Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Part E 44 (2008) 124–135

TRANSPORTATION RESEARCH PART E

www.elsevier.com/locate/tre

Quay crane scheduling with non-interference constraints in port container terminals

Der-Horng Lee [a,*], Hui Qiu Wang [a], Lixin Miao [b]

[a] *Department of Civil Engineering, National University of Singapore, Singapore 117576, Singapore*
[b] *Research Center for Modern Logistics, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, PR China*

Received 8 March 2006; received in revised form 23 June 2006; accepted 26 August 2006

Abstract

The quay crane scheduling problem studied in this paper is to determine a handling sequence of holds for quay cranes assigned to a container vessel considering interference between quay cranes. This paper provides a mixed integer programming model for the considered quay crane scheduling problem that is NP-complete in nature. A genetic algorithm is proposed to obtain near optimal solutions. Computational experiments are conducted to examine the proposed model and solution algorithm. The computational results show that the proposed genetic algorithm is effective and efficient in solving the considered quay crane scheduling problem.

Keywords: Port container terminal; Quay crane scheduling; NP-completeness; Genetic algorithm

1. Introduction

Nowadays competition between port container terminals, especially between geographically close ones, is rapidly increasing. How to improve the competitiveness of port container terminals is, therefore, an immediate challenge, with which port operators are confronted. In terms of port competitiveness, the makespan of a container vessel, which is the latest completion time among all handling tasks of the container vessel, is a critical success factor (Steenken et al., 2004). In reality, quay crane scheduling significantly affects the makespan of a container vessel since quay cranes are the interface between land side and water side in any port container terminals. Thus, this paper aims to study quay crane scheduling problem to enhance the competitiveness of port container terminals.

As illustrated in Fig. 1, container vessels are typically divided longitudinally into holds that open to the deck through a hatch. Holds are about eight containers deep, and containers can also be stacked (about six high) on deck (Daganzo, 1989). The interference between quay cranes is that quay cranes cannot cross over

[*] Corresponding author. Tel.: +65 6516 2131; fax: +65 6779 1635.
E-mail address: cveleedh@nus.edu.sg (D.-H. Lee).

1366-5545/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.tre.2006.08.001

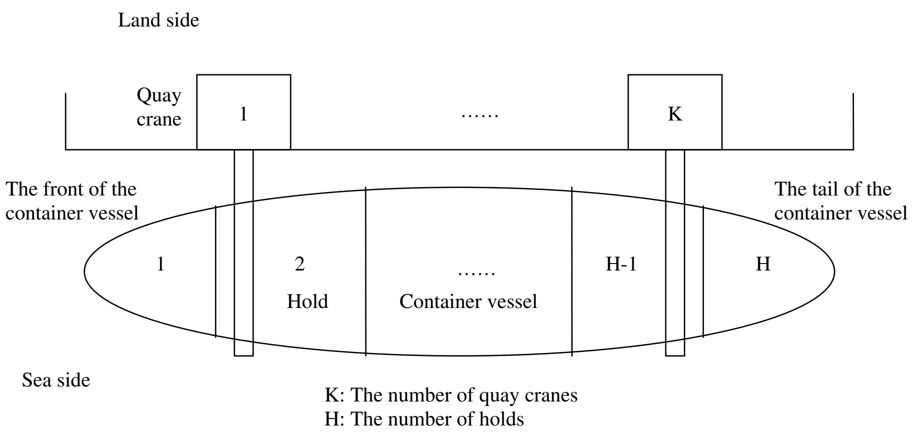
Land side
Quay crane
1
......
K
The front of the container vessel
The tail of the container vessel
1
2
Hold
......
Container vessel
H-1
H
Sea side
K: The number of quay cranes
H: The number of holds

Fig. 1. The illustration of the QCSNIP.

each other because they are on the same track. In practice, only one quay crane can work on a hold at any time. Generally, a quay crane can move to another hold until it completes the current one. The average processing time of a hold is about 2 h and the travel time of a quay crane between two holds is about 1 min. The quay crane scheduling problem in port container terminals is to determine a handling sequence of holds for quay cranes assigned to a container vessel in fulfilling pre-specified objectives and satisfying various constraints. For instance, there are 10 holds in a container vessel, and two quay cranes are allocated to handle the container vessel. Table 1 illustrates a feasible quay crane schedule for this instance. It shows the handling sequence of holds for every quay crane, the processing time of each hold and the time schedule for handling every hold.

Daganzo (1989) studied the static and dynamic quay crane scheduling problems for multiple container vessels. Daganzo (1989) assumed that container vessels were to divide into holds, and only one quay crane could work on a hold at a time. Quay cranes could be moved freely and quickly from hold to hold, and container vessels could not depart until all their holds had been handled. The objective was to serve all these container vessels, while minimizing their aggregate cost of delay. Exact and approximate solution methods for quay crane scheduling were presented in Daganzo (1989). Furthermore, Peterkofsky and Daganzo (1990) developed a branch and bound solution method for the static quay crane scheduling problem. Nevertheless, both papers did not consider the interference between quay cranes, which means the quay cranes could unrealistically cross over each other.

Lim et al. (2004) augmented the static quay crane scheduling problem for multiple container vessels by taking into account non-interference constraints. They assumed that containers from a given area on a container vessel were a job, and there was a profit value when a job was assigned to a quay crane. The objective was to find a crane-to-job matching which maximized the total profit. Dynamic programming algorithms, a probabilistic tabu search, and a squeaky wheel optimization heuristic were proposed in solving the problem.

Table 1
An illustration of a quay crane schedule

Quay Crane 1				Quay Crane 2			
Operation sequence	Hold number	Processing time of a hold (min)	Completion time of the quay crane (min)	Operation sequence	Hold number	Processing time of a hold (min)	Completion time of the quay crane (min)
1	1	98	98	1	2	81	81
2	3	119	217	2	5	178	259
3	4	52	269	3	10	171	430
4	9	101	370	4	8	162	592
5	7	114	484				
6	6	81	565				

However, it is difficult to define a profit value associated with a crane-to-job assignment in practice, and hence this research cannot be applied in port container terminals easily.

Kim et al. (2004) studied the load-sequencing problem for outbound containers in port container terminals which was to determine the pick up sequence by transfer cranes in the yard and the loading sequence of slots in the container vessel by quay cranes simultaneously. A beam search algorithm was proposed to solve this problem. Kim and Park (2004) discussed the quay crane scheduling problem with non-interference constraints in which only single container vessel was considered. Kim and Park (2004) defined a task as an unloading or loading operation for a collection of adjacent slots on one single container vessel. The objective was to minimize the weighted sum of the makespan of the container vessel and the total completion time of all quay cranes. Kim and Park (2004) proposed a branch and bound method and a heuristic algorithm called 'greedy randomized adaptive search procedure (GRASP)' for the solution of the quay crane scheduling problem. Nonetheless, Kim and Park (2004) did not discuss computational complexity of the studied problem to justify why the heuristic algorithm was adopted.

This paper focuses on the Quay Crane Scheduling with Non-Interference constraints Problem (QCSNIP) for any one single container vessel. This work was stimulated from Kim and Park (2004). Section 2 provides a more concise mathematical model than Kim and Park (2004) for the QCSNIP. Moreover, Kim and Park (2004) did not discuss computational complexity of the QCSNIP, but this paper discusses it and proves that the QCSNIP is NP-complete in Section 3. Because there exists no polynomial time algorithm for the exact solution of the QCSNIP, Section 4 proposes a genetic algorithm rather than GRASP of Kim and Park (2004) to obtain its near optimal solutions. The results of computational experiments in Section 5 show that the proposed genetic algorithm is effective and efficient in solving the QCSNIP.

2. Model formulation

This section proposes a mixed integer programming model for the QCSNIP. According to configuration of container vessels, one single container vessel is divided into holds. Fig. 1 illustrates the QCSNIP and shows that both quay cranes and holds are arranged in an increasing order from the front to the tail of the container vessel. The following assumptions are imposed in formulating the QCSNIP:

1. Quay cranes are on the same track and thus cannot cross over each other.
2. Only one quay crane can work on a hold at a time until it completes the hold.
3. Compared with processing time of a hold by a quay crane, travel time of a quay crane between two holds is small and hence it is ignored.

In order to formulate the QCSNIP, the following parameters and decision variables are introduced:

Parameters

K	the number of quay cranes;
H	the number of holds;
p_h	the processing time of hold h by a quay crane $(1 \leqslant h \leqslant H)$;
M	a sufficiently large positive constant number;

Decision variables

$X_{h,k}$	1, if hold h is handled by quay crane k; 0, otherwise $(1 \leqslant h \leqslant H, 1 \leqslant k \leqslant K)$;
$Y_{h,h'}$	1, if hold h finishes no later than hold h' starts; 0, otherwise $(1 \leqslant h,h' \leqslant H)$;
C_h	the completion time of hold h $(1 \leqslant h \leqslant H)$.

The QCSNIP can be formulated as follows:
Minimize:

$$\max_h C_h \tag{1}$$

Subject to:

$$C_h - p_h \geqslant 0 \quad \forall 1 \leqslant h \leqslant H \tag{2}$$

$$\sum_{k=1}^{K} X_{h,k} = 1 \quad \forall 1 \leqslant h \leqslant H \tag{3}$$

$$C_h - (C_{h'} - p_{h'}) + Y_{h,h'}M > 0 \quad \forall 1 \leqslant h, h' \leqslant H \tag{4}$$

$$C_h - (C_{h'} - p_{h'}) - (1 - Y_{h,h'})M \leqslant 0 \quad \forall 1 \leqslant h, h' \leqslant H \tag{5}$$

$$M(Y_{h,h'} + Y_{h',h}) \geqslant \sum_{k=1}^{K} kX_{h,k} - \sum_{l=1}^{K} lX_{h',l} + 1 \quad \forall 1 \leqslant h < h' \leqslant H \tag{6}$$

$$X_{h,k}, Y_{h,h'} = 0 \text{ or } 1 \quad \forall 1 \leqslant h, h' \leqslant H, \ \forall 1 \leqslant k \leqslant K \tag{7}$$

The objective function (1) minimizes the makespan of handling one single container vessel, which is the latest completion time among all holds. Constraints (2) define the property of the decision variable C_h. Constraints (3) ensure that every hold must be performed only by one quay crane. Constraints (4) and (5) define the properties of decision variables $Y_{h,h'}$: Constraints (4) indicate that $Y_{h,h'} = 1$ if $C_h \leqslant C_{h'} - p_{h'}$, which means $Y_{h,h'} = 1$ when hold h finishes no later than hold h' starts; Constraints (5) indicate that $Y_{h,h'} = 0$ if $C_h > C_{h'} - p_{h'}$, which means $Y_{h,h'} = 0$ when hold h finishes after hold h' starts. Finally, the interference between quay cranes can be avoided by imposing Constraints (6). Suppose that holds h and h' are performed simultaneously and $h < h'$, then this means that $Y_{h,h'} + Y_{h',h} = 0$. Note that both quay cranes and holds are arranged in an increasing order from the front to the tail of the container vessel. Thus, if quay crane k handles hold h and quay crane l handles hold h', then $k + 1 \leqslant l$.

3. Proof of NP-completeness

This section discusses computational complexity of the QCSNIP to justify why heuristic algorithms are adopted. As well known, if a problem is proved to be NP-complete, then there exists no polynomial time algorithm for its exact solution. Hence, heuristic algorithms are needed to obtain near optimal solutions for the problem. In this section, the proposed QCSNIP is proved to be NP-complete.

With respect to computational complexity, the decision version of a problem is as hard as the corresponding optimization version; the decision version of a problem has a natural and formal counterpart, which is a suitable object to be studied in a mathematically precise theory of computation. Consequently the theory of NP-completeness is designed to be applied only to the decision version (Garey and Johnson, 1979). The optimization version of the QCSNIP is presented in Section 2, and the decision version is defined as follows:

Parameter:

Z^+ the set of positive integer.

Instance: There are H holds and K quay cranes. The processing time of hold h by a quay crane is $p_h \in Z^+$ $(1 \leqslant h \leqslant H)$. There is a given number $C \in Z^+$.

Question: Is there a quay crane schedule for these K quay cranes handling these H holds such that no interference between quay cranes exists and the makespan of the quay crane schedule $\leqslant C$?

The decision version of the QCSNIP is proved to be NP-complete as the following four steps:

Theorem. *QCSNIP is NP-complete.*

Proof

Step 1: Showing that the QCSNIP is in NP.

If a quay crane schedule for the QCSNIP is given, its feasibility can be checked in polynomial time. Checking whether the quay crane schedule satisfies the non-interference constraints can be done in $O(H^2)$ time. Checking whether the makespan of the quay crane schedule $\leqslant C$ can be done in $O(H)$ time. Therefore, the QCSNIP is in NP.

Step 2: Selecting a known NP-complete problem.

PARTITION is a known NP-complete problem (Garey and Johnson, 1979). The decision version of the PARTITION is defined as follows:

Instance: There are H elements in a finite set $S = \{s_1, s_2, \ldots, s_H\}$. For each element $s_h \in S$, $s_h \in Z^+$ and the sum of all elements $\sum_{s_h \in S} s_h = D$.

Question: Can the set S be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_h \in S_1} s_h = \sum_{s_h \in S_2} s_h = D/2$?

A numerical example of the PARTITION is provided as follows. There is a finite set $S = \{95, 71, 136, 114, 192, 75, 123\}$ and the sum of all elements $\sum_{s_h \in S} s_h = D = 806$. The answer to Question is Yes because the set S can be partitioned into two disjoint subsets $S_1 = \{95, 123, 71, 114\}$ and $S_2 = \{75, 136, 192\}$ such that $\sum_{s_h \in S_1} s_h = \sum_{s_h \in S_2} s_h = D/2 = 403$.

Step 3: Constructing a transformation from the PARTITION to the QCSNIP.

The PARTITION is transformed to the QCSNIP as follows. A QCSNIP instance corresponding to an arbitrary PARTITION instance has K quay cranes and $H + K$ holds; the given number C is set as D; the following Eqs. (8)–(10) indicate the processing time of each hold which means the processing time of Hold 1 and Hold $H + 2$ is set as $D/2$, the processing time of Hold 2 to Hold $H + 1$ is set as s_1 to s_H, respectively, and the processing time of Hold $H + 3$ to Hold $H + K$ is set as D. Fig. 2 illustrates this transformation. It shows K quay cranes, $H + K$ holds and the processing time of each hold.

$$p_1 = p_{H+2} = D/2 \tag{8}$$

$$p_{h+1} = s_h \quad \forall 1 \leqslant h \leqslant H \tag{9}$$

$$p_h = D \quad \forall H + 3 \leqslant h \leqslant H + K \tag{10}$$

Then, it must be proved that the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_h \in S_1} s_h = \sum_{s_h \in S_2} s_h = D/2$ if and only if all the $H + K$ holds can be completed by K quay cranes in D time without interference between quay cranes.

First, suppose that the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_h \in S_1} s_h = \sum_{s_h \in S_2} s_h = D/2$. Then K quay cranes can be scheduled without interference as follows: Quay Crane 1 handles all the Holds $h + 1$, where $s_h \in S_1$ and then Hold 1; Quay Crane 2 handles Hold $H + 2$, and then all the Holds $h + 1$, where $s_h \in S_2$; Quay Cranes 3 to Quay Crane K handle Hold $H + 3$ to Hold $H + K$, respectively. Obviously, there is no interference in this schedule and the latest completion time among all holds is D. Hence, if the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_h \in S_1} s_h = \sum_{s_h \in S_2} s_h = D/2$, all the $H + K$ holds can be completed by K quay cranes in D time without interference between quay cranes.

Conversely, suppose all the $H + K$ holds can be completed by K quay cranes in D time without interference between quay cranes, then all the K quay cranes are fully utilized as the sum of the processing time of all the holds is KD. Thus, the completion time of each quay crane must be D. Furthermore, there is no interference in the above mentioned quay crane schedule. According to it, the sum of the processing time of all the holds except Hold 1 handled by Quay Crane 1 must be $D/2$ and the sum of the processing time of all the holds except Hold $H + 2$ handled by Quay Crane 2 must be $D/2$ as well, which means that the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_h \in S_1} s_h = \sum_{s_h \in S_2} s_h = D/2$. Hence, if all the $H + K$ holds can be completed by K quay cranes in D time without interference between quay cranes, the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_h \in S_1} s_h = \sum_{s_h \in S_2} s_h = D/2$.

Step 4: Proving that the above mentioned transformation is a polynomial transformation.

The above mentioned transformation can be done in $O(H + K)$ time.

Therefore, PARTITION $\propto$ QCSNIP, and the theorem is proved. □

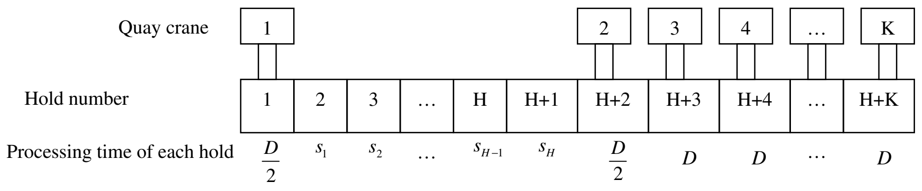

Fig. 2. The illustration of the transformation from the PARTITION to the QCSNIP.

4. A genetic algorithm

As proved in the previous section, QCSNIP is NP-complete, and thus there exists no polynomial time algorithm for the exact solution of QCSNIP. This paper employs a genetic algorithm (GA) to obtain near optimal solutions. GA is a search algorithm based on the mechanics of natural selection and natural genetics. In

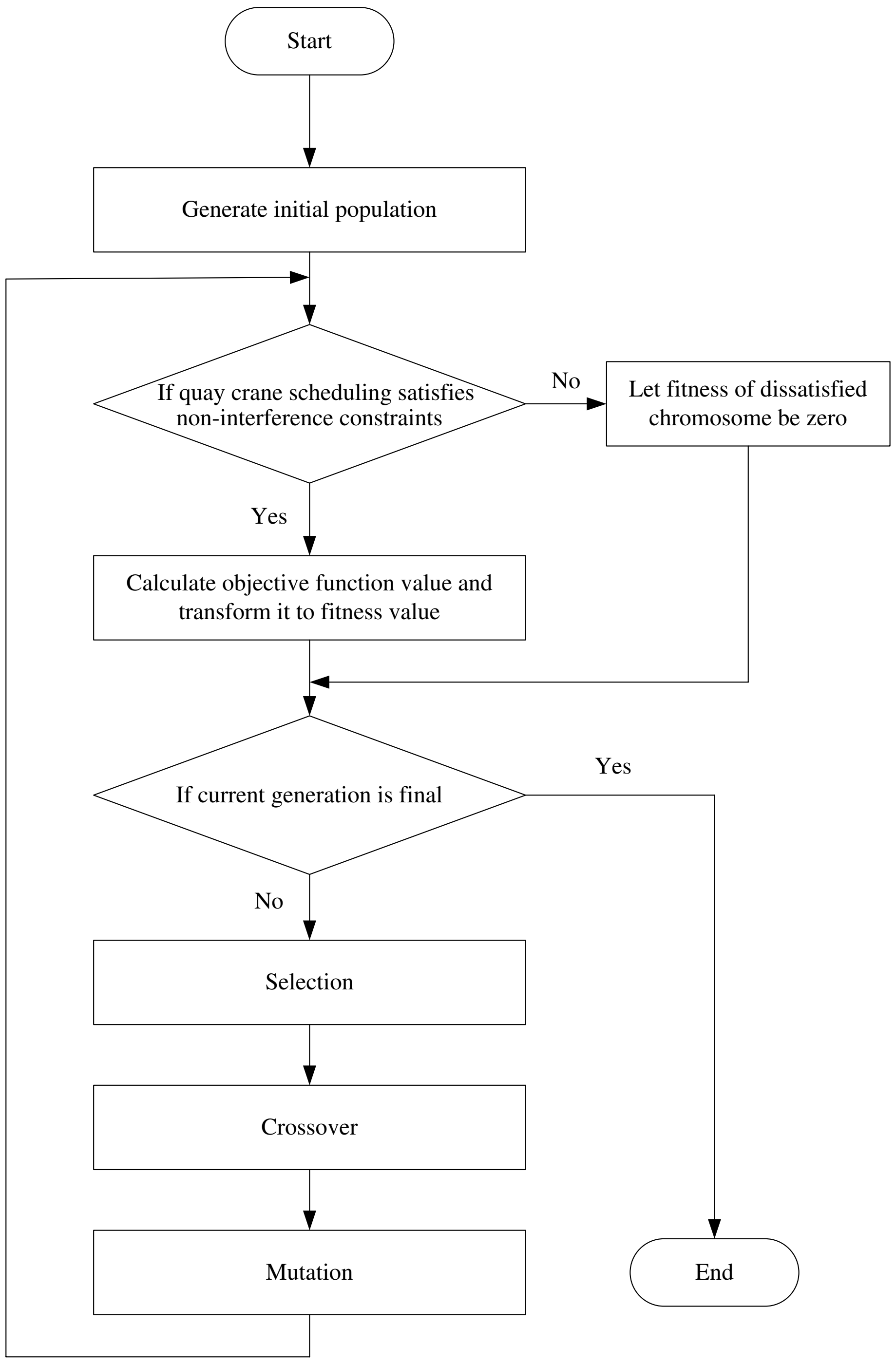
Start
Generate initial population
If quay crane scheduling satisfies non-interference constraints
No
Let fitness of dissatisfied chromosome be zero
Yes
Calculate objective function value and transform it to fitness value
If current generation is final
Yes
No
Selection
Crossover
Mutation
End

Fig. 3. The flowchart of the proposed GA.

general, there are three common genetic operators in a GA: selection, crossover, and mutation. The procedure of the proposed GA is illustrated in Fig. 3 and the details of the proposed GA are elaborated as follows.

4.1. Chromosome representation

Parameter:
ΔH the largest integer $\leqslant H/K$.

In this paper, the position of each quay crane is measured in terms of the hold number. For example, Quay Crane 1 is on Hold 1. The initial position of Quay Crane k is on Hold $1 + (k-1)\Delta H$ ($1 \leqslant k \leqslant K$).

A chromosome of the GA represents a sequence of holds. Fig. 4 provides a sample chromosome, in which a gene is a hold number. Based on the sequence of holds represented by the chromosome, a quay crane schedule can be constructed using the following procedure.

Step 1: Based on the current position of each quay crane, determine which quay cranes can handle the first unassigned hold in the chromosome without interference with the other quay cranes. If there is only one quay crane available, this hold is assigned to this quay crane. Then, this hold is deleted from the chromosome, and the position and the completion time of the assigned quay crane are updated. If there are two quay cranes available, go to Step 2.

Step 2: Compare the completion time of the two available quay cranes to finish their assigned holds, and assign this hold to the quay crane with earlier completion time. Then, this hold is deleted from the chromosome, and the position and the completion time of the assigned quay crane are updated. If their completion time is equal, go to Step 3.

Step 3: Compare the distance between this hold and these two available quay cranes, and assign this hold to the quay crane with the shorter distance. Then, this hold is deleted from the chromosome, and the position and the completion time of the assigned quay crane are updated. If their distance is equal, go to Step 4.

Step 4: Assign this hold to the quay crane with the smaller number. Then, this hold is deleted from the chromosome, and the position and the completion time of the assigned quay crane are updated.

Step 5: Steps 1–4 are repeated until all the holds in the chromosome are assigned.

Gene: hold number 1-9

Chromosome	7	2	3	8	6	4	1	9	5
Sequence	1	2	3	4	5	6	7	8	9

Fig. 4. An illustration of the chromosome representation.

Chromosome	7	12	5	3	6	1	10	8	11	2	4	9

	Position of Quay Crane	Completion Time of Quay Crane
Quay Crane 1	1	0
Quay Crane 2	5	0
Quay Crane 3	9	0

Chromosome	12	5	3	6	1	10	8	11	2	4	9

	Position of Quay Crane	Completion Time of Quay Crane
Quay Crane 1	1	0
Quay Crane 2	**7**	**114**
Quay Crane 3	9	0

Fig. 5. An illustration of constructing a quay crane schedule from a chromosome.

Fig. 5 shows a numerical example of the above mentioned quay crane scheduling procedure. There are three quay cranes and 12 holds. The initial position of Quay Crane 1, Quay Crane 2 and Quay Crane 3 are on Hold 1, Hold 5, and Hold 9, respectively. The initial completion time of three quay cranes is all 0. The first unassigned hold in the chromosome is Hold 7, of which the processing time is 114.

Step 1: Quay Crane 2 and Quay Crane 3 can handle Hold 7 without interference with the other quay cranes. Since there are two quay cranes available, go to Step 2.
Step 2: Since the completion time of Quay Crane 2 and Quay Crane 3 is both 0, go to Step 3.
Step 3: Since the distance between Hold 7 and Quay Crane 2, Quay Crane 3 is both 1 hold, go to Step 4.
Step 4: Assign Hold 7 to Quay Crane 2. Then, Hold 7 is deleted from the chromosome, the position of Quay Crane 2 is on Hold 7 and the completion time of Quay Crane 2 is 114.
Step 5: Hold 12 is the first unassigned hold in the chromosome. Repeat Steps 1–4 until all the holds in the chromosome are assigned.

4.2. Fitness evaluation and selection

Most of the quay crane schedules obtained from the above mentioned procedure do not violate the non-interference constraints. However, every quay crane schedule must be checked whether it satisfies the non-interference constraints as follows. According to a quay crane schedule constructed from a chromosome, Constraints (4) and (5), $Y_{h,h'}, \forall 1 \leqslant h, h' \leqslant H$ can be obtained and then the quay crane schedule can be checked whether it satisfies Constraints (6). If it satisfies Constraints (6), the fitness value of its corresponding chromosome is set to be the reciprocal of its objective function value, as shown in Eq. (11); otherwise, the fitness value of its corresponding chromosome is zero.

$$\text{Fitness} = 1/\max_h C_h \tag{11}$$

In this paper, a roulette wheel approach is adopted as the selection procedure. It belongs to the fitness-proportional selection and can select a new population with respect to the probability distribution based on fitness values (Gen and Cheng, 1996).

4.3. Crossover

Generally, the above mentioned chromosome representation will yield illegal offspring by one-point, two-point or multipoint crossover in the sense of that some holds may be missed while some holds may be duplicated in the offspring. Therefore, this paper adopts ‘order crossover’ (Gen and Cheng, 1996), in which repairing procedure is embedded to resolve the illegitimacy of offspring. ‘Order crossover’ works as follows:

Step 1: Select a substring from one parent randomly.
Step 2: Produce a proto-child by copying the substring into its corresponding positions.
Step 3: Delete the holds which are already in the substring from the second parent. The resulted sequence of holds contains the holds that the proto-child needs.
Step 4: Place the holds into the unfixed positions of the proto-child from left to right according to the order of the sequence to produce an offspring.

The procedure is illustrated in Fig. 6. It gives an example of making two offspring from the same parents.

4.4. Mutation

Mutation forces the GA to search new areas, and helps the GA avoid premature convergence and find the global optimal solution. Generally, in the mutation all individuals in the population are checked bit by bit and the bit values are randomly reversed according to a pre-specified rate. However, in this paper the mutation

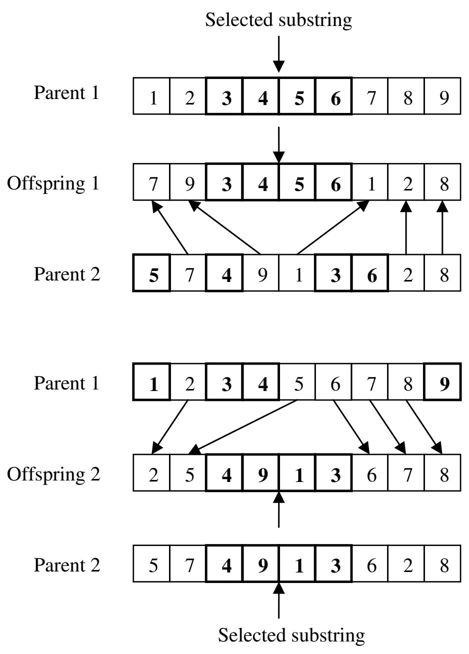

Fig. 6. An illustration of the order crossover.

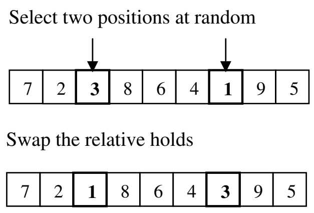

Fig. 7. An illustration of the mutation.

selects chromosomes randomly in terms of the probability of mutation and chooses two positions of the selected chromosome at random then swaps the holds on these positions as illustrated in Fig. 7.

5. Computational experiments

A series of computational experiments are conducted to examine the performance of the proposed model and GA. The GA is coded in C++ and executed in a Pentium IV 1.7 GHz PC with 256MB RAM. As a comparison, CPLEX (a commercial software for exactly solving integer programming) is employed to exactly solve random instances with small sizes and executed in the same PC.

5.1. Random instances with small sizes

Six random instances with small sizes are created, and the processing time of a hold is randomly generated from a uniform distribution of $U(30,180)$. Based on the preliminary tests, the population size, the probability of crossover, the probability of mutation, and the limit of generations of the GA are set as 150, 0.25, 0.1, and 100, respectively, in these computational experiments. As shown in Table 2, the computational time of CPLEX grows exponentially as the instance size increases since the QCSNIP is NP-complete. Moreover, it is obvious that the proposed GA can obtain the optimal solution in short time (for example, the computational time of these six instances is all around 5 s) when the instance size is small.

Table 2
Results of random instances with small sizes

Experiment No.	Size (holds × cranes)	CPLEX		GA	
		Value	CPU (s)	Value	CPU (s)
1	6 × 2	341	10.87	341	5.41
2	6 × 3	282	128.20	282	5.28
3	7 × 2	436	437.39	436	5.33
4	7 × 3	299	8014.58	299	5.53
5	8 × 2	448	11889.95	448	5.79
6	8 × 3	330	344951.97	330	5.48

5.2. Random instances with large sizes

There are forty random instances with large sizes generated. The processing time of a hold is randomly generated from a uniform distribution of $U(30,180)$. According to the preliminary tests, the population size, the probability of crossover, the probability of mutation, and the limit of generations of the GA are set as 300, 0.25, 0.2, and 1000, respectively, in these computational experiments.

In order to evaluate the performance of the proposed GA in solving the instance with large size, the lower bound corresponding to the instance can be calculated firstly by relaxing the non-interference constrains. The mathematical model of the relaxed problem is formulated as follows:

Decision variable:

c_k the completion time of quay crane k $(1 \leqslant k \leqslant K)$.

Minimize:

$$\max_k c_k \tag{12}$$

Subject to:

$$\sum_{k=1}^{K} X_{h,k} = 1 \quad \forall 1 \leqslant h \leqslant H \tag{13}$$

$$c_k \geqslant \sum_{h=1}^{H} X_{h,k} p_h \quad \forall 1 \leqslant k \leqslant K \tag{14}$$

The objective function (12) minimizes the makespan of handling one single container vessel without considering interference between quay cranes. Constraints (13) ensure that every hold must be performed only by one quay crane. Constraints (14) define the property of the decision variable c_k. Then, the mathematical model of the relaxed problem can be exactly solved by CPLEX. The objective function value of the optimal solution to the relaxed problem obtained from CPLEX is the lower bound to the original problem.

As observed in Table 3, the gaps between solutions obtained from the proposed GA and lower bounds are all small (for example, the maximum gap among the forty instances is merely 2.66%, the minimum gap is 0, and the average gap is 0.41%), and all the computational time of these forty instances is short (for example, the computational time of these forty instances is all around 120 s). Based on these 40 computational experiments, it is clear that near optimal solutions obtained from the proposed GA are of high quality. The performance of the proposed GA is thus satisfactory in solving large size instances.

The obtained lower bound may come from an infeasible solution to the original problem, because it is the objective function value of the optimal solution to the relaxed problem. In Table 3, the gaps of twelve instances are zero, which means the lower bound is, by chance, equal to the objective function value of the optimal solution to the original problem in these twelve instances. Therefore, the proposed GA achieves the optimal solution to the original problem for these 12 instances.

Table 3
Results of random instances with large sizes

Experiment No.	Size (holds × cranes)	Lower bound	GA		Gap[a] (%)
			Value	CPU (s)	
1	16 × 3	650	653	105.91	0.46
2	16 × 4	488	501	110.29	2.66
3	17 × 3	617	621	122.79	0.65
4	17 × 4	463	469	106.60	1.30
5	18 × 3	599	602	109.72	0.50
6	18 × 4	450	454	107.63	0.89
7	19 × 3	740	741	108.35	0.14
8	19 × 4	555	559	109.79	0.72
9	20 × 3	672	674	109.98	0.30
10	20 × 4	504	511	111.73	1.39
11	21 × 3	793	793	107.03	0
12	21 × 4	595	597	107.83	0.34
13	22 × 3	796	796	108.33	0
14	22 × 4	597	599	109.70	0.34
15	23 × 3	794	794	112.56	0
16	23 × 4	595	603	111.68	1.34
17	24 × 3	786	786	117.46	0
18	24 × 4	590	591	111.31	0.17
19	25 × 3	942	943	109.91	0.11
20	25 × 4	707	712	112.45	0.71
21	26 × 3	819	820	111.03	0.12
22	26 × 4	615	617	115.07	0.33
23	27 × 3	985	986	115.78	0.10
24	27 × 4	739	742	123.96	0.41
25	28 × 3	908	908	125.01	0
26	28 × 4	681	683	125.15	0.29
27	29 × 3	1065	1065	122.36	0
28	29 × 4	799	802	129.11	0.38
29	30 × 3	996	996	117.48	0
30	30 × 4	747	749	118.79	0.27
31	31 × 3	1141	1141	119.19	0
32	31 × 4	856	861	120.97	0.58
33	32 × 3	1041	1041	116.93	0
34	32 × 4	781	783	117.28	0.26
35	33 × 3	1213	1213	122.07	0
36	33 × 4	910	917	122.93	0.77
37	34 × 3	1009	1009	126.84	0
38	34 × 4	757	761	126.72	0.53
39	35 × 3	1288	1288	122.49	0
40	35 × 4	966	968	122.35	0.21

[a] Gap = (solution obtained from the proposed GA—lower bound) × 100/lower bound.

The lower bound is the objective function value of the optimal solution to the relaxed problem, which does not consider interference between quay cranes. The proposed GA obtains the near optimal solution to the original problem. As shown in Table 3, the larger gaps are observed for smaller container vessels with fewer holds handled by more quay cranes. The reason for it can be that interference between quay cranes more significantly affects scheduling more quay cranes for smaller container vessels with fewer holds.

Furthermore, in practical quay crane scheduling, the number of quay cranes ranges from 2 to 4, and the number of holds ranges from 10 to 25. The proposed GA can be considered as an appropriate approach to scheduling quay cranes in port container terminals.

According to the computational experiments with small and large sizes, the proposed GA is concluded to be effective and efficient in solving the proposed QCSNIP and can be capable in solving practical quay crane scheduling problem for port container terminals.

6. Conclusions

The contributions of this paper to the literature are that it has provided a mixed integer programming model for the proposed QCSNIP, proved that the QCSNIP is NP-complete and proposed a genetic algorithm to obtain near optimal solutions for the QCSNIP. In addition, computational experiments have been performed to examine the proposed model and GA. The results showed that the proposed GA has been effective and efficient in solving the QCSNIP.

In this paper, factors such as the travel time of a quay crane between two holds and the handling priority of every hold were not taken into account. The incorporation of these factors into the QCSNIP can be a topic for future research.

Acknowledgements

The authors would like to express their heartfelt appreciation to the two anonymous referees and Assistant Professor Qiang Meng from the Department of Civil Engineering of National University of Singapore for the valuable comments that improve the content and the presentation of this paper. The helpful suggestions of Professor Wayne K. Talley to improve this paper are also gratefully acknowledged.

References

Daganzo, C.F., 1989. The crane scheduling problem. Transportation Research Part B 23, 159–175.

Garey, M.R., Johnson, D.S., 1979. Computers and intractability: a Guide to the Theory of NP-completeness. W.H. Freeman, San Francisco.

Gen, M., Cheng, R., 1996. Genetic Algorithms and Engineering Design. John Wiley, New York.

Kim, K.H., Kang, J.S., Ryu, K.R., 2004. A beam search algorithm for the loading sequencing of outbound containers in port container terminals. OR Spectrum 26, 93–116.

Kim, K.H., Park, Y.M., 2004. A crane scheduling method for port container terminals. European Journal of Operation Research 156, 752–768.

Lim, A., Rodrigues, B., Xiao, F., Zhu, Y., 2004. Crane scheduling with spatial constraints. Naval Research Logistics 51, 386–406.

Peterkofsky, R.I., Daganzo, C.F., 1990. A branch and bound solution method for the crane scheduling problem. Transportation Research Part B 24, 159–172.

Steenken, D., Voß, S., Stahlbock, R., 2004. Container terminal operation and operations research – a classification and literature review. OR Spectrum 26, 3–49.