Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 710565, 12 pages
http://dx.doi.org/10.1155/2015/710565

Research Article

Minimization of Delay and Travel Time of Yard Trucks in Container Terminals Using an Improved GA with Guidance Search

Z. X. Wang,[1] Felix T. S. Chan,[1] S. H. Chung,[1] and Ben Niu[2]

[1]*Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong*
[2]*College of Management, Shenzhen University, Shenzhen 518060, China*

Correspondence should be addressed to Ben Niu; drniuben@gmail.com

Received 4 June 2014; Revised 21 August 2014; Accepted 21 August 2014

Academic Editor: Yan-Wu Wang

Yard truck scheduling and storage allocation problems (YTS-SAP) are two important issues that influence the efficiency of a container terminal. These two problems aim to determine the routing of trucks and proper storage locations for discharging containers from incoming vessels. This paper integrates YTS and SAP as a whole and tries to minimize the weighted summation of total delay and total yard trucks travel time. A genetic algorithm (GA) is proposed to deal with the problem. In the proposed GA, guidance mutation approach and exhaustive heuristic for local searching are used in order to force the GA to converge faster and be steadier. To test the performance of the proposed GA, both small scale and large scale cases are studied. The results of these cases are compared with CPLEX for the small scale cases. Since this problem is an NP-hard problem, which CPLEX cannot solve, a simple GA is studied for comparison in large scale cases. The comparison demonstrates that the proposed GA can obtain near optimal solutions in much shorter computational time for small scale cases. In addition, the proposed GA can obtain better results than other methods in reasonable time for large scale cases.

1. Introduction

Container terminal plays a crucial role in logistics networks under the rapid development of globalization trade, transporting goods from one country to another. Efficiency of a terminal determines its competitiveness in the terminal industries. Accordingly, terminals have been devoting much effort in shortening the vessel staying time, meanwhile increasing the turnaround time by developing various decision support systems [1]. Terminal operations are usually classified into quay operations, (e.g., berth allocation, quay crane scheduling) and yard operations (e.g., yard truck scheduling, yard crane scheduling, and storage allocation) [2]. The objective of this paper is to study yard truck scheduling and storage allocation problems (YTS-SAP).

YTS refers to the scheduling of yard trucks to serve the transportation of export/import containers between the quay side and the yard side, and the corresponding route adoption. Meanwhile SAP refers to the allocation of storage locations for the import containers, which are discharged from incoming vessels. These two problems are known to be one of the most critical terminal operations as they directly influence the efficiency of the terminals [3]. For example, any delay of transporting an export container from the yard side to the quay side for the uploading operation definitely may cause delay of vessel departure. Thus, this must be avoided. Similarly, an inadequate storage plan will induce long transportation time, overload certain yard crane while idling others, and so forth [4, 5].

From the literature, it is known that YTS and SAP are highly interrelated. Back to 2008, Lee et al. [6] firstly proposed an integrated model simultaneously dealing with the YTS and SAP. However, in that model, loading operations from yard side to quay side are not being considered. Later on, Lee et al. [2] further improved the integrated model, by proposing a two-step approach, in which the first step deals with the truck-job assignment by using a hybrid insertion algorithm, while the second step deals with the storage allocation

problem. In this model, it is assumed that the number of import containers must be equal to the reserved storage location. However, this assumption may limit the practical applications. Therefore, in this paper, we will further enhance the model by bringing in the consideration of all available storage locations in the terminals. In other words, the number of storage locations can be larger than the number of import containers. This modified model will be more complicated than the previous one as the number of possible solution combinations increased dramatically.

YTS is proved to be NP-hard by Bish et al. [7]. Thus, the integrated problem is also an NP-hard problem. Accordingly, this paper proposed a new hybrid genetic algorithm (GA) to deal with the new integrated model, which can enable us to handle up to 300 containers with a reasonable computational time, greatly improved from the 20 containers in Lee et al.'s [2] model. The paper is organized as follows. Section 2 gives a literature review. Section 3 provides a mathematical model and problem description. Section 4 presents the proposed hybrid GA. Section 5 presents the computational experiments results and Section 6 concludes the paper.

2. Literature Review

Operation research in the area of storage allocation has been studied by many researchers. In the field of storage allocation of containers, K. H. Kim and H. B. Kim [8] addressed the problem of allocating storage space for import containers and a segregation policy was considered, in which the author analysed cyclic and dynamic arrival rates. The objective of this problem was to minimize the expected number of rehandled cases. K. H. Kim and H. B. Kim [9] discussed a method in which the optimal number of storage space and yard cranes for handling import containers could be determined. The authors proposed a cost model for decision making. A deterministic model and a stochastic model were proposed for the solution. Zhang et al. [10] studied the storage allocation problem. To solve the problem, a rolling-horizon approach was proposed in which the problem was decomposed into two levels. The first level aimed to balance two types of workloads among the blocks, while the second level focused on minimizing the total distance to transport the containers between their storage blocks and the vessel berthing locations. Lee et al. [11] studied the integration of terminal and yard allocation problem. The objective was to minimize the handling cost of transhipment containers in multiterminal systems. A two-level heuristic algorithm was proposed to deal with the problem.

In the area of scheduling problems in the yard side, Kim and Bae [12] discussed how to dispatch automated guided vehicles (AGVs) by utilizing information on locations and times of future delivery tasks. A mixed-integer programming model was proposed and the problem was solved by a heuristic algorithm. Ng et al. [3] considered the problem of scheduling a fleet of trucks in a container terminal to minimize the makespan. The loading and discharging jobs had sequence-dependent processing times and different ready times. The problem was formulated as a mixed integer program (MIP) and solved by using a GA. Nguyen and Kim [13] discussed the problem of dispatching automated lifting vehicles (ALV), and the problem was formulated in an MIP model, similar to multiple travelling salesman problems with precedence constraints and time windows. The problem was solved by a heuristic algorithm. Hu et al. [14] studied the performance of three types of transporting machines which are ground trolleys (GTs), transfer platforms (TPs), and frame trolleys (FTs). Zhao and Goodchild [15] explored the truck travel time reliability and the predictability. The truck routing choices were analysed by examining the relationship between the routing choice and route attributes. Yan et al. [16] investigated a knowledge-based system for yard crane scheduling problem. The proposed system was capable of making off-line planning and real-time scheduling. Javanshir and Seyedalizadeh Ganji [17] studied the problem of yard crane scheduling with noninterference constraints for single blocks. The problem was formulated as a mathematical model and solved by GA. Javanshir et al. [18] studied the problem of yard crane scheduling of multiple blocks in multiple planning periods. The problem was formulated as MIP model and solved by Lingo. He et al. [19] proposed a model to solve the yard crane scheduling problem based on the rolling-horizon technique. A hybrid parallel GA was proposed to solve the problem. Cao et al. [20] studied the integrated model for yard truck and yard crane scheduling problems. The handling sequence of outbound containers was determined in this model. The model was formulated as a MIP programming. Homayouni and Tang [21] investigated the coordination of crane scheduling and vehicle routing. A genetic algorithm is proposed for solving the mathematical model.

The yard truck scheduling problem and the storage allocation problem were studied separately in the past. Bish et al. [7] firstly studied the two problems but solved each problem separately. Bish et al. [7] considered the assumptions for each container with a number of potential locations in the yard where it could be stored and the container was moved from the vessel to the yard by using a fleet of vehicles, each of which could carry one container at a time. The problem was solved in two steps. The first step was to determine the location assignments by ignoring the vehicle schedule and the second step determined the vehicle schedule for the location arrangements obtained from the first step. The problem was solved by using a heuristic algorithm. Bish [22] further studied the problem of determining a storage location for each discharging container, dispatching vehicles to the containers, and scheduling the unloading and loading operations on each quay crane. The objective was to minimize the maximum turnaround time of a set of vessels. Bish et al. [23] developed easily implementable heuristic algorithms for solving the problem studied in 2005. Han et al. [24] studied the yard truck scheduling and storage allocation problems as a whole in transhipment terminals. A mathematical model was proposed and the model was solved by dedicated heuristic algorithms. Lee et al. [6] proposed an integer programming model to deal with the problem of yard truck scheduling and storage allocation. This paper considered the two problems as a whole instead of solving each aspect separately. The objective is to reduce congestion and idling time of the yard trucks in order to decrease the

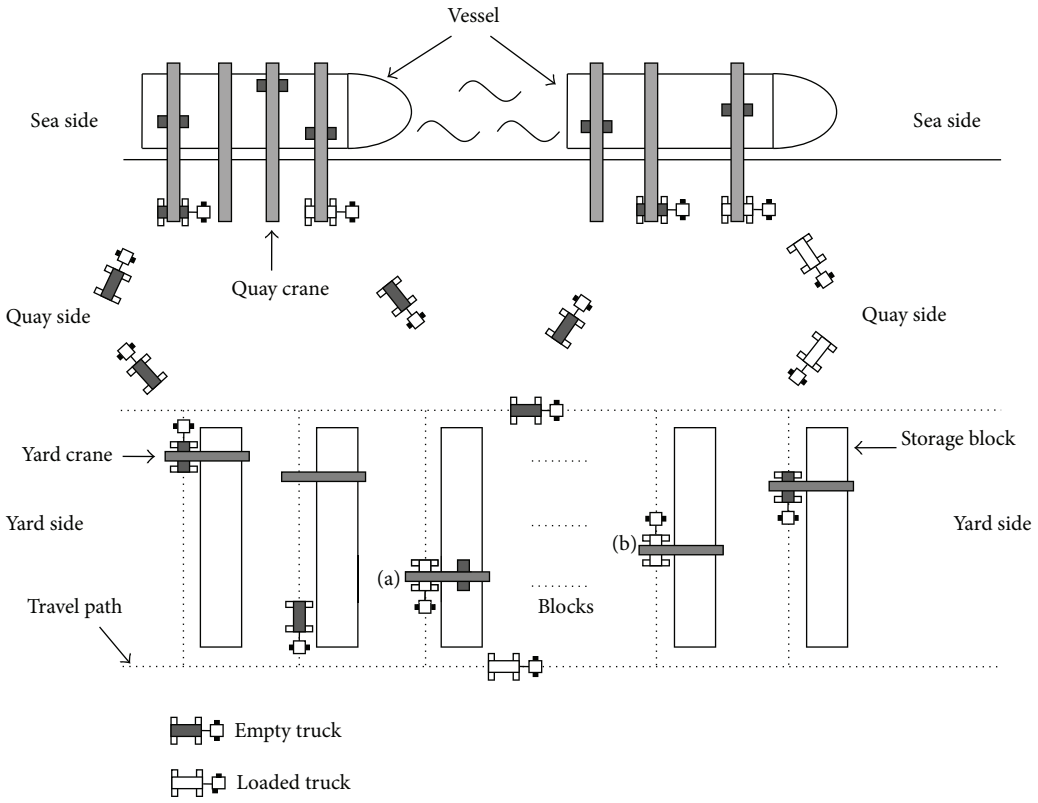

Figure 1: Outline of a container terminal.

makespan of the discharging containers. Later on, Lee et al. [2] further extended the previous study and proposed another integrated model for the yard truck scheduling and storage allocation problem. A hybrid insertion algorithm was proposed for the solution and 20 containers are considered in the computational experiments.

From the literature reviews we can clearly know that little work has been done on YTS-SAP considering all empty storage locations in the yard side and no literature study YTS-SAP for large scale instances.

3. Problem Description and Formulation

In this paper, the problem is how to schedule a fleet of trucks to load or discharge all the containers and determine the storage location for the discharging containers. First of all, we define the movement of a container from its origin to its destination as a request, denoted by i and j. There are two types of requests considered in this study: loading requests and discharging requests. For loading request, the origin is the location where a container is loaded onto a truck by a yard crane from a storage block in the yard side, while the destination is the location of the quay crane by which a container is loaded onto the vessel. For discharging request, the origin is the location of the quay crane by which a container is unloaded from a vessel, while the destination is the location where a container is unloaded from a truck to a storage block by a yard crane in the yard side as in Figure 1.

In terminal practice, a soft time window for each request $[a_i,b_i)$ is already predetermined by the terminal operator as a given data for YTS. The soft time window is a period of time which consists of the earliest possible time a_i and the due time b_i. A container can only be served after the earliest possible time and b_i can be viewed as penalty. We define the processing time (loaded travel time) t_i as the period of time that a truck processes a request i from its origin to its destination, and the setup time (empty travel time) s_{ij} is the period of time that a truck spends from the destination of the current request i to the origin of next request j. We also define the starting time w_i of request i as the time when it starts, and the completion time c_i of request i is the time when it finishes. The difference between the completion time c_i and b_i of request i is the delay of request i. Completion time of request i is $c_i = w_i + t_i$. Delay of request i is $d_i = \max\{0, c_i - b_i\}$. If request j is the successive request of request i served by the same truck, the starting time of request j is $w_j = \max\{w_i + t_i + s_{ij}, a_j\}$.

The following assumptions are made in this study.

(1) There are limited numbers of trucks and one truck serves only one route. We use dummy requests l_r and k_r to represent the initial and finial status of each route.

(2) The trucks travel between any pair of locations along the shortest and same path, so the travel times are symmetric. For example, in Figure 1, the truck travels from location (a) to location (b) and location (b) to location (a) along the same and shortest travel path.

(3) The number of storage locations is more than or equal to the number of discharging containers.

(4) The yard crane and quay crane can serve the yard truck once the yard truck arrives at the yard crane or quay crane. This means that the yard crane and quay crane are always available.

(5) Congestions among yard trucks on a guide route are not considered.

The following notations are used to describe the problem studied in this paper.

Indices

i, j: Index of request, $i \neq j$.

r: Index of route.

p, q: Index of location.

k: Index of storage location.

Problem Data

$\tau_{p,q}$: The travel time between each pair of locations (p, q).

o_i: The origin of request i.

e_i: The destination of request i.

ζ_k: The location of storage location k.

α_1: The weight of total delay of requests.

α_2: The weight of total travel time of yard trucks.

Set of Indices

J^-: The set of discharging requests with cardinality of $|J^-| = n^-$.

J^+: The set of loading requests with cardinality of $|J^+| = n^+$.

J: The set of all requests, $J = J^- \cup J^+$, with cardinality of $|J| = n$.

J': The union set of all requests and initial status, $J' = J \cup \{l_r\}$.

J'': The union set of all requests and final status, $J'' = J \cup \{k_r\}$.

R: The set of routes, $|R| = m$.

M: The set of locations of the loading containers.

N: The set of locations of the discharging containers.

K: The set of the storage locations.

L: The union set of the locations of the loading containers, the locations of the discharging containers, and the storage locations. $L = M \cup N \cup K$.

Decision Variables

$x_{ik} = 1$, if container i is allocated to storage location k.

$= 0$, otherwise.

$y_{ij} = 1$, if request i is connected to request j in the same route.

$= 0$, otherwise.

w_i: The starting time of request i.

c_i: The completion time of request i.

d_i: The delay of request i.

t_i: The processing time of the yard trucks from the origin of request i to the destination of request i. $t_i = \tau_{oi,ei}$, if request i is a loading request; $t_i = \tau_{oi,\zeta k}$, if request i is a discharging request and allocated to storage location k.

s_{ij}: The setup time of the yard trucks from the destination of request i to the origin of request j. $s_{ij} = \tau_{ei,oj}$, if request i is a loading request; $s_{ij} = \tau_{\zeta k,oj}$, if request i is a discharging request and allocated to storage location k.

The objective is to schedule the yards trucks and allocation of the loading and discharging containers, aiming at minimizing the weighted summation of the total delay and the total yard trucks travel time as model in (1). The problem formulation is modified based on the model provided by Lee et al. [2]. In our model, we consider all the available storage location in the yard side; however, Lee et al. [2] only consider the reserved storage locations for discharging containers, which means storage locations and discharging containers are equal in amount. The revised model is as shown in the following:

$$\text{Min:} \quad Z = \alpha_1 \sum_{i \in J} d_i + \alpha_2 \left(\sum_{i \in J} t_i + \sum_{i,j \in J} s_{ij} y_{ij} \right), \tag{1}$$

subject to

$$\sum_{i \in J^-} x_{ik} \leq 1 \quad \forall k \in K \tag{2}$$

$$\sum_{k \in K} x_{ik} = 1 \quad \forall i \in J^- \tag{3}$$

$$\sum_{j \in J''} y_{ij} = 1 \quad \forall i \in J' \tag{4}$$

$$\sum_{i \in J'} y_{ij} = 1 \quad \forall j \in J'' \tag{5}$$

Table 1: Sample data of containers.

Container ID	Origin	Destination	Time window a (unit: second)	Time window b (unit: second)	Type
1	(1035, 971)	(60, 665)	1362	1639	L
2	(108, 895)	(1464, 336)	716	1214	L
3	(359, 689)	(748, 1353)	284	634	L
4	(148, 391)	(1246, 312)	1320	1745	L
5	(800, 1180)	(113, 1287)	1201	1522	L
6	(767, 1015)		8	293	D
7	(496, 1210)		490	855	D
8	(1485, 414)		1160	1486	D
9	(99, 1440)		107	325	D
10	(130, 1498)		323	610	D

$$w_i \geq a_i \quad \forall i \in J' \cup J'' \tag{6}$$

$$d_i \geq w_i + t_i - b_i \quad \forall i \in J' \cup J'' \tag{7}$$

$$w_j + M\left(1 - y_{ij}\right) \geq w_i + t_i + S_{ij} \quad \forall i \in J',\ \forall j \in J'' \tag{8}$$

$$t_i = \tau_{oi,ei} \quad \forall i \in J^+ \tag{9}$$

$$t_i = \sum_{k \in K} \tau_{oi,\zeta k} x_{ik} \quad \forall i \in J^- \tag{10}$$

$$S_{ij} = \tau_{ei,oj} \quad \forall i \in J^+,\ \forall j \in J \tag{11}$$

$$S_{ij} = \sum_{k \in K} \tau_{oi,\xi i} x_{ik} \quad \forall i \in J^-,\ \forall j \in J \tag{12}$$

$$x_{ik}, y_{ij} \in \{0, 1\}, \quad \forall i \in J',\ \forall j \in J'',\ \forall k \in K \tag{13}$$

$$\begin{gathered} w_i \in \mathbf{R} \quad \forall i \in J' \cup J'' \\ t_i \in \mathbf{R} \quad \forall i \in J \\ S_{ij} \in \mathbf{R} \quad \forall i \in J,\ \forall j \in J \\ d_i \geq 0 \quad \forall i \in J' \cup J''. \end{gathered} \tag{14}$$

Constraints (2) ensure that each storage location will be assigned with at most one discharging container. Constraints (3) ensure that each discharging container will be assigned with one storage location. Constraints (4) ensure that $y_{ij} = 1$ if the yard truck processes request j after request i. Constraints (5) ensure that $y_{ij} = 1$ if the yard truck processes request i before request j. Constraints (6) ensure that requests can only be served after the earliest possible time. Constraints (7) calculate the delay of each request. Constraints (8) give the relationship of the starting time of a request and that of its successor. Constraints (9) calculate the travel time of the loading requests. Constraints (10) calculate the travel time of the discharging requests. Constraints (11) calculate the setup time of the loading requests. Constraints (12) calculate the setup time of the discharging requests. Constraints (13) ensure that x_{ik} and y_{ij} are binary variables. Constraints (14) define the range of values for w_i, t_i, s_{ij}, and d_i.

We define one more decision variable l_{ij} to linearize the nonlinear form in the objective, that is, $s_{ij}y_{ij}$. Then the objective function can be rewritten as

$$\text{Min:} \quad Z = \alpha_1 \sum_{i \in J} d_i + \alpha_2 \left(\sum_{i \in J} t_i + \sum_{i,j \in J} l_{ij} \right). \tag{15}$$

We also need to add two more constraints:

$$\begin{gathered} l_{ij} \geq y_{ij} + S_{ij} - 1 - M\left(1 - y_{ij}\right) \quad \forall i \in J,\ \forall j \in J, \\ l_{ij} \leq M \cdot y_{ij} \quad \forall i \in J,\ \forall j \in J, \\ l_{ij} \geq 0 \quad \forall i \in J,\ \forall j \in J. \end{gathered} \tag{16}$$

Then the model can be formulated as a mixed integer linear program as objective (15), subject to constraints (2)–(14) and (16).

4. Methodology

This paper proposes a hybrid GA to solve the yard truck scheduling and storage allocation problems.

4.1. Chromosome Representation. The chromosome represents a potential solution of the yard truck scheduling and storage allocation problems. A gene represents a request which contains the information of container ID, time window, origin, and destination of the request, as shown in Table 1 and Figure 2. Each chromosome consists of $|J| + |R|$ genes. Each gene may be a positive number or a negative number. A positive number represents a request and the sequence of the request prioritized from the left to the right. A negative number represents a route number. Moreover, the requests, which are between two successive negative genes, are allocated to the same truck.

A chromosome of the proposed GA can be generated using the following steps.

Step 1. Randomly allocate different storage locations for each discharging request. Then each gene contains information on the origin, destination, and sequence of each request.

Table 2: Sample data of storage locations for discharging containers.

1	2	3	4	5	6	7
(1039, 592)	(395, 686)	(18, 1263)	(635, 357)	(143, 789)	(113, 1323)	(321, 563)

Table 3: An example of chromosome encoding.

Request	9	6	10	7	−1	3	2	8	5	1	4	−2
Sequence	1	2	3	4		1	2	3	4	5	6	
Truck		Truck 1						Truck 2				

Table 4: Decoding of chromosome illustrated in Table 3.

Route 1:	$l_1 \rightarrow 9 \rightarrow 6 \rightarrow 10 \rightarrow 7 \rightarrow k_1$
Route 2:	$l_2 \rightarrow 3 \rightarrow 2 \rightarrow 8 \rightarrow 5 \rightarrow 1 \rightarrow 4 \rightarrow k_2$

Step 2. Randomly allocate all negative number genes into the chromosome and then the number of request in each route can be calculated.

Step 3. Randomly allocate all the requests to all the routes. Then the requests and the requests' sequence in each route can be obtained.

Table 3 is an example of a representation of the proposed GA for scheduling two yard trucks ($|R| = 2$) to process ten requests ($|J| = 10$) with a total length of $|R| + |J|$ of a chromosome. The $|J|$ requests are represented by a permutation of the integers from 1 to $|J|$. The $|R|$ routes are represented by the integers from $-|R|$ to -1. The decoding procedure is in the reverse order of encoding. In the example shown in Table 3, the first yard truck would sequentially process requests 9, 6, 10, 7; the second truck would process requests 3, 2, 8, 5, 1, 4, as shown in Table 4. As the sample data shown in Tables 1 and 2, if the discharging container 8 is allocated to storage location 1, the second truck may travel the coordinates (359, 689), (748, 1353), (108, 895), (1464, 336), (1485, 414), (1039, 592), (800, 1180), (113, 1287), (1035, 971), (60, 665), (148, 391), and (1246, 312) one by one.

4.2. Generation of Initial Pool. In this paper, the initial pool (with pool size P) will be generated by heuristic rules and random generation. To increase the quality of the initial pool, one of the chromosomes is generated according to the earliest possible time combining with the nearest storage location. One of the chromosomes will be generated by the earliest due time combining with the nearest storage location. The rest of the chromosomes are randomly generated.

4.3. Mating Pool and Elitist Strategy. The commonly used roulette wheel selection approach is applied for forming a mating pool. Furthermore, an elitist strategy is used to keep the best chromosome(s). The stored best chromosome found during the evolution will replace the chromosome with lowest fitness value.

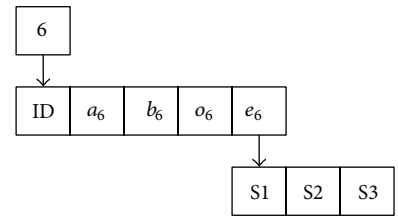

Figure 2: An example of guidance mutation method one.

Table 5: An example of crossover operation.

Parent	P_1	6	8	5	4	7	3	2	−1	1	9	10	−2
	P_2	9	2	5	−1	10	1	4	3	6	7	8	−2
After first time step 2	Ω_1	6	8	5	4	7	3	2	9				
	Ω_2	6	8	5	4	7	3	2	9				
After second time step 2	Ω_1	1	4	10									
	Ω_2	1	2	4	5	7	8	10					
Offspring	O_1	6	9	3	7	2	8	5	−1	10	4	1	−2
	O_2	6	9	3	−1	10	7	2	8	5	1	4	−2

4.4. Fitness Value. The objective is to minimize the weighted summation of the total delay and the total yard trucks travel time. Thus, the fitness value of a chromosome can be the reciprocal of its objective function value, as shown in (17). In this way, the best chromosome, which corresponds to the scheduling of the trucks and the allocation of the discharging containers with minimum weighted summation of total delay and total travel time, can be found

$$\text{Fitness} = \frac{1}{Z}. \tag{17}$$

4.5. Crossover Operation. Many studies (e.g., [25–28]) have shown that instance-specified information can make the GA searching process more effective. In the YTS-SAP, the instance-specified information is the request's earliest starting time, the request's due time, the request's processing time, and the setup time between the two requests. In the proposed GA, this instance-specified information tries to be inherited with the crossover operation. Consider the crossover operation of two parents P_1 and P_2 to reproduce two offspring O_1 and O_2. The procedure of the proposed crossover operation is shown in the following steps. Table 5 shows an example of crossover and the example uses the data shown in Table 1.

Step 1. Add all the requests in route one of both parent P_1 and parent P_2 into an empty requests set Ω_1. Delete the duplicated requests in Ω_1. Let set Ω_2 be the same as Ω_1.

Step 2. Rank the requests in Ω_1 in nondecreasing order of their earliest starting time and let set Φ be the ranked set.

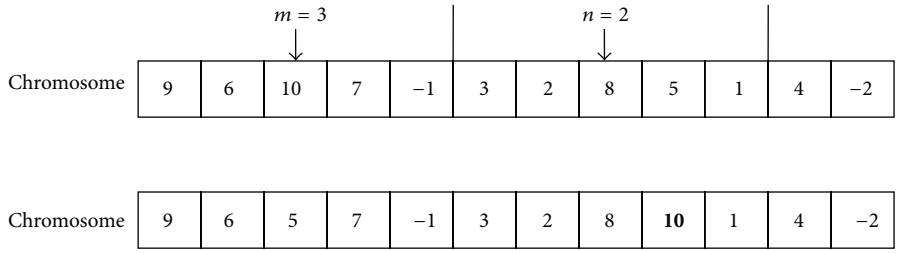

Figure 3: An example of guidance mutation method two.

Rank the requests in Ω_2 in nondecreasing order of their due time and let set Ψ be the ranked set.

Step 3. Insert the requests in the corresponding route in O_1 according to their order in set Φ and delete the inserted requests from set Ω_1. Then, insert the requests in the corresponding route in O_2 according to their order in set Ψ and delete the inserted requests from set Ω_2.

Step 4. Add all the requests in the next route for both parent P_1 and parent P_2 into set Ω_1. Then, delete the duplicated requests and delete the requests which have been inserted in O_1. Add all the requests in the next route for both parent P_1 and parent P_2 into set Ω_2. Then, delete the duplicated requests and delete the requests which have been inserted in O_2.

Step 5. Repeat Steps 2–4 until all the routes are assigned.

4.6. Fine Local Searching. To make the GA converge faster and be steadier, an exhaustive heuristic method [29, 30] is adopted. The exhaustive heuristic method is used to reinforce the GA's local searching ability. In one part of a chromosome, a set of continuous genes is selected as a segment and the number of genes formed in the segment is set to be 5, as adopted by Chung et al. [29]. This method is adopted in each chromosome part, such that each chromosome for each of the trucks in the exhaustive searching process will be executed once. Take the chromosome shown in Table 3 for example; the first part of the chromosome contains four genes that are not enough to form a segment and then the local searching will not be employed for the first part of the chromosome. If the genes 2, 8, 5, 1, and 4, which are in the second part of the chromosome, are randomly selected as a segment, all combinations of the containers sequences will be tested and then the one with best fitness value will be recorded.

4.7. Mutation Method (1)-Simple Mutation Operation. Mutation operation can help the GA prevent premature convergence and find the global optimal solution. In order to evaluate the performance of the proposed hybrid GA, a simple GA is used as a comparison. In the proposed simple GA, each chromosome contains three types of information, storage locations of the discharging containers, the sequence of requests in each route, and the amount of requests in each route. Thus, each chromosome can be mutated in three ways. The first way is to randomly choose a discharging request and change the request's storage location into another one which is an empty storage location. The second way is to randomly select two positions then swap the requests on these positions, as shown in the example in Table 6. The gene 7 and gene 9 are swapped. The third way is to change the amount of requests in the two routes, which randomly selects a request in a truck and inserts the request in another truck, as shown in an example in Table 7. Gene 7 is inserted between gene 9 and gene 10. Each of the three mutation methods will be applied once during one mutation operation.

Table 6: An example of mutation of the second way.

P_1	6	8	5	4	**7**	3	2	−1	1	**9**	10	−2
O_1	6	8	5	4	**9**	3	2	−1	1	**7**	10	−2

Table 7: An example of mutation of the third way.

P_1	6	8	5	4	**7**	3	2	−1	1	9	10	−2
O_1	6	8	5	4	3	2	−1	1	9	**7**	10	−2

4.8. Mutation Method (2)-Mutation Ways with Guidance. In the proposed hybrid GA, new mutation ways with guidance instead of the simple mutation ways will be adopted. During the mutation of the storage location, a discharging request is randomly selected first. Then, all the storage locations at which the request's travel time is within the request's due time are selected as a set, for example, storage locations 1, 2, and 3 as shown in Figure 2. Finally, randomly choose a storage location in the set to replace the origin storage location.

For the mutation way of changing two request s' positions, a request is randomly selected in one truck, recording the position m of the request. Then, randomly select another request in the range of $m+n$ to $m-n$ in another truck, where n is a positive integer. Finally, swap these two requests. Figure 3 is an example of this guidance mutation method; request 10 is selected to swap with another request. As request 10 is the third request in the first part of the chromosome, m is equal to 3. If n is set to be 2, another request is randomly selected between request 3 and request 1. In this paper, n is set to be 3.

For the mutation way of changing the amount of request s in the two trucks, a request is randomly selected in one truck, recording the position m of the request. Then, insert the request in the range of $m + n$ to $m - n$ in another truck, where n is a positive integer. Figure 4 is an example of this guidance mutation method; request 10 is selected to swap with anther request. As request 10 is the third request in the

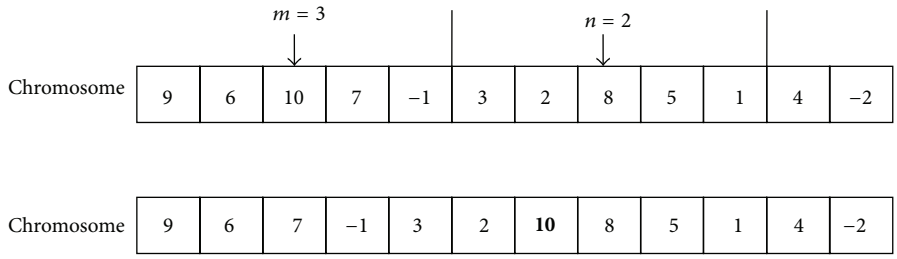

Figure 4: An example of guidance mutation method three.

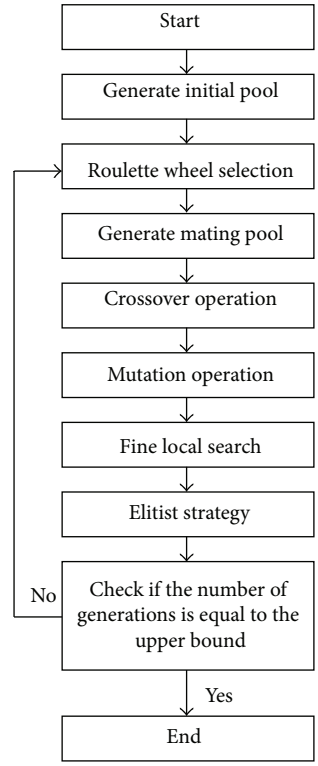

Figure 5: The flowchart of the proposed GA.

first part of the chromosome, m is equal to 3. If n is set to be 2, request 10 is randomly inserted between request 3 and request 1. In this paper, n is set to be 3 in order to avoid large change of chromosomes. The details of the proposed hybrid GA are given as shown in Figure 5.

5. Computational Experiments

In this section, a series of computational experiments are used to evaluate the performance of the proposed GA. The GA is coded by using Java Language and executed on a PC with Intel Core i7 3.4 GHz and 8 GB RAM. Instances used in the experiments are created based on the following criteria.

(1) Both the origin and destination of the loading containers, the origin of the discharging containers, and the storage locations are generated through a two-dimensional uniform distribution in the square from $(0, 0)$ to $(1500, 1500)$ (unit: meter).

(2) The earliest start time of the requests is randomly generated from a uniform distribution of $U(0, 1500)$ (unit: second) and the length of time window of requests is generated from a uniform distribution of $U(200, 500)$ (unit: second).

(3) The trucks travel at the speed of 11.11 m/s (40 km/h).

We also assume the two weight parameters α_1 and α_2 have the relation of $\alpha_1 + \alpha_2 = 1$ and α_1 is equal to 0.6 as described by Lee et al. [2].

5.1. Small Scale Problems. For small scale problems, a simple GA, which is the hybrid GA without exhaustive heuristic and guidance mutation, is used for comparison with the MIP model solved by CPLEX. The parameters of the simple GA are set as population size: 10; crossover rate P_C 0.8; mutation rate P_M 1; and maximum number of generations 2000. The number of routes is set as two. The hybrid GA is also compared with the MIP model solved by CPLEX. The

Table 8: Computational results of random instances in small scale.

Experiment number	Size (loading × discharging × storage locations)	CPLEX		Simple GA		Gap (%) between CPLEX and simple GA	Hybrid GA		Gap (%) between CPLEX and hybrid GA
		Value	CPU (s)	Value	CPU (s)		Value	CPU (s)	
1	3 × 3 × 3	177.6	7.77	177.6	3.11	0	177.6	1.83	0
2	3 × 3 × 5	157.2	27.31	157.2	2.97	0	157.2	2.01	0
3	4 × 4 × 4	209.6	148.22	209.6	3.53	0	209.6	2.61	0
4	4 × 4 × 5	209.6	7133.48	209.6	3.67	0	209.6	2.60	0
5	5 × 4 × 4	214	46350	218.5	3.70	2.1%	214	3.17	0
6	5 × 5 × 5	283.6	97612	292.4	3.81	3.1%	283.6	3.64	0
7	7 × 5 × 5	372.6	18935	384.6	3.86	3.2%	372.6	4.10	0
8	7 × 7 × 9	365	*	382.2	4.04	4.7%	365	4.78	0
9	9 × 7 × 10	385	*	406.9	4.27	5.7%	385	4.95	0
10	10 × 10 × 20	443.5	*	476.1	4.38	7.3%	443.5	5.59	0

*The computational time is longer than 10 hours.

Table 9: Number of containers and storage locations used in the instances.

	Number of loading containers	Number of discharging containers	Number of storage locations
100 containers	60	40	100
200 containers	100	100	140
300 containers	160	140	200

Table 10: Criterion of generating earliest possible time and due time for instances in large scale.

Number of distributions	Earliest possible time	Due time
1	Uniform distribution	Uniform distribution
2	Normal distribution	Uniform distribution
3	Exponential distribution	Uniform distribution
4	Uniform distribution	Normal distribution

Table 11: Computational time and generation GA used.

	Simple GA		Hybrid GA	
	CPU (s)	Generation	CPU (s)	Generation
100 containers	22	10000	74	1000
200 containers	178	30000	228	1300
300 containers	375	60000	382	1500

parameters of the hybrid GA are the same as the simple GA except that maximum number of generations is set to 200.

As is shown in Table 8, it is evident that the simple GA can obtain the optimal solution in reasonable time in the first four cases. Due to the interacting of yard truck scheduling problem and storage allocation problem, CPLEX requires hours to solve each single instance, but the simple GA, as a comparison, only uses a few seconds to solve the problem. For the last six instances, the simple GA can obtain the near optimal solution and the average gap between the simple GA and the optimal solution obtained by Branch and Bound coded in CPLEX is computed at about 4.35%. With the instances size becoming larger, the gap also becomes larger. The simple GA performs poorly with the increasing of instance size. However, the performance of the simple GA is acceptable from the practical point of view. On the other hand, the hybrid GA can always obtain optimal solutions because of guidance mutation and exhaustive heuristic for local searching. As the maximum number of generations is smaller than the simple GA, the hybrid GA is faster than the simple GA in the first six instances. However, the hybrid GA needs more time than the simple GA when the instance scale becomes larger.

5.2. Large Scale Problems. To evaluate the performance of the proposed hybrid GA in large scale problems, the simple GA is applied as a comparison for the hybrid GA.

Table 12: Computational results of random instances in large scale.

Number of containers	Criteria of forming instances	Simple GA	Hybrid GA	Gap (%)
100 containers	1	3432.4	3230.2	5%
	1	2836	2695.2	5%
	1	2997.8	2886.8	4%
	2	3470.4	2604.4	25%
	2	3997.2	3639.6	9%
	2	5752.8	5032	12%
	3	6206	5407.2	13%
	3	12239.2	11110.8	9%
	3	11013.2	8122.6	26%
	4	4507.4	2945.2	35%
	4	2916.8	2709.2	7%
	4	3068	2933.4	5%
	5	2740	2660.4	3%
	5	2665.6	2637.4	1%
	5	2857.2	2757.4	3%
200 containers	1	24863.6	21566.8	13%
	1	42080.2	35506	16%
	1	29926.6	26421.2	12%
	2	50892	50875.2	1%
	2	54559.4	51487.2	6%
	2	51953.2	47241.6	9%
	3	57543	50917.2	12%
	3	75370	60192.8	20%
	3	52303.6	41523.6	21%
	4	30470.6	16952.4	44%
	4	36708.8	34962.2	5%
	4	34427.4	32031.6	7%
	5	4893.8	4645.4	5%
	5	4861.8	4812.4	1%
	5	5020	4943.4	2%
300 containers	1	120613	117577.6	3%
	1	159931.2	151825.8	5%
	1	152231.4	145829	4%
	2	140016	133199.8	5%
	2	128797.2	123056.2	4%
	2	151134	141434.6	6%
	3	181894.6	174253.6	4%
	3	183754	176955.8	4%
	3	200072.8	191379.6	4%
	4	149158.6	123718.8	17%
	4	154976.8	143830.8	7%
	4	155862	151624	3%
	5	11003.4	9864.6	10%
	5	19459.8	18050	7%
	5	20005.4	16538.2	17%

Four different kinds of distribution combinations of the earliest possible time and the due time are applied as the criteria of generating instances to increase the variety of the

instances. We will also change the number of available trucks. The number of trucks is set as 3 at first, and then the number of trucks is set as 6 for the fifth kind of criteria. Three different kinds of instances with different sizes are formed by using each of the five criteria. The criteria of the instances are created as shown in Tables 9 and 10. The parameters of the proposed hybrid GA for large scale are set as population size 10, crossover rate P_C 0.8, and mutation rate P_M 0.9. Table 10 also shows the number of generations, which is long enough to attain a steady solution, and the computational time of the GA.

As shown in Tables 11 and 12, the proposed hybrid GA can obtain the best results and the computational time is a little longer than the simple GA. The lowest gaps between the simple GA and the new hybrid GA are 1%, 1%, and 3%, respectively, for 100 containers, 200 containers, and 300 containers. The highest gaps between the simple GA and the new hybrid GA are 35%, 44%, and 17%, respectively, for 100 containers, 200 containers, and 300 containers. The average gaps between the simple GA and the hybrid GA are 11%, 11%, and 7%, respectively, for 100 containers, 200 containers, and 300 containers. Since the hybrid GA has stronger local search ability and the mutation operation will not be totally random, the results of the hybrid GA are better than the simple one. However, the exhaustive heuristic is time consuming and it will take the hybrid genetic more time to find a solution.

6. Conclusions and Future Work

Yard truck scheduling and storage allocation are two important problems for container terminals to enhance their operation efficiency. In recent year, Lee et al. [6] proposed an integrated model simultaneously solving the two problems, and later on they further enhanced the model in Lee et al. [6]. We base on the model in Lee et al. [6] and further improve the model by considering the situation that the number of available storage locations is not equal to the number of import containers. Such improvement can make the model more practical. As the problem complexity increases dramatically, a new hybrid GA with exhaustive heuristic and guidance mutation is proposed. The crossover operation of proposed GA is based on the information of a request's ready time and due time. The mutation operator combines three new ways of mutation approach. To evaluate and demonstrate the quality of the proposed hybrid GA, both a simple GA and the hybrid GA are compared with the MIP model solved by CPLEX in small scale problems, and then the proposed hybrid genetic is compared with the simple GA by using large scale instances. It is proven that the simple GA and the hybrid GA can obtain near optimal solutions in reasonable time by using a series of computational experiments in small size problems. For large scale problems, 100, 200, and 300 containers with different numbers of storage locations and trucks are studied. The results demonstrated that the proposed hybrid GA can obtain the best solutions compared to the simple GA method.

In this paper, the number of vehicles and storage locations are assumed to be given. Given this information, yard truck routing and storage location for discharging containers are determined. However, in practical situation, the number of trucks can be flexible and the number of storage locations may dynamically change throughout the operating horizon. Therefore, the amount of trucks and storage locations can be considered as variables in the future work. Another potential further research topic is to incorporate multilayer container storage in yard side. Combined with the current model, these are expected to give a more realistic description of container terminal operations.

Conflict of Interests

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

Acknowledgments

The authors would like to thank the Teaching Company Scheme project (Project no. ZW1H (TCS162)), The Hong Kong Polytechnic University Research Committee (Project no. G-UB03), for financial and technical support, a grant from The Hong Kong Scholars Program Mainland-Hong Kong Joint Postdoctoral Fellows Program (Project no. G-YZ24), and The National Natural Science Foundation of China (Grants nos. 71471158 and 71271140). The authors also would like to thank The Hong Kong Polytechnic University Research Committee for financial and technical support.

References

[1] D. Steenken, S. Voß, and R. Stahlbock, "Container terminal operation and operations research: a classification and literature review," *OR Spectrum*, vol. 26, no. 1, pp. 3–49, 2004.

[2] D.-H. Lee, J. X. Cao, Q. Shi, and J. H. Chen, "A heuristic algorithm for yard truck scheduling and storage allocation problems," *Transportation Research E: Logistics and Transportation Review*, vol. 45, no. 5, pp. 810–820, 2009.

[3] W. C. Ng, K. L. Mak, and Y. X. Zhang, "Scheduling trucks in container terminals using a genetic algorithm," *Engineering Optimization*, vol. 39, no. 1, pp. 33–47, 2007.

[4] C. Zhang, Y.-W. Wan, J. Liu, and R. J. Linn, "Dynamic crane deployment in container storage yards," *Transportation Research Part B: Methodological*, vol. 36, no. 6, pp. 537–555, 2002.

[5] O. Sharif and N. Huynh, "Storage space allocation at marine container terminals using ant-based control," *Expert Systems with Applications*, vol. 40, no. 6, pp. 2323–2330, 2013.

[6] D. H. Lee, J. X. Cao, and Q. Shi, "Integrated model for truck scheduling and storage allocation problem at contain terminals," in *Proceeding of TRB 87th Annual Meeting Compendium of Papers DVD*, 2008.

[7] E. K. Bish, T. Leong, C. Li, J. W. C. Ng, and D. Simchi-Levi, "Analysis of a new vehicle scheduling and location problem," *Naval Research Logistics*, vol. 48, no. 5, pp. 363–385, 2001.

[8] K. H. Kim and H. B. Kim, "Segregating space allocation models for container inventories in port container terminals," *International Journal of Production Economics*, vol. 59, no. 1–3, pp. 415–423, 1999.

[9] K. H. Kim and H. B. Kim, "The optimal sizing of the storage space and handling facilities for import containers," *Transportation Research B: Methodological*, vol. 36, no. 9, pp. 821–835, 2002.

[10] C. Zhang, J. Liu, Y.-W. Wan, K. G. Murty, and R. J. Linn, "Storage space allocation in container terminals," *Transportation Research B*, vol. 37, no. 10, pp. 883–903, 2003.

[11] D.-H. Lee, J. G. Jin, and J. H. Chen, "Terminal and yard allocation problem for a container transshipment hub with multiple terminals," *Transportation Research E: Logistics and Transportation Review*, vol. 48, no. 2, pp. 516–528, 2012.

[12] K. H. Kim and J. W. Bae, "A look-ahead dispatching method for automated guided vehicles in automated port container terminals," *Transportation Science*, vol. 38, no. 2, pp. 224–234, 2004.

[13] V. D. Nguyen and K. H. Kim, "A dispatching method for automated lifting vehicles in automated port container terminals," *Computers and Industrial Engineering*, vol. 56, no. 3, pp. 1002–1020, 2009.

[14] H. Hu, B. K. Lee, Y. Huang, L. H. Lee, and E. P. Chew, "Performance analysis on transfer platforms in frame bridge based automated container terminals," *Mathematical Problems in Engineering*, vol. 2013, Article ID 593847, 8 pages, 2013.

[15] W. Zhao and A. V. Goodchild, "Truck travel time reliability and prediction in a port drayage network," *Maritime Economics and Logistics*, vol. 13, no. 4, pp. 387–418, 2011.

[16] W. Yan, Y. Huang, D. Chang, and J. He, "An investigation into knowledge-based yard crane scheduling for container terminals," *Advanced Engineering Informatics*, vol. 25, no. 3, pp. 462–471, 2011.

[17] H. Javanshir and S. R. Seyedalizadeh Ganji, "Yard crane scheduling in port container terminals using genetic algorithm," *Journal of Industrial Engineering International*, vol. 6, no. 11, pp. 39–50, 2010.

[18] H. Javanshir, S. Ghomi, and M. Ghomi, "Investigating transportation system in container terminals and developing a yard crane scheduling model," *Management Science Letters*, vol. 2, no. 1, pp. 171–180, 2012.

[19] J. He, D. Chang, W. Mi, and W. Yan, "A hybrid parallel genetic algorithm for yard crane scheduling," *Transportation Research E: Logistics and Transportation Review*, vol. 46, no. 1, pp. 136–155, 2010.

[20] J. X. Cao, D.-H. Lee, J. H. Chen, and Q. Shi, "The integrated yard truck and yard crane scheduling problem: benders' decomposition-based methods," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 3, pp. 344–353, 2010.

[21] S. M. Homayouni and S. H. Tang, "Multi objective optimization of coordinated scheduling of cranes and vehicles at container terminals," *Mathematical Problems in Engineering*, vol. 2013, Article ID 746781, 9 pages, 2013.

[22] E. K. Bish, "A multiple-crane-constrained scheduling problem in a container terminal," *European Journal of Operational Research*, vol. 144, no. 1, pp. 83–107, 2003.

[23] E. K. Bish, F. Y. Chen, Y. T. Leong, B. L. Nelson, J. W. C. Ng, and D. Simchi-Levi, "Dispatching vehicles in a mega container terminal," *OR Spectrum*, vol. 27, no. 4, pp. 491–506, 2005.

[24] Y. Han, L. H. Lee, E. P. Chew, and K. C. Tan, "A yard storage strategy for minimizing traffic congestion in a marine container transshipment hub," *OR Spectrum*, vol. 30, no. 4, pp. 697–720, 2008.

[25] J. L. Blanton Jr. and R. L. Wainwright, "Multiple vehicle routing with time and capacity constraints using genetic algorithms," in *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 452–459, 1993.

[26] P. W. Poon and J. N. Carter, "Genetic algorithm crossover operators for ordering applications," *Computers and Operations Research*, vol. 22, no. 1, pp. 135–147, 1995.

[27] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: a review of representations and operators," *Artificial Intelligence Review*, vol. 13, no. 2, pp. 129–170, 1999.

[28] C. Moon, J. Kim, G. Choi, and Y. Seo, "An efficient genetic algorithm for the traveling salesman problem with precedence constraints," *European Journal of Operational Research*, vol. 140, no. 3, pp. 606–617, 2002.

[29] S. H. Chung, F. T. S. Chan, and W. H. Ip, "Minimization of order tardiness through collaboration strategy in multifactory production system," *Systems Journal IEEE*, vol. 5, no. 1, pp. 40–49, 2011.

[30] M. Palpant, C. Artigues, and P. Michelon, "LSSPER: solving the resource-constrained project scheduling problem with large neighbourhood search," *Annals of Operations Research*, vol. 131, no. 1–4, pp. 237–257, 2004.