Scheduling Yard Cranes in a Container Terminal Using a New Genetic Approach

K.L. Mak, D. Sun

Abstract—**Effective and efficient scheduling of yard crane operations is essential to guarantee a smooth and fast container flow in a container terminal, thus leading to a high terminal throughput. This paper studies the problem of scheduling yard cranes to perform a given set of loading and unloading jobs with different ready times in a yard zone. In particular, the inter-crane interference between adjacent yard cranes which results in the movement of a yard crane being blocked by adjacent yard cranes is studied. The objective is to minimize the sum of yard crane completing times. Since the scheduling problem is NP-complete, a new hybrid optimization algorithm combining the techniques of genetic algorithm and tabu search method (GA-TS) is proposed to solve the challenging problem. Two new operators, namely the Tabu Search Crossover (TSC) and the Tabu Search Mutation (TSM), are introduced into the proposed algorithm to ensure efficient computation. A set of test problems generated randomly based on real life data is used to evaluate the performance of the proposed algorithm. Computational results clearly indicate that GA-TS can successfully locate cost-effective solutions which are on average 20% better than that located by GA. Indeed, the proposed hybrid algorithm is an effective and efficient means for scheduling yard cranes in computer terminals.**

Index Terms—**Yard Crane, inter-crane Interference, Hybrid algorithm, Genetic Algorithm, Tabu Search**

I. INTRODUCTION

With the rapid trade globalization, the marine transportation is getting more and more popular. Large numbers of cargos are moved in containers through ports. Therefore, effective and efficient management of port container terminals is quite important in marine transportation development. In addition, container ports compete with each other for better customer service. Of all the service performance measures, vessel turnaround time, which is the average time that a vessel stays in a terminal, is the key one. The most effective method to reduce the terminal turnaround time is to improve the productivity of handling activities.

It is essential to study the operational processes of a port container terminal. Fig. 1.1. shows the typical container flow between major handling equipment in a port container terminal. When a vessel arrives at the terminal, containers are normally discharged from the vessel onto trucks by quay cranes. Unloaded import containers are transported to the yard and off-loaded by yard cranes for storage. For export containers, containers are mounted onto trucks by yard cranes and transported to the quayside for loading onto the vessel by quay cranes. Steenken et al. [8] described and classified the main logistics processes and operations in container terminals and presented a survey of methods for their optimization. Stahlbock and Vo β [7] extended the study of [8].

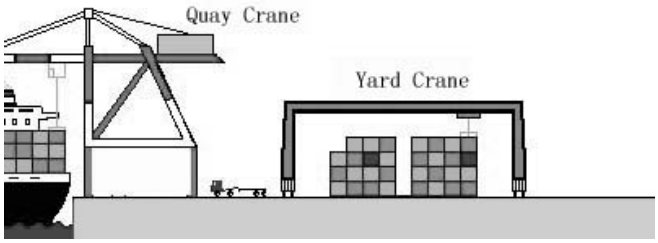


Fig. 1.1. Typical container flow in terminal operations

In a container terminal, yard cranes are important equipment for loading containers onto and unloading containers from mobile trucks, and for stacking containers in storage locations in the container yard according to the sequence predetermined by terminal planners. However, the low physical operation rate of such equipment and their frequent movements when performing handling tasks in the yard very often cause bottlenecks in the container flow in the terminal. Therefore, yard crane scheduling plays a significant role in port management. An effective yard crane scheduling can reduce truck waiting time, speed up the container flow to and from vessels. A number of researchers studied scheduling of yard cranes in a container yard.

Kim and Kim [2] considered the problem of routing a single straddle carrier, which is transportation equipment with container handling capability, to transport export containers to a loading vessel. To minimize the total container handling time of a straddle carrier, a mixed integer program was formulated. Zhang et al. [9] investigated the Rubber Tyred Gantry Cranes (RTGCs) deployment problem. The objective was to find the times and routes of cranes movement among blocks. So that the total delayed workload in the yard was minimized. A mixed integer program (MIP) model was formulated and solved by Lagrangean relaxation. Linn and Zhang [10] also studied the RTGCs deployment problem. The objective was to minimize the total workload overflow through determining the crane deployment frequency and routes over a planning horizon. A heuristic algorithm was also developed to provide a near optimal solution for crane deployment.

Rail Mounted Gantry Cranes (RMGs) are also used frequently in practice. These cranes are particularly effective for rail/road transshipments of large quantities of

K.L. Mak is Professor at the Dept. of Industrial and Manufacturing Systems Engineering (IMSE), The University of Hong Kong, Hong Kong, (e-mail: makkl@hkucc.hku.hk).

D. Sun is a PhD student at the Dept. of IMSE, HKU, Hong Kong, (e-mail: h0795497@hku.hk).

containers. For yard cranes (RMGs) running on rails, movement is restricted to a predetermined zone. Within the zone, yard cranes can move freely as long as they do not cross over each other in the zone.

Fig.1.2 shows a typical partial container yard layout with RMGs yard cranes. The yard is composed of multiple blocks called yard blocks. Each yard block consists of a contiguous stretch of slots (40-60 slots). Each ground slot, denoted as a rectangle in the diagram, can store 5-7 containers. In most container terminals, operators simplify the control of yard crane movements and reduce the amount of time in which the yard cranes occupy trucks travelling lanes by grouping adjacent yard blocks together to form zones. In the layout depicted in Fig.1.2, there are two zones in the yard, yard zone 1 and yard zone 2 which are formed by grouping blocks 1, 2 and 3, and blocks 4, 5 and 6 together, respectively.

Due to sharing of the traveling lane among two or more yard cranes in a yard zone, inter-crane interference, a planned move of a yard crane blocked by the other yard cranes, may happen. For example, in Fig. 1.2, if yard crane 2 is handling its job, yard crane 1 cannot move across yard crane 2 to handle jobs on its right hand side. Therefore, if yard crane 1 is going to handle jobs in block 3, it has to wait until yard crane 2 has completed its job.

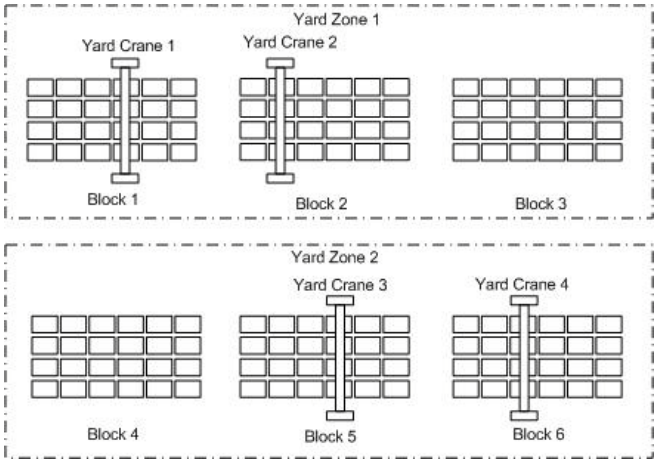


Fig.1.2. Layout of container yard

Due to the complexity of considering inter-crane interference constraints in scheduling yard cranes, Ng and Mak [11] neglected the crossing movement restriction by assuming only a yard crane in a yard zone, where the yard crane can move freely to perform all handling jobs generated by different vessels. They studied the problem of scheduling only a yard crane to perform a given set of loading/unloading jobs with different ready times. The objective is to minimize the sum of job waiting times. The authors proposed a branch and bound algorithm to locate the optimal solution of the scheduling problem. Efficient and effective algorithms are proposed to find lower bounds and upper bounds. However, since their paper only handled a yard crane scheduling, it is not directly applicable in practice.

Only a few papers addressed the routing problem regarding yard cranes with inter-crane interference constraints. Lim et al. [3] studied a model that took into account of interference between yard cranes as "non-crossing" constraint and used a tabu search heuristic for solutions. However, the paper did not consider the handling time of yard cranes, the travel time between two jobs and the waiting time due to yard crane interference. Ng. [4] addressed the scheduling problem for yard cranes considering interference among adjacent yard cranes. The paper divided the yard to several zones and used a dynamic programming model to determine the sequence of jobs for each yard crane. In [4], the time was discretized beforehand. However, in real world, the time is continuous. Jung and Kim [1] scheduled loading operations when multiple yard cranes are operating in the same block. They considered interferences between adjacent yard cranes. The objective was the minimization of the make-span of the yard crane operation. The paper used a genetic algorithm and a simulated annealing method to solve the model.

This paper studies the problem of scheduling multiple yard cranes in a yard zone to minimize the sum of the completion times of yard cranes. When calculating the completion time of each yard crane, we consider the container ready time, the handling time, the yard crane travelling time and the waiting time of yard cranes due to inter-crane interference. The remainder of the paper is organized as follows. Section 2 proposes a mixed integer program model for the scheduling problem. A new hybrid genetic algorithm and tabu search method for solving the scheduling problem is given in section 3. Section 4 presents the results of computational experiments. Conclusions are in section 5.

II. Model Development

A mixed integer program mathematical model describing the characteristics of the yard crane scheduling problem is developed. Some important assumptions for the formulation are listed below:

a) A yard truck can transport an export/import container from the yard storage place/quayside to quayside/yard storage place.

b) A job is defined as a yard crane loading/unloading an export/import container onto/from a yard truck from/to its storage yard place.

c) The handling time of a job is fixed.

d) Multiple yard cranes serve simultaneously in a yard zone which can move freely as long as they do not cross over each other. Inter-crane interference between yard cranes is taken into consideration.

There are n jobs to be handled by m identical yard cranes in a yard zone of θ slots. The yard cranes are ordered in an increasing order of their relative locations in the yard zone. The meanings of variables in the model are listed below:

r_i = the ready time of job i

l_i = the location of job i

d_{ij} = the time required for yard cranes to travel from l_i to l_j

h = the time required by a yard crane to handle one job

Decision variables:

W_i = (S_i, D_i) the handling time window for job i

D_i = the completion time of job i

S_i = the time at which the yard crane assigned starts to handle job i

t_i = the arrival time of the yard crane assigned to job i

C_k = the completion time of yard crane k

$X_{ij}^k = \begin{cases} 1 & \text{if yard crane } k \text{ performs job } i \text{ before job } j \\ 0 & \text{otherwise} \end{cases}$

Y_i = the yard crane assigned to handle job i

The mathematical model describing the characteristics of the scheduling problem is shown below:

Minimize $\sum_{k=1}^{m} C_k$

Subject to

$$\sum_{j=1}^{n} X_{0j}^k = 1, \quad k = 1,\ldots,m \tag{1}$$

$$\sum_{i=1}^{n} X_{iT}^k = 1, \quad k = 1,\ldots,m \tag{2}$$

$$\sum_{k=1}^{m}\sum_{i=0}^{n} X_{ij}^k = 1, \quad j = 1,\ldots,n \tag{3}$$

$$\sum_{j=1}^{n} X_{ij}^k - \sum_{j=1}^{n} X_{ji}^k = 0, \quad k = 1,\ldots,m \tag{4}$$

$$D_i = S_i + h, \quad i = 1,\ldots,n \tag{5}$$

$$S_i = \max\{r_i, t_i\}, \quad i = 1,\ldots,n \tag{6}$$

$$D_j - D_i \geq d_{ij} + h - (1 - X_{ij}^k)M, \ i, j = 1,\ldots,n, \text{ and } i \neq j \tag{7}$$

$$(Y_i - Y_j)(l_i - l_j) > 0 \quad \text{if } \bigcap_{i \neq j} W_i \cap W_j \neq \phi,$$

$$i, j = 1,\ldots,n, \quad \text{and } i \neq j \tag{8}$$

$$D_j + d_{jT} - C_k \leq M(1 - X_{jT}^k), \ j = 1,\ldots,n, \ k = 1,\ldots,m \tag{9}$$

$$X_{ij}^k \in \{0,1\}, \quad i, j = 1,\ldots,n, \quad k = 1,\ldots,m \tag{10}$$

$$D_i, \ S_i, \ t_i, \ C_k \geq 0, \quad i = 1,\ldots,n, \ k = 1,\ldots,m \tag{11}$$

$$Y_i \in \{1, 2, \ldots\} \quad i = 1,\ldots,n. \tag{12}$$

In equation (7), M is a big positive number.

The objective of the scheduling problem is to minimize the sum of the completion times of the yard cranes. Constraints (1) and (2), select the first and last tasks for each yard crane, respectively. Constraint (3) indicates that every job must be completed by one yard crane. Constraint (4) is a flow balance constraint for yard crane travels. Constraint (5) computes a job's completion time. Constraint (6) shows that the yard crane assigned for a job would start to handle the job after the job ready time and its arrival time to the job. Constraint (7) implies the relationship between the completion time of a job and that of its successors. By constraint (8), interference among yard cranes can be avoided. The completion time of each yard crane is defined by constraint (9). X_{ij}^k can be 0 or 1 by constraint (10). Variables in Constraint (11) have nonnegative value, while Y_i is a nonnegative integer variable by constraint (12).

III. HYBRID GENETIC ALGORITHM AND TABU SEARCH

The multiple yard cranes scheduling problem could be simplified by decomposing the whole problem into several sub-problems where the yard zone is partitioned into several ranges and all the jobs within a range are handled by a yard crane. Each subproblem is a problem of non-preemptive scheduling with different job ready times on a single machine which has been proved NP-complete by Lenstra et a. [12]. Thus, the multiple yard cranes scheduling problem must be an NP-complete problem.

NP-complete problems could not be solved optimally in polynomial-bound time. Thus, in order to find an efficient and effective solution, this paper develops a new hybrid genetic algorithm and tabu search method. Two new TS-based operators of genetic algorithm are designed. The first one is adding tabu search to the mutation operation of genetic algorithm called (TSM). Genetic algorithm and tabu search depend on disparate search principles. In general, TS is good at performing deeper exploitation since it is based on local search heuristics. GA is short of the power of deeper exploitation within the promising regions but performs well at exploring different regions due to the implicit parallelism feature. It would be desirable to hybridize them to achieve a better balance between exploration and exploitation. The TSM fulfills the objective.

The other one is the new tabu search crossover operator called (TSC). In canonical GA, all chromosomes are replaced by their offspring after the crossover procedures as the population evolves. Due to the extremely short life span of the individuals, the search algorithms therefore do not have sufficient time to sample out the useful schemata from individuals. To avoid the shortage, the parents would stay in the population to join the selection process with their offspring. However, this method would cause new problem. Some good individuals would stay permanent and give birth to new generations which would lead to premature and stalling of the search process. In order to ameliorate this situation, this paper designs the TSC. It is a memory-based strategy like tabu search. A TSC list is designed to store the individuals who have been selected as parents. The TSC list records the number of times (np) of each chromosome selected as parent. The probability to become parent would be inversely proportional to the np of each individual in the TSC list. Using this method, this paper effectively avoids premature caused by crossover operation.

The general outline of each step of the proposed algorithm is presented below:

Step 1:

Set $k = 0$, and randomly generate the initial population P_0. The size of P_0 is n_0.

Step 2:

a) Select n_0 individuals from population P_k to form population S_k by using the selection operator of conventional GA.

b) Select parent individuals to give birth to new individuals according to the individuals' crossover probabilities calculated by using the value of np of each individual in the TSC list. Population S_k and the new individuals together form population C_k. The size of C_k is n_1 ($n_1 > n_0$).

c) Apply the TSM mutation operator to population C_k to obtain population M_k.

d) Set $k = k+1$, $P_k = M_k$

Step 3: If $k <$ Total Generation, go to Step 2.

A. Representation

This paper uses the same two-part chromosome structure as in [5]. (2 4 8 9 5 3 9 6 1 | 4 3 2) illustrates an example chromosome for representing a crane schedule for assigning three yard cranes ($m = 3$) to process nine jobs ($n = 9$) using the two-part chromosome structure. There are two distinct parts in the chromosome with a total length of $m+n$. In the first part, the n jobs are represented by a permutation of the integers from 1 to n. The second part of the chromosome, which is of length m, represents the number of jobs assigned to each of the m yard cranes. The values assigned to the second part of the chromosome are constrained to be m non-negative integers whose sum must equal the number n of jobs. In the example shown above, the first yard crane would sequentially process jobs 2, 4, 8 and 9, the second yard crane should process jobs 5, 3 and 7, and the last yard crane would process jobs 6 and 1.

B. Fitness evaluation and selection operation

Inter crane interference occurs when the next job fo yard crane A is not located between yard crane A and yar crane B, and yard crane B is working on another job. In th case, yard crane A has to wait until the yard crane B ha completed its current job. Consequently, the procedure t evaluate the objective function value of an individual is a follows:

Step 1:

Calculate the objective function value of the individual without considering constraint (8). During this process, we can obtain the value of $W_i = (S_i, D_i)$ of job i ($i = 1, 2, \ldots, n$).

Step 2: (Check possible inter crane interference) -

a) Sort the n jobs in ascending order of the value of its S_i to determine the sequence ($1', 2', \ldots, n'$).

b) Find the first pair of jobs (i, j) such that $\bigcap_{i \neq j} W_i \cap W_j \neq \phi$ and constraint (8) is not satisfied.

Let $i = 1'$ to $n'-1$. For each i, check $S_j < D_i$ for $j = i+1$ to n' until the condition is satisfied and jobs (i, j) does not satisfy constraint (8). If such a pair of (i, j) exits, inter crane interference occurs. Otherwise, inter crane interference does not exit and the current result obtained is the final objective function value of the individual.

Step 3: (Conflict resolution).

Yard crane Y_j must wait until yard crane Y_i has completed job i. Then, Y_j moves from l_i to l_j and begins to handle job j. Update all the W_k affected by this operation.

Step 4:

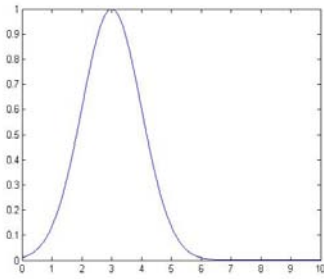
Repeat steps 2 and 3 until no inter crane interference could be found.

C. Tabu Search Crossover (TSC) operation

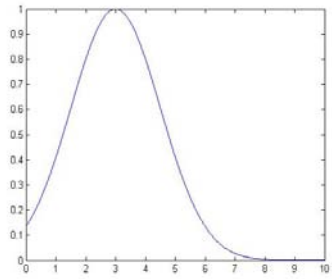
In crossover procedure, there are two questions should be answer. The first one is how to choose individuals to be parents. The other one is how to use information of two parents to generate good offspring. This section answers these two questions, respectively. The individuals are chosen as parents depending on their crossover probabilities which is computed as following:

$$Pb(np(b_i)) = \exp\left(-\frac{(np(b_i)-a)^2}{b^2}\right)$$

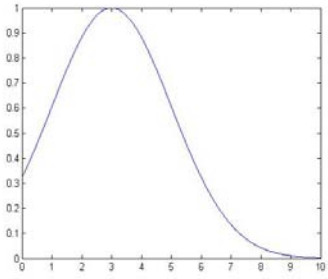
where $np(b_i)$ is the value of solution b_i in TSC list. TSC list records the number of times that an individual has been selected as parent. a and b are given parameters which control the shape of probability function $Pb(np(b_i))$. If $np(b_i) = a$, then $Pb(np(b_i)) = 1$. $Pb(np(b_i))$ gradually converges to 0 with $np(b_i)$ growing up.



(a=3, b=1)



(a=3, b=1.5)



(a=3, b=2)

Fig. 3.1 graphics of $Pb(np(b_i))$ with different parameters

The procedure to choose the individuals as parents is as following:

Step 1:

Calculate crossover probability for each individual, $Pb(np(b_i))$.

Step 2:

Generate a random number RN between 0 and 1.

Step 3:

If $RN < Pb(np(b_i))$, the individual b_i is chosen as parents. $np(b_i) = np(b_i) + 1$.

After getting parents, this paper uses the method suggested by [5] to generate offspring. Consider the crossover operation of two parents P_1 and P_2 to reproduce two offspring O_1 and O_2. In the two-part chromosome representation adopted in this paper, the first part of chromosome Q can be partitioned into m mutually exclusive sets $\Omega_1(Q), \Omega_2(Q), \ldots, \Omega_m(Q)$, where $\Omega_k(Q)$ denotes the sequence of jobs to be processed by yard crane k $(k = 1, 2, \ldots, m)$ as specified by chromosome Q. The procedure to generate O_1 is given below:

Step 1:

Rank the n jobs in non-decreasing order of their ready times and let J be the ranked set.

Step 2:

For $k = 1, 2, \ldots, m$, find the jobs that are elements of both $\Omega_k(P_1)$ and $\Omega_k(P_2)$, insert these jobs in $\Omega_k(O_1)$ according to their order in set J, and delete the inserted jobs from set J.

Step 3:

Let job i be the first job in set J. For $k = 1, 2, \ldots, m$, tentatively place the job as the last element of $\Omega_k(O_1)$ and compute the completion time $D_k(i)$ of job i. Find q where $D_q(i) = \min_{k=1,2,\ldots,m}\{D_k(i)\}$, place job i as the last element of $\Omega_q(O_1)$, and delete job i from set J.

Step 4:

Repeat step 3 until set J is a null set.

Step 5:

Determine the number of jobs for each crane and place the numbers in the second part of O_1.

The procedure for generate O_2 is the same as that for generating O_1 except that, in step 3, job i is placed as the first element of $\Omega_k(O_2)$.

D. Tabu Search Mutation (TSM) operation

Mutation forces GA searching new areas. Adding Tabu Search in mutation operation would improve the deeper exploitation for GA. By using Tabu List, it could avoid premature and finally get global optimal solution. The procedure of tabu search mutation is following:

x_0 the initial solution. The tabu list is TL. Neighbourhood of x_0 is $N(x_0)$.

Input x_0

While ($t < T$)

$TL = \phi$; Set the best solution $x = x_0$; $t = 0$;

Get set $CN(x_0) \subseteq N(x_0)$ and $CN(x_0) \bigcap TL = \phi$

get $x' \in CN(x_0)$ and x' has the best fitness value in $CN(x_0)$;

Move x to x'; $t = t + 1$;

Update (x; x_0; TL)

Output x

IV. Computation Results

The effectiveness of the hybrid Genetic Algorithm and Tabu Search method described in Section 3 is tested by compring its performance with that of the Branch and Bound algorithm and of the Genetic Algorithm.

A. Test Problem Set and Parameters

The test problems are randomly generated using the method proposed in Ng. [4]. In typical yard operations, 2 to 4 yard cranes are deployed to a yard zone that contains three 40-slot yard blocks, $h = 4$ minutes, $d_{ij} = 3 * \left| l_i - l_j \right|$ seconds and the number of jobs in a yard zone to be handled in a planning horizon of one hour ranges from 10 to 60 jobs. The mean time between successive job arrivals is approximately $4/m$ minutes. For different combinations of n, m and θ, a number of test problems were constructed with l_i randomly selected from a uniform distribution $[1, \theta]$ and r_i constructed from the inter-arrival times which are randomly generated from an exponential distribution with mean $4/m$ minutes.

The parameters of GA-TS need to be determined. This paper assumes that $a = 3$, $b = 2$ in the TSC operator. In the mutation operation, the length of the tabu list is 4, the number of neighbors searched is 5 and the termination condition is $T = 6$. In both GA-TS and GA, the crossover and mutation probabilities are fixed to 0.6 and 0.4, respectively. The initial population size is set to 50 and the stop condition is 100 iterations. The computer used is equipped with Inter Pentium 2.4GHz CPU and 512MB RAM. The calculation time is shown in seconds.

B. Comparison with Branch and Bound Algorithm

This paper uses the technique of Least Cost Branch and Bound to get the optimal solution. In such an approach, the solution space is often organized as a tree. Details of the method can be found in [6]. A small-size test problem is used in the experiment for performance evaluation. Let TC_{GT} and TC_{IP} be the best objective function values found by using the hybrid GA-TS method and the Branch and

Bound algorithm, respectively. $Time_{GT}$ and $Time_{IP}$ are the computational times of the algorithms. A set of 20 test problems is randomly generated for each combination of parameters. The results are shown in Table 1.

Table 1 Performance of proposed algorithm on small-scale test problem

Instances	n	5	5	10	10
	m	1	2	2	2
	θ	10	10	20	30
$\frac{(TC_{GT}-TC_{IP})}{TC_{IP}}*100\%$	Max	0.0	0.0	2.2	3.3
	Min	0.0	0.0	0.0	0.0
	Mean	0.0	0.0	1.3	2.1
$Time_{GT}$	Max	5.03	5.79	6.12	6.83
	Min	3.09	3.84	4.43	4.74
	Mean	3.61	4.56	5.61	6.18
$Time_{IP}$	Max	151	172	924	1057
	Min	45	50	472	561
	Mean	74	88	650	732

Comparing the results between branch and bound algorithm and GA-TS for small-scale problems, for $n=5$, GA-TS can always get the global optimum. Although in some cases GA-TS may not always reach the optimum, its solutions are quite near optimum with only average 2% above optimal solutions. Thus, the quality of solutions found by GA-TS is acceptable. In addition, by using GA-TS, it would averagely cost less than 5 seconds to solve problems when $n=5$ and 7 seconds when $n=10$. In contrast, the runtime used by Branch-and-Bound is much longer. Since the Branch and Bound algorithm requires a n-element permutation, the solution space tree is a permutation tree with $n!$ leaves. Searching through all nodes of the tree would cost $\Omega(n!)$ time. Thus, the runtime of Branch-and-Bound algorithm increases by geometric rate as the number of jobs growing. In this experiment, using Branch-and-Bound algorithm, the average CPU time required is 74 seconds to solve the problems when $n=5$. As to the problems when $n=10$, it increases to 650 seconds. Therefore, when the number of jobs is large, Branch-and-Bound algorithm is not suitable as a solution approach.

C. Comparison with Genetic Algorithm

Results given by hybrid Genetic Algorithm and Tabu Search method and canonical Genetic Algorithm are presented in table 2. For each combination of parameters, 20 random test problems are generated. TC_{GA} is the objective function value found by Genetic Algorithm. The results show the performance of GA-TS clearly outperforms GA. Almost all solutions obtained by GA-TS are better than the corresponding solutions obtained by GA except one. The GA-TS solutions are on average 20% better than solutions found by GA.

Table 2 Comparison between GA-TS and GA

Instances			$(TC_{GA}-TC_{GT})/TC_{GT}*100\%$		
n	m	θ	Max	Min	Mean
10	2	40	11.8	0.0	5.7
20	2	40	33.3	6.7	22.6
30	3	40	39.6	5.5	21.9
40	3	40	41.4	11.7	23.8
50	4	40	24.2	6.0	10.6
60	4	40	23.4	2.0	11.6
10	2	80	17.0	0.0	8.9
20	2	80	41.9	4.7	28.0
30	3	80	44.9	4.1	23.1
40	3	80	26.1	-0.1	14.6
50	4	80	36.5	4.3	27.1
60	4	80	14.1	1.9	7.0
10	2	120	33.9	4.0	20.6
20	2	120	42.6	18.0	22.9
30	3	120	53.8	3.1	35.7
40	3	120	44.6	10.7	35.3
50	4	120	58.6	9.0	40.2
60	4	120	25.6	7.8	10.2

V. CONCLUSION

This paper studys the yard crane scheduling problem with inter-crane interference. The objective is to minimize the sum of the completion time of the yard cranes. When caculating the completion time of each yard crane, we consider the container ready time, handling time, the yard crane travelling time and the waiting time of yard cranes due to inter-crane interference. Since the problem is NP-complete, a new hybrid Genetic Algorithm and Tabu Search method has been developed to resolve the challenging problem. The computation results show the new algorithm is superior to GA with its solutions on average 20% better than solutions found by GA. Tabu Search Crossover (TSC) and Tabu Search Mutation (TSM) are effective to avoid premature and speed up convergence. Comparing with branch and bound algorithm, GA-TS can always give a reasonable solution within limited time.

REFERENCE

[1] S.H. Jung, and K. H. Kim. "Load scheduling for multiple quay cranes in port container terminals" *Journal of Intelligent Manufacturing*, 17, 2006, pp. 479-492

[2] K.H. Kim, and K.Y. Kim. "An optimal routing algorithm for a transfer crane in port container terminals", *Transportation Science*, 33, 1999, pp. 17 – 33.

[3] A. Lim, F. Xiao, B. Rodrigues, and Y. Zhu. "Crane scheduling using tabu search". In *14th IEEE International Conference on Tools with Artificial Intelligence*, Washington DC, USA 2002.

[4] W. C. Ng, "Crane scheduling in container yards with inter-crane interference". *European Journal of Operational Research*, 164, 2005, pp. 64–78.

[5] W.C. Ng, K.L Mak,. and Y.X Zhang, "Scheduling trucks in container terminals using a genetic algorithm". *Engineering Optimization*, 39, 2007, pp. 33-47.
[6] S. Sahni, *Data Structures, Algorithms, and Applications in C++*. United States: McGraw-Hill, Inc., 1998.
[7] R. Stahlbock, S. Vo β , " Operations research at container terminals: a literature update". *OR Spectrum*, 30(1), 2008, pp. 1 – 52
[8] D. Steenken, S. Voß, and R. Stahbock, "Container terminal operation and operations research – a classification and literature review". *OR Spectrum*, 26, 2004, pp. 282-292.
[9] C. Zhang, Y. Wan, J. Liu, R.J. Linn, "Dynamic crane deployment in container storage yards", *Transportation Research Part B*, 36, 2002, pp. 537 – 555.
[10] R. J. Linn and C. Zhang, "A heuristic for dynamic yard crane deployment in a container terminal", *IIE Transactions*, 35, 2003, pp. 161-174.
[11] W.C. Ng and K.L. Mak, "Yard crane scheduling in port container terminals". *Applied Mathematical Modeling*, 29, 2005, pp. 263-276
[12] J.K. Lenstra, A.H.G. Rinnooy and P.Brucker "Complexity of machine scheduling problems", *Annals of Discrete Mathematics*, 1, 1997, pp. 343-362