The Josai Journal of Business Administration (2013), Vol. 10, No.1, 29–41
©2013, Josai University

An Improved Model for the Integrated Yard Crane and Yard Truck Scheduling Problem

Meilong Le
Shanghai Maritime University

Toshitake Kohmura
Josai University

Abstract

This paper investigates the integrated *yard crane* (YC) and *yard truck* (YT) scheduling problems. The YCs actual operation constraints were considered. For example, when several YCs share a bi-directional lane, a non-crossing limitation is used. There are other constraints, such as fixed YC separation distances and job-precedence constraints. Based on these constraints, we formulated a model to minimize the make-span. Due to computational intractability, we introduce a method called *multi-layer genetic algorithm* (MLGA) to solve the model, and then conduct computational experiments to evaluate the effectiveness of the method.

Key Words: Yard crane; Yard truck; Scheduling; Integrated scheduling; Multi-layer genetic algorithm.

1. Introduction

It has been 50 years since the first regular sea container service came into use in the international transportation market. It began with container service between the U.S. East Coast and points in the Caribbean and Central and South America. Container services have enjoyed rapid development. Following are some data to show the development of container use.

Through the graph, we can see that containerization trends share a steady development, except for 2009, because of the economic crisis in 2008. Statistics show that container transportation has a bright future. The increasing use of containers calls for higher requirements on seaport container terminals. It also requires a lot of paperwork, which is concentrated in the equipment and resource management of container terminals. As equipment and resource management is very complex, researchers have always divided these problems into several sub-problems, such as berth allocation, quay crane (QC) allocation, YT scheduling, YC scheduling and yard resources planning. But as the resources in container terminals are highly correlated with each other, research in integrated scheduling is really vital to the efficient improvement of container terminals. The YC and YT integrated scheduling problems are investigated in this paper. Based on integrated scheduling, YC operational constraints and job-precedence constraints are introduced first in the paper, which makes the model more reasonable and practical.

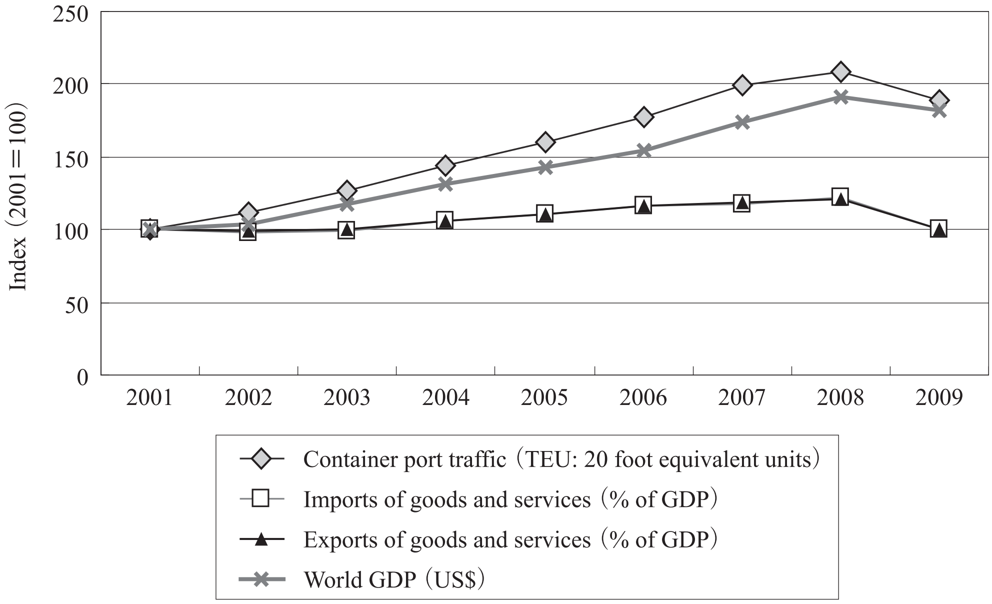

Fig. 1　Containerization trends in recent years (from 2000 to 2009).
Source: World Bank WDI Database

This paper is organized as follows. In section 2, we give a brief literature review in related research fields. The problem description and model are given in section 3. We introduce our method-MLGA in section 4. Computational experiments are conducted to evaluate the method in section 5. Section 6 is the conclusion section.

2. Literature review

Container terminal operation research is often researched now. As we study YC and YT integrated scheduling in this paper, we give a review of previous research on YC and YT scheduling and examples of studies looking at integrated scheduling. Most previous research focused on optimizing a single sub-problem.

A Single YC Scheduling Problem. Kim and Kim (1999) considered the loading operations of a YC, with the single YC scheduling problem as a traveling salesman problem. The YC moved among the container groups and then finished the partial tours one by one. By connecting partial tours, a complete tour for a YC could be obtained. Ng and Mak (2005) also studied a single YC scheduling problem, different from Kim and Kim (1999). They considered the ready time of each job (container). They modeled the problem as an integer program with the objective to minimize the total waiting time of all jobs, and solved the problem by using a branch and bound algorithm. X. Guo, et al. (2011) modeled the problem with the objective to minimize the average waiting time of all YTs, and then developed two algorithms to solve the model. One is the Modified A* search, and the other is Prioritized Recursive Backtracking with heuristics. The latter overcomes the limitations of the previous two. The first one is of memory usage; the second is a large time requirement.

Multiple Scheduling without Interference. C. Zhang, et al. (2002) focus on multiple YC scheduling without interference considered. The problem was formulated as a mixed integer programming model, and its objective function was to minimize the total delayed workload, then the model was solved by Lagrangean relaxation. Similar to C. Zhang, et al. (2002), Cheung, et al. (2002) also formulated it as a mixed integer program, and they had the same objective. The difference between them is that Cheung et al. (2002) solved the problem by Lagrangean decomposition, and also developed a method called successive piecewise-linear approximation. D. H. Lee, et al. (2007) studied two YCs scheduling problems, and formulated a mathematical model to minimize the later finishing time of the two YCs in the last sub-tour. They solved the model by using a simulated annealing algorithm.

Other Multiple Scheduling Problems. W. C. Ng (2005) studied the problem of scheduling multiple YCs in a yard zone with only one bi-directional travelling lane. He considered the interference between multiple YCs, and then formulated an integer program to minimize the total completion time, and then developed a dynamic programming-based heuristic to solve the scheduling problem. W. Li, et al. (2009) also focused on multiple YC's scheduling problems. Compared to W. C. Ng (2005), W. Li, et al. (2009) considered one more factor, named the fixed YC separation distances.

Fleet Sizing and Vehicle Routing Problem. P. H. Koo, et al. (2004) proposed a two-phased procedure to solve the problem of fleet sizing and vehicle routing problems. In phase one, it constructed a model to minimize the total empty vehicle travel time and to get a lower bound on the fleet size by using a network flow method; in phase two, a tabu search method is used to improve the fleet size and to find the vehicle routing. E. Nishimura, et al. (2005) show the two models based on different situations, a single trailer and multiple trailers, respectively. In a single trailer situation, it was considered a traveling salesman problem.

Loading and Unloading Problem. E. K. Bish (2003) considered the operations of two ships, one unloading and the other loading, and the number of unloaded and loaded containers as equal. The model was formulated in two steps. Step one is the assignment and matching step, which is constructed to find out the combined trip; step two is the scheduling step. This step dispatches the vehicles to finish the jobs by using the results of step one. Kim and Bae (2004) studied the AGV scheduling problem in static situations, and formulated a mixed integer program to minimize the total cost, including travel cost and penalty cost for the delay in the completion time. W. C. Ng, et al. (2007) studied the problem of scheduling a fleet of trucks to perform a set of transportation jobs with different ready times. The problem was formulated as a model to minimize the total finished time of all jobs, and was solved by a genetic algorithm.

Most of the previous studies focus on the scheduling of only one kind of equipment due to computational intractability. But the synchronization of different handling equipment is crucial to the efficiency of container terminals.

Integrated QC and YT Scheduling. J. X. Cao, et al. (2010) studied integrated QC and YT scheduling problems for inbound containers. The jobs were assumed to be operated under the gang structure mode, i.e., several YTs served the same QC. The problem was formulated as a mixed integer program with the objective to minimize the make-span for dispatching a set of containers allocated to the QC. The model was solved by a genetic algorithm and a modified Johnson's Rule-based heuristic algorithm. L. Chen, et al. (2007) studied the integrated scheduling of three kinds of handling equipments. They considered the problem as a Hybrid Flow Shop Scheduling problem, and then solved it by a tabu search algorithm. Lau and Zhao (2008) also focused on the integrated scheduling of three kinds of equipment, and formulated the problem as a mixed integer program with the objective to minimize the AGVs travel time, the delay of QC operations and to minimize the total travel time of YCs. J. X. Cao, et al. (2010) studied the integrated scheduling of YT and YC. The problem was formulated as a mixed integer program with the objective to minimize the last job's completion time. Then the model was solved by two methods, one was a general Benders'cut-based method and the other was a combinatorial Benders' cut-based method.

After reviewing other research, we decided to try to find a solution to the problems of scheduling when there is more than one sub-problem.

3. Problem description and formulation

Fig. 2 shows a typical layout of container terminals. The figure showing a brief introduction of loading and unloading operations, is given below.

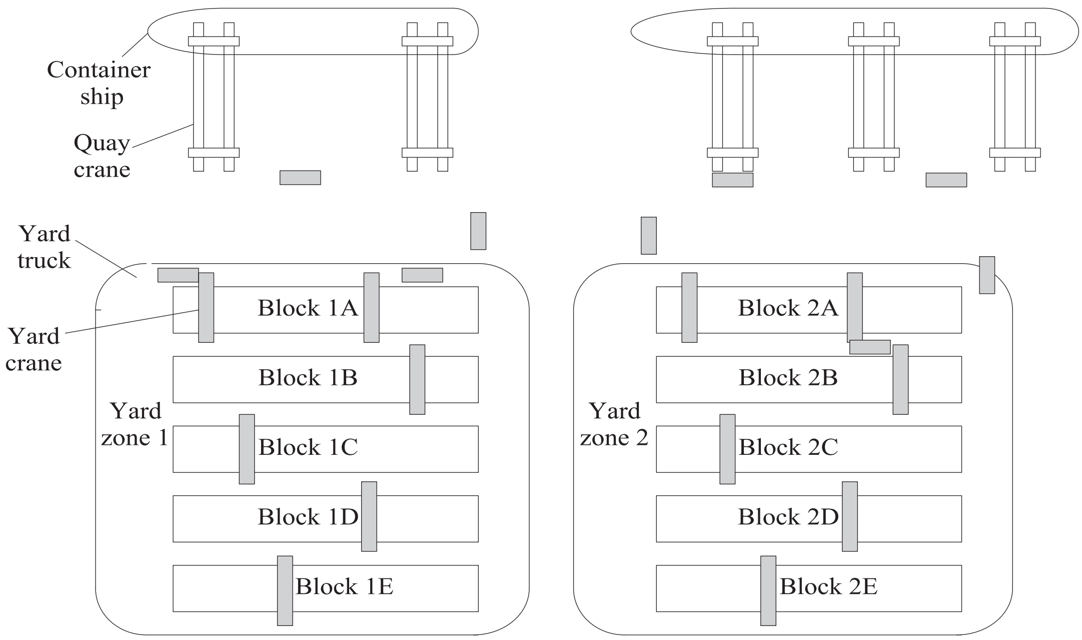

Fig. 2 A typical layout of container terminals

When there is a ship to be served, containers of the ship should be unloaded first, then the loading operation should be carried out. During the unloading operation, containers are unloaded onto YTs through the QCs, and then YTs transport them to a specific block in container yard, and YCs lift the containers up, and put them into specific slots. When we put the operations sequence in reverse, we get the loading operation.

There is a job-precedence constraint during the unloading and loading operations. For example, the containers above the deck should be served by QCs before those on the under deck during unloading operation, and when doing the loading operation, the sequence is reversed.

When several YCs share the same block, there are some limits. The non-crossing constraint is one limit; this means that if YC A is in the left position of YC B, YC A cannot cross YC B to work in the right position of YC B. The fixed YC separation distance limitation means that a safe distance is needed between YCs; this distance is set to be 160 feet or 8 slots (W. Li et al. 2009).

To investigate the integrated YC and YT scheduling problem based on the above constraints, we formulated this problem into a model which minimized the make-span. In order to build the model, we have followed these assumptions:

(1) Only loading operations are considered

(2) Positions of containers to be handled are given

(3) YT can only transport a 20 feet container at a time
(4) Interference between YTs are not considered
(5) Each container can be loaded in any sequence except the special ones (should be satisfied with the job-precedence constraint)
(6) QCs are always available

In this paper, we refer to the operation of each container as a job, and there are n jobs in all. Let φ be the set of jobs, so $|\varphi| = n$. Additionally, we define two dummy jobs, indexed by 0 and $n+1$, to represent the initial and final states of YCs and YTs, respectively.

Let $\varphi_1 = \varphi \cup \{0\}$; $\varphi_2 = \varphi \cup \{n+1\}$ and $\varphi_3 = \varphi_1 \cup \varphi_2$.

Following are indices used in the model:

i, j — indices of jobs, $i, j = 0, 1, , \ldots, n, n+1$

l — index of bays, $l = 1, 2, \ldots, \theta$. Bays in the block are ordered from 1 to θ.

p — index of periods, $p = 0, 1, 2, \ldots, \pi$. Suppose that there exists a known upper boundary for the make-span of the optimal schedule. Partition the upper bound into π periods with the length of each period equal to the time required for a yard crane to travel a single bay. See W. C. Ng (2005)

m — indice of YCs, $m = 1, 2, \ldots, M$.

k — indice of YTs, $k = 1, 2, \ldots, K$.

The following parameters are used in the model:

o_i — bay number of job i, which represents the initial location of job i.

d_i — destination of job i.

OP — Set of pair of jobs. If $(i, j) \in OP$, then job i should be completed earlier than job j.

$p(l)$ — if a YC is located in bay l at period p, $p(l)$ denotes the set of bays the YC can possibly be in at period p-1.

$s(l)$ — if a YC is located in bay l at period p, $s(l)$ denotes the set of bays the YC can possibly be in at period p+1.

δ_{uv} — Distance between location u and location v.

v_1 — speed of YC (1 bay/period)

v_2 — speed of YT (bay/period)

s_{ij}^1 — setup time for YC to travel from job i to job j. So $s_{ij}^1 = \dfrac{\delta_{o_i, o_j}}{v_1}$.

s_{ij}^2 — setup time for YT to travel from job i to job j. So $s_{ij}^2 = \dfrac{\delta_{d_i, o_j}}{v_2}$.

t_i — transport time of job i. So $t_i = \dfrac{\delta_{o_i, d_i}}{v_2}$.

τ — QC processing time of each job

τ_i — YC processing time of job i

c_i — completion time of job i

r_i — ready time of job i, which is decided by YC ready time and YT ready time. r_{ik}^2 denotes the ready time of YT k to handle job i, r_{im}^1 denotes the ready time of YC m to handle job i. Then we get a ready time for job i, where $r_i = \max(r_{im}^1, r_{ik}^2)$.

Following are the decision variables

$$Y_{m,l,p} = \begin{cases} 1 & \textit{if YC m is located in bay l at period p} \\ 0 & \textit{otherwise} \end{cases}$$

$$X^1_{i,j,m} = \begin{cases} 1 & \textit{if job i is immediately handled before job j by YC m} \\ 0 & \textit{otherwise} \end{cases}$$

$$X^2_{i,j,k} = \begin{cases} 1 & \textit{if job i is immediately handled before job j by YT k} \\ 0 & \textit{otherwise} \end{cases}$$

$$W^1_{i,m,p} = \begin{cases} 1 & \textit{if YC m completes handling of job i at period p} \\ 0 & \textit{otherwise} \end{cases}$$

$$W^2_{i,k,p} = \begin{cases} 1 & \textit{if YT k completes handling of job i at period p} \\ 0 & \textit{otherwise} \end{cases}$$

So the completion time of job $c_i = \sum_{p=1}^{\pi} \sum_{k=1}^{k} p^* W^2_{i,k,p}$ i .

Distances are quantified with the length of a bay, and time is quantified with the length of a period. For example, suppose that the length of a bay is 20 feet and a period is 4 seconds. If the distance between location u and location v is 200 feet long, we say that the distance between location u and v is 10 (200/20). And if YT k takes 200 seconds to travel from location a to location b, then we say YT k takes 50 periods to travel from location a to b.

The model is stated as follows, followed by a brief explanation.

Objective function

$$\min \quad C_{n+1} \qquad (1)$$

Subject to

$$C_{n+1} \geq c_i, i \in \varphi \qquad (2)$$

$$\sum_{j \in \varphi_2} X^1_{i,j,m} = 1, \ \forall i \in \varphi_1, \forall m \qquad (3)$$

$$\sum_{i \in \varphi_1} X^1_{i,j,m} = 1, \ \forall j \in \varphi_2, \forall m \qquad (4)$$

$$\sum_{m=1}^{M} \sum_{j \in \varphi_2} x^1_{0,j,m} = M \qquad (5)$$

$$\sum_{m=1}^{M} \sum_{i \in \varphi_1} x^1_{i,n+1,m} = M \qquad (6)$$

$$\sum_{m=1}^{M} \sum_{p=1}^{\pi} W^1_{i,m,p} = 1 \qquad (7)$$

$$\sum_{j \in \varphi_2} X^2_{i,j,k} = 1, \qquad \forall i \in \varphi_1, \forall k \qquad (8)$$

$$\sum_{i \in \varphi_1} X^2_{i,j,k} = 1, \qquad \forall j \in \varphi_2, \forall k \qquad (9)$$

$$\sum_{k=1}^{K} \sum_{j \in \varphi_2} x_{0,j,k}^2 = K \tag{10}$$

$$\sum_{k=1}^{K} \sum_{i \in \varphi_1} x_{i,n+1,k}^2 = K \tag{11}$$

$$\sum_{k=1}^{K} \sum_{p=1}^{\pi} W_{i,k,p}^2 = 1 \tag{12}$$

$$\sum_{l=1}^{\theta} Y_{m,l,p} = 1, \ \forall m, p \tag{13}$$

$$\sum_{m=1}^{M} Y_{m,l,p} \leq 1, \ \forall l, p \tag{14}$$

$$Y_{m,l,p} \leq \sum_{l' \in p(l)} Y_{m,l',p-1}, \quad \forall m, l; p = 1, 2, 3, \ldots, \pi \tag{15}$$

$$Y_{m,l,p} \leq \sum_{l' \in s(l)} Y_{m,l',p+1}, \quad \forall m, l; p = 0, 1, 2, \ldots, \pi-1 \tag{16}$$

$$\sum_{p'=0}^{\pi_i} Y_{m,o_i,p-p'} - \tau_i - 1 \geq N\,(W_{i,m,p}^1 - 1) \tag{17}$$

$$N\,(1 - Y_{m,l,p}) \geq \sum_{q=l}^{\theta} Y_{m-1,q,p}; \quad \forall p, m = 2, 3, \ldots, M, \ l = 1, 2, \ldots, \theta \tag{18}$$

$$\sum_{l \in L} l \cdot Y_{m+1,l,p} - \sum_{l' \in L} l' \cdot Y_{m,l',p} \geq 8 \tag{19}$$

$$c_i \leq c_j \qquad (i, j) \in OP \tag{20}$$

$$r_{jm}^1 - r_i + N(1 - X_{i,j,m}^1) \geq \begin{cases} 0 & for\ i = 0,\ j \in \varphi \\ \tau_i + s_{ij}^1 & for\ i, j \in \varphi,\ and\ i \neq j \\ 0 & for\ i \in \varphi,\ j = n+1 \\ \infty & for\ i = 0,\ j = n+1 \end{cases} \tag{21}$$

$$r_{jk}^2 - r_i + N(1 - X_{i,j,k}^2) \geq \begin{cases} 0 & for\ i = 0,\ j \in \varphi \\ \tau_i + t_i + \tau + s_{ij}^2 & for\ i, j \in \varphi,\ and\ i \neq j \\ 0 & for\ i \in \varphi,\ j = n+1 \\ \infty & for\ i = 0,\ j = n+1 \end{cases} \tag{22}$$

$$r_i = \max(r_{im}^1, r_{ik}^2) \tag{23}$$

$$\sum_{m=1}^{M}\sum_{p=1}^{\pi} p^{*} W^{1}_{i,m,p} \geq r_i + \tau_i \qquad (24)$$

$$\sum_{p'=1}^{\pi} p'^{*} W^{2}_{i,k,p'} \geq \sum_{p=1}^{\pi} p^{*} W^{1}_{i,m,p} + t_i + \tau \qquad (25)$$

$$X^{1}_{i,j,m}, X^{2}_{i,j,k}, Y_{m,l,p} \in \{0,1\}, \ \forall \mathrm{i} \in \varphi_1, j \in \varphi_2, k \in K, m \in M, l \in L \qquad (26)$$

Where N is a big positive number.

The objective is to minimize the makespan. Constraint (2) ensures that the make-span would be longer than the completion time of any job. Constraint (3) means for each job $i \in \varphi_1$, there is a succeeding job assigned to the same YC m as job i. Constraint (4) means for each job $i \in \varphi_2$, there is a preceding job assigned to the same YC m as job j. Constraints (5) and (6) promise that there are M YCs being deployed. Constraint (7) guarantees that each job would be served by a YC in a period. Constraints (8)-(11) are similar to Constraints (3)-(6). Constraint (12) guarantees that each job would be finished by a YT in a period. Constraint (13) implies that every YC would be located in a bay in any period. Constraint (14) promises there is at most one YC in a bay in a period. Constraints (15) and (16) state the relationship between locations visited by a YC in successive periods. Constraint (17) ensures that during a loading operation, the YC should stay in the bay throughout the operation. Constraint (18) is the non-crossing constraint. The fixed YC distance separation constraint is stated as constraint (19). Constraint (20) is job-precedence constraint. Constraint (21) depicts the relationship between ready time of a job and the ready time of YC. Constraint (22) depicts the relationship between the ready time of a job and the ready time of YT. Constraint (23) calculate the ready time of job i. Constraint (24) calculate the completion time of job i at YC. Constraint (25) calculates the completion time of job i.

4. Multi-layer genetic algorithm

As the model above is non-linear, exact algorithms can hardly get a solution. So we introduced a multi-layer genetic algorithm (MLGA), (H. Y. K. Lau, et al. 2008) to solve the model. The procedures of the method are described in Fig. 3. We have two layers in the MLGA. The main-layer is used to find the job sequences of YCs, and the sub-layer is used to find job sequences of YTs. But the generation of a sub-layer is restricted by the main layer, because the job sequences of YTs depend on the job sequences of YCs.

4.1. *Structure of individuals*

Individuals of the main-layer represent candidates of job sequences of YCs. An individual of a sub-layer represents a candidate of job sequences of YTs. The initial main-layer solution was randomly generated while considering job-precedence constraints, and individuals of the sub-layer were generated by considering main-layer individuals. Here we take 2 YCs, 5 YTs and 10 jobs as an example to illustrate the individual. As Fig. 4 shows, in the main-layer, 0 is used to partition the jobs into two job sequences. It means that the first YC operational sequence is (1, 5, 2, 7, 10) and the second YC operational sequence is (8, 3, 9, 6, 4). Similar to this, in the sub-layer a set of jobs (1, 5) is carried out by YT 1. YT 2 will transport job 2. YT 3 is asked to serve (7, 10). YT 4 (8, 3, 9), and lastly YT 5 needs to transport (6, 4).

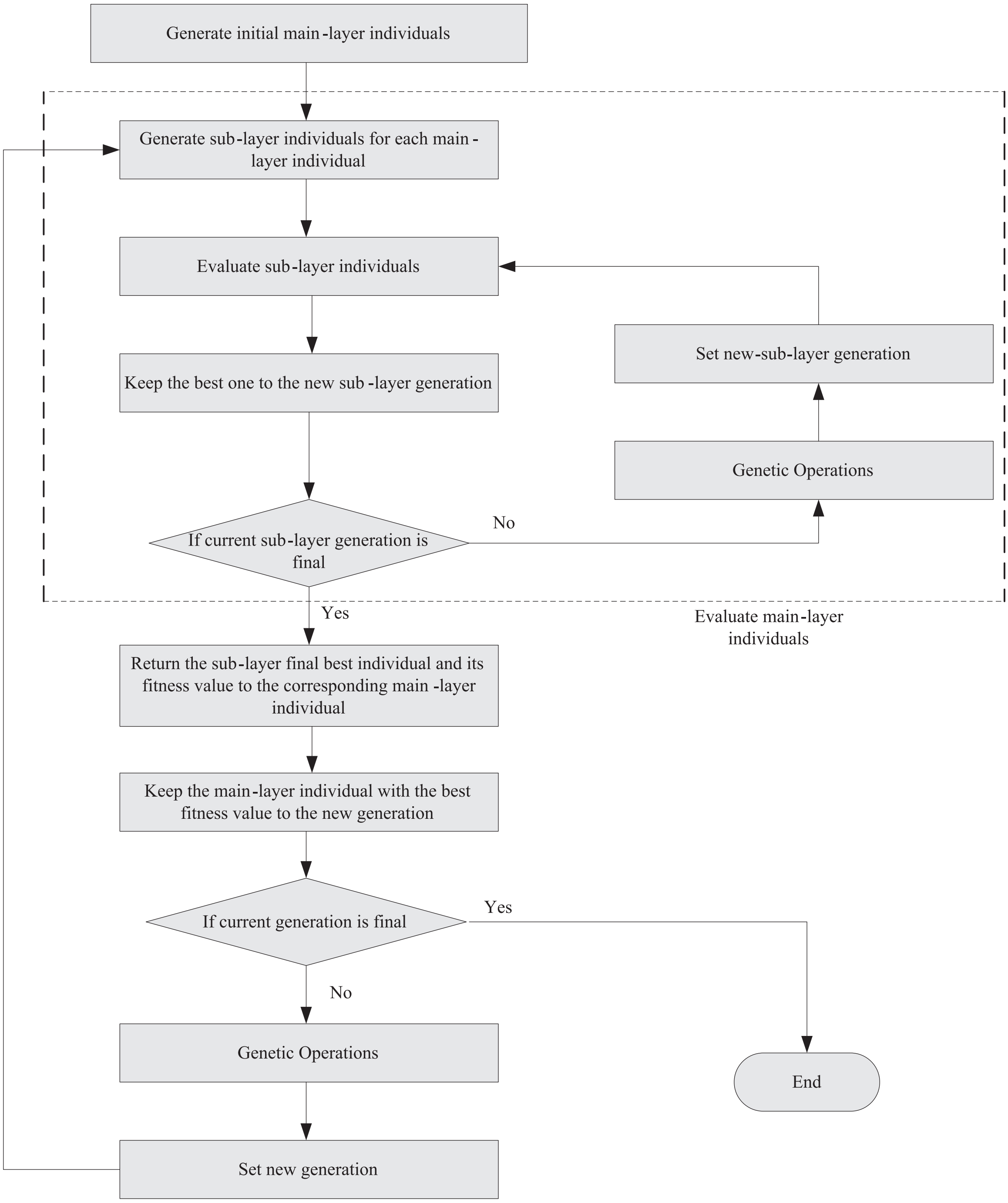

Fig. 3 The MLGA procedures.

1	5	2	7	10	0	8	3	9	6	4

Main-layer individual

1	5	0	2	0	7	10	0	8	3	9	0	6	4

Sub-layer individual

Fig. 4 Structure of a individual

4.2. *Fitness evaluation and selection*

The individual is evaluated based on:

$eval = 1/C_{n+1}$.

In this paper, we accept a roulette wheel approach.

4.3. *Crossover*

"Ordered crossover" is used to process the main-layer individuals in the paper. It works as follows:

Step 1 Randomly select a substring in one parent

Step 2 Produce a proto-child by copying the sub-string into the corresponding positions in the child.

Step 3 Delete the holds which are already in the substring from the second parent. The resulting sequence of holds contains the holds needed by the proto-child.

Step 4 Place the holds into the unfixed positions of the proto-child from left to right according to the order of the sequence used to produce an offspring.

Fig. 5 is used to illustrate the "ordered crossover".

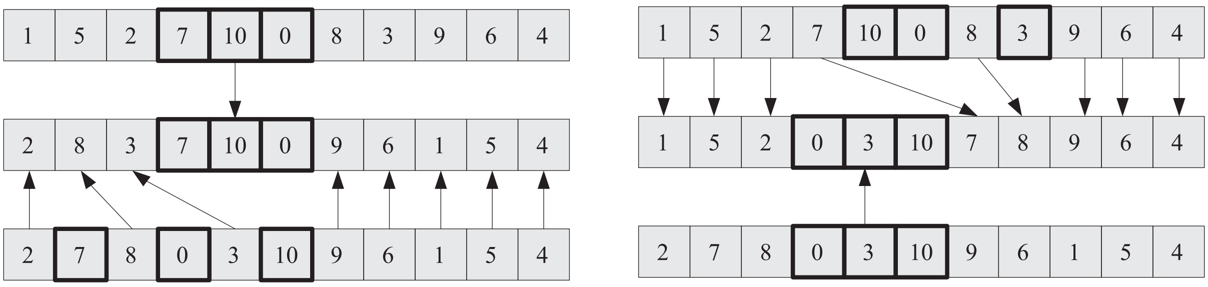

Fig. 5 Illustration of ordered crossover

Since the structures of the individuals of the sub-layer are related to the structure of main-layer, their structures are much more complex than those of the main-layer. The ordered crossover is not appropriate for them. Here we create a new sub-layer child individual by propagating two parent individuals. This method works as follows:

Step 1 Select one parent, and put its 0 into the corresponding position of child.

Step 2 Select one substring of the same parent, and then put it into the corresponding position of child.

Step 3 Delete 0 and same jobs of substring from the other parent.

Step 4 Place the remaining jobs into unfixed positions from left to right.

We use Fig. 6 to illustrate it.

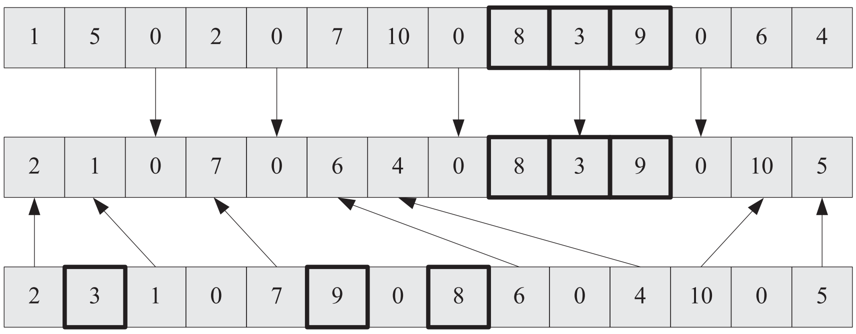

Fig. 6 An illustration of the cross method for sub-layer individuals

4. 4. *Mutation*

By selecting two jobs and then altering their positions, mutation is conducted. Taking a main-layer individual as an example, Fig. 7 illustrates the mutation.

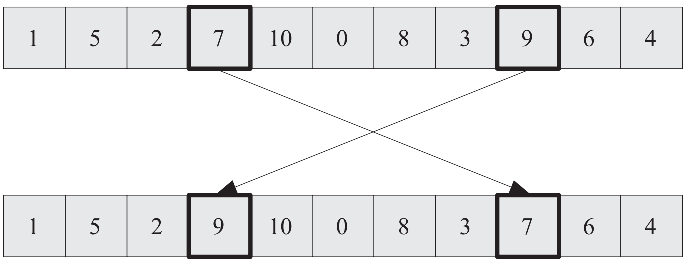

Fig. 7 An illustration of mutation

5. Computational experiments

We first give a detailed result of an example. Fig. 8 shows a layout of 11 jobs in a block. In the experiment, there are 2 YCs and 4 YTs.

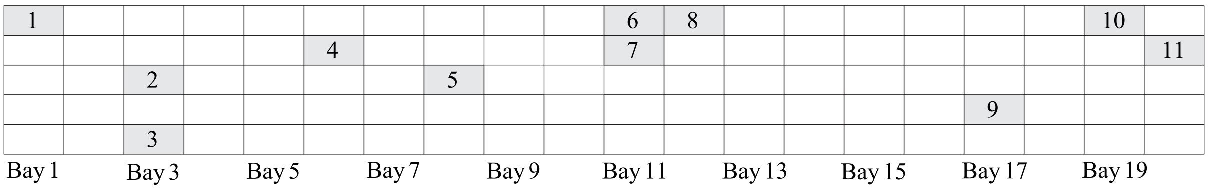

Fig. 8 Layout of 11 jobs

We get the result by MLGA, with the following details:

Job sequence of YC 1: 0−1−2−3−4−5−7−12 (where 0 and 12 represent the initial and final states respectively)

Job sequence of YC 2: 0−6−8−9−10−11−12

Job sequence of YT 1: 0−1−9−7−12

Job sequence of YT 2: 0−6−3−5−12

Job sequence of YT 3: 0−2−10−12

Job sequence of YT 4: 0−8−4−11−12

Make-span: c (12) =703

Operation track positions have been kept, and we can see them in Fig. 9:

We can know the positions of these two YCs and the distance between them at any period. So we can check to see if the result meets the requirements of the non-crossing constraint and the fixed YC separation distance constraint. For example, YC 1 is located in bay 1 at period 0 while YC 2 is located in bay 11, so the distance between them is 10 bays long.

To test the effectiveness of MLGA, we also generate another 15 examples to evaluate it. These examples are based on a container terminal in Ningbo, China. We assume that the speed of YTs is 5 m/s, and the travel speed of YCs is 1.5 m/s. The processing time of a job by YCs ranges from 90 s to 120 s. It takes 90 s to handle a job by QC. The number of jobs, YTs and YCs are listed in Table 1. The results are listed in Table 2.

We know that below a certain number of jobs, the increase in the amount of equipment has a positive influence to the reduction of make-span through Table 1 and 2. From Example 1 to Example 4, the number of YTs was set to be 2, 4, 6 and 8 respectively. From the result, we can see that the increase in the number of YTs makes a significant contribution to the reduction of make-span. But as the number of YTs increases, the reduction of make-span decreases.

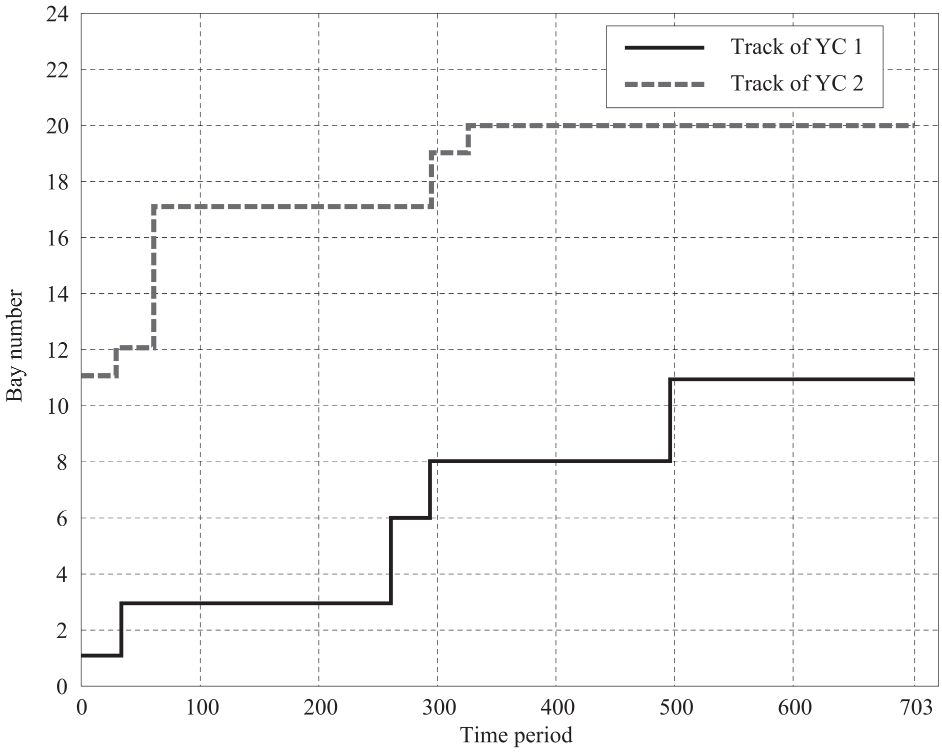

Fig. 9 Tracks of YCs

Table 1 Details of computational examples

Examples	Number of jobs	Number of YTs	Number of YCs
1	10	2	2
2	10	4	2
3	10	6	2
4	10	8	2
5	20	10	2
6	20	3	3
7	20	6	3
8	20	9	3
9	20	12	3
10	30	6	3
11	30	9	3
12	30	12	3
13	30	15	3
14	50	15	3
15	50	18	3

Table 2 Results of examples

Examples	Result (period)	CPU time (secs)
1	1155	55
2	658	64
3	461	72
4	430	75
5	608	67
6	1498	83
7	833	87
8	462	92
9	373	112
10	1148	104
11	670	106
12	523	118
13	428	127
14	683	135
15	475	148

6. Conclusion

An integrated YC and YT scheduling problem based on several constraints is investigated in the paper. These constraints include non-crossing interference, fixed YC separation distance and job-precedence constraint. Due to the computational intractability, we introduced a method called MLGA (Multi-layer genetic algorism) to solve the model. Computational experiments were conducted to evaluate the effectiveness of MLGA, and we see that MLGA can consistently get a satisfactory solution in an acceptable time.

Acknowledgment

The work was partially supported by research grants from Shanghai Municipal Natural Science Foundation [No. 10190502500], Shanghai Maritime University Start-up Funds, Shanghai Science & Technology Commission Project [No. 09DZ2250400] and Shanghai Education Commission Project [No. J50604].

References

Bish, E. K., 2003, A multiple-crane-constrained scheduling problem in a container terminal, *European Journal of Operational Research,* v. 144-1, p. 83-107, January 2003.

Cao, J. X., Lee, D. H., Chen, J. H. and Shi, Q., 2010. The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods, *Transportation Research Part E: Logistics and Transportation Review,* v. 46-3, p. 344-353, May 2010.

Cao, J. X., Shi, Q., Lee, D. H., 2010, Integrated Quay Crane and Yard Truck Schedule Problem in Container Terminals, *TSINGHUA SCIENCE AND TECHNOLOGY,* v. 15-4, p. 467-474, August 2010.

Chen, L., Bostel, N., Dejax, P., Cai, J. and Xi. L., 2007, A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal, *European Journal of Operational Research,* v. 181-1, p. 40-58, August 2007.

Cheung, R. K., Li, C., Lin, W., 2002. Inter-block Crane Deployment in Container Terminals, *TRANSPORTATION SCIENCE* v. 36-1, p. 79-93, February 2002.

Guo, X., Huang, S. Y., Hsu, W. J., and Low, M. Y. H., 2011, Dynamic yard crane dispatching in container terminals with predicted vehicle arrival information, *Advanced Engineering Informatics.*

Kim, K. H. and Kim, K. Y., 1999, An Optimal Routing Algorithm for a Transfer Crane in Port Container Terminals, *TRANSPORTATION SCIENCE,* v. 33-1, p. 17-33, February 1999.

Kim, K. H., Bae, J. W., 2004. A Look-Ahead Dispatching Method for Automated Guided Vehicles in Automated Port Container Terminals, *TRANSPORTATION SCIENCE,* v. 38-2, p. 224-234, May 2004.

Koo, P. H., Lee, W. S. and Jang, D. W., 2004. Fleet sizing and vehicle routing for container transportation in a static environment, *OR Spectrum,* v. 26, p. 193-209, 2004.

Lau, H. Y. K., and Zhao, Y., 2008, Integrated scheduling of handling equipment at automated container terminals, *International Journal of Production Economic,* v. 112-2, p. 665-682, April 2008.

Lee, D. H., Cao, Z. and Meng, Q., 2007, Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. *International Journal of Production Economics,* v. 107-1, p. 115-124, May 2007.

Li, W., Wu, Y., Petering, M. E. H., Goh, M. and de Souza, R., 2009. Discrete time model and algorithms for container yard crane scheduling, *European Journal of Operational Research,* v. 198-1, p. 165-172, October 2009.

Ng, W. C., and Mak, K. L., 2005, Yard crane scheduling in port container terminals, *Applied Mathematical Modeling,* v. 29-3, p. 263-276, March 2005.

Ng, W. C., 2005, Crane Scheduling in container yards with inter-crane interference, *European Journal of Operational Research,* v. 164-1, p. 64-78, July 2005.

Ng, W. C., Mak, K. L. and Zhang, Y. X., 2007, Scheduling trucks in container terminals using a genetic algorithm. *Engineering Optimization,* v. 39-1, p. 33-47, January 2007.

Nishimura, E., Imai, A., and Papadimitriou, S., 2005. Yard trailer routing at a maritime container terminal. *Transportation Research Part E: Logistics and Transportation Review,* v. 41-1, p. 53-76, January 2005.

Zhang, C., Wan, Y., Liu, J., Linn, R. J., 2002. Dynamic crane deployment in container storage yards. *Transportation Research Part B,* v. 36, p. 537-555, 2002.