vol. 73, pp. 64–70, 2015.

[36] M.-W. Li, W.-C. Hong, J. Geng, and J. Wang, "Berth and quay crane coordinated scheduling using multi-objective chaos cloud particle swarm optimization algorithm," *Neural Computing and Applications*, vol. 28, no. 11, pp. 3163–3182, 2017.

[37] A. Wang, S. K. Kumar, A. Gunasekaran, and M. K. Tiwari, "Sustainable maritime inventory routing problem with time window constraints," *Engineering Applications of Artificial Intelligence*, vol. 61, pp. 77–95, 2017.

[38] W. Tiwari, Y. Wang, P. Gupta et al., "A novel hybrid heuristic algorithm for a new uncertain mean-variance-skewness portfolio selection model with real constraints," *Applied Intelligence*, vol. 48, pp. 2996–3018, 2018.

[39] T. G. Crainic, M. Hewit, M. Toulouse et al., "Service network design with resource constraints," *Transportation Science*, vol. 50, no. 4, 2016.

[40] E. Li and H. Wang, "An alternative adaptive differential evolutionary algorithm assisted by expected improvement criterion and cut-HDMR expansion and its application in time-based sheet forming design," *Advances in Engineering Software*, vol. 97, pp. 96–107, 2016.

[41] P. Liu, G. Cui, Y. Xiao et al., "A new heuristic algorithm with the step size adjustment strategy for heat exchanger network synthesis," *Energy*, vol. 143, pp. 24–37, 2018.

[42] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, pp. 512–520, Wiley, Hoboken, NJ, USA, 2000.

[43] M. Gen, R. Cheng, and L. Lin, *Network Models and Optimization: Multiobjective Genetic Algorithm Approach*, pp. 710–715, Springer Nature, Switzerland, 2008.

[44] C. Sangsawang, K. Sethanan, T. Fujimoto, and M. Gen, "Metaheuristics optimization approaches for two-stage reentrant flexible flow shop with blocking constraint," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2395–2410, 2015.

[45] A. Gen, D. Srinivasan, S. Biswas, and T. Reindl, "A genetic algorithm-differential evolution based hybrid framework: case study on unit commitment scheduling problem," *Information Sciences*, vol. 354, pp. 275–300, 2016.

[46] Y. Reindl and M. Gen, "Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics," *Fuzzy Optimization and Decision Making*, vol. 2, no. 2, pp. 161–175, 2003.

[47] T. Jamrus, C.-F. Chien, M. Gen, and K. Sethanan, "Multistage production distribution under uncertain demands with

integrated discrete particle swarm optimization and extended priority-based hybrid genetic algorithm," *Fuzzy Optimization and Decision Making*, vol. 14, no. 3, pp. 265–287, 2015.

[48] T. Sethanan, C. F. Chien, M. Gen et al., "Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 1, pp. 32–41, 2017.

[49] M. Mussetta, "Genetical swarm optimization: self-adaptive hybrid evolutionary algorithm for electromagnetics," *IEEE Transactions on Antennas & Propagation*, vol. 55, no. 3, pp. 781–785, 2007.

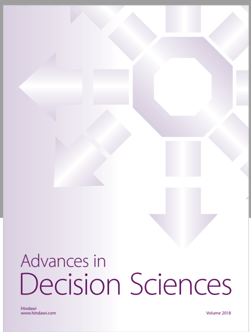
[50] X. Zhou, X. Zhao, and Y. Liu, "A multiobjective discrete bat algorithm for community detection in dynamic networks," *Applied Intelligence*, vol. 1, pp. 1–13, 2018.

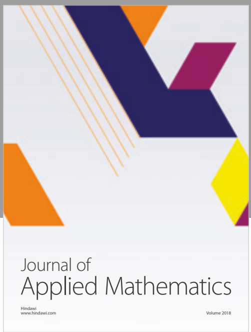
[51] K. E. Heraguemi, N. Kamel, and H. Drias, "Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies," *Applied Intelligence*, vol. 45, no. 4, pp. 1–13, 2016.

[52] N. S. Jaddi, S. Abdullah, and A. R. Hamdan, "Multi-population cooperative bat algorithm-based optimization of artificial neural network model," *Information Sciences*, vol. 294, pp. 628–644, 2015.

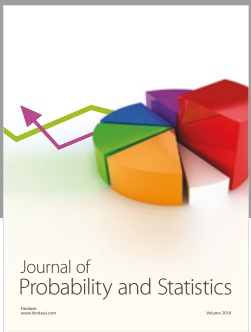
[53] T. Vidal, T. G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei, "A hybrid genetic algorithm for multidepot and periodic vehicle routing problems," *Operations Research*, vol. 60, no. 3, pp. 611–624, 2012.

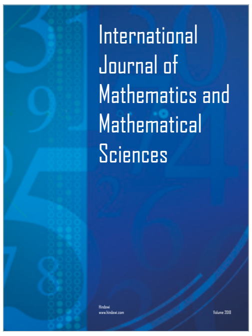
Advances in
Operations Research


Advances in
Decision Sciences

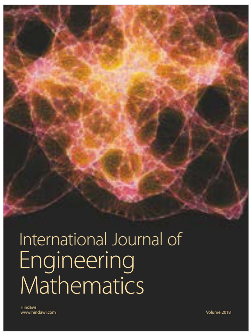

Journal of
Applied Mathematics

The Scientific
World Journal

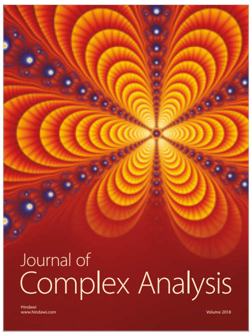

Journal of
Probability and Statistics

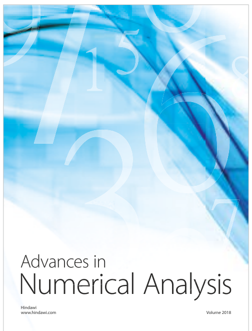

International
Journal of
Mathematics and
Mathematical
Sciences

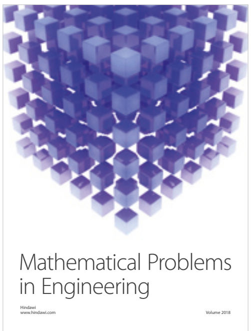
Journal of
Optimization

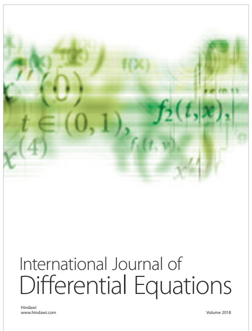

International Journal of
Engineering
Mathematics

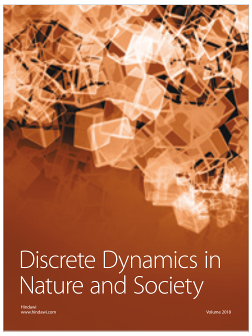
International Journal of
Analysis

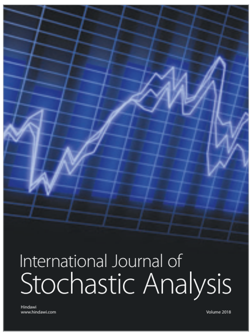

Journal of
Complex Analysis


Advances in
Numerical Analysis


Mathematical Problems
in Engineering

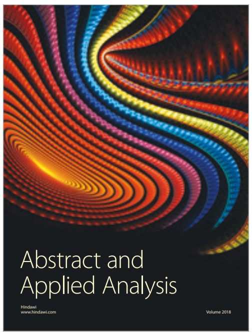

International Journal of
Differential Equations

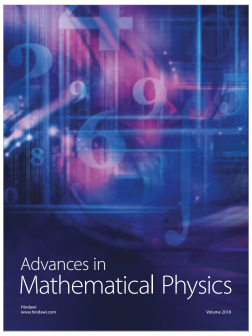

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics