BERTH ALLOCATION AND QUAY CRANE SCHEDULING IN PORT CONTAINER TERMINALS

WANG HUIQIU

NATIONAL UNIVERSITY OF SINGAPORE

2007

BERTH ALLOCATION AND QUAY CRANE SCHEDULING

IN PORT CONTAINER TERMINALS

WANG HUIQIU

(M.Eng., Tsinghua University)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF CIVIL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

ACKNOWLEDGEMENT

My deepest appreciation goes to my supervisor Associate Professor Lee Der-Horng for his invaluable guidance, constructive suggestion and continuous support throughout the course of my Ph.D. study in National University of Singapore. My gratitude also goes to Assistant Professor Meng Qiang for his great encouragement and inspiration on both my academic research and personal life.

The author would also like to thank Prof. IMAI Akio and Prof. TEO Chung Piaw for their precious guidance and suggestions on his academic research work.

I would like to thank Mr. Foo Chee Kiong and all other technicians and administrative staffs for their friendship and kind assistance.

Particularly, thanks also are extended to my colleagues in the ITVS Lab, Cao Zhi, Dong Meng, Cao Jinxin, Bian Wen, Huang Yikai, Alvina Kek Geok, Khoo Hooi Ling, Fung Chau Ha Jenice, Huang Yongxi, Deng Weijia, Cheng Shihua and Fery Pierre Geoffroy Julien, for their encouragement and help in the past three years. I also wish to record my gratitude to all others who have assisted me in one way or other.

Special thanks go to National University of Singapore for providing me with a research scholarship covering the entire period of my graduate studies.

Finally, the most sincere gratitude is due to my parents and wife for their endless love and support through all the time.

TABLE OF CONTENTS

SUMMARY

Rapidly increasing competition between port container terminals, especially between geographically close ones, has forced them to improve their efficiency. Since berths and quay cranes are the interface between sea side and land side in any port container terminal, their operations significantly influence the efficiency of port container terminals. This research focused on optimizing berth allocation and quay crane scheduling in port container terminals to enhance their efficiency. In this research, analytical models, approximation algorithms, genetic algorithms were proposed to ameliorate berth and quay crane operations.

A quay crane scheduling with non-crossing constraints problem was first investigated in this thesis. A mixed integer programming model was provided for this problem that is NP-complete in nature. Therefore, there exists no polynomial time algorithm for its exact solution unless P=NP. An approximation algorithm and a genetic algorithm were then developed to obtain its near optimal solutions. In addition, worst-case analysis for the approximation algorithm was performed and computational experiments were conducted to examine the proposed model and solution algorithms. The results showed that both the approximation algorithm and the genetic algorithm were effective and efficient in solving the problem.

A quay crane scheduling with safety distance and non-crossing constraints problem was then addressed. A mixed integer programming model was built for this problem which

was proved to be NP-complete. For obtaining its near optimal solutions, an approximation algorithm based on a dynamic programming and a genetic algorithm were proposed. Worst-case analysis for the approximation algorithm and computational experiments for examining the proposed model and solution algorithms were performed. The results showed that both the approximation algorithm and the genetic algorithm were effective and efficient in solving the problem.

In the third part of this thesis, a quay crane scheduling with handling priority and non-crossing constraints problem was studied. This problem was formulated as a mixed integer programming model and was proved to be NP-complete. An approximation algorithm was proposed to obtain its near optimal solution. Moreover, worst-case analysis for the approximation algorithm was performed and computational experiments were conducted. The results showed that the approximation algorithm was effective and efficient in solving the problem.

Finally, an integrated discrete berth allocation and quay crane scheduling problem was discussed. A mixed integer programming model including two parts was proposed for this problem which was proved to be NP-complete. A genetic algorithm containing an approximation algorithm for quay crane scheduling was designed for obtaining its near optimal solutions. The computational results showed that the proposed genetic algorithm was effective and efficient in solving the problem.

This research considered quay crane scheduling with non-crossing, safety distance, and handling priority, which may contribute to the theory of parallel machine scheduling. The proposed scheduling methods in this research may improve the efficiency of berth and quay crane operations in port container terminals. Furthermore, results of this research should enhance our understanding of combined optimization of berth allocation and quay crane scheduling. This knowledge may further increase the overall efficiency of port operations when comparing to optimizing berth allocation or quay crane scheduling individually.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

The proportion of cargo transported by containers has steadily increased due to the advantages of container transport such as less product packaging, less damaging, higher productivity, and easier transshipment between different modes (Vis and de Koster, 2003). In container transport, port container terminals play a very important role as they are the interface between sea container transport and land container transport. However, the competition between port container terminals has considerably increased, caused by huge growth rates on major maritime container routes (Günther and Kim, 2006). To succeed in the fierce competition, a crucial competitive advantage is the high efficiency of operations in port container terminals (Steenken et al., 2004). Therefore, many studies on port operations have been conducted to enhance the efficiency of port container terminals. The rest of the chapter provides an overview of port operations, literature review on berth allocation, literature review on quay crane scheduling, the research objectives, and ends with the organization of the thesis.

1.1 OVERVIEW OF PORT OPERATIONS

When a container ship is moored in its allocated berth, the assigned quay cranes start to unload containers from the container ship. The typical operation flow of unloading a container is described as follows. A quay crane unloads a container from the container ship to a container truck. The container truck then transports the container to the assigned location in the yard. A yard crane finally loads the container from the container truck to the designated slot. The process of loading a container to a container ship is reversed.

Thus, port operations generally consist of berth allocation, quay crane scheduling, ship stowage planning, container truck scheduling, yard storage planning, and yard crane scheduling.

Berth allocation and quay crane scheduling significantly influence the efficiency of port operations since berths and quay cranes are the interface between sea side and land side in any port container terminal. Singapore Container Terminal is one of the busiest container terminals in terms of container throughput in the world. However, in order to succeed in the intense competition, Port of Singapore Authority attempts to optimize their berth and quay crane operations. Therefore, the emphasis of this thesis is on berth allocation and quay crane scheduling problems to enhance the efficiency of port container terminals.

1.1.1 Overview of Berth Allocation

Berth allocation is to determine the berthing time and position of every container ship considering some factors including the length and draft of each container ship, the arrival time of each container ship, the number of containers to be unloaded and loaded, and the storage location of outbound containers to be loaded onto the corresponding container ship. As shown in Figure 1.1, the entire wharf in a port container terminal is partitioned into several berths and a container ship is moored within the allocated berth in practice. This leads to the discrete berth allocation problem (Imai et al., 2005). However, sometimes container ships are allowed to be moored across the berth boundary to enhance the efficiency of berth usage which leads to the continuous berth allocation problem (Imai et al., 2005).

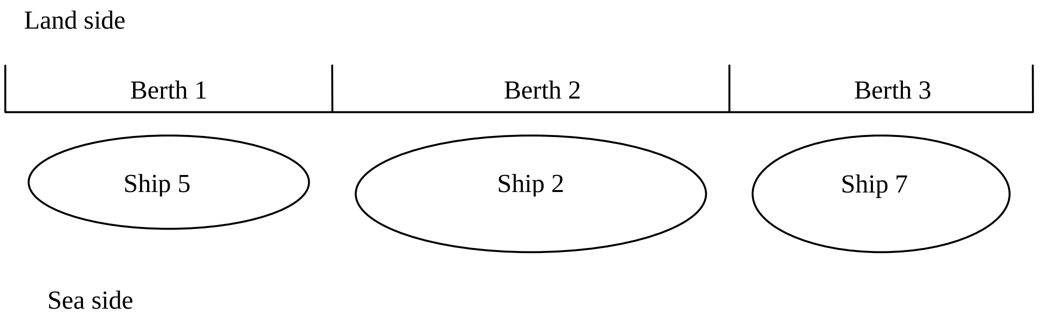

Figure 1.1 An Illustration of Berth Allocation

1.1.2 Overview of Quay Crane Scheduling

As illustrated in Figure 1.2, a container ship is typically divided longitudinally into ship bays that consist of holds and decks. Holds are about eight containers deep and containers can also be stacked (about six high) on decks. Quay cranes are operated on the same tracks and thus cannot cross over each other. Furthermore, only one quay crane can work on a ship bay at any time and a quay crane usually moves to the next assigned ship bay until it completes the current one. The average processing time of a ship bay is about three hours and the travel time of a quay crane between two ship bays is about one minute.

In practice, there are requirements of maintaining safety distance between any two quay cranes in operation. For example, as a rule two adjacent operating quay cranes must be apart from each other by one ship bay. Moreover, different ship bay has different handling priority. For instance, according to a survey of port container terminals in China, in practice some port operators prefer to assign a high handling priority to a ship bay with long processing time.

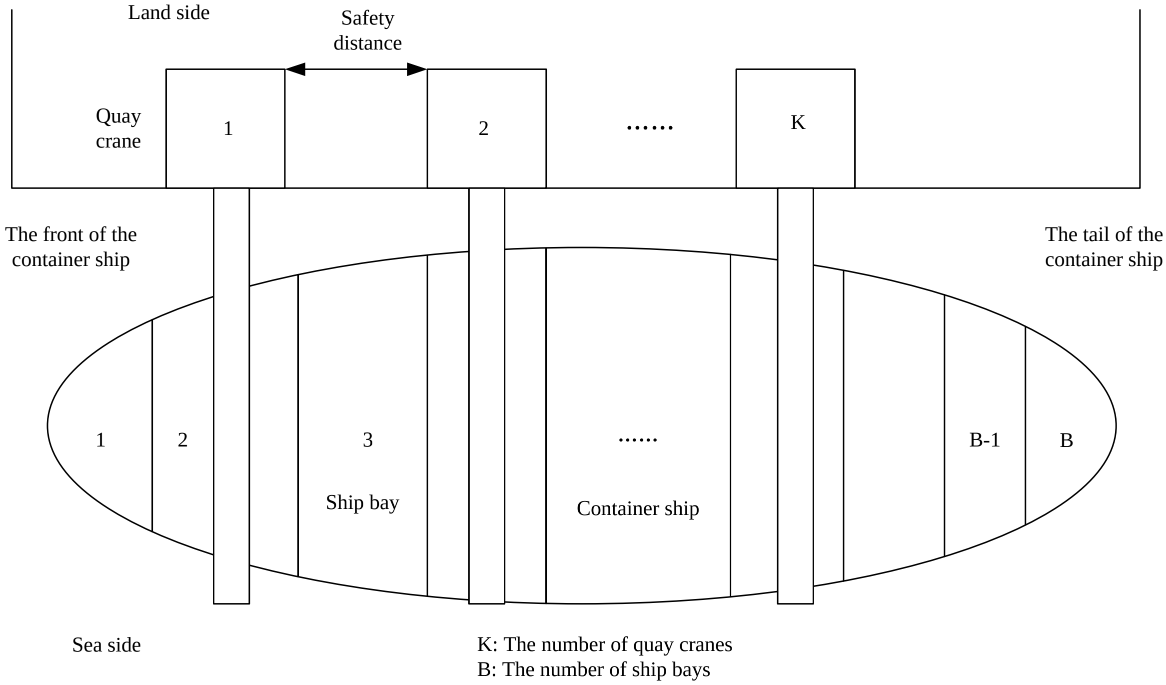

Figure 1.2 An Illustration of Quay Crane Scheduling

Quay crane scheduling is to determine a handling sequence of ship bays for quay cranes assigned to a container ship in fulfilling pre-specified objectives and satisfying various constraints such as non-crossing, safety distance, and handling priority of every ship bay. Table 1.1 illustrates a feasible quay crane schedule for the instance in which a container ship with ten ship bays is handled by two quay cranes and the safety distance between the two quay cranes is one ship bay. The handling priority of Ship Bay 3 is higher than Ship Bay 2, and the handling priority of Ship Bay 4 is higher than Ship Bay 10. The handling sequence of ship bays for every quay crane, the processing time of each ship bay, and the time schedule for handling every ship bay are shown in Table 1.1.

Table 1.1 An Illustration of a Quay Crane Schedule

Quay Crane 1				Quay Crane 2			
Operation Sequence	Ship Bay Number	Processing Time of a Ship Bay (min)	Completion Time of the Quay Crane (min)	Operation Sequence	Ship Bay Number	Processing Time of a Ship Bay (min)	Completion Time of the Quay Crane (min)
1	1	98	98	**1**	3	81	81
2	2	119	217	**2**	5	103	184
3	4	76	293	**3**	8	214	398
4	6	137	430	**4**	10	93	491
5	7	65	495				
6	9	81	576				

1.2 LITERATURE REVIEW ON BERTH ALLOCATION

1.2.1 Discrete Berth Allocation Problem

A discrete berth allocation problem was addressed by Lai and Shih (1992). They employed a discrete event simulation model to analyze four berth allocation policies based on the data from a major port container terminal in Hong Kong. The three policies proposed by Lai and Shih (1992) were dominated by first-come-first-served rule. The simulation results showed that compared with the current berth allocation policy, the three proposed policies improved the operational efficiency. In fact, the first-come-first-served rule is questionable as it cannot maximize the efficiency of port container terminals. It is possible that the efficiency of port container terminals may be further enhanced if the first-come-first-served rule is not considered.

Imai et al. (1997) assumed that the berth allocation was made for container ships already arrived before a given planning horizon that was a static berth allocation problem. Imai et al. (1997) did not take into account the first-come-first-served rule which may cause the dissatisfaction of container ships with the order of service. Thus, their model had two objectives: the minimization of total service time (waiting time plus handling time) of every container ship, and the minimization of the dissatisfaction of container ships with the order of service. A weighting method was developed by Imai et al. (1997) to identify a set of non-inferior solutions for the problem. Nevertheless, the assumption of static berth allocation may not always hold in practice. It is possible that some container ships may arrive at a port container terminal after the beginning time of the planning horizon.

Imai et al. (2001) assumed that some container ships arrived at the port container terminal after the beginning time of the planning horizon that was a dynamic berth allocation problem. Their objective was to minimize total service time of every container ship. A sub-gradient optimization procedure based on the Lagrangian relaxation of the original problem was proposed by Imai et al. (2001) to obtain near optimal solutions. Nishimura et al. (2001) extended the dynamic berth allocation problem proposed by Imai et al. (2001) with considerations of water depth, berth length, container ship draft, and container ship length. A genetic algorithm was developed by Nishimura et al. (2001) to obtain near optimal solutions. Computational experiments showed that compared with the sub-gradient optimization procedure based on Lagrangian relaxation (Imai et al., 2001), the genetic algorithm was effective. Finally, Nishimura et al. (2001) used actual data from Kobe port during one month of February 1996 to test the proposed genetic algorithm and the results showed that the genetic algorithm seemed adaptable to real world applications. Imai et al. (2003) augmented the dynamic berth allocation problem proposed in 2001 by considering service priority of every container ship. Imai et al. (2003) first attempted to adopt a sub-gradient optimization procedure based on Lagrangian relaxation to solve the problem, but enormous computational effort was expected. Then, they employed a genetic algorithm to obtain near optimal solutions. Computational experiments were conducted by Imai et al. (2003) to show the importance of considering service priority of every container ship. In reality, the handling time of a container ship at a berth is related to its quay crane schedule, but the above mentioned research work did not take into account the relationship between berth allocation and quay crane scheduling. Hence, the incorporation of quay crane scheduling into berth allocation should be further investigated.

1.2.2 Continuous Berth Allocation Problem

Another continuous berth allocation problem was discussed by Lim (1998). His objective was to find the exact location of each container ship in the berth and to minimize the maximum amount of space used in the berth at any time. Lim (1998) showed that the problem is NP-complete, transformed the problem to a restricted form of the two-dimensional packing problem, and used a graph theoretical representation to capture the problem. A heuristic was proposed by Lim (1998) for the problem and experimental results showed that the heuristic performed well on historical test data from the Port of Singapore Authority for six months. However, Lim (1998) implied that container ships could be berthed immediately when they arrived at a port container terminal, but this may not always be possible. When the port container terminal is busy, it is likely that some container ships may have to wait for available berths.

Li et al. (1998) studied a static berth allocation problem which was to minimize the makespan of the schedule (the latest completion time among all container ships). Li et al. (1998) assumed that a larger container ship required a longer processing time, preemption of container ships was not allowed, and the processing time of a container ship was independent of the other container ships processed at the same time. Li et al. (1998) considered three cases: the first case assumed that the physical position of any container ship could not be changed during the processing of the container ship; the second case assumed that the physical position of the container ships could be changed at any time; and the third case assumed that the berth was only partially available for an initial time period for the non-fixed position case. Since these three cases were all strongly NP-hard, they developed generalized First-Fit-Decreasing heuristics to approximately solve them and performed worst-case analysis for the proposed algorithms. Computational experiments showed that the heuristics developed by Li et al. (1998) were effective in producing a near optimal solution. Guan et al. (2002) addressed a similar static berth allocation problem to Li et al. (1998), but with a different objective of minimizing the total weighted completion time of container ships. Guan et al. (2002) showed that the proposed problem was NP-hard, designed a heuristic for the problem, and performed worst-case analysis for the heuristic. Moreover, Guan and Cheung (2004) extended the static berth allocation problem proposed by Guan et al. (2002) to a dynamic berth allocation problem. Their objective was to minimize total weighted service time of every container ship. Guan and Cheung (2004) developed a tree search procedure for obtaining the optimal solution and proposed a composite heuristic for solving large size problems.

Li et al. (1998), Guan et al. (2002), and Guan and Cheung (2004) all assumed that a larger container ship required a longer processing time, but this assumption may not always hold in practice. It is probable that some large container ships may not have much unloading and loading work in a port container terminal.

Park and Kim (2002) investigated a dynamic berth allocation problem which was to minimize the penalty costs resulting from delays in the departures of container ships and the additional handling costs resulting from non-optimal locations of container ships in a wharf. Park and Kim (2002) developed a sub-gradient optimization technique. Furthermore, Kim and Moon (2003) proposed a simulated annealing algorithm for the same problem as Park and Kim (2002). However, it may be difficult to define the best berthing location of each container ship, the additional handling cost resulting from non-optimal location of each container ship, and the penalty cost resulting from delay in the departure of each container ship in practice. Hence, the aforementioned research may not be applied in port container terminals easily.

Imai et al. (2005) addressed a dynamic berth allocation problem which assumed that the handling time of a container ship depended on its berthing location. Their objective was to minimize the total service time of all container ships. Imai et al. (2005) developed a heuristic algorithm with two stages for the proposed problem. Nonetheless, minimization of the total service time may be for container ships rather than for port container terminals. Thus, if the emphasis is on the efficiency of port container terminals, minimization of the makespan may be better than minimization of the total service time.

Moorthy and Teo (2006) studied a dynamic berth allocation problem which considered uncertainties of the arrival time and processing time of container ships. Their objective was to minimize the expected delays and the connectivity cost. Moorthy and Teo (2006) proposed a sequence pair based simulated annealing algorithm to solve the problem. However, it may be difficult to define the connectivity cost in reality.

1.3 LITERATURE REVIEW ON QUAY CRANE SCHEDULING

A static and a dynamic quay crane scheduling problem for multiple container ships were studied by Daganzo (1989). The objective was to serve all these container ships, while minimizing their aggregate cost of delay. Exact and approximate solution methods 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 non-crossing constraints between quay cranes, which means the quay cranes may unrealistically cross over each other.

Liu et al. (2006) augmented the dynamic quay crane scheduling problem proposed by Daganzo (1989) by taking into account the non-crossing and safety distance constraints. Their objective was to minimize the maximum relative tardiness of multiple container ships. Liu et al. (2006) applied a heuristic decomposition approach to solve the problem. However, they did not consider the handling priority of every ship bay, which means the quay crane schedule obtained from their method may not always fulfill the operational requirements.

Lim et al. (2004a) discussed a quay crane scheduling problem for single container ship. Lim et al. (2004a) assumed that containers from a given area on a container ship were a job and there was a profit value when a job was assigned to a quay crane. Their 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 by Lim et al. (2004a) for solving the problem. Nonetheless, it may be difficult to define a profit value associated with a crane-to-job assignment in practice, and hence this research may not be applied in port container terminals easily.

Kim and Park (2004) addressed a quay crane scheduling problem for single container ship. Kim and Park (2004) defined a task as an unloading or loading operation for a collection of adjacent slots on single container ship. Their objective was to minimize the weighted sum of the makespan of handling the container ship (that was the latest completion time among all tasks) 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' for the solution. Moreover, Moccia et al. (2006) reformulated the same problem as Kim and Park (2004) and developed a branch-and-cut algorithm to solve small size instances exactly. Nonetheless, both papers did not discuss computational complexity of the studied problem to justify why heuristic algorithms were necessary.

Ng and Mak (2006) discussed a quay crane scheduling problem for single container ship. Their objective was to minimize the makespan of handling the container ship (that was the latest completion time among all ship bays). A heuristic was proposed by Ng and Mak (2006) for solving this problem. Zhu and Lim (2006) provided a different mathematical model, a branch-and-bound algorithm, and a simulated annealing algorithm for the same problem as Ng and Mak (2006). Moreover, Lim et al. (2004b) devised a highly optimized backtracking scheme and a simulated annealing algorithm with a stochastic neighborhood, and Lim et al. (2004c) proposed a dynamic programming algorithm and approximation algorithms for solving the same problem as Zhu and Lim (2006). However, the aforementioned research work did not consider the safety distance constraints between quay cranes, which means the quay crane schedule obtained from their methods may not always be feasible in practice.

Park and Kim (2003) proposed an integer programming model for scheduling berth and quay cranes. A two-phase solution procedure was developed for solving the problem. In the first phase, the berthing position and time of each container ship as well as the number of quay cranes assigned to each container ship at each time segment were determined by using a sub-gradient optimization technique. The second phase determined which quay crane was assigned to which container ship at each time segment by using a dynamic programming technique. Park and Kim (2003) assumed that the handling time of a container ship was inversely proportional to the number of quay cranes assigned to the container ship, but this assumption may not be true. Due to the non-crossing and safety distance constraints between quay cranes, the relationship between the handling time of a container ship and the number of quay cranes assigned to the container ship may be nonlinear.

In sum, the three vital influential factors in practical quay crane scheduling, which are non-crossing, safety distance, and handling priority of each ship bay, were not investigated sufficiently in the existing studies on quay crane scheduling. In reality, the handling time of a container ship at a berth is related to its quay crane schedule. However, few studies on integrated berth allocation and quay crane scheduling were conducted.

1.4 RESEARCH OBJECTIVES

The main objectives of this thesis were to:

1. Formulate the quay crane scheduling with non-crossing constraints problem; discuss computational complexity of the proposed problem; propose an approximation algorithm for the problem and perform worst-case analysis for the proposed approximation algorithm; develop a genetic algorithm to obtain near optimal solutions for the problem; conduct computational experiments to examine the proposed mathematical model and solution methods.
2. Formulate the quay crane scheduling with safety distance and non-crossing constraints problem; discuss computational complexity of the proposed problem; propose an approximation algorithm for the problem and perform worst-case analysis for the proposed approximation algorithm; develop a genetic algorithm to obtain near optimal solutions for the problem; conduct computational experiments to examine the proposed mathematical model and solution methods.

3. Formulate the quay crane scheduling with handling priority and non-crossing constraints problem; discuss computational complexity of the proposed problem; propose an approximation algorithm for the problem and perform worst-case analysis for the proposed approximation algorithm; conduct computational experiments to examine the proposed mathematical model and solution method.
4. Formulate the integrated discrete berth allocation and quay crane scheduling problem; discuss computational complexity of the proposed problem; develop a genetic algorithm to obtain near optimal solutions for the problem; conduct computational experiments to examine the proposed mathematical model and solution method.

Although continuous berth allocation can enhance the efficiency of berth usage, the incorporation of quay crane scheduling into continuous berth allocation is beyond the scope of this thesis. This is due to the fact that most of port container terminals adopt discrete berth allocation for safety and convenience.

This thesis considers quay crane scheduling with non-crossing, safety distance, and handling priority, which may contribute to the theory of parallel machine scheduling. The proposed scheduling methods in this thesis may improve the efficiency of berth and quay crane operations in port container terminals. Furthermore, results of this thesis should enhance our understanding of combined optimization of berth allocation and quay crane scheduling. This knowledge may further increase the overall efficiency of port operations when comparing to optimizing berth allocation or quay crane scheduling individually.

1.5 ORGANIZATION OF THE THESIS

This thesis consists of six chapters.

Chapter 1 is the introductory chapter which provides an overview of port operations, literature review on berth allocation, literature review on quay crane scheduling, the research objectives, and ends with the organization of the thesis.

Chapter 2 provides a mixed integer programming model for the quay crane scheduling with non-crossing constraints problem that is NP-complete in nature. Therefore, there exists no polynomial time algorithm for its exact solution unless P=NP. An approximation algorithm and a genetic algorithm are then developed to obtain its near optimal solutions. In addition, worst-case analysis for the approximation algorithm is performed and computational experiments are conducted to examine the proposed model and solution algorithms.

Chapter 3 presents a mixed integer programming model for the quay crane scheduling with safety distance and non-crossing constraints problem which is proved to be NP-complete. For obtaining its near optimal solutions, an approximation algorithm based on a dynamic programming and a genetic algorithm are proposed. Worst-case analysis for the approximation algorithm and computational experiments for examining the proposed model and solution algorithms are performed.

Chapter 4 provides a mixed integer programming model for the quay crane scheduling with handling priority and non-crossing constraints problem that is proved to be NP-complete. An approximation algorithm is proposed to obtain its near optimal solution. Moreover, worst-case analysis for the approximation algorithm is performed and computational experiments are conducted.

Chapter 5 presents a mixed integer programming model including two parts for the integrated discrete berth allocation and quay crane scheduling problem which is proved to be NP-complete. A genetic algorithm containing an approximation algorithm for quay crane scheduling is designed for obtaining its near optimal solutions and computational experiments for examining the genetic algorithm are performed.

Chapter 6 provides a conclusion of this thesis. The recommendations for future research and the contributions of this thesis are also presented.

CHAPTER 2 QUAY CRANE SCHEDULING WITH NON-CROSSING CONSTRAINTS

As shown in previous discussions, quay cranes are operated on the same tracks and thus they cannot cross over each other. To consider this vital influential factor, this chapter addresses the Quay Crane Scheduling with Non-Crossing constraints Problem (QCSNCP).

2.1 MODEL FORMULATION

This chapter proposes a mixed integer programming model for QCSNCP. According to the configuration of container ships, one single container ship is divided into ship bays. Figure 1.2 shows that both quay cranes and ship bays are arranged in an increasing order from the front to the tail of the container ship. The following assumptions are imposed in formulating the QCSNCP:

1. Quay cranes are operated on the same tracks and thus cannot cross over each other.
2. Only one quay crane can work on a ship bay at a time until it completes the ship bay.
3. Compared with the processing time of a ship bay by a quay crane, the travel time of a quay crane between two ship bays is small and hence it is not considered.

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

Parameters:

K the number of quay cranes;

B the number of ship bays;

p_b the processing time of ship bay b by a quay crane ($1 \le b \le B$);

M a sufficiently large positive number (constant);

Decision variables:

$X_{b,k}$ 1, if ship bay b is handled by quay crane k; 0, otherwise ($1 \le b \le B,\ 1 \le k \le K$);

$Y_{b,b'}$ 1, if ship bay b finishes no later than ship bay b' starts; 0, otherwise ($1 \le b,\ b' \le B,\ b \ne b'$);

C_b the completion time of ship bay b ($1 \le b \le B$).

The QCSNCP can be formulated as follows:

Minimize:

$$\max_b \ C_b \tag{2.1}$$

Subject to:

$$C_b - p_b \ge 0 \quad \forall 1 \le b \le B \tag{2.2}$$

$$\sum_{k=1}^{K} X_{b,k} = 1 \quad \forall 1 \le b \le B \tag{2.3}$$

$$C_b - (C_{b'} - p_{b'}) + Y_{b,b'} M > 0 \quad \forall 1 \le b,\ b' \le B,\ b \ne b' \tag{2.4}$$

$$C_b - (C_{b'} - p_{b'}) - (1 - Y_{b,b'}) M \le 0 \quad \forall 1 \le b,\ b' \le B,\ b \ne b' \tag{2.5}$$

$$M(Y_{b,b'} + Y_{b',b}) \ge \sum_{k=1}^{K} k X_{b,k} - \sum_{k'=1}^{K} k' X_{b',k'} + 1 \quad \forall 1 \le b < b' \le B \tag{2.6}$$

$$X_{b,k},\ Y_{b,b'} = 0 \text{ or } 1 \quad \forall 1 \le b,\ b' \le B,\ b \ne b',\ \forall 1 \le k \le K \tag{2.7}$$

The objective function (2.1) minimizes the makespan of handling one single container ship, which is the latest completion time among all ship bays. Constraints (2.2) define the property of the decision variable C_b. Constraints (2.3) ensure that every ship bay must be performed only by one quay crane. Constraints (2.4) and (2.5) define the properties of decision variables $Y_{b,b'}$: Constraints (2.4) indicate that $Y_{b,b'} = 1$ if $C_b \leq C_{b'} - p_{b'}$, which means $Y_{b,b'} = 1$ when ship bay b finishes no later than ship bay b' starts; Constraints (2.5) indicate that $Y_{b,b'} = 0$ if $C_b > C_{b'} - p_{b'}$, which means $Y_{b,b'} = 0$ when ship bay b finishes after ship bay b' starts. Finally, crossing between quay cranes can be avoided by imposing Constraints (2.6). Suppose that ship bays b and b' are performed simultaneously and $b < b'$, and this means that $Y_{b,b'} + Y_{b',b} = 0$. Note that both quay cranes and ship bays are arranged in an increasing order from the front to the tail of the container ship. Thus, if quay crane k handles ship bay b and quay crane k' handles ship bay b', then $k + 1 \leq k'$. For example, Ship Bay 3 and Ship Bay 8 are performed simultaneously, and thus $Y_{3,8} + Y_{8,3} = 0$. If Ship Bay 3 is assigned to Quay Crane 4 and Ship Bay 8 is assigned to Quay Crane 2, Constraint (2.6) $0 \geq 4 - 2 + 1 = 3$ does not satisfy. This means that Constraint (2.6) does not allow the aforementioned quay crane schedule to avoid the crossing between quay cranes.

2.2 PROOF OF NP-COMPLETENESS

This chapter discusses computational complexity of the QCSNCP 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 unless P=NP. Hence heuristic algorithms are needed to obtain near optimal solutions for the problem. In this chapter, the proposed QCSNCP 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 QCSNCP is presented in Section 2.1, and its decision version is defined as follows:

Parameter:

Z^+ the set of positive integer.

Instance: There are B ship bays and K quay cranes. The processing time of ship bay b by a quay crane is $p_b \in Z^+$ ($1 \le b \le B$). There is a given number $C \in Z^+$.

Question: Is there a quay crane schedule for these K quay cranes handling these B ship bays such that no crossing between quay cranes exists and the makespan of the quay crane schedule $\le C$?

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

Theorem 2.1: QCSNCP is NP-complete.

Proof:

***Step 1*: Showing that the QCSNCP is in NP.**

If a quay crane schedule for the QCSNCP is given, its feasibility can be checked in polynomial time. Checking whether the quay crane schedule satisfies the non-crossing constraints can be done in $O(B^2)$ time. Checking whether the makespan of the quay crane schedule $\leq C$ can be done in $O(B)$ time. Therefore, the QCSNCP 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 B elements in a finite set $S = \{s_1, s_2, \cdots, s_B\}$. For each element $s_b \in S$, $s_b \in Z^+$ and the sum of all elements $\sum_{s_b \in S} s_b = D$.

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

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 $\sum_{s_b \in S} s_b = D = 806$. 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 $\sum_{s_b \in S_1} s_b = \sum_{s_b \in S_2} s_b = D/2 = 403$.

***Step 3*: Constructing a transformation from the PARTITION to the QCSNCP.**

The PARTITION is transformed to the QCSNCP as follows. A QCSNCP instance corresponding to an arbitrary PARTITION instance has K quay cranes and $B+K$ ship bays; the given number C is set as D; the following Equations (2.8)-(2.10) indicate the processing time of each ship bay which means the processing time of Ship Bay 1 and Ship Bay $B+2$ is set as $D/2$, the processing time of Ship Bay 2 to Ship Bay $B+1$ is set as s_1 to s_B respectively, and the processing time of Ship Bay $B+3$ to Ship Bay $B+K$ is set as D. Figure 2.1 illustrates this transformation. It shows K quay cranes, $B+K$ ship bays and the processing time of each ship bay.

$$p_1 = p_{B+2} = D/2 \tag{2.8}$$

$$p_{b+1} = s_b \quad \forall 1 \le b \le B \tag{2.9}$$

$$p_b = D \quad \forall B+3 \le b \le B+K \tag{2.10}$$

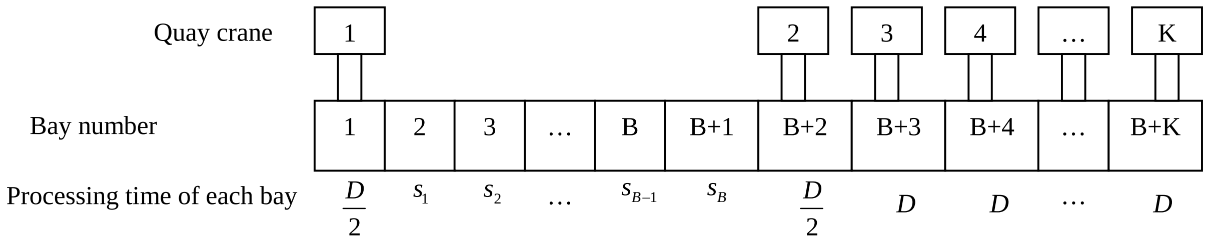

Figure 2.1 The Illustration of the Transformation from the PARTITION to the QCSNCP

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

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

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

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

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

Therefore, $\text{PARTITION} \propto \text{QCSNCP}$, and the Theorem 2.1 is proved.

2.3 AN APPROXIMATION ALGORITHM

As proved in the previous section, QCSNCP is NP-complete, and thus there exists no polynomial time algorithm for the exact solution to QCSNCP unless P=NP. This section proposes an approximation algorithm to obtain its near optimal solution which is elaborated as follows:

Parameters:

AT the average working time of a quay crane;

k quay crane number ($1 \le k \le K$);

b_1, b_2 ship bay number ($1 \le b_1 \le b_2 \le B$).

***Step 0*:** Set $k = 1$, $b_1 = b_2 = 1$.

***Step 1*:** Calculate $AT = \sum_{b=1}^{B} p_b / K$.

***Step 2*:** If $\sum_{b=b_1}^{b_2} p_b \le AT$, then $b_2 = b_2 + 1$ and repeat Step 2; if $\sum_{b=b_1}^{b_2} p_b > AT$, then go to Step 3.

***Step 3*:** If $\left|\sum_{b=b_1}^{b_2} p_b - AT\right| > \left|\sum_{b=b_1}^{b_2-1} p_b - AT\right|$ and $\left|\sum_{b=b_1}^{b_2-1} p_b - AT\right| < AT/(K-1)$, then assign Ship Bay b_1 to Ship Bay $b_2 - 1$ to Quay Crane k, set $b_1 = b_2$, $k = k+1$, and go to Step 4; otherwise, assign Ship Bay b_1 to Ship Bay b_2 to Quay Crane k, set $b_1 = b_2 + 1,\ b_2 = b_2 + 1,\ k = k+1$, and go to Step 4.

***Step 4*:** If $k \le K-1$, then go to Step 2; if $k = K$, then assign Ship Bay b_1 to Ship Bay B to Quay Crane K and go to End.

Figure 2.2 shows a numerical example of the approximation algorithm, which has two quay cranes and six ship bays.

***Step 0*:** Set $k = 1, b_1 = b_2 = 1$.

***Step 1*:** Calculate $AT = \sum_{b=1}^{B} p_b / K = (112+187+90+241+71+132)/2=416.5$.

***Step 2*:** Since $\sum_{b=1}^{3} p_b = 112+187+90=389<416.5 = AT$ and $\sum_{b=1}^{4} p_b = 112+187+90+241=630 > 416.5 = AT$, go to Step 3.

***Step 3*:** Since $\left|\sum_{b=1}^{4} p_b - AT\right| = 630 - 416.5 = 213.5 > 27.5 = 416.5 - 389 = \left|\sum_{b=1}^{3} p_b - AT\right|$ and $\left|\sum_{b=1}^{3} p_b - AT\right| = 27.5 < 416.5 = AT/(K-1)$, assign Ship Bay 1 to Ship Bay 3 to Quay Crane 1, set $b_1 = b_2 = 4,\ k = k+1 = 2$, and go to Step 4.

***Step 4*:** Since $k = 2 = K$, assign Ship Bay 4 to Ship Bay 6 to Quay Crane 2 and go to End.

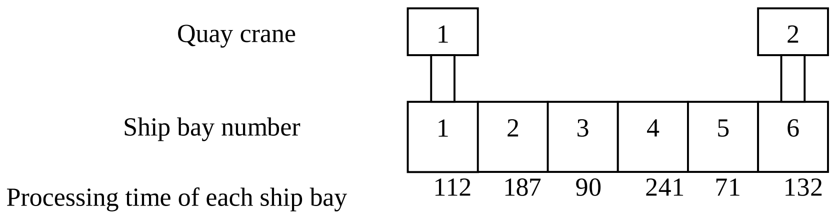

Figure 2.2 A Numerical Example of the Approximation Algorithm

Worst-case analysis for the approximation algorithm is performed as follows:

Parameters:

c_k the completion time of quay crane k ($1 \le k \le K$);

Z the objective function value of the solution obtained by the approximation algorithm;

Z^* the objective function value of the optimal solution to the QCSNCP.

Theorem 2.2: $Z / Z^* \le 2$.

Proof:

Note that $Z = \max_k c_k$. Assume the completion time of Quay Crane l $(1 \le l \le K-1)$ is the latest and Ship Bay i to Ship Bay $i+j$ are assigned to Quay Crane l, and thus $Z = c_l = p_i + p_{i+1} + ... + p_{i+j-1} + p_{i+j}$. According to the approximation algorithm, $p_i + p_{i+1} + ... + p_{i+j-1} \le AT \le p_i + p_{i+1} + ... + p_{i+j-1} + p_{i+j}$, and hence $Z \le AT + p_{i+j}$. From the objective function (2.1) and the property of C_b that is $C_b \ge p_b$ $\forall 1 \le b \le B$, it is clear that $Z^* \ge p_b$ $\forall 1 \le b \le B$. Therefore, $p_{i+j} \le Z^*$. Obviously $AT \le Z^*$, and thus

$Z \le AT + p_{i+j} \le 2Z^*$. On the other hand, since in Step 3 $\left|\sum_{b=b_1}^{b_2-1} p_b - AT\right| < AT/(K-1)$, the completion time of Quay Crane K $c_K < 2AT \le 2Z^*$. Thus, the Theorem 2.2 is proved.

As shown in Figure 2.3, the error bound of 2 is tight for the proposed approximation algorithm in terms of the instance which has K quay cranes and $2K$ ship bays (assume $K > 3$). The processing time of the leftmost K ship bays is all $K-1$ and the processing time of the rightmost K ship bays is all 1. The optimal schedule is to assign two ship bays to each quay crane, one from the leftmost K ship bays and the other from the rightmost K ship bays. The optimal makespan is K. The approximation algorithm is to assign Ship Bay 1 to Ship Bay $K-1$ to Quay Crane 1 to Quay Crane $K-1$ respectively and to assign Ship Bay K to Ship Bay $2K$ to Quay Crane K. The makespan obtained by the approximation algorithm is $2K-1$. Therefore, $Z/Z^* = (2K-1)/K \to 2$ as $K \to \infty$.

Bay number	1	…	K	K+1	…	2K
Processing time of each bay	K-1	K-1	K-1	1	1	1

Figure 2.3 A Tight Instance for the Approximation Algorithm

2.4 COMPUTATIONAL EXPERIMENTS FOR THE APPROXIMATION ALGORITHM

A series of computational experiments are conducted to examine the performance of the proposed model and approximation algorithm. The Approximation Algorithm (AA) is coded in C++ and executed in a Pentium IV 1.7GHz PC with 256MB RAM.

There are twenty random instances generated in which the processing time of a ship bay is randomly generated from a uniform distribution of $U(30,300)$. In order to evaluate the performance of the proposed approximation algorithm in solving the instance, the lower bound corresponding to the instance can be calculated by relaxing the non-crossing constrains. The mathematical model of the relaxed problem is formulated as follows:

Minimize:

$$\max_{k} \ c_k \tag{2.11}$$

Subject to:

$$\sum_{k=1}^{K} X_{b,k} = 1 \quad \forall 1 \le b \le B \tag{2.12}$$

$$c_k \ge \sum_{b=1}^{B} X_{b,k} p_b \quad \forall 1 \le k \le K \tag{2.13}$$

$$X_{b,k} = 0 \text{ or } 1 \quad \forall 1 \le b \le B, \ \forall 1 \le k \le K \tag{2.14}$$

The objective function (2.11) minimizes the makespan of handling one single container ship without considering the non-crossing constraints. Constraints (2.12) ensure that every ship bay must be performed only by one quay crane. Constraints (2.13) define the property of the decision variable c_k. The mathematical model of the relaxed problem can be exactly solved by CPLEX (a commercial software for integer programming). 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 2.1, the gaps between solutions obtained from the proposed approximation algorithm and lower bounds are all small (for example the maximum gap among the twenty instances is 11.18%, the minimum gap is 1.59%, and the average gap is 7.08%), and all the computational time of these twenty instances is within one second. Therefore, the proposed approximation algorithm is concluded to be effective and efficient in solving the proposed QCSNCP.

Table 2.1 The Results of Computational Experiments for the Approximation Algorithm

Experiment No	Size (bays×cranes)	Lower Bound	AA	Gap [a] (%)
1	16×3	953	990	3.88
2	16×4	754	766	1.59
3	17×3	960	1044	8.75
4	17×4	667	714	7.05
5	18×3	964	1024	6.22
6	18×4	723	795	9.96
7	19×3	906	941	3.86
8	19×4	861	933	8.36
9	20×3	915	998	9.07
10	20×4	686	727	5.98
11	21×3	1134	1181	4.14
12	21×4	850	937	10.24
13	22×3	1453	1487	2.34
14	22×4	1011	1116	10.39
15	23×3	1312	1441	9.83
16	23×4	984	1080	9.76
17	24×3	1372	1476	7.58
18	24×4	1216	1352	11.18
19	25×3	1484	1532	3.23
20	25×4	1113	1204	8.18

[a] Gap = (solution obtained from the proposed AA - lower bound)×100/lower bound

2.5 A GENETIC ALGORITHM

This chapter employs a genetic algorithm (GA) to obtain near optimal solutions to the QCSNCP. GA is a search algorithm based on the mechanisms of natural selection and genetics. In general, there are three common genetic operators in a GA: selection, crossover, and mutation. The procedure of the proposed GA is illustrated in Figure 2.4 and the details of the proposed GA are elaborated as follows.

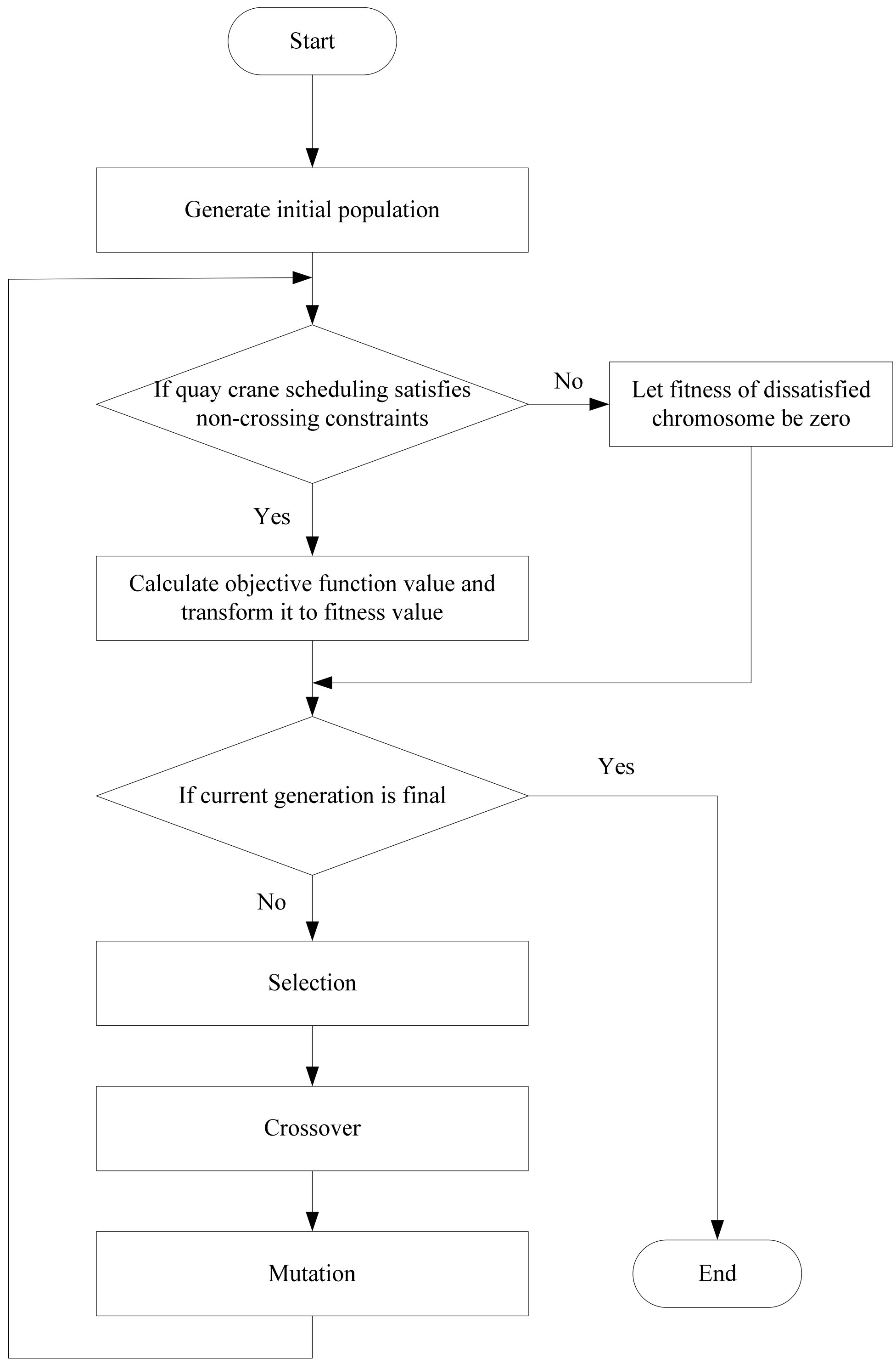

Figure 2.4 The Flowchart of the Proposed GA

2.5.1 Chromosome Representation and Decoding Procedure

Parameter:

ΔB the largest integer $\leq B / K$.

In this chapter, the position of each quay crane is measured in terms of the ship bay number. For example, Quay Crane 1 is on Ship Bay 1. The initial position of Quay Crane k in the proposed GA is on Ship Bay $1+(k-1)\Delta B$ ($\forall 1 \leq k \leq K$).

A chromosome of the GA represents a sequence of ship bays. Figure 2.5 provides a sample chromosome, in which a gene is a ship bay number. Based on the sequence of ship bays represented by the chromosome, a quay crane schedule can be constructed using the following procedure.

Gene: ship bay number 1-9

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

Figure 2.5 An Illustration of the Chromosome Representation

***Step 1*:** Based on the current position of each quay crane, determine which quay cranes can handle the first unassigned Ship Bay b in the chromosome without crossing other quay cranes. If there is only one Quay Crane k available, Ship Bay b is assigned to Quay Crane k . Then, Ship Bay b is deleted from the chromosome, the position of Quay Crane k is set as Ship Bay b , the completion time of Quay Crane k is set as

$c_k = c_k + p_b$, and go to Step 5. If there are two quay cranes available that are Quay Crane k and Quay Crane $k+1$, go to Step 2.

Step 2**:** Compare the completion time of the two available quay cranes to finish their assigned ship bays and assign this ship bay to the quay crane with earlier completion time. Suppose $c_k < c_{k+1}$, and thus assign Ship Bay b to Quay Crane k. Then, Ship Bay b is deleted from the chromosome, the position of Quay Crane k is set as Ship Bay b, the completion time of Quay Crane k is set as $c_k = c_k + p_b$, and go to Step 5. If their completion time is equal that is $c_k = c_{k+1}$, go to Step 3.

Step 3**:** Compare the distance between this ship bay and these two available quay cranes and assign this ship bay to the quay crane with the shorter distance. Suppose Quay Crane k with the shorter distance, and thus assign Ship Bay b to Quay Crane k. Then, Ship Bay b is deleted from the chromosome, the position of Quay Crane k is set as Ship Bay b, the completion time of Quay Crane k is set as $c_k = c_k + p_b$, and go to Step 5. If their distance is equal, go to Step 4.

Step 4**:** Assign this ship bay to the quay crane with the smaller number, and thus assign Ship Bay b to Quay Crane k. Then, Ship Bay b is deleted from the chromosome, the position of Quay Crane k is set as Ship Bay b, the completion time of Quay Crane k is set as $c_k = c_k + p_b$, and go to Step 5.

Step 5**:** If there are unassigned ship bays in the chromosome, go to Step 1; otherwise, go to End.

Figure 2.6 shows a numerical example of the above mentioned procedure for constructing a quay crane schedule from a chromosome. There are three quay cranes and twelve ship bays. The initial position of Quay Crane 1, Quay Crane 2, and Quay Crane 3 are on Ship Bay 1, Ship Bay 5, and Ship Bay 9 respectively. The initial completion time of three quay cranes is all 0. The first unassigned ship bay in the chromosome is Ship Bay 7, of which the processing time is 114.

Step 1**:** Quay Crane 2 and Quay Crane 3 can handle Ship Bay 7 without crossing 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 Ship Bay 7 and Quay Crane 2, Quay Crane 3 is both 1 ship bay, go to Step 4.

Step 4**:** Assign Ship Bay 7 to Quay Crane 2. Then, Ship Bay 7 is deleted from the chromosome, the position of Quay Crane 2 is on Ship Bay 7, the completion time of Quay Crane 2 is 114, and go to Step 5.

Step 5**:** Since Ship Bay 12 is the first unassigned ship bay in the chromosome, go to Step 1.

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

Figure 2.6 An Illustration of Constructing a Quay Crane Schedule from a Chromosome

2.5.2 Fitness Evaluation and Selection

Most of the quay crane schedules obtained from the above mentioned procedure do not violate the non-crossing constraints. However, every quay crane schedule must be checked whether it satisfies the non-crossing constraints as follows. According to a quay crane schedule constructed from a chromosome, Constraints (2.4) and Constraints (2.5), $Y_{b,b'}\ \forall 1 \le b,\ b' \le B$ can be obtained and then the quay crane schedule can be checked whether it satisfies Constraints (2.6). If it satisfies Constraints (2.6), the fitness value of its corresponding chromosome is set to be the reciprocal of its objective function value, as shown in Equation (2.15); otherwise, the fitness value of its corresponding chromosome is zero.

$$Fitness = 1 \Big/ \max_b \ C_b \qquad (2.15)$$

In this chapter, 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).

2.5.3 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 ship bays may be missed while some ship bays may be duplicated in the offspring. Therefore, this chapter 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 ship bays which are already in the substring from the second parent. The resulted sequence of ship bays contains the ship bays that the proto-child needs.

***Step 4*:** Place the ship bays into the unfixed positions of the proto-child from left to right according to the order of the sequence to produce an offspring.

The 'order crossover' is illustrated in Figure 2.7 that presents an example of producing two offspring from the same parents.

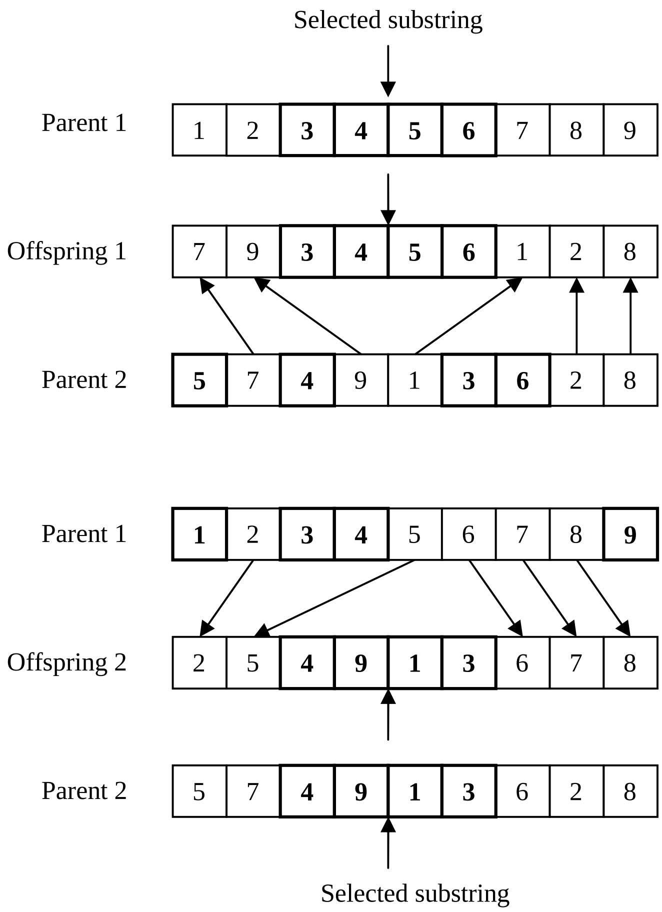

Figure 2.7 An Illustration of the Order Crossover

2.5.4 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 chapter 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 ship bays on these positions as illustrated in Figure 2.8.

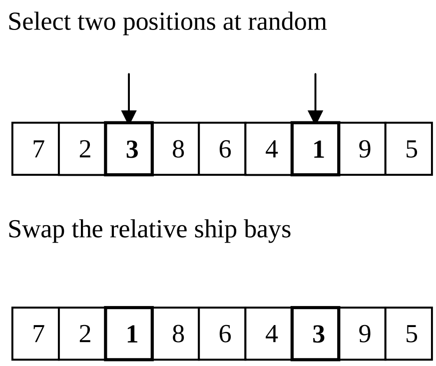

Figure 2.8 An Illustration of the Mutation

2.6 COMPUTATIONAL EXPERIMENTS FOR THE GENETIC ALGORITHM

A series of computational experiments are conducted to examine the performance of the proposed model and GA as well. The GA is coded in C++ and executed in a Pentium IV 1.7GHz 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.

2.6.1 Random Instances with Small Sizes

Six random instances with small sizes are created, and the processing time of a ship bay 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 2.2, the computational time of CPLEX grows exponentially as the instance size increases since the QCSNCP 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 five seconds) when the instance size is small.

Table 2.2 Results of Random Instances with Small Sizes

Experiment No	Size (bays×cranes)	CPLEX		GA	
		Value	CPU (sec)	Value	CPU (sec)
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

2.6.2 Random Instances with Large Sizes

There are forty random instances with large sizes generated. The processing time of a ship bay 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 obtained by the same method which is elaborated in Section 2.4 (Page 28-29).

As observed in Table 2.3, 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 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 one hundred and twenty seconds). Based on these forty 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 2.3, 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 twelve instances.

The lower bound is the objective function value of the optimal solution to the relaxed problem, which does not consider the non-crossing constraints between quay cranes. The proposed GA obtains the near optimal solution to the original problem. As shown in Table 2.3, the larger gaps are observed for smaller container ships with fewer ship bays handled by more quay cranes. The reason for it can be that the non-crossing constraints between quay cranes more significantly affect scheduling more quay cranes for smaller container ships with fewer ship bays.

According to the computational experiments with small and large sizes, the proposed GA is concluded to be effective and efficient in solving the proposed QCSNCP.

Table 2.3 Results of Random Instances with Large Sizes

Experiment No	Size (bays×cranes)	Lower Bound	GA Value	GA CPU (sec)	Gap [a] (%)
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

2.7 SUMMARY

This chapter provides a mixed integer programming model for the proposed QCSNCP, proves that the QCSNCP is NP-complete, and proposes an approximation algorithm and a genetic algorithm to obtain near optimal solutions for the QCSNCP. Worst-case analysis for the AA is performed and computational experiments are conducted to examine the proposed model, AA and GA. The results show that both the proposed AA and GA are effective and efficient in solving the QCSNCP.

CHAPTER 3 QUAY CRANE SCHEDULING WITH SAFETY DISTANCE AND NON-CROSSING CONSTRAINTS

As discussed in Chapter 1, there are requirements of maintaining safety distance between any two quay cranes in operation. Based on Chapter 2, this chapter studies the Quay Crane Scheduling with Safety Distance and non-crossing constraints Problem (QCSSDP).

3.1 MODEL FORMULATION

This chapter proposes a mixed integer programming model for the QCSSDP. According to the configuration of container ships, one single container ship is divided into ship bays. Figure 1.2 shows that both quay cranes and ship bays are arranged in an increasing order from the front to the tail of the container ship. The following assumptions are imposed in formulating the QCSSDP:

1. Quay cranes are operated on the same tracks and thus cannot cross over each other.
2. There are requirements of maintaining safety distance between any two quay cranes in operation.
3. Only one quay crane can work on a ship bay at a time until it completes the ship bay.
4. Compared with the processing time of a ship bay by a quay crane, the travel time of a quay crane between two ship bays is small and hence it is not considered.

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

Parameters:

K the number of quay cranes;

B the number of ship bays;

p_b the processing time of ship bay b by a quay crane ($1 \le b \le B$);

$sd_{k,k'}$ the required safety distance between quay crane k and quay crane k' ($1 \le k,\ k' \le K$);

M a sufficiently large positive constant number;

Decision variables:

$X_{b,k,i}$ 1, if ship bay b is handled as the i th ship bay by quay crane k; 0, otherwise ($1 \le b \le B, 1 \le k \le K, 1 \le i \le B$);

$Y_{b,b'}$ 1, if ship bay b finishes no later than ship bay b' starts; 0, otherwise ($1 \le b,\ b' \le B,\ b \ne b'$);

C_b the completion time of ship bay b ($1 \le b \le B$).

The QCSSDP can be formulated as follows:

Minimize:

$$\max_b \ C_b \tag{3.1}$$

Subject to:

$$\sum_{k=1}^{K} \sum_{i=1}^{B} X_{b,k,i} = 1 \quad \forall 1 \le b \le B \tag{3.2}$$

$$\sum_{b=1}^{B} X_{b,k,i} \le 1 \quad \forall 1 \le k \le K, \forall 1 \le i \le B \tag{3.3}$$

$$C_b \ge \sum_{b'=1}^{B} \sum_{k=1}^{K} \sum_{i'=1}^{i} p_{b'} X_{b',k,i'} X_{b,k,i} \quad \forall 1 \le b \le B, \forall 1 \le i \le B \tag{3.4}$$

$$C_b - (C_{b'} - p_{b'}) + Y_{b,b'}M > 0 \quad \forall 1 \le b,\ b' \le B,\ b \ne b' \tag{3.5}$$

$$C_b - (C_{b'} - p_{b'}) - (1 - Y_{b,b'})M \le 0 \quad \forall 1 \le b,\ b' \le B,\ b \ne b' \tag{3.6}$$

$$M(Y_{b,b'} + Y_{b',b}) \ge \sum_{k=1}^{K}\sum_{i=1}^{B} kX_{b,k,i} - \sum_{k'=1}^{K}\sum_{i'=1}^{B} k'X_{b',k',i'} + 1 \quad \forall 1 \le b < b' \le B \tag{3.7}$$

$$M(Y_{b,b'} + Y_{b',b}) \ge b + sd_{\sum_{k=1}^{K}\sum_{i=1}^{B} kX_{b,k,i},\ \sum_{k'=1}^{K}\sum_{i'=1}^{B} k'X_{b',k',i'}} + 1 - b' \quad \forall 1 \le b < b' \le B \tag{3.8}$$

$$X_{b,k,i}, Y_{b,b'} = 0 \text{ or } 1 \quad \forall 1 \le b,\ b' \le B, b \ne b', \forall 1 \le k \le K, \forall 1 \le i \le B \tag{3.9}$$

The objective function (3.1) minimizes the makespan of handling one single container ship, which is the latest completion time among all ship bays. Constraints (3.2) ensure that every ship bay must be handled only by one quay crane. Constraints (3.3) enforce that every quay crane handles up to one ship bay at any time. Constraints (3.4) define the properties of decision variables C_b. Constraints (3.5) and (3.6) define the properties of decision variables $Y_{b,b'}$: Constraints (3.5) indicate that $Y_{b,b'} = 1$ if $C_b \le C_{b'} - p_{b'}$, which means $Y_{b,b'} = 1$ when ship bay b finishes no later than ship bay b' starts; Constraints (3.6) indicate that $Y_{b,b'} = 0$ if $C_b > C_{b'} - p_{b'}$, which means $Y_{b,b'} = 0$ when ship bay b finishes after ship bay b' starts. The crossing between quay cranes can be avoided by imposing Constraints (3.7). Suppose that ship bays b and b' are performed simultaneously and $b < b'$, then this means that $Y_{b,b'} + Y_{b',b} = 0$. Note that both quay cranes and ship bays are arranged in an increasing order from the front to the tail of the container ship. Thus, if quay crane k handles ship bay b and quay crane k' handles ship bay b', then $k + 1 \le k'$. Constraints (3.8) guarantee the safety distance between any two quay cranes in operation. Suppose that ship bay b is handled by quay crane k and at the same time ship bay b' is handled by quay crane k', then this means that

$Y_{b,b'} + Y_{b',b} = 0$, $\sum_{k=1}^{K}\sum_{i=1}^{B} kX_{b,k,i} = k$, and $\sum_{k'=1}^{K}\sum_{i'=1}^{B} k'X_{b',k',i'} = k'$. Therefore, the distance between quay crane k and quay crane k', which is $b' - b - 1$, must be no less than the required safety distance $sd_{k,k'}$.

3.2 PROOF OF NP-COMPLETENESS

This chapter discusses computational complexity of the QCSSDP to justify why heuristic algorithms are adopted. The optimization version of the QCSSDP is presented in Section 3.1 and the decision version is defined as follows:

Parameter:

Z^+ the set of positive integer.

Instance: There are B ship bays and K quay cranes. The processing time of ship bay b by a quay crane is $p_b \in Z^+$ ($1 \le b \le B$). There is a given number $C \in Z^+$.

Question: Is there a quay crane schedule for these K quay cranes handling these B ship bays such that the safety distance and non-crossing constraints are satisfied and the makespan of the quay crane schedule $\le C$?

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

Theorem 3.1: QCSSDP is NP-complete.

Proof:

***Step 1*: Showing that the QCSSDP is in NP.**

If a quay crane schedule for the QCSSDP is given, its feasibility can be checked in polynomial time. Checking whether the quay crane schedule satisfies the safety distance constraints can be done in $O(B^2)$ time. Checking whether the quay crane schedule satisfies the non-crossing constraints can be done in $O(B^2)$ time. Checking whether the makespan of the quay crane schedule $\leq C$ can be done in $O(B)$ time. Therefore, the QCSSDP 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 in Section 2.2 (Page 21).

***Step 3*: Constructing a transformation from the PARTITION to the QCSSDP.**

The PARTITION is transformed to the QCSSDP as follows. A QCSSDP instance corresponding to an arbitrary PARTITION instance has 2 quay cranes and $B+4$ ship bays; the given number C is set as D; the safety distance between the two quay cranes is one ship bay; the following Equations (3.10)-(3.12) indicate the processing time of each ship bay which means the processing time of Ship Bay 1 and Ship Bay $B+4$ is set as $\frac{D}{2}$, the processing time of Ship Bay 2 and Ship Bay $B+3$ is set as 0, and the processing time of Ship Bay 3 to Ship Bay $B+2$ is set as s_1 to s_B respectively. Figure 3.1 illustrates this transformation, which shows 2 quay cranes, $B+4$ ship bays, and the processing time of each ship bay.

$$p_1 = p_{B+4} = \frac{D}{2} \tag{3.10}$$

$$p_2 = p_{B+3} = 0 \tag{3.11}$$

$$p_{b+2} = s_b \quad \forall 1 \le b \le B \tag{3.12}$$

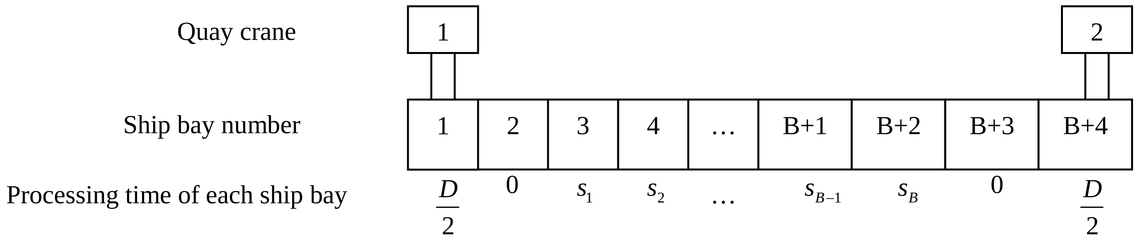

Figure 3.1 The Illustration of the Transformation from the PARTITION to the QCSSDP

Then, it must be proved that the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_b \in S_1} s_b = \sum_{s_b \in S_2} s_b = \frac{D}{2}$ if and only if all the $B+4$ ship bays can be completed by 2 quay cranes in D time with satisfying the safety distance and non-crossing constraints.

First, suppose that the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_b \in S_1} s_b = \sum_{s_b \in S_2} s_b = \frac{D}{2}$. Then 2 quay cranes can be scheduled with satisfying the safety distance and non-crossing constraints as follows: Quay Crane 1 handles all the Ship Bays $b+2$, where $s_b \in S_1$ and then Ship Bay 1; Quay Crane 2 handles Ship Bay $B+4$ and then all the Ship Bays $b+2$, where $s_b \in S_2$. Obviously, the safety distance and non-crossing constraints are satisfied in this schedule and the latest completion time among all ship bays is D. Hence, if the set S can be partitioned into two disjoint subsets S_1 and S_2

such that $\sum_{s_b \in S_1} s_b = \sum_{s_b \in S_2} s_b = \frac{D}{2}$, all the $B+4$ ship bays can be completed by 2 quay cranes in D time with satisfying the safety distance and non-crossing constraints.

Conversely, suppose all the $B+4$ ship bays can be completed by 2 quay cranes in D time with satisfying the safety distance and non-crossing constraints, then both the 2 quay cranes are fully utilized as the total processing time of all ship bays is $2D$. Thus, the completion time of each quay crane must be D. Furthermore, the safety distance and non-crossing constraints are satisfied in the above mentioned quay crane schedule. According to this quay crane schedule, the total processing time of all ship bays except Ship Bay 1 handled by Quay Crane 1 must be $\frac{D}{2}$ and the total processing time of all ship bays except Ship Bay $B+4$ handled by Quay Crane 2 must be $\frac{D}{2}$ as well, which means that the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_b \in S_1} s_b = \sum_{s_b \in S_2} s_b = \frac{D}{2}$. Hence, if all the $B+4$ ship bays can be completed by 2 quay cranes in D time with satisfying the safety distance and non-crossing constraints, the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_b \in S_1} s_b = \sum_{s_b \in S_2} s_b = \frac{D}{2}$.

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

The above mentioned transformation can be done in $O(B)$ time.

Therefore, $\text{PARTITION} \propto \text{QCSSDP}$, and the Theorem 3.1 is proved.

3.3 AN APPROXIMATION ALGORITHM

As proved in the previous section, QCSSDP is NP-complete, and thus there exists no polynomial time algorithm for the exact solution to the QCSSDP unless P=NP. This section proposes an approximation algorithm to obtain its near optimal solution which is elaborated as follows.

Approximation Algorithm: assign adjacent ship bays, $b_{k-1}+1,\ b_{k-1}+2,\ ...,\ b_k-1,\ b_k$, to quay crane k ($\forall 1 \le k \le K$). Note that $b_0 = 0$ and $b_K = B$. A dynamic programming algorithm is then proposed to determine the best partition points, $b_1,\ b_2,\ ...,\ b_{K-2},\ b_{K-1}$, which minimizes the latest completion time among all ship bays.

Parameters:

c_k the completion time of quay crane k ($1 \le k \le K$);

$MC[k,\ b]$ the minimum latest completion time when ship bays $1,\ 2,\ ...,\ b-1,\ b$ are assigned to quay cranes $1,\ 2,\ ...,\ k-1,\ k$ in the above mentioned adjacent manner;

$TP[b_1,\ b_2] = \sum_{b=b_1}^{b_2} p_b$ the total processing time of ship bays $b_1,\ b_1+1,\ ...,\ b_2-1,\ b_2$.

Dynamic programming equations for determining the best partition points, $b_1,\ b_2,\ ...,\ b_{K-2},\ b_{K-1}$, are as follows:

$$MC[1,\ b] = TP[1,\ b] \quad \forall 1 \le b \le B \tag{3.13}$$

$$MC[k,\ b] = \min_{k-1 \le b_{k-1} \le b-1} \max\{MC[k-1,\ b_{k-1}],\ TP[b_{k-1}+1,\ b]\}\ \forall 2 \le k \le K, \forall k \le b \le B \quad (3.14)$$

Based on the best partition points, quay crane k handles the assigned ship bays according to the sequence of $b_{k-1}+1,\ b_{k-1}+2,\ ...,\ b_k-1,\ b_k$ ($\forall 1 \le k \le K$). The obtained quay crane schedule obviously satisfies the non-crossing constraints. Then, check whether the obtained quay crane schedule satisfies the safety distance constraints. The possible scenario of violating the safety distance constraints is described as follows. Assume the safety distance between two adjacent quay cranes in operation is one ship bay. When quay crane k has already completed ship bays $b_{k-1}+1,\ b_{k-1}+2,\ ...,\ b_k-1$ and is ready to handle ship bay b_k, quay crane $k+1$ is still handling ship bay b_k+1. Consequently, quay crane k can start to handle ship bay b_k until quay crane $k+1$ finishes ship bay b_k+1 which means quay crane k has to wait due to the safety distance constraint. Therefore, if the obtained quay crane schedule does not satisfy the safety distance constraints, the completion time of the corresponding quay cranes must be adjusted to include waiting time. Otherwise, the completion time of every quay crane is equal to the total processing time of its assigned ship bays.

Figure 3.2 shows a numerical example of the approximation algorithm in which there are two quay cranes and four ship bays. Assume the safety distance between the two quay cranes is one ship bay. According to Equation (3.13), $MC[1,\ 1] = TP[1,\ 1] = 196$, $MC[1,\ 2] = TP[1,\ 2] = 302$, $MC[1,\ 3] = TP[1,\ 3] = 392$, and $MC[1,\ 4] = TP[1,\ 4] = 460$.

According to Equation (3.14),

$MC[2,\ 4] = \min\ \{\max\{MC[1,\ 1],\ TP[2,\ 4]\},\ \max\{MC[1,\ 2],\ TP[3,\ 4]\},$
$\max\{MC[1,\ 3],\ TP[4,\ 4]\}\} = \min\ \{\max\{196,\ 264\},\ \max\{302,\ 158\},\ \max\{392,\ 68\}\}$
$= \min\ \{264,\ 302,\ 392\} = 264.$

Therefore, Ship Bay 1 is assigned to Quay Crane 1, and Ship Bay 2, 3, and 4 are assigned to Quay Crane 2. Based on the best partition points, Quay Crane 1 handles Ship Bay 1 and Quay Crane 2 handles the assigned ship bays according to the sequence of 2, 3, and 4. However, the obtained quay crane schedule does not satisfy the safety distance constraints. Hence, the completion time of Quay Crane 1 is $c_1 = 196 + \text{waiting time} = 196 + 106 = 302$, the completion time of Quay Crane 2 is $c_2 = 106 + 90 + 68 = 264$, and the makespan of this quay crane schedule is 302.

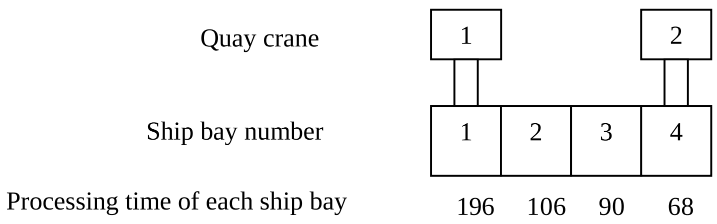

Figure 3.2 A Numerical Example of the Approximation Algorithm

Assume the safety distance between two adjacent quay cranes in operation is one ship bay. Worst-case analysis for the approximation algorithm is performed as follows.

Parameters:

Z_1 the objective function value of the solution to the QCSNCP obtained by the approximation algorithm proposed in Chapter 2;

Z_2 the objective function value of the solution to the QCSNCP obtained by the approximation algorithm proposed in Chapter 3;

Z_3 the objective function value of the solution to the QCSSDP obtained by the approximation algorithm proposed in Chapter 3;

Z_1^* the objective function value of the optimal solution to the QCSNCP;

Z_2^* the objective function value of the optimal solution to the QCSSDP.

Theorem 3.2: $Z_3 / Z_2^* \leq 3$

Proof:

Both the approximation algorithm proposed in Chapter 2 and the approximation algorithm proposed in Chapter 3 assign ship bays to quay cranes in the aforementioned adjacent manner. According to the Theorem 2.2, $Z_1 \leq 2Z_1^*$. Since the approximation algorithm proposed in Chapter 3 optimizes the partition points, $b_1, b_2, ..., b_{K-2}, b_{K-1}$, $Z_2 \leq Z_1 \leq 2Z_1^*$. With considering the safety distance constraints, the worst case is $Z_3 = Z_2 + \text{waiting time}$. Since $\text{waiting time} \leq \max_b p_b \leq Z_1^*$, $Z_3 \leq 3Z_1^*$. Obviously, $Z_1^* \leq Z_2^*$, and hence $Z_3 \leq 3Z_2^*$. The Theorem 3.2 is proved.

3.4 COMPUTATIONAL EXPERIMENTS FOR THE APPROXIMATION ALGORITHM

A series of computational experiments are conducted to examine the performance of the proposed approximation algorithm which is coded in C++ and executed in a Pentium IV 1.7GHz PC with 256MB RAM.

There are twenty random instances generated in which the processing time of a ship bay is randomly generated from a uniform distribution of $U(30,300)$. Assume that the safety distance between two adjacent quay cranes in operation is one ship bay.

In order to evaluate the performance of the proposed approximation algorithm in solving the instance, the lower bound corresponding to the instance can be calculated by relaxing the safety distance and non-crossing constraints. The mathematical model of the relaxed problem is formulated in Section 2.4 (Page 28-29) that 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 3.1, the gaps between solutions obtained from the proposed approximation algorithm and lower bounds are all small (for example the maximum gap among the twenty instances is 12.50%, the minimum gap is 2.16%, and the average gap is 6.74%), and all the computational time of these twenty instances is within one second. Therefore, the proposed approximation algorithm is concluded to be effective and efficient in solving the proposed QCSSDP.

Table 3.1 The Results of Computational Experiments for the Approximation Algorithm

Experiment No	Size (bays×cranes)	Lower Bound	AA	Gap [a] (%)
1	16×3	771	818	6.10
2	16×4	600	675	12.50
3	17×3	928	948	2.16
4	17×4	787	862	9.53
5	18×3	951	1030	8.31
6	18×4	714	758	6.16
7	19×3	1016	1071	5.41
8	19×4	762	801	5.12
9	20×3	1188	1292	8.75
10	20×4	778	851	9.38
11	21×3	882	932	5.67
12	21×4	661	699	5.75
13	22×3	1211	1248	3.06
14	22×4	982	1075	9.47
15	23×3	1022	1109	8.51
16	23×4	910	942	3.52
17	24×3	1257	1302	3.58
18	24×4	943	998	5.83
19	25×3	1292	1411	9.21
20	25×4	969	1035	6.81

[a] Gap = (solution obtained from the proposed AA - lower bound)×100/lower bound

3.5 A GENETIC ALGORITHM

This chapter employs a genetic algorithm (GA) to obtain near optimal solutions to the QCSSDP. The procedure of the proposed GA is illustrated in Figure 3.3 and the details of the proposed GA are elaborated as follows.

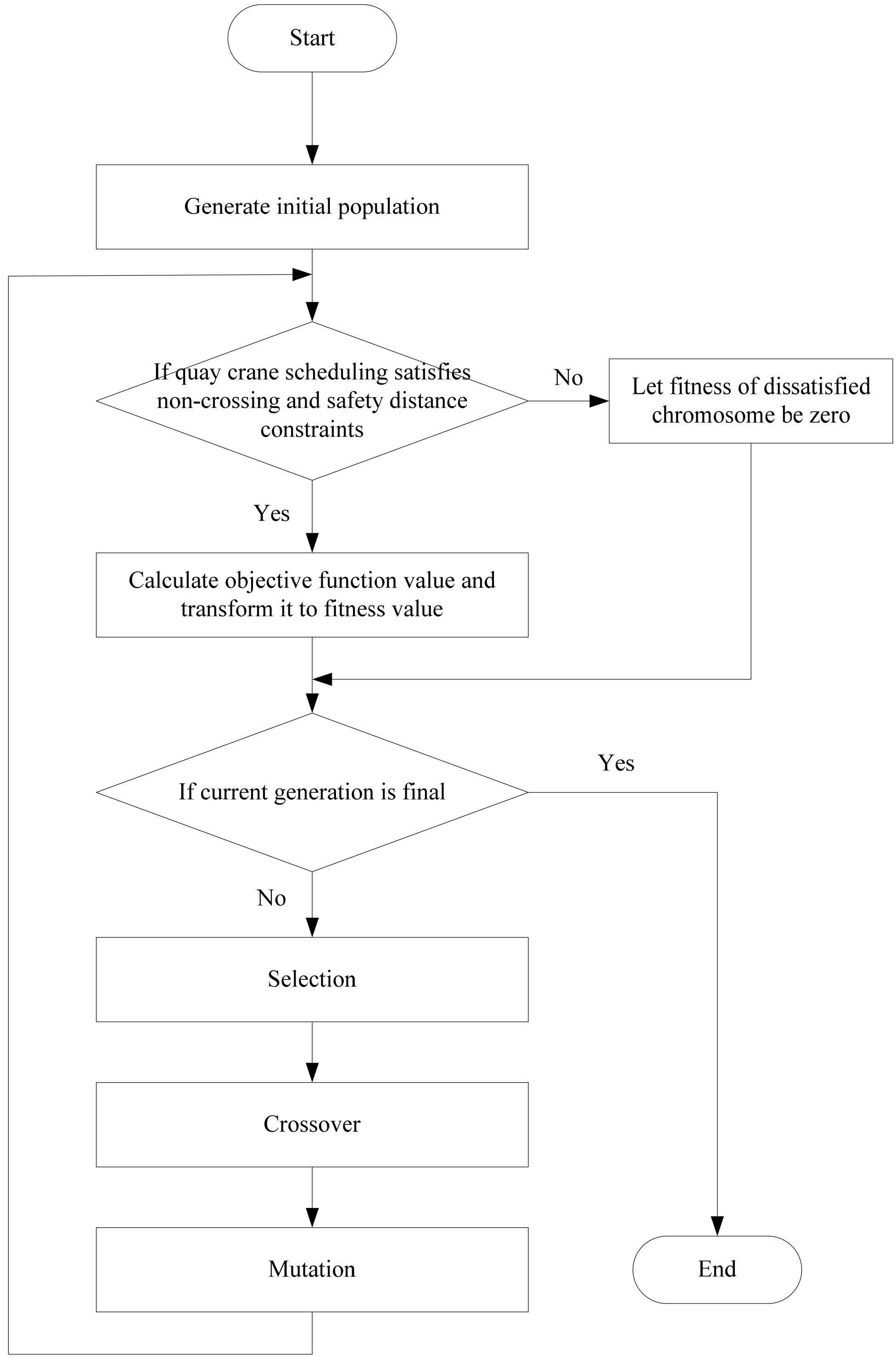

Figure 3.3 The Flowchart of the Proposed GA

3.5.1 Chromosome Representation and Decoding Procedure

The chromosome representation is the same as the one described in Section 2.5.1 (Page 32). Based on the sequence of ship bays 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 Ship Bay b in the chromosome without crossing other quay cranes. If there is only one Quay Crane k available, Ship Bay b is assigned to Quay Crane k. Then, Ship Bay b is deleted from the chromosome, the position of Quay Crane k is set as Ship Bay b, the completion time of Quay Crane k is set as $c_k = c_k + p_b$, and go to Step 5. If there are two quay cranes available that are Quay Crane k and Quay Crane $k+1$, go to Step 2.

***Step 2*:** Compare the completion time of the two available quay cranes to finish their assigned ship bays and assign this ship bay to the quay crane with earlier completion time. Suppose $c_k < c_{k+1}$, and thus assign Ship Bay b to Quay Crane k. Then, Ship Bay b is deleted from the chromosome and the position of Quay Crane k is set as Ship Bay b. Check whether the safety distance between Quay Crane k and Quay Crane $k+1$ is satisfied. If it is satisfied, the completion time of Quay Crane k is set as $c_k = c_k + p_b$ and go to Step 5. If it is not satisfied, the completion time of Quay Crane k is set as $c_k = c_{k+1} + p_b$ and go to Step 5. If their completion time is equal that is $c_k = c_{k+1}$, go to Step 3.

***Step 3*:** Compare the distance between this ship bay and these two available quay cranes and assign this ship bay to the quay crane with the shorter distance. Suppose Quay Crane

k with the shorter distance, and thus assign Ship Bay b to Quay Crane k. Then, Ship Bay b is deleted from the chromosome, the position of Quay Crane k is set as Ship Bay b, the completion time of Quay Crane k is set as $c_k = c_k + p_b$, and go to Step 5. If their distance is equal, go to Step 4.

Step 4**:** Assign this ship bay to the quay crane with the smaller number, and thus assign Ship Bay b to Quay Crane k. Then, Ship Bay b is deleted from the chromosome, the position of Quay Crane k is set as Ship Bay b, the completion time of Quay Crane k is set as $c_k = c_k + p_b$, and go to Step 5.

Step 5**:** If there are unassigned ship bays in the chromosome, go to Step 1; otherwise, go to End.

Figure 3.4 illustrates a numerical example of the above mentioned procedure of constructing a quay crane schedule from a chromosome. There are three quay cranes and twelve ship bays. The current position of Quay Crane 1, Quay Crane 2, and Quay Crane 3 are on Ship Bay 1, Ship Bay 5, and Ship Bay 8 respectively. The current completion time of Quay Crane 1, Quay Crane 2, and Quay Crane 3 is 163, 94, and 157 respectively. The first unassigned ship bay in the chromosome is Ship Bay 7, of which the processing time is 114. The safety distance between two adjacent quay cranes in operation is one ship bay.

Step 1**:** Quay Crane 2 and Quay Crane 3 can handle Ship Bay 7 without crossing other quay cranes. Since there are two quay cranes available, go to Step 2.

Step 2**:** Since $c_2 = 94 < 157 = c_3$, assign Ship Bay 7 to Quay Crane 2. Then, Ship Bay 7 is deleted from the chromosome and the position of Quay Crane 2 is set as Ship Bay 7. Check whether the safety distance between Quay Crane 2 and Quay Crane 3 is satisfied.

Since the current position of Quay Crane 2 and Quay Crane 3 are on Ship Bay 7 and Ship Bay 8 respectively, the safety distance constraint is not satisfied. Therefore, the completion time of Quay Crane 2 is set as $c_2 = c_3 + p_7 = 157 + 114 = 271$, and go to Step 5.

***Step 5*:** Since Ship Bay 12 is the first unassigned ship bay in the chromosome, go to Step 1.

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	163
Quay Crane 2	5	94
Quay Crane 3	8	157

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	163
Quay Crane 2	**7**	**271**
Quay Crane 3	8	157

Figure 3.4 An Illustration of Constructing a Quay Crane Schedule from a Chromosome

3.5.2 Fitness Evaluation

Most of the quay crane schedules obtained from the above mentioned procedure do not violate the safety distance and non-crossing constraints. However, every quay crane schedule must be checked whether it satisfies the safety distance and non-crossing constraints as follows. According to a quay crane schedule constructed from a chromosome, Constraints (3.5), and Constraints (3.6), $Y_{b,b'}$, $\forall 1 \le b,\ b' \le B, b \ne b'$ can be obtained and then the quay crane schedule can be checked whether it satisfies Constraints

(3.7) and Constraints (3.8). If it satisfies Constraints (3.7) and Constraints (3.8), the fitness value of its corresponding chromosome is set to be the reciprocal of its objective function value, as shown in Equation (3.15); otherwise, the fitness value of its corresponding chromosome is zero.

$$Fitness = \frac{1}{\max_{b} C_b} \tag{3.15}$$

3.5.3 Selection, Crossover and Mutation

The roulette wheel selection, the order crossover, and the mutation are elaborated in Section 2.5.2 (Page 36), Section 2.5.3 (Page 36-37), and Section 2.5.4 (Page 37-38), respectively.

3.6 COMPUTATIONAL EXPERIMENTS FOR THE GENETIC ALGORITHM

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

3.6.1 Random Instances with Small Sizes

Six random instances with small sizes are created, and the processing time of a ship bay is randomly generated from a uniform distribution of $U(30,300)$. Assume that the safety distance between two adjacent quay cranes in operation is one ship bay. Then, Constraints (3.8) are reduced to:

$$M(Y_{b,b'}+Y_{b',b}) \geq b + 2(\sum_{k'=1}^{K}\sum_{i'=1}^{B} k'X_{b',k',i'} - \sum_{k=1}^{K}\sum_{i=1}^{B} kX_{b,k,i}) - b' \quad \forall 1 \leq b < b' \leq B \qquad (3.16)$$

The simplified mathematical model of the QCSSDP can be exactly solved by CPLEX when the instance size is small.

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 3.2, the computational time of CPLEX grows exponentially as the instance size increases since the QCSSDP 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 five seconds) when the instance size is small.

Table 3.2 Results of Random Instances with Small Sizes

Experiment No	Size (bays×cranes)	CPLEX		GA	
		Value	CPU (sec)	Value	CPU (sec)
1	8×2	517	28.44	517	5.29
2	8×3	381	105.86	381	5.54
3	9×2	959	376.67	959	5.69
4	9×3	704	1293.88	704	5.09
5	10×2	753	3698.51	753	5.52
6	10×3	586	14685.32	586	5.03

3.6.2 Random Instances with Large Sizes

There are forty random instances with large sizes generated. The processing time of a ship bay is randomly generated from a uniform distribution of $U(30,300)$. Assume that the safety distance between two adjacent quay cranes in operation is one ship bay. 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 1,000 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 by relaxing the safety distance and non-crossing constraints. The mathematical model of the relaxed problem is formulated in Section 2.4 (Page 28-29) that 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 3.3, 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 4.52%, the minimum gap is 0.15%, and the average gap is 1.56%), and all the computational time of these forty instances is short (for example the computational time of these forty instances is all around one hundred and ten seconds). Based on these forty 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. According to the computational experiments with small and large sizes, the proposed GA is concluded to be effective and efficient in solving the proposed QCSSDP.

Furthermore, the lower bound is the objective function value of the optimal solution to the relaxed problem, which does not consider the safety distance and non-crossing constraints. The proposed GA obtains near optimal solution to the original problem. As shown in Table 3.3, the larger gaps between solutions obtained from the proposed GA and lower bounds are observed for smaller container ships with fewer ship bays handled by more quay cranes. The reason for it can be that the safety distance and non-crossing constraints more significantly affect scheduling more quay cranes for smaller container ships with fewer ship bays.

Table 3.3 Results of Random Instances with Large Sizes

Experiment No	Size (bays×cranes)	Lower Bound	GA		Gap [a] (%)
			Value	CPU (sec)	
1	11×3	590	603	102.71	2.20
2	11×4	516	529	103.25	2.52
3	12×3	755	773	102.14	2.38
4	12×4	548	572	104.23	4.38
5	13×3	819	854	103.11	4.27
6	13×4	620	648	102.65	4.52
7	14×3	687	693	105.38	0.87
8	14×4	516	533	104.68	3.29
9	15×3	699	718	105.21	2.72
10	15×4	525	546	105.63	4.00
11	16×3	960	972	102.01	1.25
12	16×4	642	648	102.04	0.93
13	17×3	743	752	104.65	1.21
14	17×4	799	817	104.73	2.25
15	18×3	1114	1122	103.92	0.72
16	18×4	669	677	105.56	1.20
17	19×3	1029	1037	104.88	0.78
18	19×4	741	768	104.20	3.64
19	20×3	1091	1100	106.19	0.82
20	20×4	681	699	105.84	2.64
21	21×3	1497	1501	119.59	0.27
22	21×4	968	988	107.65	2.07
23	22×3	1143	1162	106.40	1.66
24	22×4	886	892	110.43	0.68
25	23×3	1234	1238	109.03	0.32
26	23×4	1002	1013	113.89	1.10
27	24×3	1642	1648	109.28	0.37
28	24×4	1049	1057	116.64	0.76
29	25×3	1261	1268	108.61	0.56
30	25×4	888	892	117.12	0.45
31	26×3	1415	1424	109.69	0.64
32	26×4	1115	1135	113.86	1.79
33	27×3	1317	1321	111.12	0.30
34	27×4	1017	1022	112.75	0.49
35	28×3	1603	1610	111.37	0.44
36	28×4	1196	1200	113.70	0.33
37	29×3	1530	1542	112.50	0.78
38	29×4	1147	1166	113.23	1.66
39	30×3	1365	1367	112.23	0.15
40	30×4	1193	1204	113.70	0.92

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

3.7 SUMMARY

This chapter provides a mixed integer programming model for the proposed QCSSDP, proves that the QCSSDP is NP-complete, and proposes an approximation algorithm and a genetic algorithm to obtain near optimal solutions for the QCSSDP. Worst-case analysis for the AA is performed and computational experiments are conducted to examine the proposed model, AA and GA. The results show that both the proposed AA and GA are effective and efficient in solving the QCSSDP. In addition, in practical quay crane scheduling, the number of quay cranes ranges from two to four, and the number of ship bays ranges from ten to twenty-five. Based on the computational experiments, the proposed AA and GA can be considered as appropriate approaches to scheduling quay cranes in port container terminals to enhance their efficiency.

CHAPTER 4 QUAY CRANE SCHEDULING WITH HANDLING PRIORITY AND NON-CROSSING CONSTRAINTS

As discussed in Chapter 1, different ship bay has different handling priority in practice. Based on Chapter 2, this chapter investigates the Quay Crane Scheduling with Handling Priority and non-crossing constraints Problem (QCSHPP).

4.1 MODEL FORMULATION

This chapter proposes a mixed integer programming model for the QCSHPP. According to the configuration of container ships, one single container ship is divided into ship bays. Figure 1.2 shows that both quay cranes and ship bays are arranged in an increasing order from the front to the tail of the container ship. The following assumptions are imposed in formulating the QCSHPP:

1. Every ship bay has its own handling priority.
2. Quay cranes are operated on the same tracks and thus cannot cross over each other.
3. Only one quay crane can work on a ship bay at a time until it completes the ship bay.
4. Compared with the processing time of a ship bay by a quay crane, the travel time of a quay crane between two ship bays is small and hence it is not considered.

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

Parameters:

K the number of quay cranes;

B the number of ship bays;

p_b the processing time of ship bay b by a quay crane ($1 \le b \le B$);

α_b the weight of ship bay b ($1 \le b \le B$);

M a sufficiently large positive constant number;

Decision variables:

$X_{b,k,i}$ 1, if ship bay b is handled as the ith ship bay by quay crane k; 0, otherwise ($1 \le b \le B, 1 \le k \le K, 1 \le i \le B$);

$Y_{b,b'}$ 1, if ship bay b finishes no later than ship bay b' starts; 0, otherwise ($1 \le b,\ b' \le B, b \ne b'$);

C_b the completion time of ship bay b ($1 \le b \le B$).

The QCSHPP can be formulated as follows:

Minimize:

$$\sum_{b=1}^{B} \alpha_b C_b \tag{4.1}$$

Subject to:

$$\sum_{k=1}^{K} \sum_{i=1}^{B} X_{b,k,i} = 1 \quad \forall 1 \le b \le B \tag{4.2}$$

$$\sum_{b=1}^{B} X_{b,k,i} \le 1 \quad \forall 1 \le k \le K, \forall 1 \le i \le B \tag{4.3}$$

$$C_b \ge \sum_{b'=1}^{B} \sum_{k=1}^{K} \sum_{i'=1}^{i} p_{b'} X_{b',k,i'} X_{b,k,i} \quad \forall 1 \le b \le B, \forall 1 \le i \le B \tag{4.4}$$

$$C_b - (C_{b'} - p_{b'}) + Y_{b,b'} M > 0 \quad \forall 1 \le b,\ b' \le B, b \ne b' \tag{4.5}$$

$$C_b - (C_{b'} - p_{b'}) - (1 - Y_{b,b'})M \leq 0 \quad \forall 1 \leq b,\ b' \leq B, b \neq b' \tag{4.6}$$

$$M(Y_{b,b'} + Y_{b',b}) \geq \sum_{k=1}^{K}\sum_{i=1}^{B} kX_{b,k,i} - \sum_{k'=1}^{K}\sum_{i'=1}^{B} k'X_{b',k',i'} + 1 \quad \forall 1 \leq b < b' \leq B \tag{4.7}$$

$$X_{b,k,i}, Y_{b,b'} = 0 \text{ or } 1 \quad \forall 1 \leq b,\ b' \leq B, b \neq b', \forall 1 \leq k \leq K, \forall 1 \leq i \leq B \tag{4.8}$$

The objective function (4.1) minimizes the sum of the weighted completion time of every ship bay. Constraints (4.2) ensure that every ship bay must be handled only by one quay crane. Constraints (4.3) enforce that every quay crane handles up to one ship bay at a time. Constraints (4.4) define the properties of decision variables C_b. Constraints (4.5) and (4.6) define the properties of decision variables $Y_{b,b'}$: Constraints (4.5) indicate that $Y_{b,b'} = 1$ if $C_b \leq C_{b'} - p_{b'}$, which means $Y_{b,b'} = 1$ when ship bay b finishes no later than ship bay b' starts; Constraints (4.6) indicate that $Y_{b,b'} = 0$ if $C_b > C_{b'} - p_{b'}$, which means $Y_{b,b'} = 0$ when ship bay b finishes after ship bay b' starts. Finally, the crossing between quay cranes can be avoided by imposing Constraints (4.7). Suppose that ship bays b and b' are performed simultaneously and $b < b'$, then this means that $Y_{b,b'} + Y_{b',b} = 0$. Note that both quay cranes and ship bays are arranged in an increasing order from the front to the tail of the container ship. Thus, if quay crane k handles ship bay b and quay crane k' handles ship bay b', then $k + 1 \leq k'$.

4.2 PROOF OF NP-COMPLETENESS

This chapter discusses the computational complexity of QCSHPP to justify why the heuristic algorithms are adopted. The optimization version of the QCSHPP is presented in Section 4.1 and its decision version is defined as follows:

Parameter:

Z^+ the set of positive integer.

Instance: There are B ship bays and K quay cranes. The processing time of ship bay b by a quay crane is $p_b \in Z^+$ ($1 \le b \le B$). There is a given number $E \in Z^+$.

Question: Is there a quay crane schedule for these K quay cranes handling these B ship bays such that the non-crossing constraints are satisfied and $\sum_{b=1}^{B} \alpha_b C_b \le E$?

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

Theorem 4.1: QCSHPP is NP-complete.

Proof:

***Step 1*: Showing that the QCSHPP is in NP.**

If a quay crane schedule for the QCSHPP is given, its feasibility can be checked in polynomial time. Checking whether the quay crane schedule satisfies the non-crossing constraints can be done in $O(B^2)$ time. Checking whether $\sum_{b=1}^{B} \alpha_b C_b \le E$ can be done in $O(B)$ time. Therefore, the QCSHPP 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 in Section 2.2 (Page 21).

***Step 3*: Constructing a transformation from the PARTITION to the QCSHPP.**

The PARTITION is transformed to the QCSHPP as follows. A QCSHPP instance corresponding to an arbitrary PARTITION instance has 2 quay cranes and $B+2$ ship bays; the given number E is set as $\sum_{1\le b\le b'\le B} s_b s_{b'} + \frac{3}{4}D^2$; the following Equations (4.9) and (4.10) indicate the processing time of each ship bay which means the processing time of Ship Bay 1 and Ship Bay $B+2$ is set as $\frac{D}{2}$ and the processing time of Ship Bay 2 to Ship Bay $B+1$ is set as s_1 to s_B respectively. Figure 4.1 illustrates this transformation, which shows 2 quay cranes, $B+2$ ship bays, and the processing time of each ship bay. Equations (4.11) denote the weight of every ship bay.

$$p_1 = p_{B+2} = \frac{D}{2} \tag{4.9}$$

$$p_{b+1} = s_b \quad \forall 1 \le b \le B \tag{4.10}$$

$$\alpha_b = p_b \quad \forall 1 \le b \le B+2 \tag{4.11}$$

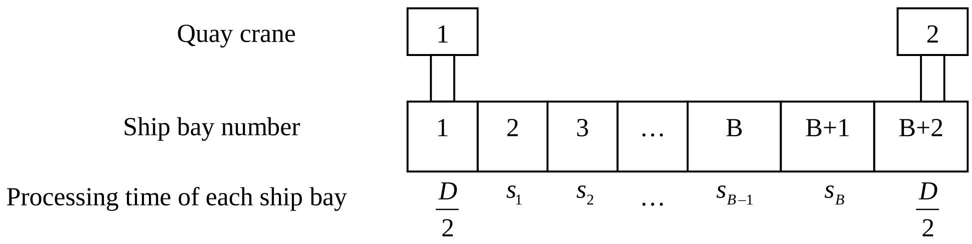

Figure 4.1 The Illustration of the Transformation from the PARTITION to the QCSHPP

Then, it must be proved that the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_b \in S_1} s_b = \sum_{s_b \in S_2} s_b = \frac{D}{2}$ if and only if all the $B+2$ ship bays can be completed by 2 quay cranes with satisfying the non-crossing constraints and $\sum_{b=1}^{B+2} \alpha_b C_b \le E$.

First, suppose that the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_b \in S_1} s_b = \sum_{s_b \in S_2} s_b = \frac{D}{2}$. Then 2 quay cranes can be scheduled with satisfying the non-crossing constraints as follows: Quay Crane 1 handles all the Ship Bays $b+1$, where $s_b \in S_1$ and then Ship Bay 1; Quay Crane 2 handles Ship Bay $B+2$ and then all the Ship Bays $b+1$, where $s_b \in S_2$. Let $e = \sum_{s_b \in S_1} s_b - \frac{D}{2}$. Since $\alpha_b = p_b \;\; \forall 1 \le b \le B+2$, the value of $\sum_{b=1}^{B+2} \alpha_b C_b$ is not influenced by the ordering of the ship bays handled by the quay cranes and only depends on the choice of S_1 (Lenstra et al., 1977). Thus, the objective function value $\sum_{b=1}^{B+2} \alpha_b C_b$ of this schedule can be expressed as Equation (4.12).

$$\begin{aligned}\sum_{b=1}^{B+2}\alpha_b C_b &= \sum_{1\le b\le b'\le B} s_b s_{b'} - \left(\sum_{s_b\in S_1} s_b\right)\left(\sum_{s_b\in S_2} s_b\right) + \left(\sum_{s_b\in S_1} s_b + \frac{D}{2}\right)\frac{D}{2} + \left(\sum_{s_b\in S_2} s_b + \frac{D}{2}\right)\frac{D}{2} \\ &= \sum_{1\le b\le b'\le B} s_b s_{b'} - (\frac{D}{2}+e)(\frac{D}{2}-e) + D^2 \\ &= \sum_{1\le b\le b'\le B} s_b s_{b'} + \frac{3}{4}D^2 + e^2 \\ &= E + e^2\end{aligned} \tag{4.12}$$

As known $\sum_{s_b\in S_1} s_b = \sum_{s_b\in S_2} s_b = \frac{D}{2}$, therefore $e = \sum_{s_b\in S_1} s_b - \frac{D}{2} = 0$ and $\sum_{b=1}^{B+2}\alpha_b C_b = E$. Hence, if the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_b\in S_1} s_b = \sum_{s_b\in S_2} s_b = \frac{D}{2}$, all the $B+2$ ship bays can be completed by 2 quay cranes with satisfying the non-crossing constraints and $\sum_{b=1}^{B+2}\alpha_b C_b \le E$.

Conversely, suppose all the $B+2$ ship bays can be completed by 2 quay cranes with satisfying the non-crossing constraints and $\sum_{b=1}^{B+2}\alpha_b C_b \le E$. In terms of assigning Ship Bay 1 and Ship Bay $B+2$, there are four possible cases: both Ship Bay 1 and Ship Bay $B+2$ are assigned to Quay Crane 1 or Quay Crane 2; Ship Bay 1 is assigned to Quay Crane 2 and Ship Bay $B+2$ is assigned to Quay Crane 1; Ship Bay 1 is assigned to Quay Crane 1 and Ship Bay $B+2$ is assigned to Quay Crane 2. Due to the non-crossing constraints, $\sum_{b=1}^{B+2}\alpha_b C_b$ obtained from any quay crane schedule $\ge E + e^2$. As known there is at least one quay crane schedule with $\sum_{b=1}^{B+2}\alpha_b C_b \le E$ and satisfying the non-crossing constraints, therefore $e = 0$ which means $\sum_{s_b \in S_1} s_b = \frac{D}{2}$. Hence, if all the $B+2$ ship bays can be completed by 2 quay cranes with satisfying the non-crossing constraints and $\sum_{b=1}^{B+2} \alpha_b C_b \le E$, the set S can be partitioned into two disjoint subsets S_1 and S_2 such that $\sum_{s_b \in S_1} s_b = \sum_{s_b \in S_2} s_b = \frac{D}{2}$.

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

The above mentioned transformation can be done in $O(B)$ time.

Therefore, $\text{PARTITION} \propto \text{QCSHPP}$, and the Theorem 4.1 is proved.

4.3 AN APPROXIMATION ALGORITHM

As proved in the previous section, QCSHPP is NP-complete, and thus there exists no polynomial time algorithm for the exact solution to the QCSHPP unless P=NP. This section proposes an Approximation Algorithm (AA) to obtain its near optimal solution which is elaborated as follows.

Lemma 4.1: For a single quay crane, the sum of the weighted completion time of every ship bay is optimized if the ship bays are handled in a non-increasing order of α_b / p_b .

Proof:

For any sequencing of the ship bays, $\alpha_i / p_i < \alpha_{i+1} / p_{i+1}$ implies that interchanging the i th and $i+1$ st ship bays will reduce the sum of the weighted completion time of every ship bay by $\alpha_{i+1} p_i - \alpha_i p_{i+1} > 0$. If starting with an optimal solution, there can be no profitable interchanges; and thus the ship bay with the highest ratio will necessarily be processed first, that with the second highest ratio next, and so forth in a non-increasing order of α_b / p_b. Hence the Lemma 4.1 is proved (Smith, 1956).

Approximation Algorithm: assign adjacent ship bays, $b_{k-1}+1,\ b_{k-1}+2,\ ...,\ b_k-1,\ b_k$, to quay crane k ($\forall 1 \le k \le K$). Note that $b_0 = 0$ and $b_K = B$. Then, quay crane k handles the assigned ship bays in a non-increasing order of α_b / p_b. Furthermore, a dynamic programming algorithm is proposed to determine the best partition points, $b_1,\ b_2,\ ...,\ b_{K-2},\ b_{K-1}$, which minimizes the sum of the weighted completion time of every ship bay.

Parameters:

$WC[k,\ b]$ the minimum sum of the weighted completion time of ship bays $1,\ 2,\ ...,\ b-1,\ b$ when they are assigned to quay cranes $1,\ 2,\ ...,\ k-1,\ k$ in the above mentioned adjacent manner;

$TC[b_1,\ b_2] = \sum_{b=b_1}^{b_2} \alpha_b C_b$ the sum of the weighted completion time of ship bays $b_1,\ b_1+1,\ ...,\ b_2-1,\ b_2$ when they are handled by a quay crane in a non-increasing order of α_b / p_b.

Dynamic programming equations for determining the best partition points, $b_1,\ b_2,\ ...,\ b_{K-2},\ b_{K-1}$, are as follows:

$$WC[1,\ b] = TC[1,\ b] \quad \forall 1 \le b \le B \qquad (4.13)$$

$$WC[k,\ b] = \min_{k-1 \le b_{k-1} \le b-1} \left\{ WC[k-1,\ b_{k-1}] + TC[b_{k-1}+1,\ b] \right\} \quad \forall 2 \le k \le K,\ \forall k \le b \le B \qquad (4.14)$$

Figure 4.2 shows a numerical example of the approximation algorithm in which there are two quay cranes and four ship bays. According to Equation (4.13) and note that the assigned ship bays are handled by the quay crane in a non-increasing order of α_b / p_b, $WC[1,\ 1] = TC[1,\ 1] = 348$, $WC[1,\ 2] = TC[1,\ 2] = 2372$, $WC[1,\ 3] = TC[1,\ 3] = 2750$, and $WC[1,\ 4] = TC[1,\ 4] = 4795$. According to Equation (4.14),

$$WC[2,\ 4] = \min\ \{WC[1,\ 1] + TC[2,\ 4],\ WC[1,\ 2] + TC[3,\ 4],\ WC[1,\ 3] + TC[4,\ 4]\}$$
$$= \min\ \{348 + 3404,\ 2372 + 828,\ 2750 + 625\} = \min\ \{3752,\ 3200,\ 3375\} = 3200$$

Therefore, Quay Crane 1 handles Ship Bay 2 and then Ship Bay 1; Quay Crane 2 handles Ship Bay 4 and then Ship Bay 3. In this quay crane schedule, $\sum_{b=1}^{4} \alpha_b C_b = 3200$.

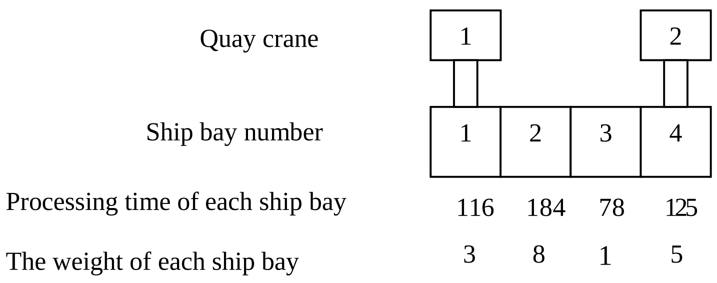

Figure 4.2 A Numerical Example of the Approximation Algorithm

The worst-case analysis for the proposed approximation algorithm is performed as follows.

Parameters:

Z the objective function value of the solution to the QCSHPP obtained by the proposed approximation algorithm;

Z_1^* the objective function value of handling B ship bays in a non-increasing order of α_b / p_b by a single quay crane;

Z_K^* the objective function value of the optimal solution to handling B ship bays by K quay cranes without considering the non-crossing constraints;

Z^* the objective function value of the optimal solution to the QCSHPP.

Lemma 4.2: $Z^* \geq Z_K^* \geq \frac{K+B}{K(B+1)} Z_1^*$

Proof:

As proved in Eastman et al. (1964), $Z_K^* \geq \frac{K+B}{K(B+1)} Z_1^*$. The QCSHPP considers the non-crossing constraints, therefore $Z^* \geq Z_K^*$. The Lemma 4.2 is proved.

Theorem 4.2: $Z/Z^* \leq \frac{K(B+1)}{K+B}$

Proof:

When B ship bays are handled by K quay cranes using the proposed approximation algorithm, ship bay b ($\forall 1 \leq b \leq B$) is completed no later than it is completed in the schedule of handling B ship bays in a non-increasing order of α_b / p_b by a single quay crane. Therefore, $Z \le Z_1^*$. A tight instance of $Z = Z_1^*$ has 2 quay cranes and 3 ship bays. The processing time and weight of each ship bay are $p_1 = 100,\ \alpha_1 = 1;\ p_2 = 120,\ \alpha_2 = 0;\ p_3 = 270,\ \alpha_3 = 0$. In this case, $Z = Z_1^* = 100$. Based on Lemma 4.2, $Z^* \ge \frac{K+B}{K(B+1)} Z_1^* \ge \frac{K+B}{K(B+1)} Z$. Thus, the Theorem 4.2 is proved.

4.4 COMPUTATIONAL EXPERIMENTS

A series of computational experiments are conducted to examine the performance of the proposed model and approximation algorithm. The approximation algorithm is coded in C++ and executed in a Pentium IV 1.7GHz PC with 256MB RAM.

There are forty random instances generated in which the processing time of a ship bay is randomly generated from a uniform distribution of $U(30, 300)$ and the weight of a ship bay is randomly generated from a uniform distribution of $U(1,10)$.

In order to evaluate the performance of the proposed approximation algorithm in solving the instance and according to Lemma 4.2, $\frac{K+B}{K(B+1)} Z_1^*$ is adopted as the lower bound corresponding to the instance.

As observed in Table 4.1, the gaps between solutions obtained from the proposed approximation algorithm and lower bounds are all small (for example the maximum gap among the forty instances is 16.45%, the minimum gap is 7.24%, and the average gap is

12.14%), and all the computational time of these forty instances is within one second. Therefore, the proposed approximation algorithm is concluded to be effective and efficient in solving the proposed QCSHPP.

Table 4.1 The Results of Computational Experiments

Experiment No	Size (bays×cranes)	Lower Bound	AA	Gap[1] (%)
1	11×3	15126	16707	10.45
2	12×3	22158	24878	12.28
3	13×3	19218	21217	10.40
4	14×3	17550	19052	8.56
5	15×3	21502	23290	8.32
6	16×3	29659	32991	11.23
7	17×3	30215	32409	7.26
8	18×3	32817	35193	7.24
9	19×3	37686	41078	9.00
10	20×3	40313	44077	9.34
11	11×4	13657	15484	13.38
12	12×4	16926	19115	12.93
13	13×4	24183	26417	9.24
14	14×4	29829	33004	10.64
15	15×4	19398	22198	14.43
16	16×4	25328	29125	14.99
17	17×4	22295	25327	13.60
18	18×4	25368	28638	12.89
19	19×4	26914	30760	14.29
20	20×4	31162	34594	11.01
21	21×5	29312	34134	16.45
22	22×5	31752	35969	13.28
23	23×5	36756	42405	15.37
24	24×5	47536	52616	10.69
25	25×5	43101	48568	12.68
26	26×5	70507	76101	7.93
27	27×5	34727	39800	14.61
28	28×5	61564	68031	10.50
29	29×5	40510	46147	13.92
30	30×5	48162	53071	10.19
31	21×6	26738	30746	14.99
32	22×6	33091	38101	15.14
33	23×6	43494	49497	13.80
34	24×6	34208	39138	14.41
35	25×6	32213	37408	16.13
36	26×6	42839	49118	14.66
37	27×6	41107	46015	11.94
38	28×6	45470	51469	13.19
39	29×6	48933	56062	14.57
40	30×6	59505	65260	9.67

[1]Gap=(solution obtained from the proposed AA-lower bound)×100/lower bound

Furthermore, in practical quay crane scheduling, the number of quay cranes ranges from two to five, and the number of ship bays ranges from ten to twenty-five. The proposed approximation algorithm may be considered as a suitable approach in scheduling quay cranes in port container terminals taking into account handling priority of every ship bay and to enhance the efficiency of port operations.

4.5 SUMMARY

This chapter provides a mixed integer programming model for the proposed QCSHPP, proves that the QCSHPP is NP-complete, and proposes an approximation algorithm to obtain near optimal solution for the QCSHPP. In addition, the worst-case analysis for the approximation algorithm is performed and computational experiments are conducted to examine the proposed model and approximation algorithm. The results show that the proposed approximation algorithm is effective and efficient in solving the QCSHPP.

CHAPTER 5 INTEGRATED DISCRETE BERTH ALLOCATION AND QUAY CRANE SCHEDULING

As discussed in Chapter 1, the handling time of a container ship at a berth is related to its quay crane schedule. To consider the relationship between berth allocation and quay crane scheduling, this chapter studies the Integrated discrete Berth Allocation and Quay Crane Scheduling Problem (IBAQCSP).

5.1 MODEL FORMULATION

This chapter proposes a mixed integer programming model which includes two parts for the IBAQCSP. The first part is a dynamic berth allocation model based on discrete locations. The handling time of every container ship at each berth in the first part is obtained from the second part which is a quay crane scheduling with non-crossing constraints model. The following assumptions are imposed in formulating the first part:

1. Each berth can handle only one container ship at a time until the container ship is completed.
2. There are no physical or technical restrictions such as container ship and berth length, and container ship draft and water depth.
3. The handling time of a container ship at a berth depends on the quay crane schedule for the container ship.
4. Container ships can arrive at a port container terminal during the planning horizon and every container ship cannot be handled before it arrives.

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

Parameters:

Q the number of berths;

S the number of container ships;

a_s the arrival time of container ship s ($1 \le s \le S$);

$H_{s,q}$ the handling time of container ship s at berth q ($1 \le s \le S$, $1 \le q \le Q$);

M a sufficiently large positive number (constant);

Decision variables:

$x_{s,q}$ 1, if container ship s is assigned to berth q; 0, otherwise ($1 \le s \le S$, $1 \le q \le Q$);

y_s the berthing time of container ship s ($1 \le s \le S$);

$z_{s,s'}$ 1, if container ship s finishes no later than container ship s' starts; 0, otherwise ($1 \le s,\ s' \le S,\ s \ne s'$);

c_s the completion time of container ship s ($1 \le s \le S$).

The first part can be formulated as follows:

Minimize:

$$\max_{s} \ c_s \tag{5.1}$$

Subject to:

$$\sum_{q=1}^{Q} x_{s,q} = 1 \quad \forall 1 \le s \le S \tag{5.2}$$

$$c_s = y_s + \sum_{q=1}^{Q} H_{s,q} x_{s,q} \quad \forall 1 \le s \le S \tag{5.3}$$

$$c_s - y_{s'} + z_{s,s'}M > 0 \quad \forall 1 \le s,\ s' \le S,\ s \ne s' \tag{5.4}$$

$$c_s - y_{s'} - (1 - z_{s,s'})M \le 0 \quad \forall 1 \le s,\ s' \le S,\ s \ne s' \tag{5.5}$$

$$M\left[(x_{s,q} - 1) + (x_{s',q} - 1)\right] \le z_{s,s'} + z_{s',s} - 1 \quad \forall 1 \le s,\ s' \le S,\ s \ne s',\ \forall 1 \le q \le Q \tag{5.6}$$

$$y_s \ge a_s \quad \forall 1 \le s \le S \tag{5.7}$$

$$x_{s,q},\ z_{s,s'} = 0 \text{ or } 1 \quad \forall 1 \le s,\ s' \le S,\ s \ne s',\ \forall 1 \le q \le Q \tag{5.8}$$

The objective function (5.1) of the first part minimizes the makespan of handling all container ships, which is the latest completion time among all container ships. Constraints (5.2) ensure that every container ship must be allocated only to one berth. Constraints (5.3) define the property of the decision variable c_s. Constraints (5.4) and (5.5) define the properties of decision variables $z_{s,s'}$: Constraints (5.4) indicate that $z_{s,s'} = 1$ if $c_s \le y_{s'}$, which means $z_{s,s'} = 1$ when container ship s finishes no later than container ship s' starts; Constraints (5.5) indicate that $z_{s,s'} = 0$ if $c_s > y_{s'}$, which means $z_{s,s'} = 0$ when container ship s finishes after container ship s' starts. Constraints (5.6) guarantee that any two container ships do not conflict with each other in terms of the berthing time. Suppose that both container ship s and container ship s' are assigned to the same berth q, then $0 \le z_{s,s'} + z_{s',s} - 1$ which means $z_{s,s'} + z_{s',s} \ne 0$. Therefore, if two container ships are allocated to the same berth, Constraints (5.6) assure that they are not handled simultaneously. Constraints (5.7) enforce that every container ship cannot berth before it arrives.

The handling time of container ship s at berth q, $H_{s,q}$ can be obtained from the second part. The following assumptions are imposed in formulating the second part:

1. The number of quay cranes at each berth is fixed.
2. Quay cranes are operated on the same tracks and thus cannot cross over each other.
3. Only one quay crane can work on a ship bay at a time until it completes the ship bay.
4. Compared with the processing time of a ship bay by a quay crane, the travel time of a quay crane between two ship bays is small and hence it is not considered.

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

Parameters:

K_q the number of quay cranes at berth q ($1 \le q \le Q$);

B_s the number of ship bays in container ship s ($1 \le s \le S$);

P_{b_s} the processing time of ship bay b in container ship s by a quay crane ($1 \le b_s \le B_s$);

Decision variables:

X_{b_s,k_q} 1, if ship bay b in container ship s is handled by quay crane k at berth q; 0, otherwise ($1 \le b_s \le B_s, 1 \le k_q \le K_q$);

Y_{b_s,b'_s} 1, if ship bay b in container ship s finishes no later than ship bay b' in container ship s starts; 0, otherwise ($1 \le b_s,\ b'_s \le B_s, b_s \ne b'_s$);

C_{b_s} the completion time of ship bay b in container ship s ($1 \le b_s \le B_s$).

The second part can be formulated as follows:

Minimize:

$$H_{s,q} \tag{5.9}$$

Subject to:

$$H_{s,q} \geq C_{b_s} \quad \forall 1 \leq b_s \leq B_s \tag{5.10}$$

$$C_{b_s} \geq P_{b_s} \quad \forall 1 \leq b_s \leq B_s \tag{5.11}$$

$$\sum_{k_q=1}^{K_q} X_{b_s,k_q} = 1 \quad \forall 1 \leq b_s \leq B_s \tag{5.12}$$

$$C_{b_s} - (C_{b'_s} - P_{b'_s}) + Y_{b_s,b'_s} M > 0 \quad \forall 1 \leq b_s,\ b'_s \leq B_s,\ b_s \neq b'_s \tag{5.13}$$

$$C_{b_s} - (C_{b'_s} - P_{b'_s}) - (1 - Y_{b_s,b'_s}) M \leq 0 \quad \forall 1 \leq b_s,\ b'_s \leq B_s,\ b_s \neq b'_s \tag{5.14}$$

$$M(Y_{b_s,b'_s} + Y_{b'_s,b_s}) \geq \sum_{k_q=1}^{K_q} k_q X_{b_s,k_q} - \sum_{k'_q=1}^{K_q} k'_q X_{b'_s,k'_q} + 1 \quad \forall 1 \leq b_s < b'_s \leq B_s \tag{5.15}$$

$$X_{b_s,k_q},\ Y_{b_s,b'_s} = 0 \text{ or } 1 \quad \forall 1 \leq b_s,\ b'_s \leq B_s,\ b_s \neq b'_s,\ \forall 1 \leq k_q \leq K_q \tag{5.16}$$

The objective function (5.9) of the second part minimizes the handling time of container ship s at berth q. Constraints (5.10) define $H_{s,q}$ as the latest completion time among all ship bays of container ship s when it is handled at berth q. Constraints (5.11) define the property of the decision variable C_{b_s}. Constraints (5.12) ensure that every ship bay of container ship s must be performed only by one quay crane at berth q. Constraints (5.13) and (5.14) define the properties of decision variables Y_{b_s,b'_s}. Constraints (5.13) indicate that $Y_{b_s,b'_s} = 1$ if $C_{b_s} \leq C_{b'_s} - P_{b'_s}$, which means $Y_{b_s,b'_s} = 1$ when ship bay b of container ship s finishes no later than ship bay b' of container ship s starts. Constraints (5.14) indicate that $Y_{b_s,b'_s} = 0$ if $C_{b_s} > C_{b'_s} - P_{b'_s}$, which means $Y_{b_s,b'_s} = 0$ when ship bay b of container ship

s finishes after ship bay b' of container ship s starts. Finally, the crossing between quay cranes at berth q can be avoided by imposing Constraints (5.15). Suppose that ship bays b and b' of container ship s are performed simultaneously and $b_s < b'_s$, then this means that $Y_{b_s,b'_s} + Y_{b'_s,b_s} = 0$. Note that both quay cranes at a berth and ship bays of a container ship are arranged in an increasing order from the front to the tail of the container ship (refer to Figure 1.2 in Page 4). Thus, if quay crane k at berth q handles ship bay b of container ship s and quay crane k' at berth q handles ship bay b' of container ship s, then $k_q + 1 \le k'_q$.

5.2 PROOF OF NP-COMPLETENESS

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

Theorem 5.1: IBAQCSP is NP-complete.

Proof:

Restrict the number of berths $Q = 1$, the number of container ships $S = 1$, and the arrival time of the single container ship $a = 0$. Then, the resulting restricted IBAQCSP is identical to the Quay Crane Scheduling with Non-Crossing constraints Problem

(QCSNCP). The QCSNCP is proved to be NP-complete in Section 2.2. Hence the Theorem 5.1 is proved.

5.3 A GENETIC ALGORITHM

As proved in the previous section, IBAQCSP is NP-complete, and thus there exists no polynomial time algorithm for the exact solution to IBAQCSP unless P=NP. This section employs a Genetic Algorithm (GA) to obtain its near optimal solutions. The procedure of the proposed GA is illustrated in Figure 5.1 and the details of the proposed GA are elaborated as follows.

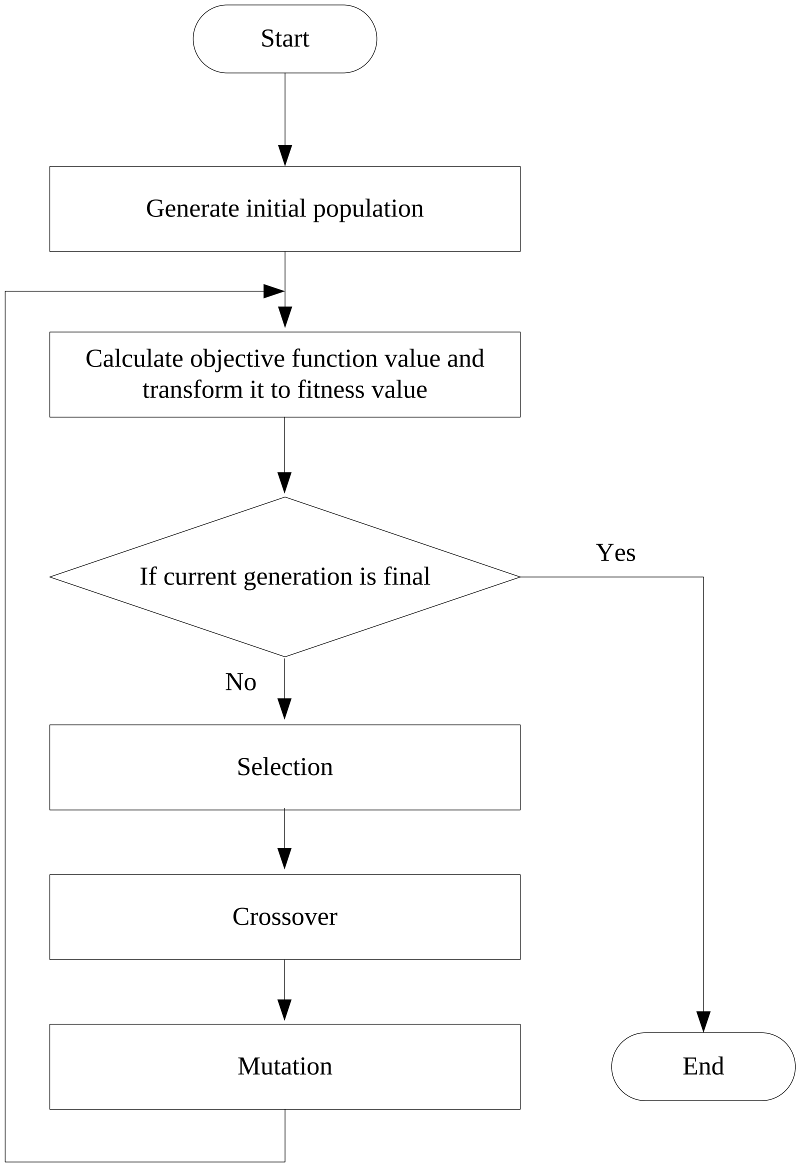

Figure 5.1 The Flowchart of the Proposed GA

5.3.1 Chromosome Representation and Decoding Procedure

In this chapter, berths are numbered in an increasing order from the left to the right (as illustrated in Figure 1.1 in Page 3) and container ships are numbered according to their arrival time. If a container ship arrives earlier, its number is smaller. A chromosome of the GA represents a sequence of container ships. Figure 5.2 provides a sample chromosome, in which a gene is a container ship number. Based on the sequence of container ships represented by the chromosome, a berth allocation can be constructed using the following procedure.

Gene: container ship number 1-9

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

Figure 5.2 An Illustration of the Chromosome Representation

Step 1**:** Based on the current completion time of each berth to finish its already allocated container ships and the arrival time of the first unassigned container ship in the chromosome, determine which berths can handle this container ship immediately. If there is no idle berth when this container ship arrives, go to Step 2.1. Otherwise, go to Step 3.1.

Step 2.1**:** If there is only one berth with the earliest completion time, this container ship has to wait and is allocated to this berth. Then, this container ship is deleted from the chromosome, the completion time of the assigned berth is updated, and go to Step 4. If there are two or more berths with the earliest completion time, go to Step 2.2.

Step 2.2**:** If there is only one berth with the largest number of quay cranes, this container ship has to wait and is allocated to this berth. Then, this container ship is deleted from the chromosome, the completion time of the assigned berth is updated, and go to Step 4. If there are two or more berths with the largest number of quay cranes, go to Step 2.3.

Step 2.3**:** This container ship has to wait and is allocated to the berth with the smallest number. Then, this container ship is deleted from the chromosome, the completion time of the assigned berth is updated, and go to Step 4.

Step 3.1**:** If there is only one idle berth, this container ship is allocated to this berth. Then, this container ship is deleted from the chromosome, the completion time of the assigned berth is updated, and go to Step 4. If there are two or more idle berths, go to Step 3.2.

Step 3.2**:** If there is only one idle berth with the largest number of quay cranes, this container ship is allocated to this berth. Then, this container ship is deleted from the chromosome, the completion time of the assigned berth is updated, and go to Step 4. If there are two or more idle berths with the largest number of quay cranes, go to Step 3.3.

Step 3.3**:** This container ship is allocated to the idle berth with the smallest number. Then, this container ship is deleted from the chromosome, the completion time of the assigned berth is updated, and go to Step 4.

Step 4**:** If there are unassigned container ships in the chromosome, go to Step 1; otherwise, go to End.

Note that when updating the completion time of the assigned berth in the aforementioned procedure, the handling time of this container ship at the assigned berth is needed. This time is obtained from an approximation algorithm for the quay crane scheduling with non-crossing constraints problem which is elaborated later.

Figure 5.3 shows a numerical example of the aforementioned procedure which is to construct a berth allocation from a chromosome. The number of quay cranes at each berth and the current completion time of each berth are indicated in Figure 5.3. The first unassigned container ship in the chromosome is Container Ship 3 whose arrival time is 714.

Step 1**:** When Container Ship 3 arrives, Berth 1, Berth 2, and Berth 4 are idle. Therefore, go to Step 3.1.

Step 3.1**:** Since there are three idle berths, go to Step 3.2.

Step 3.2**:** Since the number of quay cranes at both Berth 2 and Berth 4 is 3, go to Step 3.3.

Step 3.3**:** Container Ship 3 is allocated to Berth 2. Then, Container Ship 3 is deleted from the chromosome. The handling time of Container Ship 3 at Berth 2 is 595 (that is obtained from an approximation algorithm for the quay crane scheduling with non-crossing constraints problem which is elaborated later). Thus, the completion time of Berth 2 = the arrival time of Container Ship 3 + the handling time of Container Ship 3 at Berth 2 = 714 + 595 = 1309. Go to Step 4.

Step 4**:** Since Container Ship 5 is the first unassigned container ship in the chromosome, go to Step 1.

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

	The Number of Quay Cranes	Completion Time of Berth
Berth 1	2	689
Berth 2	3	709
Berth 3	4	1021
Berth 4	3	615

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

	The Number of Quay Cranes	Completion Time of Berth
Berth 1	2	689
Berth 2	3	**1309**
Berth 3	4	1021
Berth 4	3	615

Figure 5.3 An Illustration of Constructing a Berth Allocation from a Chromosome

5.3.2 Fitness Evaluation and Selection

Based on the aforementioned procedure, the makespan of a berth allocation can be obtained. As shown in Equation (5.17), the reciprocal of this makespan is set to be the fitness value of the chromosome from which the berth allocation is constructed.

$$\text{Fitness value} = 1 \Big/ \max_{s} \; c_s \qquad (5.17)$$

In this chapter, 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).

5.3.3 Crossover

Generally, the aforementioned chromosome representation will yield illegal offspring by one-point, two-point or multipoint crossover in the sense of that some container ships may be missed while some container ships may be duplicated in the offspring. Therefore, this chapter 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 container ships which are already in the substring from the second parent. The resulted sequence of container ships contains the container ships that the proto-child needs.

Step 4**:** Place the container ships into the unfixed positions of the proto-child from left to right according to the order of the sequence to produce an offspring.

The ‘order crossover’ is illustrated in Figure 5.4 that presents an example of producing two offspring from the same parents.

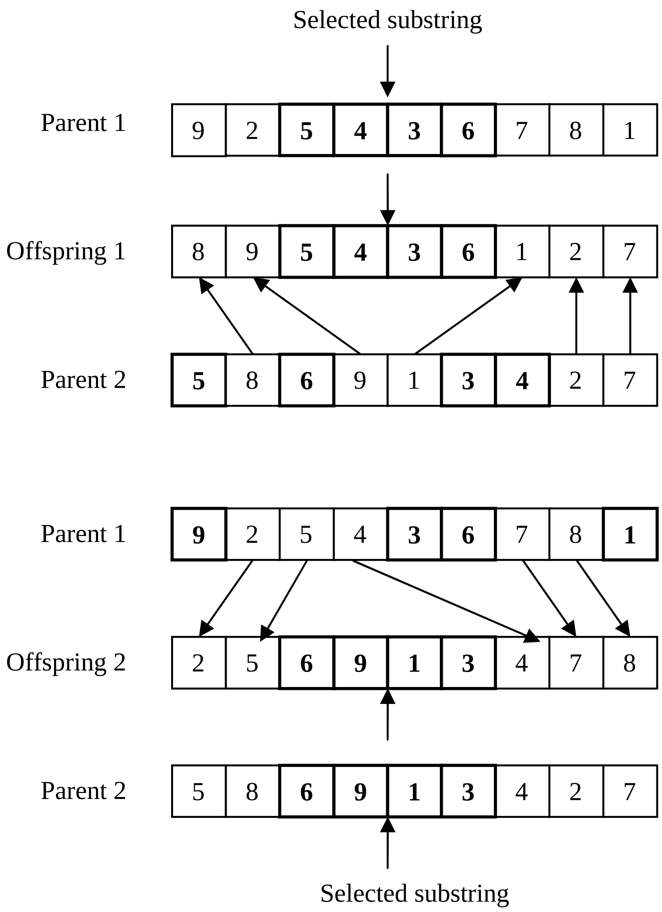

Figure 5.4 An Illustration of the Order Crossover

5.3.4 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 chapter 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 container ships on these positions as illustrated in Figure 5.5.

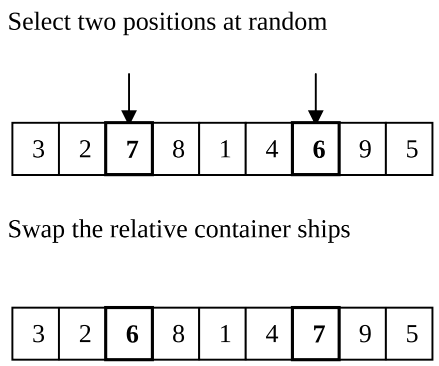

Figure 5.5 An Illustration of the Mutation

5.3.5 An Approximation Algorithm for Quay Crane Scheduling

As mentioned previously, when updating the completion time of the assigned berth in the procedure of constructing a berth allocation from a chromosome, the handling time of this container ship at the assigned berth is needed. In this chapter, this handling time is obtained from an approximation algorithm for the Quay Crane Scheduling with Non-Crossing constraints Problem (QCSNCP) which is elaborated as follows.

Approximation Algorithm: assume that container ship s is allocated to berth q, the number of ship bays in container ship s is B, and the number of quay cranes at berth q is K. The quay crane schedule for container ship s at berth q can be constructed as follows. Assign adjacent ship bays, $b_{k-1}+1,\ b_{k-1}+2,\ ...,\ b_k-1,\ b_k$, to quay crane k ($\forall 1 \le k \le K$). Note that $b_0 = 0$ and $b_K = B$. A dynamic programming algorithm is then proposed to determine the best partition points, $b_1,\ b_2,\ ...,b_{K-2},\ b_{K-1}$, which minimizes the latest completion time among all ship bays.

Parameters:

$MC[k,\ b]$ the minimum latest completion time when ship bays $1,\ 2,\ ...,b-1,\ b$ are assigned to quay cranes $1,\ 2,\ ...,k-1,\ k$ in the above mentioned adjacent manner;

$TP[b_1,\ b_2]=\sum_{b=b_1}^{b_2}P_b$ the total processing time of ship bays $b_1,\ b_1+1,\ ...,b_2-1,\ b_2$.

Dynamic programming equations for determining the best partition points, $b_1,\ b_2,\ ...,b_{K-2},\ b_{K-1}$, are as follows:

$$MC[1,\ b]=TP[1,\ b] \quad \forall 1\le b\le B \tag{5.18}$$

$$MC[k,\ b]=\min_{k-1\le b_{k-1}\le b-1}\max\{MC[k-1,\ b_{k-1}],\ TP[b_{k-1}+1,\ b]\}\ \forall 2\le k\le K, \forall k\le b\le B \tag{5.19}$$

The makespan of the quay crane schedule obtained from the approximation algorithm is $MC[K,\ B]$, which is the handling time of container ship s at berth q.

Figure 5.6 shows a numerical example of the approximation algorithm in which there are two quay cranes at a berth and four ship bays in a container ship. According to Equation (5.18), $MC[1,\ 1]=TP[1,\ 1]=187$, $MC[1,\ 2]=TP[1,\ 2]=281$, $MC[1,\ 3]=TP[1,\ 3]=387$, and $MC[1,\ 4]=TP[1,\ 4]=461$. According to Equation (5.19),

$$\begin{aligned}&MC[2,\ 4]=\min\ \{\max\{MC[1,\ 1],\ TP[2,\ 4]\},\ \max\{MC[1,\ 2],\ TP[3,\ 4]\},\\&\max\{MC[1,\ 3],\ TP[4,\ 4]\}\}=\min\ \{\max\{187,\ 274\},\ \max\{281,\ 180\},\ \max\{387,\ 74\}\}\\&=\min\ \{274,\ 281,\ 387\}=274.\end{aligned}$$

Therefore, Ship Bay 1 is assigned to Quay Crane 1, and Ship Bay 2, 3, and 4 are assigned to Quay Crane 2. The makespan of this quay crane schedule is $MC[2,\ 4]=274$, which is the handling time of the container ship at the berth.

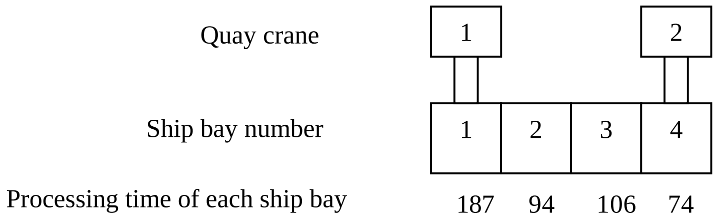

Figure 5.6 A Numerical Example of the Approximation Algorithm

Worst-case analysis for the approximation algorithm is performed as follows.

Parameters:

Z_1 the objective function value of the solution to the QCSNCP obtained by the approximation algorithm proposed in Chapter 2;

Z_2 the objective function value of the solution to the QCSNCP obtained by the approximation algorithm proposed in Chapter 5;

Z^* the objective function value of the optimal solution to the QCSNCP.

Theorem 5.2: $Z_2 / Z^* \le 2$

Proof:

Both the approximation algorithm proposed in Chapter 2 and the approximation algorithm proposed in Chapter 5 assign ship bays to quay cranes in the aforementioned adjacent manner. According to the Theorem 2.2, $Z_1 \le 2Z^*$. Since the approximation algorithm proposed in Chapter 5 optimizes the partition points, $b_1, b_2, ..., b_{K-2}, b_{K-1}$, $Z_2 \le Z_1 \le 2Z^*$. The Theorem 5.2 is proved.

5.4 COMPUTATIONAL EXPERIMENTS

A series of computational experiments are conducted to examine the performance of the proposed GA. The GA is coded in C++ and executed in a Pentium IV 3.6GHz PC with 2GB RAM.

There are forty random instances systematically generated. Two port container terminals are examined whose configurations are summarized in Table 5.1. The planning horizon is one week and the number of container ships arriving at each port container terminal during one week is indicated in Table 5.2 and Table 5.3. The arrival time of every container ship, the number of ship bays in every container ship, and the processing time of each ship bay in every container ship are randomly generated from uniform distribution of $U(0,10080)$, $U(10,30)$, and $U(30,180)$, respectively. For each problem size, four instances are generated by using different random seeds. 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 500, 0.5, 0.3, and 2,000 respectively in these computational experiments.

Table 5.1 The Configurations of Two Port Container Terminals

Port container terminal 1		Port container terminal 2	
Berth number	The number of quay cranes at each berth	Berth number	The number of quay cranes at each berth
1	2	1	2
2	3	2	4
3	4	3	3
4	3	4	4
		5	3
		6	5

In order to evaluate the performance of the proposed GA, the lower bound corresponding to the instance can be obtained from the following equations. Equation (5.20) denotes the lower bound of the handling time of container ship s at the berth with the largest number of quay cranes, lb_s. Equation (5.21) indicates the lower bound of the makespan of handling all container ships, LB.

$$lb_s = \max \ \{\sum_{b_s=1}^{B_s} P_{b_s} \Big/ \text{the largest number of quay cranes, } \max_{b_s}\{P_{b_s}\}\} \quad \forall 1 \le s \le S \qquad (5.20)$$

$$LB = \max_s \ \{a_s + lb_s\} \qquad (5.21)$$

As shown in Table 5.2, for port container terminal 1, the maximum gap between the near optimal solution obtained from the genetic algorithm and the lower bound among these twenty instances is 27.97%, the minimum gap is 0.18%, and the average gap is 12.41%. As shown in Table 5.3, for port container terminal 2, the maximum gap between the near optimal solution obtained from the genetic algorithm and the lower bound among these twenty instances is 26.16%, the minimum gap is 0.23%, and the average gap is 9.68%. As observed in Table 5.2 and Table 5.3, when the number of container ships arriving at a port container terminal during one week increases, the gap between the near optimal solution obtained from the genetic algorithm and the lower bound grows. However, it does not always indicate that the gap between the near optimal solution and the optimal solution increases. This may be due to the following reasons. When calculating the lower bound, it is assumed that each container ship can be berthed immediately when it arrives at a port container terminal. However, it is possible that some container ships may have to wait for available berths when the number of container ships becomes larger. In this case, the gap between the optimal solution and the lower bound may become larger as well. Therefore, the gap between the near optimal solution and the optimal solution may still be small. As seen in Table 5.2 and Table 5.3, all the computational time of these forty instances is within seven seconds. Based on the aforementioned analysis, the proposed GA is concluded to be effective and efficient in solving the proposed IBAQCSP.

In general, the number of berths ranges from two to six in port container terminals, the number of container ships arriving during one week ranges from twenty to sixty, the number of quay cranes at a berth ranges from two to four, and the number of ship bays in a container ship ranges from ten to twenty-five. Hence, the random instance in the computational experiments is very close to the reality. Based on the computational results, the proposed GA may be considered as an appropriate approach to scheduling berths and quay cranes in port container terminals to enhance their efficiency.

Table 5.2 Computational Results of Port Container Terminal 1

Experiment No	Size (ships×berths)	Lower Bound	GA		Gap [a] (%)
			Value	CPU (sec)	
1	25×4	10915	11235	3.27	2.93
2	25×4	10404	11008	3.23	5.81
3	25×4	10481	10815	3.22	3.19
4	25×4	10763	10981	3.22	2.03
5	30×4	10010	10705	3.73	6.94
6	30×4	10321	10930	3.58	5.90
7	30×4	10530	10549	3.61	0.18
8	30×4	10920	11214	3.58	2.69
9	35×4	10880	12430	3.95	14.25
10	35×4	10492	11960	3.94	13.99
11	35×4	10698	11723	3.98	9.58
12	35×4	10746	12018	4.06	11.84
13	40×4	10923	12992	4.42	18.94
14	40×4	10858	12498	4.41	15.10
15	40×4	10727	13032	4.44	21.49
16	40×4	10775	13237	4.39	22.85
17	45×4	10831	13169	4.86	21.59
18	45×4	10803	13187	4.84	22.07
19	45×4	10717	13715	4.86	27.97
20	45×4	10926	12991	4.89	18.90

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

Table 5.3 Computational Results of Port Container Terminal 2

Experiment No	Size (ships×berths)	Lower Bound	GA		Gap [a] (%)
			Value	CPU (sec)	
1	40×6	10639	10900	4.39	2.45
2	40×6	10534	10632	4.38	0.93
3	40×6	10744	10769	4.41	0.23
4	40×6	10686	10721	4.44	0.33
5	45×6	10635	11372	4.84	6.93
6	45×6	10583	11422	4.86	7.93
7	45×6	10602	11219	4.88	5.82
8	45×6	10698	11687	4.97	9.24
9	50×6	10572	11119	5.34	5.17
10	50×6	10731	11765	5.36	9.64
11	50×6	10621	11596	5.33	9.18
12	50×6	10652	10756	5.36	0.98
13	55×6	10670	12322	5.88	15.48
14	55×6	10667	12133	6.03	13.74
15	55×6	10692	12571	5.88	17.57
16	55×6	10702	11814	5.83	10.39
17	60×6	10659	13447	6.36	26.16
18	60×6	10642	12522	6.33	17.67
19	60×6	10678	12201	6.39	14.26
20	60×6	10638	12704	6.38	19.42

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

5.5 SUMMARY

This chapter provides a mixed integer programming model including two parts for the proposed IBAQCSP, proves that the IBAQCSP is NP-complete, and proposes a genetic algorithm containing an approximation algorithm for quay crane scheduling to obtain near optimal solution for the IBAQCSP. In addition, computational experiments are conducted to examine the proposed genetic algorithm. The results show that the proposed genetic algorithm is effective and efficient in solving the IBAQCSP.

CHAPTER 6 CONCLUSIONS

6.1 CONCLUDING REMARKS

The main purpose of this thesis was to enhance the efficiency of berth and quay crane operations in port container terminals. In the first part of this thesis, an innovative work on the Quay Crane Scheduling with Non-Crossing constraints Problem (QCSNCP) was discussed. This part provided a mixed integer programming model for the QCSNCP that was NP-complete in nature. Since there were no polynomial time algorithms for the exact solution to NP-complete problems unless P=NP, an approximation algorithm and a genetic algorithm were proposed to obtain its near optimal solutions. Furthermore, worst-case analysis for the approximation algorithm was performed and computational experiments were conducted to examine the proposed model and solution algorithms. The computational results showed that the proposed approximation algorithm and genetic algorithm were effective and efficient in solving the QCSNCP.

In the second part of this thesis, an original work on the Quay Crane Scheduling with Safety Distance and non-crossing constraints Problem (QCSSDP) was presented. A mixed integer programming model was provided for the QCSSDP which was proved to be NP-complete. An approximation algorithm and a genetic algorithm were proposed to obtain near optimal solutions for the QCSSDP. In addition, worst-case analysis for the approximation algorithm was performed, and computational experiments for the approximation algorithm and the genetic algorithm were conducted. The computational results showed that both the approximation algorithm and the genetic algorithm were effective and efficient in solving the QCSSDP.

In the third part of this thesis, a novel work on the Quay Crane Scheduling with Handling Priority and non-crossing constraints Problem (QCSHPP) was described. The QCSHPP was formulated as a mixed integer programming model and proved to be NP-complete. Thus, an approximation algorithm was designed for obtaining near optimal solution to the QCSHPP. Moreover, worst-case analysis for the approximation algorithm was performed and computational experiments were conducted. The computational results showed that the proposed approximation algorithm was effective and efficient in solving the QCSHPP.

In the last part of this thesis, an original work on the Integrated discrete Berth Allocation and Quay Crane Scheduling Problem (IBAQCSP) was addressed. A mixed integer programming model including two parts was provided for the IBAQCSP which was proved to be NP-complete. A genetic algorithm containing an approximation algorithm for quay crane scheduling was then proposed to obtain near optimal solution to the IBAQCSP. Finally, computational experiments were performed to examine the performance of the proposed GA and the results showed that the proposed GA was effective and efficient in solving the IBAQCSP.

6.2 RECOMMENDATIONS FOR FUTURE RESEARCH

1. As the first attempt, the Quay Crane Scheduling with Handling Priority and non-crossing constraints Problem (QCSHPP), and Integrated discrete Berth Allocation

and Quay Crane Scheduling Problem (IBAQCSP) did not consider the safety distance constraints. This implies that quay crane schedules obtained from these models may not always satisfy operational requirements in port container terminals. Therefore, the incorporation of the safety distance constraints into the QCSHPP and IBAQCSP may be explored in future research.

2. Compared with the processing time of a ship bay by a quay crane, the travel time of a quay crane between two ship bays is small and hence it was not considered in this thesis. However, the travel time of quay cranes exists in reality. Further research may take this factor into account so that the attained quay crane scheduling model may be more adaptable in practice.

3. The integrated discrete berth allocation and quay crane scheduling problem assumed that the number of quay cranes at each berth was fixed. In fact, quay cranes can be transferred among berths to increase operational efficiency. The incorporation of quay crane transfer into the current study may be a promising topic for future research.

4. Compared to the discrete berth allocation, the continuous berth allocation can further enhance the efficiency of berth usage. It may be interesting to study the Integrated Continuous Berth Allocation and Quay Crane Scheduling Problem (ICBAQCSP) in the future. The berthing position, the berthing time, the number of assigned quay cranes, and the quay crane schedule for every container ship may be determined

simultaneously in the ICBAQCSP so that the efficiency of port operations may be further improved.

6.3 RESEARCH CONTRIBUTIONS

1. A comprehensive literature review on berth allocation and quay crane scheduling is provided and the details of practical berth and quay crane operations are elaborated in this thesis. It may serve as a reference for researchers who are interested in port operations.

2. Traditional parallel machine scheduling problems do not consider the non-crossing and safety distance constraints. This thesis investigates the parallel quay crane scheduling problems with the non-crossing and safety distance constraints. It may contribute to the theory of parallel machine scheduling.

3. This thesis proves that all the proposed problems are NP-complete. Theoretically speaking, there are no polynomial time algorithms for the exact solution to all these problems unless P=NP. Researchers who are interested in these problems may take these proofs as references and focus on developing heuristic algorithms for these problems.

4. Computational experiments show that both the approximation algorithms and genetic algorithms proposed by this thesis are effective and efficient in scheduling berths and

quay cranes. Port container terminals may adopt these scheduling methods in practice to enhance their operational efficiency.

5. The study on the IBAQCSP should enhance our understanding of combined optimization of berth allocation and quay crane scheduling. This knowledge may further increase the overall efficiency of port operations when comparing to optimizing berth allocation or quay crane scheduling individually.

6. The proposed scheduling methods are coded into computer programs. These source codes may be employed as the key components of the future software for optimizing port operations.

REFERENCES

Daganzo, C.F. (1989) The crane scheduling problem. *Transportation Research Part B*, Vol. 23, 159-175.

Eastman, W.L., Even, S. and Isaacs, I.M. (1964) Bounds for the optimal scheduling of *n* jobs on *m* processors. *Management Science*, Vol. 11, 268-279.

Garey, M.R. and Johnson, D.S. (1979) *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, San Francisco.

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

Guan, Y., Xiao, W.Q., Cheung, R.K. and Li, C.L. (2002) A multiprocessor task scheduling model for berth allocation: heuristic and worst-case analysis. *Operations Research Letters*, Vol. 30, 343-350.

Guan, Y. and Cheung, R.K. (2004) The berth allocation problem: models and solution methods. *OR Spectrum*, Vol. 26, 75-92.

Günther, H.O. and Kim, K.H. (2006) Container terminals and terminal operations. *OR Spectrum*, Vol. 28, 437-445.

Imai, A., Nagaiwa, K. and Chan, W.T. (1997) Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation*, Vol. 31, 75-94.

Imai, A., Nishimura, E. and Papadimitriou, S. (2001) The dynamic berth allocation problem for a container port. *Transportation Research Part B*, Vol. 35, 401-417.

Imai, A., Nishimura, E. and Papadimitriou, S. (2003) Berth allocation with service priority. *Transportation Research Part B*, Vol. 37, 437-457.

Imai, A., Sun, X., Nishimura, E. and Papadimitriou, S. (2005) Berth allocation in a container port: using a continuous location space approach. *Transportation Research Part B*, Vol. 39, 199-221.

Kim, K.H. and Moon, K.C. (2003) Berth scheduling by simulated annealing. *Transportation Research Part B*, Vol. 37, 541-560.

Kim, K.H. and Park, Y.M. (2004) A crane scheduling method for port container terminals. *European Journal of Operation Research*, Vol. 156, 752-768.

Lai, K.K. and Shih, K. (1992) A study of container berth allocation. *Journal of Advanced Transportation*, Vol. 26, 45-60.

Lenstra, J.K., Rinnooy Kan, A.H.G. and Brucker, P. (1977) Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, Vol. 1, 343-362.

Li, C.L., Cai, X. and Lee, C.Y. (1998) Scheduling with multiple-job-on-one-processor pattern. *IIE Transactions*, Vol. 30, 433-445.

Lim, A. (1998) The berth planning problem. *Operations Research Letters*, Vol. 22, 105-110.

Lim, A., Rodrigues, B., Xiao, F. and Zhu, Y. (2004a) Crane scheduling with spatial constraints. *Naval Research Logistics*, Vol. 51, 386-406.

Lim, A., Rodrigues, B. and Xu, Z. (2004b) Solving the crane scheduling problem using intelligent search schemes. *Lecture Notes in Computer Science*, Vol. 3258, 747-751.

Lim, A., Rodrigues, B. and Xu, Z. (2004c) Approximation schemes for the crane scheduling problem. *Lecture Notes in Computer Science*, Vol. 3111, 323-335.

Liu, J., Wan, Y. and Wang, L. (2006) Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. *Naval Research Logistics*, Vol. 53, 60-74.

Moccia, L., Cordeau, J.F., Gaudioso, M. and Laporte, G. (2006) A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal. *Naval Research Logistics*, Vol. 53, 45-59.

Moorthy, R. and Teo, C.P. (2006) Berth management in container terminal: the template design problem. *OR Spectrum*, Vol. 28, 495-518.

Ng, W.C. and Mak, K.L. (2006) Quay crane scheduling in container terminals. *Engineering Optimization*, Vol. 38, 723-737.

Nishimura, E., Imai, A. and Papadimitriou, S. (2001) Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research*, Vol. 131, 282-292.

Park, K.T. and Kim, K.H. (2002) Berth scheduling for container terminals by using a sub-gradient optimization technique. *Journal of the Operational Research Society*, Vol. 53, 1054-1062.

Park, Y.M. and Kim, K.H. (2003) A scheduling method for berth and quay cranes. *OR spectrum*, Vol. 25, 1-23.

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

Smith, W.E. (1956) Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, Vol. 3, 59-66.

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

Vis, I.F.A. and de Koster, R. (2003) Transshipment of containers at a container terminal: an overview. *European Journal of Operational Research*, Vol. 147, 1-16.

Zhu, Y. and Lim, A. (2006) Crane scheduling with non-crossing constraint. *Journal of the Operational Research Society*, Vol. 57, 1464-1471.

APPENDIX: Recent Research Accomplishments

[1] Lee, D.H., Wang, H.Q. and Miao, L. (2008) Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E: Logistics and Transportation Review*, Vol.44, 124-135.

[2] Lee, D.H., Wang, H.Q. and Miao, L. (2007) Quay crane scheduling with handling priority in port container terminals. *Engineering Optimization*, in press.

[3] Lee, D.H., Wang, H.Q. and Miao, L. (2007) An approximation algorithm for quay crane scheduling with non-crossing and safety distance constraints in port container terminals. *Journal of the Eastern Asia Society for Transportation Studies*, in press.

[4] Lee, D.H., Song, L. and Wang, H.Q. (2007) An optimization approach for security operations towards sustainable seaports. *International Journal of Sustainable Transportation*, in press.

[5] Lee, D.H., Song, L., Wang, H.Q. and Miao, L. (2006) An interactive scheduling method for berths and quay cranes in port container terminals. *Journal of the Institution of Engineers, Singapore*, in press.

[6] Lee, D.H., Wang, H.Q. and Miao, L. (2007) Quay crane scheduling with safety distance and non-crossing constraints in port container terminals. *Transportation Research Part C: Emerging Technology*, Submitted for review.

[7] Lee, D.H., Wang, H.Q. and Miao, L. (2007) An approximation algorithm for quay crane scheduling with handling priority in port container terminals. *International Journal of Production Economics*, Submitted for review.

[8] Lee, D.H., Wang, H.Q. and Miao, L. (2008) Integrated discrete berth allocation and quay crane scheduling in port container terminals. *Applied Mathematics and Computation*, Submitted for review.

[9] Lee, D.H., Wang, H.Q., Shon, Z.Y. and Miao, L. (2007) An approximation algorithm for quay crane scheduling with non-crossing constraints in port container terminals. *Transportation Research Record: Journal of the Transportation Research Board*, Submitted for review.

[10] Lee, D.H., Song, L., Wang, H.Q. and Miao, L. (2005) A genetic algorithm for a bi-level programming model of berth allocation and quay crane scheduling. *Proceedings of the 85th Transportation Research Board (TRB) Annual Meeting, January 22-26, 2006, Washington, DC, U.S.*

[11] Lee, D.H., Wang, H.Q. and Miao, L. (2007) An approximation algorithm for quay crane scheduling with non-interference constraints in port container terminals. *Proceedings of the Sixth Triennial Symposium on Transportation Analysis (TRISTAN VI), June 10-15, 2007, Thailand*.

[12] Wang, H.Q., Lee, D.H., Cao, Z. and Miao, L. (2006) Quay crane scheduling with non-crossing and safety distance constraints in port container terminals. *Proceedings of the 36th International Conference on Computers and Industrial Engineering (ICCIE 2006), June 20-23, 2006, Taipei, China*.

[13] Lee, D.H., Wang, H.Q. and Miao, L. (2006) Quay crane scheduling in port container terminals: a heuristic algorithm and worst-case analysis. *Proceedings of the 11th World Conference on Transportation Research (WCTR 2007), June 24-28, 2007, Berkeley, U.S.*

[14] Lee, D.H., Song, L. and Wang, H.Q. (2005) A bi-level programming model and solutions of berth allocation and quay crane scheduling. *Proceedings of the First International Conference on Transportation Logistics (T-Log 2005), July 27-29, 2005, Singapore*.

[15] Lee, D.H., Wang, H.Q. and Miao, L. (2006) A genetic algorithm for quay crane scheduling with handling priority in port container terminals. *Proceedings of the Second International Conference on Transportation Logistics (T-Log 2007), July 4-6, 2007, Shenzhen, China*.

[16] Lee, D.H., Wang, H.Q. and Miao, L. (2007) An approximation algorithm for quay crane scheduling with non-crossing and safety distance constraints in port container terminals. *Proceedings of the 7th International Conference of Eastern Asia Society for Transportation Studies, September 24-27, 2007, Dalian, China*.

[17] Lee, D.H., Wang, H.Q., Shon, Z.Y. and Miao, L. (2007) An approximation algorithm for quay crane scheduling with non-crossing constraints in port container terminals. *Proceedings of the 87th Transportation Research Board (TRB) Annual Meeting, January 13-17, 2008, Washington, DC, U.S*.