OPTIMIZING YARD CRANE OPERATIONS IN PORT CONTAINER TERMINALS

CAO ZHI

NATIONAL UNIVERSITY OF SINGAPORE

2006

OPTIMIZING YARD CRANE OPERATIONS IN PORT CONTAINER TERMINALS

CAO ZHI

B.Eng. (*Tsinghua University*)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF CIVIL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2006

ACKNOWLEDGEMENTS

The author wishes to express his deepest appreciation to both of his supervisors, Associate Professor Lee Der-Horng and Assistant Professor Meng Qiang, for their rigorous scientific guidance, invaluable constant advice, constructive suggestion, and continuous support throughout the course of his Ph.D. study in NUS, and their care and advice on his personal matters as well.

The author would also like to thank Dr. Tan Kok Choon and Prof. IMAI Akio for their precious guidance and suggestions on his academic research work.

The author is pleased to thank Mr. Foo Chee Kiong and all other technicians and administrative staffs for their friendship and kind assistance.

Particularly, the author would like to thank his colleagues in the ITVS Lab, Alvina Kek Geok Hoon , Wang Huiqiu, Huang Yikai, Dong Meng, Bian Wen, Cheng Shihua, Deng Weijia, Fery Pierre Geoffroy Julien, Xie Chenglin, Huang Yongxi, Liu Nan, Sun Yueping, Pan Xiaohong, Yao Li, Huang Wei, Fan Tao, Song Liying and Zheng Weizhong. The author is highly appreciated to the encouragement and help from his peers in the past three years. A special note of thankfulness is also expressed to others who have helped him in one way or other.

Special thanks are due to the National University of Singapore for providing the author with a research scholarship covering the entire period of his graduate studies.

Last but not the least, the author would like to take this opportunity to express his deep-hearted gratitude to his parents and wife for their endless love and support through all the time.

TABLE OF CONTENTS

OPTIMIZING YARD CRANE OPERATIONS IN PORT CONTAIENR TERMINALS

(SUMMARY)

In modern business logistics, both the number of container ports and the competition among them have become prominent with the steady progress of containerization over the past 20 years, which makes the efficiency of port operation an important factor in succeeding in the fierce competition.

This thesis focuses on one of the critical aspects of the container terminal operations, the scheduling of yard cranes. Despite the fact that the yard crane scheduling plays an important role in determining the over efficiency of the terminal operation, the related reports in the literature only studied the problem partially. Therefore a comprehensive study on the scheduling problem of yard cranes in port container terminals is highly desired.

A simplified multiple yard crane scheduling problem, two yard crane scheduling problem, is first studied as a preliminary work. Based on that, the typical multiple yard crane scheduling problem is then intensively studied. Subsequently, the results is extended to two problems derived from the standard multiple yard crane scheduling problem, the scheduling of multiple yard cranes in terminals with buffer areas and the deployment of double rail mounted gantry cranes in yard truck based terminals. In the end a study on the simultaneous scheduling problem of quay crane and yard crane is presented. All these problems are successively formulated by mathematical models. Several solution techniques are developed to solve these problems.

The results of the study indicates that compared to the widely used meta-heuristic algorithms, the relatively simple greedy heuristics algorithm is a more effective solution technique for solving the scheduling problem of the multiple yard crane system. Therefore it can be adopted by the container terminal operators to improve the efficiency of their operations. The influence of using buffer area in container terminals has also been examined in the study. The results suggests that the productivity of yard cranes could be enhanced and the loading operation at the yard area can be expedited at the expense of using more land space and more yard trucks. This result can be used by the terminal operators as a reference when deciding whether to use buffer areas in their terminals. The deployment strategy of the double rail mounted gantry crane system in yard truck based container terminals is also investigated. Using this system in traditional yard truck based container terminals can eliminate the interference of yard cranes. As a result the productivity of the cranes can be improved. The operational strategy of the double rail mounted gantry crane system proposed outperformed the SA algorithm through numerical experiments. A simultaneous scheduling of quay crane and yard crane was also successfully accomplished in the study. Being the first study of its kind, this study can be used to improve the overall performance of quay cranes and yard cranes. It can also work as one component of the wholly integrated container terminal operating system which is to be developed in the future research.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

With the steady progress of world trade, marine transportation has experienced immense growth over the past 20 years. Container, as the foundation of the unit-load-concept, has achieved undoubted importance in international marine transportation. Today among the world's seaborne cargo, more than 60% is transported in containers and this proportion is still growing. Figure 1.1 shows this containerization trend in the past decade. As a result, the number of container shipments has increased dramatically over the past decade, which causes higher demand for the throughput of container terminals and leads to intense competition among these terminals, especially the geographically close ones such as the port of Singapore and the Tanjung Pelepas port of Malaysia. To accommodate the increased demand and succeed in the fierce competition in the container logistics industry, the container terminal operators need to improve the efficiency of their port operations by means of implementing new management strategies and adopting advanced technologies.

In general, after arrival at a container terminal, the containership is allocated to a berth equipped with quay cranes to load and unload containers. The unloaded inbound containers are distributed to the yard area by yard trucks and stacked in the container blocks by yard cranes. The outbound containers arriving by road or railway are handled in a converse way. Figure 1.2 illustrates the standard flow of containers in port container terminals.

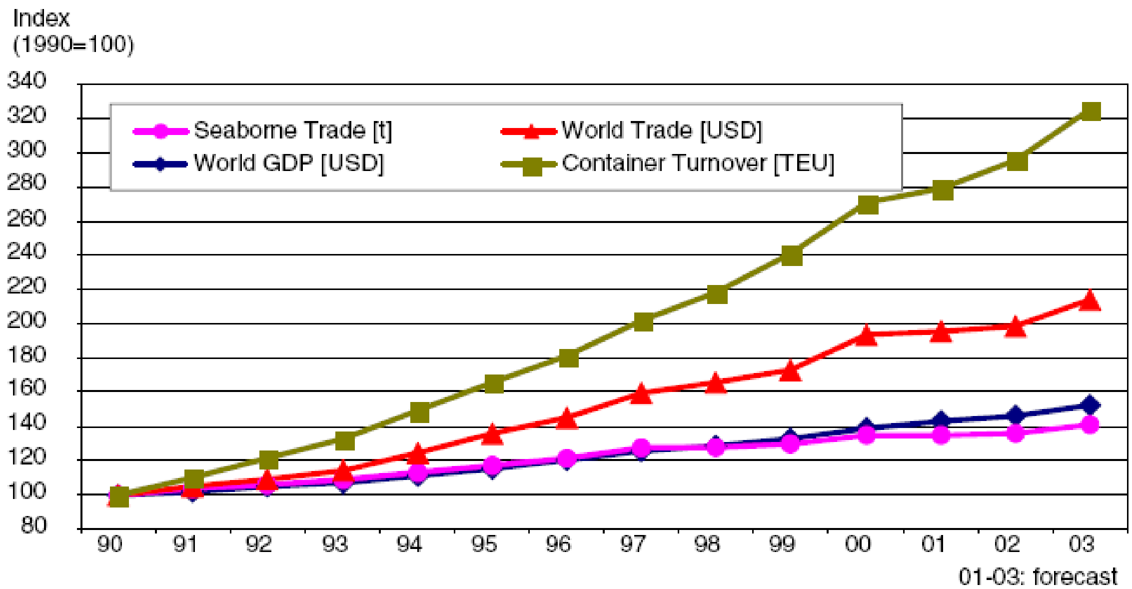


Figure 1.1 Containerization Trend: High Growth Rate of Container Turnover (Steenken et al., 2004)

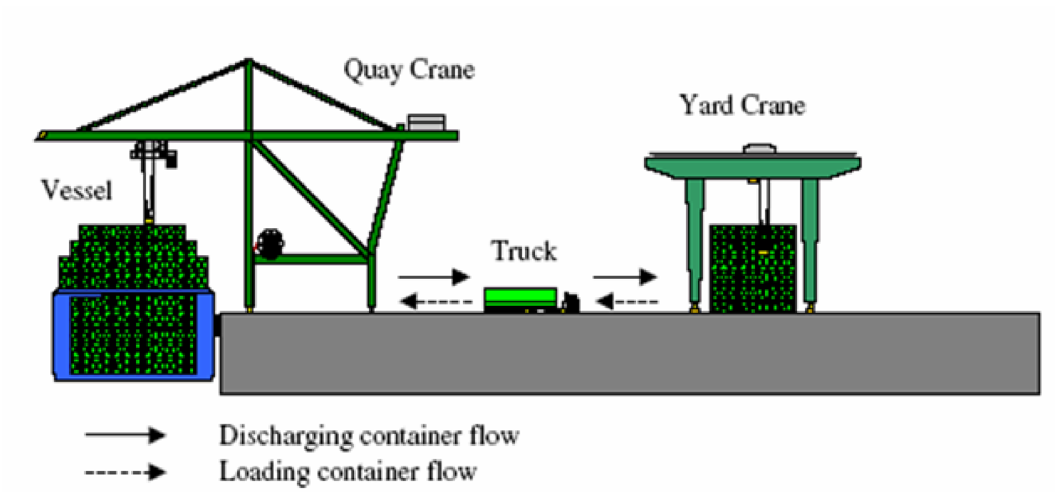


Figure 1.2 Container Flow in a Port Container Terminals (Ng, 2005)

1.2 RESEARCH OBJECTIVES AND SCOPE

This thesis will present a comprehensive study on the multiple yard crane scheduling problem in which both the inter-crane interference constraint and the container loading sequence constraint are considered. A mathematical model will be developed for the formulation of the problem. Exact algorithms will be designed to solve small-scale problems while meta-heuristic algorithms as well as customized heuristic algorithms will be designed to solve large scale problems. The performance of all the algorithms will be examined through numerical experiments.

A study on the scheduling problem of yard cranes in container terminals with buffer areas will also be presented in this thesis. An integer programming model will be proposed to formulate the problem. A heuristic algorithm based on greedy principle will also be developed as a solving technique to the model. Sample test problems will be generated to examine the effect on the yard crane operation time by reserving buffer areas in the stacking area.

Double rail mounted gantry crane (DRMG) system is a new container handling technology which consists of two cranes of different size. Since the two cranes can pass each other during operations, the productivity of the system will be higher than the traditional type of crane system. This thesis will study the operation strategy of the DRMG system in yard truck based container terminals. A mathematical model will be developed for the problem formulation and a set of operation rules will be proposed to conduct the

DRMG scheduling.

This thesis will also present a simultaneous study on the quay crane scheduling problem and yard crane scheduling problem. In the study, the work schedule of a quay crane will act as the container loading sequence requirement for the yard cranes serving the quay crane. An integer programming model will be developed to formulate the quay crane scheduling and the related yard crane scheduling. A simulated annealing algorithm will be designed to solve the proposed model. Different weights of quay crane operation time and yard crane operation time will be examined through numerical experiments.

This thesis may provide a better way to conduct the scheduling of yard cranes in port container terminals. As a result the overall efficiency of the port operation can be enhanced. The study of reserving buffer areas in the stack area can also help the terminal operators to decide whether to use buffer areas in the yard or not. The study of DRMG system can be used as a reference in the future deployment of DRMG system in yard truck based container terminals. This thesis may also clarify the relationship between different weights of quay crane and yard crane operation time and the related quay crane and yard crane scheduling. Hence it could help the terminal operators to determine the proper work schedules of quay cranes and yard cranes to satisfy different time requirements.

1.3 ORGANIZATION OF THE THESIS

This thesis consists of eight chapters.

Chapter 1 is the introductory chapter which provides the general background of the research and lays out the objective and scope of the research.

Chapter 2 reviews the past research works related to this research as well as some mathematical techniques which will be used in conducting the research.

Chapter 3 describes a mathematical model developed to formulate the two yard crane scheduling problem which is a simplified version of the multiple yard crane scheduling problem. In the problem, two yard cranes are working for one loading plan in two different blocks, free of inter-crane interference, at the same time. A simulated annealing revised algorithm is designed to solve the proposed model and the performance of the algorithm is tested through a series of numerical experiments.

Chapter 4 extends the study in Chapter 3 to the general case of multiple yard crane scheduling problem in which the interference of cranes needs to be considered. An integer programming model is developed to formulate the problem and several heuristic algorithms are proposed to solve the problem. Computational experiments are conducted to measure the performance of the algorithms.

Chapter 5 provides a study on scheduling multiple yard crane systems in port container terminals with buffer areas. The existence of buffer areas relaxes the loading sequence requirement of yard cranes and therefore affects the scheduling of yard cranes. A mathematical model is also developed for the problem formulation. The influence of buffer areas on the terminal operations is investigated through numerical experiments.

Chapter 6 investigates the operational strategies of DRMG system in yard truck based container terminals. The deployment of the DRMG system will help to avoid the inter-crane interference so that the productivity of yard cranes can be enhanced. The problem is formulated as an integer programming model. A heuristic approach is designed to conduct the scheduling of DRMG system.

Chapter 7 proposes the concept of simultaneous scheduling of quay crane and yard crane. The quay crane scheduling problem and its related yard crane scheduling problem are studied at the same time so that a holistic view of the container terminal facility operation is achieved. An integer programming model is developed to model the proposed problem. A genetic algorithm is also designed as the solution technique.

Chapter 8 provides a conclusion of this thesis. Contributions of the research and the recommendations for future study are also appended at the end.

CHAPTER 2

LITERATURE REVIEW

2.1 CONTAINER TERMINAL OPERATIONS

In general port operations can be divided into two main parts: quayside operation and landside operation. The quayside operation consists of berth allocation, stowage planning and quay crane scheduling. The landside operation includes yard storage planning, internal transport planning and yard crane scheduling. Although much research has been conducted on the different aspects of port operations, yard crane scheduling, being one key component of port operations, has not been studied systemically. Therefore this thesis will present a comprehensive study on the yard crane scheduling problem. Since the different components of port operations are closely related to each other, an overview of the aforementioned quayside and landside operations is first introduced in the following section.

2.1.1 Overview of Port Operations

Before the arrival of a containership, the port operator must allocate a berth to the ship. To conduct the berth allocation, the operator needs to consider the technical data of the ship, the quay availability and the yard situation to choose an appropriate berth to the ship. Once a berth is allocated to the ship, the terminal operator will start the ship stowage planning process, in which, dedicated containers identified by numbers will be assigned to the respective slots in the ship. After constructing the stowage plan, the operators then can determine the number of quay cranes to serve the ship and the work schedule of each quay crane. Figure 2.1 shows the quay cranes in operation.

Figure 2.1 Quay Cranes in Operations (Linn et al., 2003)

At the same time the yard storage planning will also be carried out. In this process, a specific position in the yard characterized by the numbers of block, yard bay, slot and tier will be assigned to an inbound or outbound container. Based on the work schedule of quay cranes and the yard storage plan, the terminal operator then can develop the work schedules of yard cranes as well as the internal transport plan of yard trucks, which is used to transport containers between quay cranes and yard cranes. Figure 2.2 shows the yard cranes in a container terminal.

Container terminals can be classified into two categories according to the nature of their operations, namely transshipment terminal and import-export terminal, also called gate terminal. In transshipment terminals, usually several clusters of yard-bays will be reserved for the arrival of a vessel so that the inbound containers can be stacked in these clusters and transported to the connecting vessel later from there. In this operation, since the containers are located close to each other, the yard cranes need not to traverse much. However in import-export terminal, outbound containers are usually scattered in the container blocks in the stacking area. The yard cranes therefore need to traverse along the container blocks to reach the containers. Moreover, the containers picked up by the yard cranes must satisfy the work schedules of quay cranes, which makes the scheduling of yard crane in handling outbound containers a complicated problem that requires intensive study efforts of researchers. In contrast an inbound container is normally stacked next to the previous one. The yard cranes do not need to traverse much along the container blocks to stack the inbound container, which makes the scheduling of yard cranes in handling inbound containers a relatively simple problem. Hence the scheduling problem of yard cranes in loading outbound containers in import-export terminals will be the focus of this thesis.

Several researches have been conducted on the yard crane scheduling problem. The following section will provides a detailed report on the studies on the yard crane scheduling problem.

Figure 2.2 Yard Crane in a Container Terminal (Linn et al., 2003)

2.1.2 Literature Review on Yard Crane Operations

2.1.2.1 Single yard crane scheduling

Since the yard crane scheduling problem is of great importance in determining the overall efficiency of container port operations, a number of studies have been conducted in this area.

Kim and Kim (1999) proposed a mixed integer programming (MIP) model to formulate the routing problem of a single yard crane loading export containers out of the stack onto waiting yard trucks. Based on the MIP formulation, an optimizing algorithm was also developed. However the algorithm was only applied to small scale problems in the study.

Narasimhan and Palekar (2002) proved that the above single yard crane routing problem is

NP-complete in nature. A heuristic algorithm and an exact branch-and-bound algorithm for the problem were developed and tested by numerical experiments. The computational results showed that the exact algorithm is not practical for large scale problems due to intolerable computational time.

To deal with the excessive computational time requirement, Kim and Kim (1999) proposed a beam search algorithm for the problem solution. The same authors (2003) compared the performance of the beam search algorithm and genetic algorithm on the problem. It was found through numerical experiments that the proposed beam search algorithm consistently outperformed a genetic algorithm.

Kim et al. (2003) also studied the single yard crane scheduling problem from a different perspective by investigating the delay of yard trucks which need to be served by yard cranes. The loading sequence requirement is represented in terms of the delay cost of yard trucks. The performance of various sequencing methods on the proposed problem were tested through a simulation study.

2.1.2.2 Multiple yard crane scheduling

All the above studies focused on the single yard crane scheduling problem in which only one yard crane is used to serve one quay crane. However, because of the different technical performances between quay crane and yard crane (quay crane: 50-60 boxes/hr, yard crane: 20 moves/hr), two or even more yard cranes are deployed to serve one quay crane in many container terminals. Thus it is necessary to study the scheduling problem of multiple yard crane system to enhance the efficiency of yard crane operations.

Recently, Kim et al. (2005) studied the load scheduling problem of two yard cranes in the same container block. In the study each yard crane was dedicated to one quay cranes. However it is possible to further increase the efficiency of yard crane operations if the two yard cranes are free to work for any of the two quay cranes.

Ng (2005) studied the scheduling problem of multiple yard crane systems and proposed a heuristic algorithm to minimize the operation time. Nevertheless, the loading sequence requirement of the containers is not considered in the study.

Despite the significance of the scheduling problem of multiple yard crane system in practical operation, only the aforementioned two studies are available in literature. Therefore in-depth studies on the scheduling problem of multiple yard crane system are highly desired. In most import-export terminals, outbound containers are scattered in the container blocks. To fetch the appropriate containers satisfying the loading sequence requirement, the yard cranes need to traverse extensively along the container blocks. However, an inbound container is normally stacked next to the previous one. The yard cranes do not need to traverse much along the container blocks to stack the inbound container, which makes the scheduling of yard cranes in handling inbound containers a relatively simple problem. Therefore, only the scheduling problem of yard cranes in loading outbound containers will be considered in this thesis.

DRMG system represents a new container handling technology in port container terminals.

The only work regarding the operation of DRMG in literature is conducted by Kim et al. (2002). The authors carried out a simulation study on the operation rules of DRMG in an Automated Guided Vehicles (AGV) based container terminal. Hence a systemic study on the operation of DRMG system will be of significant meaning in the future deployment of the system.

In practical operation, physical or virtual buffer areas will be reserved in the stacking area of some container terminals. The containers picked up by the yard cranes ahead of schedule then can be temporarily stored in the buffer areas till they can be handled by the quay cranes. Using such buffer areas will help to increase the utilization of the yard cranes and expedite the loading operation at the stacking area. Nevertheless no research has been conducted on the scheduling problem of yard cranes in container terminals with buffer areas. A study on this problem will be of practical importance in operating yard cranes in container terminals with buffer areas.

2.1.3 Literature Review on Quay Crane Scheduling

The work schedule of quay cranes usually serves as the guideline for the yard crane operations. Hence the scheduling of yard cranes will be significantly affected by the scheduling of quay cranes. Several researches have been done on the quay crane scheduling problem.

Daganzo (1989) developed a MIP formulation for the quay crane scheduling problem. They used exact method to solve small-scale problems and proposed a heuristic procedure for large-scale problems. One important issue in operating quay cranes, the interference problem of quay cranes, was not taken into account in this study.

Lim et al. (2004) augmented the static QC scheduling problem for multiple container vessels by taking into account non-interference constraints. Dynamic programming algorithms, a probabilistic tabu search, and a squeaky wheel optimization heuristic were proposed in solving the problem. However, it is difficult to define a profit value associated with a crane-to-job assignment in practice.

Kim and Park (2004) discussed the QC scheduling problem with non-interference constraints in which only single container vessel was considered. A branch-and-bound method and a heuristic algorithm called greedy randomized adaptive search procedure (GRASP) were designed for the proposed QC scheduling problem.

Based on the earlier study of Daganzo, Park and Kim (2003) combined the quay crane deployment problem with the berth allocation problem. The combined problem was solved by a two-phase solution procedure. The study demonstrated that a detailed working schedule for each quay can be constructed after the preliminary solution of the berth allocation phase is determined. Only the static berth allocation problem, which assumes all the ships have arrived at the terminal before the berth allocation starts, is considered in the study.

Bish (2003) studied a different combined problem which consisted of scheduling quay crane, dispatching yard trucks and determining the storage location for inbound containers and developed a heuristic method to solve the proposed multiple-crane-constrained vehicle scheduling and location problem. The paper presented an integrated study on the three components of port operation, quay crane scheduling, internal transportation and yard storage planning, which could help to achieve better overall performance of the three components compared to studying the components separately.

In spite of the fact that the operation of quay cranes is closely related to the operation of yard cranes, no simultaneous study on these two problems is available in literature. Hence a holistic study, which takes into account both the quay crane scheduling problem and the yard scheduling problem, is highly needed in research.

2.2 META-HEURISTIC ALGORITHMS

2.2.1 Genetic Algorithm

Genetic algorithm (GA) is a directed random search techniques which is developed by Holland (1975) and presented in his book "Adaptation in Natural and Artificial Systems". The method is based on imitating the mechanism of natural genetics and natural species selection process.

When applying GA to solve an optimization problem, first the solutions of the problem need to be encoded into chromosomes. Several encoding methods such as binary encoding, real-number encoding, etc., are generally adapted according to the nature of the problem.

To find the optimal solution of the problem, three genetic operators, crossover, mutation and selection, are used to explore the search space. Crossover is usually used to explore the search space beyond a local optimum while mutation is usually used to improve the preliminary solution. Selection is the process to choose promising chromosomes from the current generation as the parent chromosomes in next generation. Fig 2.3 provides the flowchart of the GA algorithm.

2.2.2 Simulated Annealing Algorithm

Simulated Annealing (SA) is first proposed by Kirkpatrick (1983) inspired by the physical process of the annealing of solids. In the natural annealing process, first the solid is heated up to a high temperature. At that temperature all the molecules of the material have high energies and randomly arrange themselves into a liquid state. Then the temperature decreases at a certain rate which will reduce the molecules' energies and their freedom to arrange themselves. Finally, the temperature goes down to such a level that all the molecules lose their freedom to arrange themselves then the material crystallizes. During the annealing, if the temperature decreases at a proper rate, the material can obtain a regular internal structure at the minimum energy state. But if the temperature goes down too fast, the irregularities and defects will appear in the solid and the system will be at a local minimum energy state.

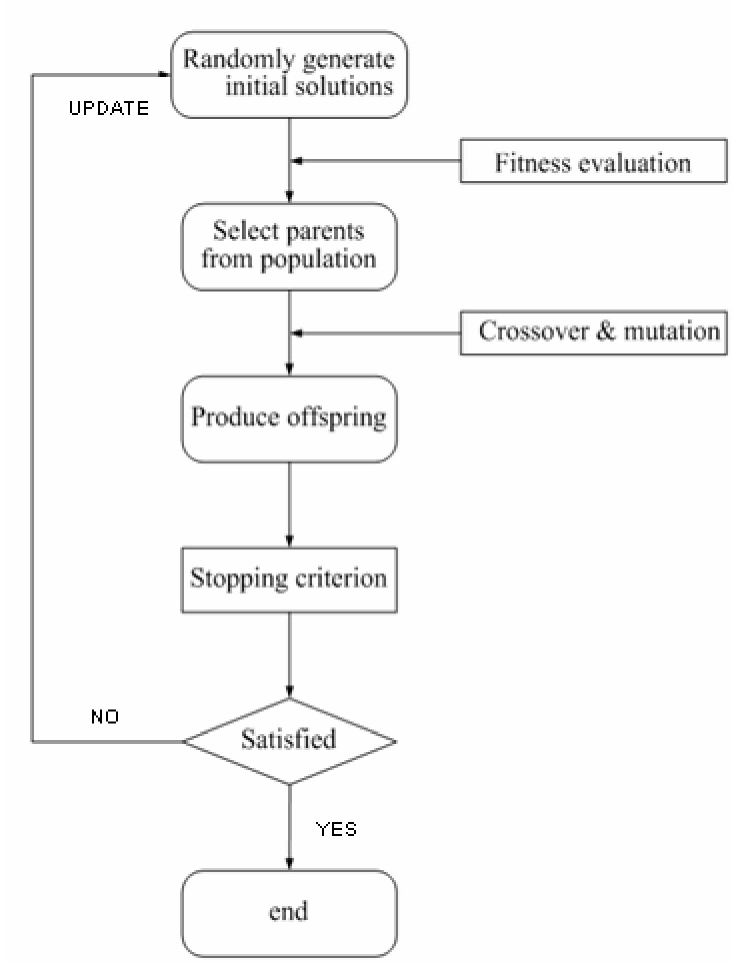


Figure 2.3 An Illustration of the Process of Genetic Algorithm.

In analogy to the annealing process, the feasible solutions of the optimization problem correspond to the states of the material, the objective function values computed at these solutions are represented by the energies of the states, the optimal solution to the problem can be viewed as the minimum energy state of the material and the suboptimal solutions correspond to the local minimum energy states. A flowchart of a typical SA algorithm is provided in figure 2.4.

There are two driving issues for the SA algorithm, acceptance criterion for the new solutions and the temperature update scheme. Metropolis' criterion is used as the acceptance criterion for the new solutions. In this criterion, a random number r in [0, 1] is generated from a uniform distribution and let Δ equal to the difference between the objective function values computed by the current solution and the new solution, then if $r \le e^{-\Delta/T}$, where T represents the current temperature, the new solution will be accepted to replace the current solution, otherwise it will be rejected.

As to the temperature update scheme, a number of rules have been proposed. A commonly used one is the geometric cooling rule. In this rule, the temperature will be updated as following,

$$T_{i+1} = cT_i, \qquad i = 0,1\cdots$$

where c is a constant smaller than but close to 1.

2.2.3 Tabu-search Algorithm

The tabu-search algorithm was developed independently by Glover (1986) and Hansen (1986) for solving combinatorial optimization problems. The algorithm is an iterative search approach characterized by the use of flexible memory. The three main components of a tabu-search algorithm are forbidding strategy, freeing strategy and short-term strategy (Glover (1989), Glover (1990)).

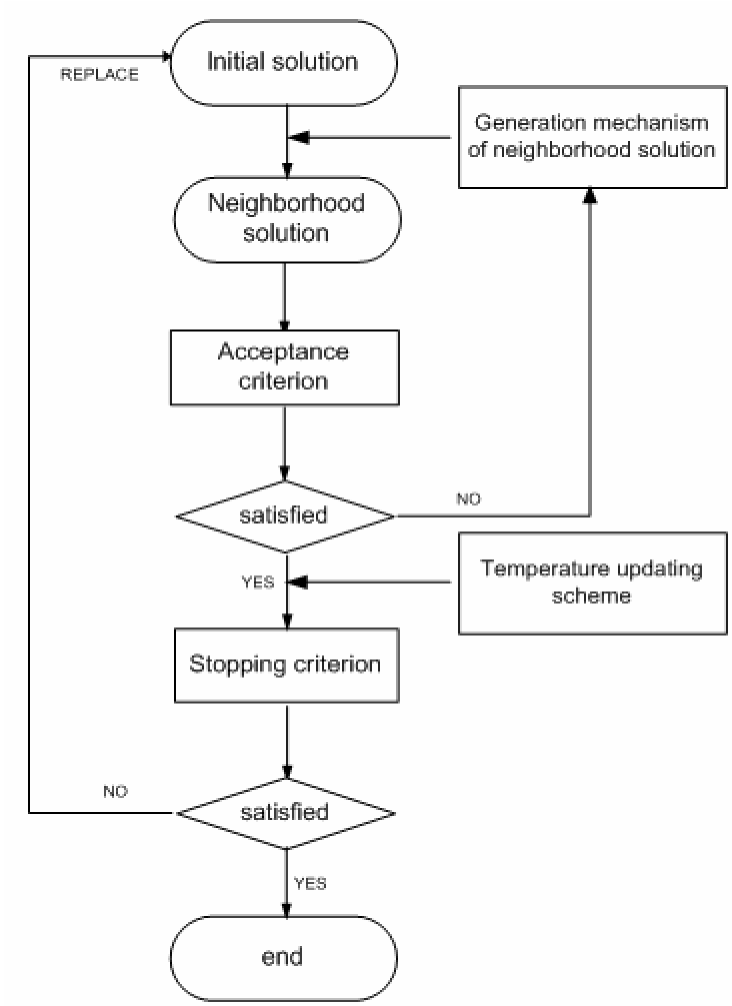


Figure 2.4 An Illustration of the Process of Simulated Annealing Algorithm.

The forbidding strategy is used to prevent the cycling problem occurred in search process by forbidding certain searching moves. The tabu list is constructed by registering the previous moves. Ideally the tabu list should record all the moves in previous iterations. However this might require too much memory space and computational effort. In practical use of tabu-search algorithm, normally only the moves occur in previous *n* iterations are stored in the tabu list and are therefore forbidden in the searching process. A critical problem here is to determine a proper value of *n*, which is also called the tabu list length or tabu list size. If the value is too small, the probability of cycling is high, while if it is too large, the search might be driven away from a good solution region before the region is completely explored. The freeing strategy controls which moves will be released from the tabu list. A first-in-first-out (FIFO) procedure is commonly used as the freeing strategy. In this procedure, once the tabu list is full each new move is written over the oldest move. The short term strategy, also called overall strategy, manages the interplay between the forbidding and freeing strategies. A flowchart of a standard tabu search algorithm is provided in figure 2.5.

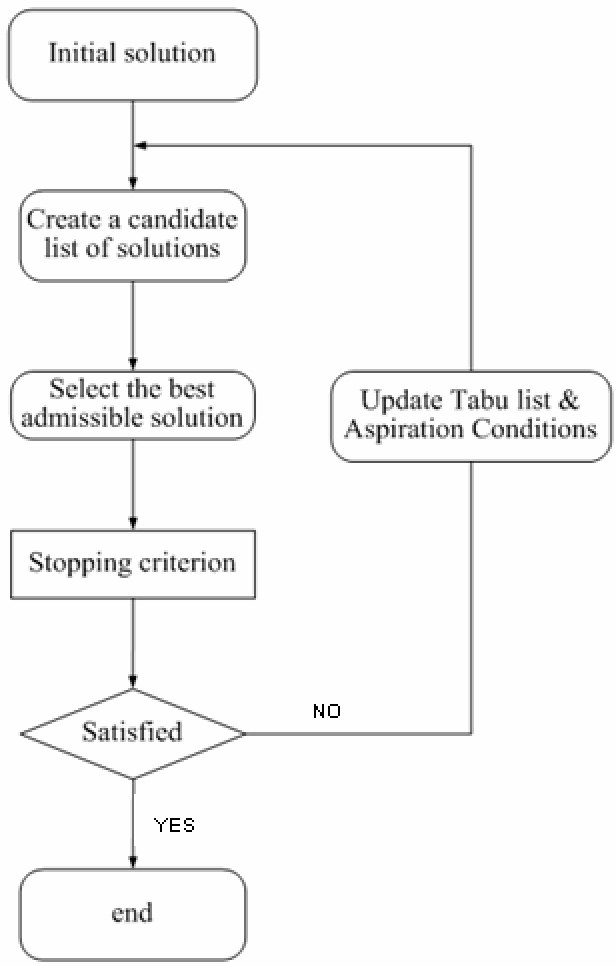


Figure 2.5 An Illustration of the Process of Tabu-Search Algorithm.

CHAPTER 3

SCHEDULING OF MULTIPLE YARD CRANE SYSTEMS (I)

3.1 INTRODUCTION

This chapter presents a preliminary work on the multiple yard crane scheduling (MYCS) problem. A simplified version of the MYCS problem, two yard crane system scheduling (TYCS) problem, is studied by confining each yard crane (YC) in its dedicated working range. The problem is formulated by a mathematical model and solved by a designed simulated annealing (SA) algorithm. The performance of the SA algorithm is also evaluated through numerical experiments. To ease the understanding of the problem formulation, a detailed description of the TYCS is provided in the following section.

3.2 TWO YARD CRANE SCHEDULING PROBLEM

Figure 3.1 briefly illustrates the loading operation in a container loading system using a two YC system. In the problem, the load plan of the quay crane (QC) and the container block plans are known beforehand. YC *A* and YC *B* are used to serve QC *A* at Block *1* and *2* respectively. They will perform the loading jobs according to the load plan of QC *A* together.

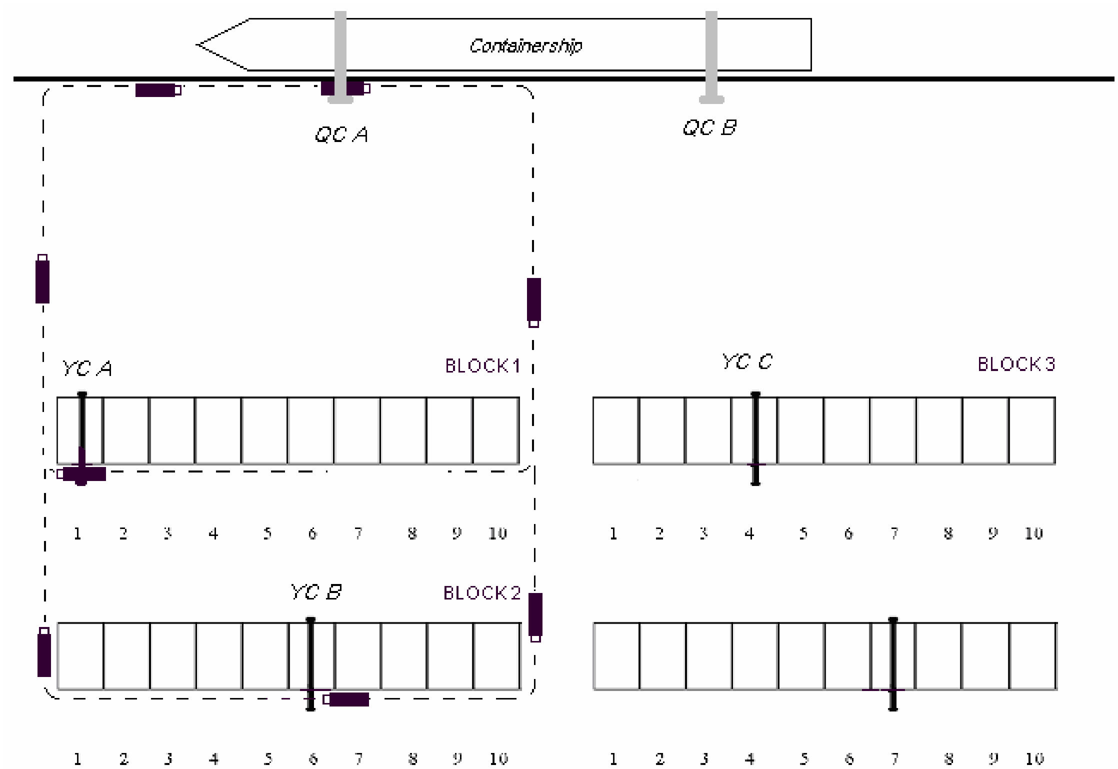


Figure 3.1 A Layout of a Container Loading System

Following is an example used to illustrate the problem.

Table 3.1 Quay Crane Load Plan

Sequence	1	2	3	4	5	6
Container type	A	C	B	A	C	B
Number of containers	20	18	22	24	30	26

Table 3.2 Plan of Container Block 1

Yard-bay number	1	2	3	4	5	6	7	8	9	10
Container type	A		B	C		C	A	B		A
Number of containers	8		15	10		15	8	12		4

Table 3.3 Plan of Container Block 2

Yard-bay number	1	2	3	4	5	6	7	8	9	10
Container type		B		A	C	A			C	B
Number of containers		11		10	8	14			15	10

Table 3.1 is a sample load plan of a quay crane which is also the loading sequence requirement of the containers. Table 3.2 and 3.3 are the container block plans which show where these containers are stacked in the container blocks. According to the load plan of

the quay crane, the YCs need to pick up 20 containers of type A together at first. One possible schedule of the two YCs could be YC A: 1(6) – 7(5); YC B: 4(4) – 6(5) (YC 1 first visits Yard-bay *1* and pick up 6 containers there then visits yard-bay *7* and pick up 5 containers. At the same time, YC 2 will visit Yard-bay *4* and *6* and pick up 4 and 5 containers respectively). Alternative schedules could be YC 1: 1(5) – 10(4); YC 2 6(8) – 4(3) and so on. After all the 20 containers of type A are picked up, the YCs then can start to work for sequence 2, picking up 18 containers of type C, and so on. It's obvious that different schedules of YCs will lead to different finishing time of the loading process.

The two decision factors in the problem are the yard-bay visiting sequences of the two YCs and the number of containers picked up at each visit. To decide the bay visiting sequences of the two YCs is actually to find the routing paths of the two YCs which can be represented on networks. Figure 3.2 is the sample network of YC A on which the numbers in each node are the bay numbers representing the location of the container bays. Thus to determine the bay visiting sequence of the YC is just to find a routing from node I to node F.

Since the loading jobs are distributed among the two YCs, making their working schedule dependent on one another, the schedules of the two YCs need to be coordinated to minimize the overall loading time.

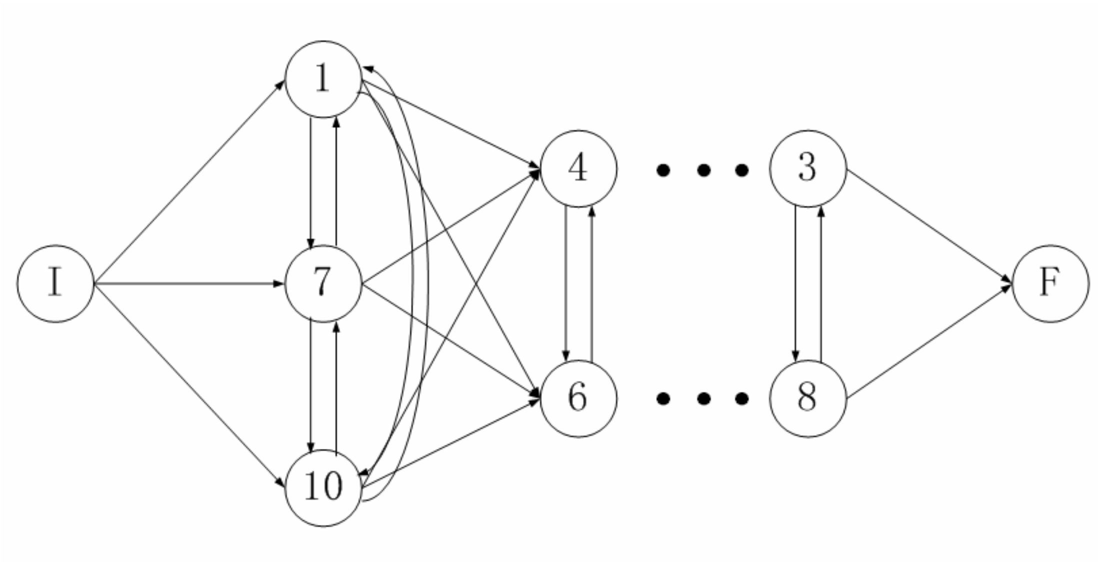


Figure 3.2 A Sample Network of the Routing of One YC

3.3 MATHEMATICAL FORMULATION

To simplify the mathematical model for the TYCS problem, three types of reasonable assumptions are first made.

i. There is only one type of container stacked in one yard-bay, which is the common practice in allocating space in the stack area of container terminals.

ii. The time required for an YC to load a container is assumed to be the same for all the containers despite the exact storage positions of individual containers.

iii. YCs will not travel between two blocks during the loading process.

To formulate the problem, a "sub-tour" (subsequence) is first defined as a sequence of containers that needs to be picked up together, which is according to load plan of the quay crane. A sub-tour represents a set of containers picked up by the YCs for one loading sequence of the quay crane.

The following notations are introduced to formulate the problem

Parameters

N_j the initial number of containers stacked at Yard-bay *j* in block 1

N'_j the initial number of containers stacked at Yard-bay *j* in block 2

s sub-tour number

r^s the number of containers requested in Sub-tour *s*

n the number of sub-tours for the whole loading process

m the number of container types

t_{As} the loading time of YC A in Sub-tour *s*

t_{Bs} the loading time of YC B in Sub-tour *s*

T_{As} the ending time of Sub-tour *s* for YC A

T_{Bs} the ending time of Sub-tour *s* for YC B

c_s the type of containers loaded in Sub-tour *s*

$B(c)$ the set of yard-bay numbers which contains containers of type *c* and are served by YC A

$B'(c)$ the set of yard-bay numbers which contains containers of type *c* and are served by YC B.

$S(c)$ the set of sub-tour numbers, for which the container type is *c*

I_A the initial location of YC A

I_B the initial location of YC B

F_A the final location of YC A

F_B the final location of YC B

Constants

T_D the travel time for YC to move per the distance of a bay

T_L the loading time of one container

Decision variables

z_{ij}^s =1 if YC A moves from Yard-bay *i* to *j* just before starting Sub-tour *s*

=0 otherwise

x_{ij}^s =1 if YC A moves from Yard-bay *i* to *j* during Sub-tour *s*

=0 other wise

w_{ij}^s =1 if YC B moves from Yard-bay *i* to *j* just before starting Sub-tour *s*

=0 otherwise

y_{ij}^s =1 if YC B moves from Yard-bay *i* to *j* during Sub-tour *s*

=0 otherwise

r_{jA}^s the number of containers picked up at Yard-bay *j* during Sub-tour *s* by YC A

r_{jB}^s the number of containers picked up at Yard-bay *j* during Sub-tour *s* by YC B

Loading time in one sub-tour

The loading time of YC A in Sub-tour *s* can be expressed by the following equation, the first two terms are the travel time before and during the sub-tour respectively and the second two terms are the handling time of containers.

$$t_{As} = \sum_{i \in B(c_{s-1}), j \in B(c_s)} T_D \cdot |i-j| \cdot z_{ij}^s + \sum_{i,\, j\, \in\, B(c_s)} T_D \cdot |i-j| \cdot x_{ij}^s + \sum_{i \in B(c_{s-1}), j \in B(c_s)} T_L \cdot r_{jA}^s \cdot z_{ij}^s + \sum_{i, j \in B(c_{s+1})} T_L \cdot r_{jA}^s \cdot x_{ij}^s \tag{3.1}$$

Similarly, equation (2) is the loading time of YC B in Sub-tour *s*

$$t_{Bs} = \sum_{i \in B'(c_{s-1}), j \in B'(c_s)} T_D \cdot |i-j| \cdot w_{ij}^s + \sum_{i,\, j\, \in\, B'(c_s)} T_D \cdot |i-j| \cdot y_{ij}^s + \sum_{i \in B'(c_{s-1}), j' \in B(c_s)} T_L \cdot r_{jB}^s \cdot w_{ij}^s + \sum_{i, j \in B'(c_s)} T_L \cdot r_{jB}^s \cdot y_{ij}^s \tag{3.2}$$

Ending time of Sub-tour *1*

Since Sub-tour *1* starts at time 0, the ending time of sub-tour *1* is the same as the loading time in the sub-tour, which can be represented as

$$T_{A1} = \sum_{i \in I_A, j \in B(c_1)} T_D \cdot |i-j| \cdot z_{ij}^1 + \sum_{i,\, j\, \in\, B(c_1)} T_D \cdot |i-j| \cdot x_{ij}^1 + \sum_{i \in I_A, j \in B(c_1)} T_L \cdot r_{jA}^1 \cdot z_{ij}^1 + \sum_{i, j \in B(c_1)} T_L \cdot r_{jA}^1 \cdot x_{ij}^1 \tag{3.3}$$

$$T_{B1} = \sum_{i \in I_B, j \in B'(c_1)} T_D \cdot |i-j| \cdot w_{ij}^1 + \sum_{i,\, j\, \in\, B'(c_1)} T_D \cdot |i-j| \cdot y_{ij}^1 + \sum_{i \in I_B, j \in B'(c_1)} T_L \cdot r_{jB}^1 \cdot w_{ij}^1 + \sum_{i, j \in B'(c_1)} T_L \cdot r_{jB}^1 \cdot y_{ij}^1 \tag{3.4}$$

Ending time relationship between two successive sub-tours

It is further defined that

$$\delta(x) = \begin{cases} 1 & x > 0 \\ 0 & x \le 0 \end{cases} \tag{3.5}$$

$$T_{\Delta s}^A = \min(|T_{As} - T_{Bs}|, T_D \cdot |i-j| \cdot z_{ij}^{s+1}) \quad i \in B(c_s), j \in B(c_{s+1}) \tag{3.6}$$

$$T_{\Delta s}^B = \min(|T_{As} - T_{Bs}|, T_D \cdot |i-j| \cdot w_{ij}^{s+1}) \quad i \in B'(c_s), j \in B'(c_{s+1}) \tag{3.7}$$

$T_{\Delta s}^A$ ($T_{\Delta s}^B$) is the time YC A(B) can possible save in Sub-tour *s+1* if it finishes the jobs in

Sub-tour *s* early than YC B(A).

Thus the ending time of the two YCs in sub-tour *s+1* can be formulated as follows,

$$T_{A(s+1)} = \delta(T_{As} - T_{Bs}) \cdot T_{As} + \delta(T_{Bs} - T_{As}) \cdot T_{Bs} + t_{A(s+1)} - \delta(T_{Bs} - T_{As}) \cdot T_{\Delta s}^{A} \quad (3.8)$$

$$T_{B(s+1)} = \delta(T_{As} - T_{Bs}) \cdot T_{As} + \delta(T_{Bs} - T_{As}) \cdot T_{Bs} + t_{B(s+1)} - \delta(T_{As} - T_{Bs}) \cdot T_{\Delta s}^{B} \quad (3.9)$$

Therefore the objective function can be interpreted in following equation, which is to minimize the later finishing time of the two YCs in the last sub-tour.

$$Min \ \max(T_{An}, T_{Bn}) \quad (3.10)$$

Subject to

$$\sum_{i \in I_A, j \in B(c_1)} z_{ij}^{1} = 1 \quad (3.11)$$

$$\sum_{i \in I_B, j \in B'(c_1)} w_{ij}^{1} = 1 \quad (3.12)$$

$$\sum_{i \in B(c_n), j \in F_A} z_{ij}^{n+1} = 1 \quad (3.13)$$

$$\sum_{i \in B'(c_n), j \in F_B} w_{ij}^{n+1} = 1 \quad (3.14)$$

$$\Big(\sum_{j \in B(c_{t-1})} z_{ji}^{s} + \sum_{k \in B(c_t)} x_{ki}^{s}\Big) - \Big(\sum_{j \in B(c_{t+1})} z_{ij}^{s+1} + \sum_{k \in B(c_t)} x_{ik}^{s}\Big) = 0 \quad i \in B(c_t), \ \ t = 1,2...n \quad (3.15)$$

$$\Big(\sum_{j \in B'(c_{t-1})} w_{ji}^{s} + \sum_{k \in B'(c_t)} y_{ki}^{s}\Big) - \Big(\sum_{j \in B'(c_{t+1})} w_{ij}^{s+1} + \sum_{k \in B'(c_t)} y_{ik}^{s}\Big) = 0 \quad i \in B(c_t), \ \ t = 1,2...n \quad (3.16)$$

$$\sum_{i \notin S, j \in S} x_{ij}^{s} \ge 1 \quad \forall S \subseteq B(c_s) \setminus \{0\}, \ \mathrm{S} \ne 0, \quad s = 1,2...n \quad (3.17)$$

$$\sum_{i \notin S, j \in S} y_{ij}^{s} \ge 1 \quad \forall \ S \subseteq B'(c_s) \setminus \{0\}, \ \mathrm{S} \ne 0, \quad s = 1,2...n \quad (3.18)$$

$$r_{jA}^{s} \le M\Big(\sum_{i \in B(c_{s-1})} z_{ij}^{s} + \sum_{k \in B(c_s)} x_{kj}^{s}\Big) \quad j \in B(c_s), \quad s = 1,2...n \quad (3.19)$$

$$r_{jB}^{s} \leq M(\sum_{j \in B'(c_{s-1})} w_{ij}^{s} + \sum_{k \in B'(c_s)} y_{kj}^{s}) \qquad j \in B'(c_s), \quad s = 1,2...n \tag{3.20}$$

$$\sum_{j \in B(c_s)} r_{jA}^{s} + \sum_{j \in B'(c_s)} r_{jB}^{s} = r^{s} \qquad s = 1,2...n \tag{3.21}$$

$$\sum_{s \in S(c)} r_{jA}^{s} = N_j \qquad \mathrm{c} = 1,2...m, \quad j \in B(c) \tag{3.22}$$

$$\sum_{s \in S(c)} r_{jB}^{s} = N'_j \qquad \mathrm{c} = 1,2...m, \quad j \in B'(c) \tag{3.23}$$

$$r_{jA}^{s} \geq 0, \qquad j \in B(c_s) \quad s = 1,2...n \tag{3.24}$$

$$r_{jB}^{s} \geq 0, \qquad j \in B(c_s) \quad s = 1,2...n \tag{3.25}$$

$$(\sum_{j \in B(c_{s-1})} z_{ji}^{s} + \sum_{k \in B(c_s)} x_{ki}^{s}) \leq 1 \qquad i \in B(c_s), \quad s = 1,2...n \tag{3.26}$$

$$(\sum_{j \in B'(c_{s-1})} w_{ji}^{s} + \sum_{k \in B'(c_s)} y_{ki}^{s}) \leq 1 \qquad i \in B(c_s), \quad s = 1,2...n \tag{3.27}$$

, M is a big positive number.

Constraints (3.11) to (3.16) are the flow conservation constraints. A feasible solution of one YC corresponds to a path from its source node to its terminal node: (3.11) and (3.12) are the outflow constraints at the source node; (3.13) and (3.14) are the inflow constraints at the terminal node: (3.15) and (3.16) are the flow conservation constraints for the other nodes. Constraints (3.17) and (3.18) are to ensure the connectivity of the solutions, which eliminate the isolated cycles form the solution set. Constraints (3.19) and (3.20) are to ensure that only when YC visits a bay can it pick up containers there, where M is a sufficient large number. Constraint (3.21) guarantee that the number of containers picked up in one sub-tour is equal to the number of containers requested by the load plan. Constraints (3.22) to (3.25) are to ensure that the total number of containers picked up at

Bay j is equal to the initial number of containers stored at Bay j. Constraints (3.26) and (3.27) are to ensure that the YC can only visit one bay at most once in one sub-tour which characterizes the optimal solution of the TYCS problem.

3.4 SIMULATED ANNEALING ALGORITHM FOR TYCS PROBLEM

It has been proven in literatures that the single YC scheduling problem is an NP-complete problem. Needless to say, the TYCS problem is also an NP-complete problem which makes exact algorithm not practical to solve the large scale cases. Hence, heuristic algorithms are required to solve the TYCS problem efficiently. In this study, simulated annealing (SA) algorithm, one of the commonly used meta-heuristics, are used to solve the proposed TYCS problem.

3.4.1 Encoding Method

To use SA algorithm in solving the TYCS problem, an encoding method to represent the feasible solutions is first introduced. The feasible solutions for the TYCS problem are coded into strings of integer numbers in this study. Each string consists of certain number of sections according to the number of sub-tours and each section includes four subsections, first two subsections indicating the bay visiting sequence of YC A and the number of containers picked up by it at each visit, last two subsections indicating the information of YC B.

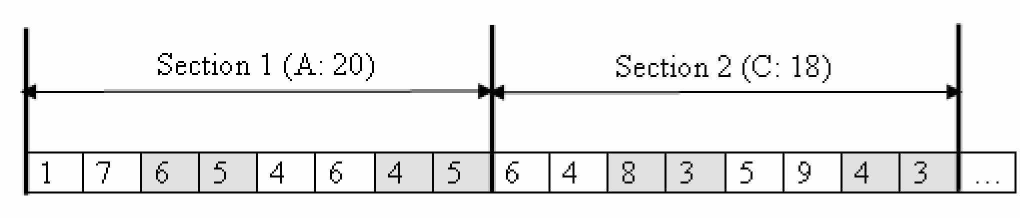


Figure 3.3 A Sample of Part of the Feasible Solution

Figure 3.3 is an example of part of the feasible solution, which contains two sections corresponding to Sub-tour 1 (picking up 20 containers of type A) and Sub-tour 2 (picking up 18 containers of type C) respectively. In Section 1, the first two subsections mean that YC A will visit bays in the sequence of 1-7 and pick up 6 and 5 containers at each visit accordingly. While the last two subsections show that at the same time YC B will visit bays in the sequence of 4-6 and pick up 4 and 5 containers at each visit. After both the two YCs finish their work in Section 1, they will start to work for the Section 2.

3.4.2 Generation Mechanism of Neighborhood Solution

To implement the SA algorithm, we need to generate a sequence of iterations, of which each is composed of changing the current solution in a designed way to create a neighborhood solution.

The Generation mechanism of neighborhood solution deployed here is as follows: A cut point is randomly chosen among the points between the first and the last string of sections then all the elements behind the cut point are regenerated according to the constraints. Figure 3.4 illustrates this process: A cut point is chosen after the Section 1, then all the sections after the cut point will be regenerated.

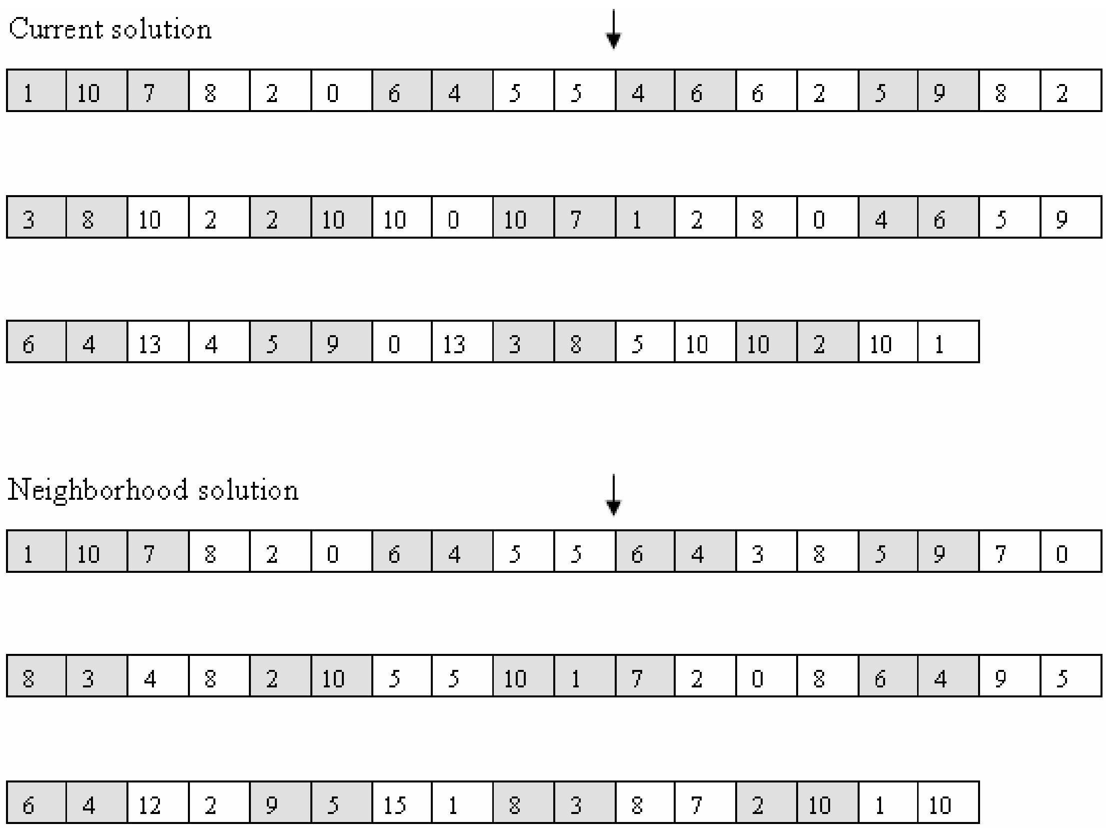


Figure 3.4 An Illustration of the Generation Mechanism of Neighborhood Solutions

3.4.3 Acceptance Criterion for the Neighborhood Solution

Once a neighborhood solution is generated, the following criterion is adopted to judge whether to accept it or not.

Let $\Delta = f(s) - f(s_0)$ (3.28)

s_0 represents the current solution and s represents the neighborhood solution generated from current solution.

$f(*)$ represents the objective function value computed from the solution $(*)$.

A random number r in $[0,1)$ is generated from a uniform distribution and if

$r \le e^{-\Delta/T_i}$, T_i represents the current temperature (3.29)

Then the neighborhood will be accepted as the current solution. If not, the current solution will remain unchanged.

3.4.4 Temperature Updating Scheme

The temperature updating scheme adopted here is first introduced by Lundy and Mees (1986), which outperforms the commonly used geometric updating scheme

$$T_{i+1} = cT_i, \quad i = 0,1\ldots \tag{3.30}$$

in a preliminary numerical experiment. In this scheme the temperature is updated by the following formula:

$$T_{i+1} = \frac{T_i}{1+\beta T_i}, \quad i = 1,\ldots,K-1 \tag{3.31}$$

Where β is the rate parameter in terms of the initial temperature, T_1, stopping temperature, T_K and iteration number, K.

$$\beta = \frac{T_1 - T_K}{(K-1)T_1 T_K} \tag{3.32}$$

3.4.5 Stopping Criterion

The stopping temperature, T_K, and iteration number, K are used to control the stoppage of the SA process.

3.5 NUMERICAL EXAMPLES

3.5.1 Experiment Design

To measure the performance of the proposed SA algorithm, 40 sample problems are first generated as follows,

1) Generate the load plan of the quay crane:
    a) The total number of containers for each problem ranges from 200 to 600. 10 sample problems are randomly generated for each interval of 100 (e.g. 200 – 300, 300 – 400, …).
    b) For each sample problem, the containers are randomly classified into five types, namely A, B, C, D and E
    c) Each type is then further divided into 2 or 3 groups.
    d) The load plan of the quay crane is finally generated by joining these groups in a random sequence.
2) Allocate the containers required by the quay crane in the stack area: Containers are randomly allocated in two container blocks, each of which consists of 25 yard-bays subjected to the constraint that only one type of container can be stacked in one yard-bay

Computer programs are written in C++ language to perform the numerical test of the SA algorithm. All the programs are executed on a DELL P IV (3.0GHz) PC and are completed within one minute.

3.5.2 Solution Sensitivity to SA Parameters

It is found through a preliminary test that 1,000,000 is a proper value of the initial temperature, T_1. The tested value of stopping temperature, T_K are 0.1, 0.5, 1, 2, 3, 4, 5 and the tested value of iteration number, K are 5,000, 10,000, 20,000, 50,000, 60,000. A sample problem is solved using different combinations of the two parameters so that the best combination could be found.

Figure 3.5 illustrates the average loading time (ALT) obtained by SA algorithm with different combinations of parameters. Figure 3.6 illustrates the best results of loading time obtained by SA algorithm with different combinations of parameters.

It is noted that T_K =0.5 and K =50,000 is the best performed combination of SA parameters both in the average and best objective function value tests. Thus this set of parameters is used to solve the other sample problems and the results obtained are used to compare against the estimated lower bound.

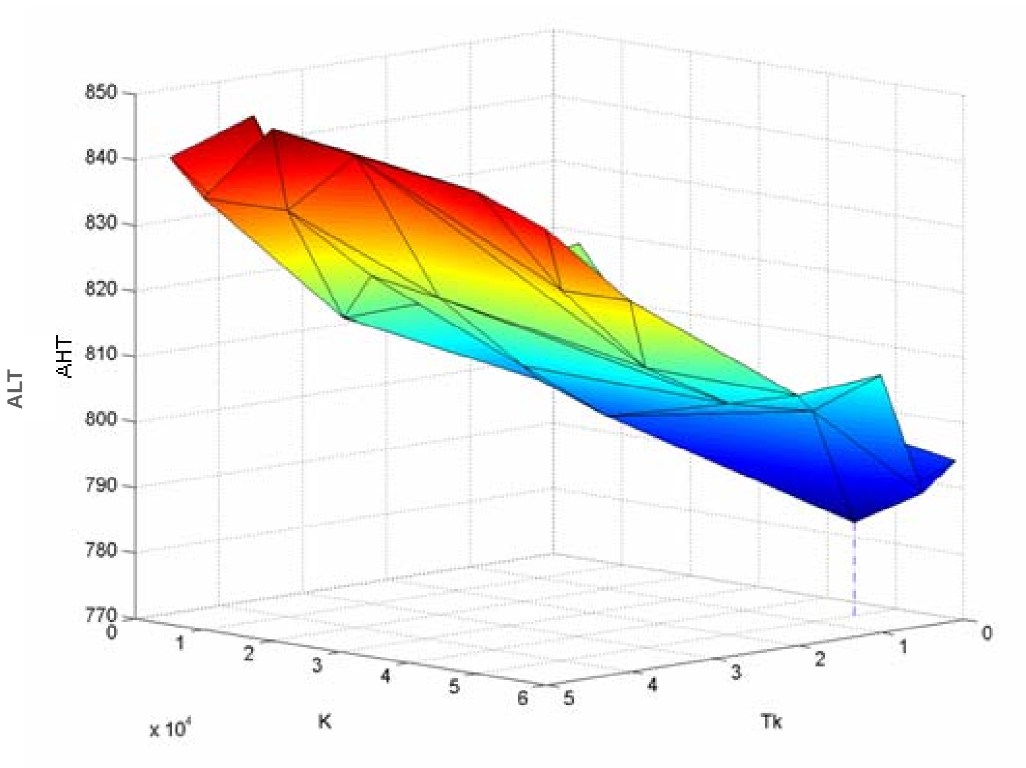


Figure 3.5 The Average Loading Time for Different Values of Parameters

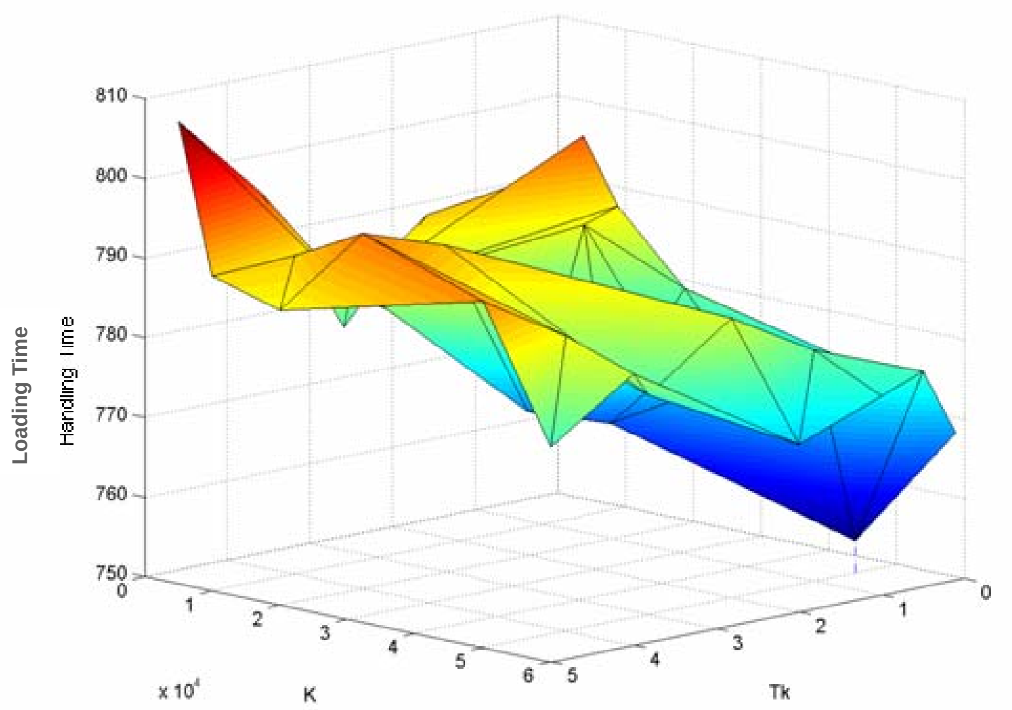


Figure 3.6 The Shortest Loading Time for Different Values of Parameters

3.5.3 Lower Bound Estimation and Results Comparison

The minimum loading time of one container block can be calculated as follows,

$$\text{TIME}_{1(2)} = \text{Min travel time}_{1(2)} + \text{Min handling time}_{1(2)}$$

Where,

$\text{TIME}_{1(2)}$ is the minimum loading time of block 1 (2)

$\text{Min travel time}_{1(2)} = T_D \times$ the total length of block 1 (2)

$\text{Min handling time}_{1(2)} = T_L \times$ the total number of containers in block 1 (2)

Thus, the lower bound of the operation time (T_{LB}) of the whole container loading process can be obtained by taking the greater one of TIME_1 and TIME_2

To evaluate the performance of the proposed SA algorithm, 40 sample problems are generated. The results obtained from SA (T_{SA}) are compared with the estimated lower bound, which is illustrated in table 3.4.

Table 3.4 Performance of the SA Algorithm

No. of containers	$(T_{SA} - T_{LB}) / T_{LB} \times 100\%$		
	Max	Min	Mean
200-300	14.83%	8.19%	9.19%
300-400	11.95%	5.79%	10.08%
400-500	14.29%	6.22%	10.64%
500-600	14.78%	6.67%	10.20%

On average, the result obtained from simulated annealing algorithm is 10.03% worse than the estimated lower bound. Considering the bounds are estimated in a very loose way, the results of SA is quite satisfactory. It is also noted that the performance of the proposed SA algorithm is independent on the number of containers loaded.

3.6 SUMMARY

In this chapter a simplified version of the MYCS problem, TYCS problem, is investigated. A mathematical formulation for the problem is provided. Also, a SA algorithm is proposed to solve the problem. In order to evaluate the performance of the SA algorithm, numerical experiments are performed with a number of generated test examples. The computational results show that the completion time found by the SA is on average 10.03% above the lower bound and the performance of the algorithm is irrelevant to the number of containers loaded.

CHAPTER 4

SCHEDULING OF MULTIPLE YARD CRANE SYSTEMS (II)

4.1 INTRODUCTION

This chapter extends the study in the previous chapter to a more general case in which, multiple YCs are used to load a sequence of containers from one or more container blocks. An integer programming model is developed for the problem formulation. In the model, the work schedules of different YCs are decided simultaneously to minimize the loading time. It is noted that the YC scheduling problem is NP-complete by nature. This research develops a greedy heuristic and a Simulated Annealing algorithm to solve the proposed model. The performance of the two algorithms is illustrated through presented numerical examples.

4.2 MULTIPLE YARD CRANE SCHEDULING PROBLEM

In a multiple yard crane scheduling (MYCS) problem, several YCs will work at several container blocks to serve one QC. Figure 4.1 shows a typical container loading system using multiple YC systems in which YC 1, 2 and 3 are working at two blocks (Block 1 and 2) to serve QC 1 at the same time. Hence they will perform the loading jobs according to the load plan of QC 1. The main character which distinguishes the MYCS problem from TYCS problem is that the YCs in MYCS problem are free to travel between the container blocks. Following is an example used to illustrate the MYCS problem.

Table 4.1 A Sample Load Plan

Sequence	1	2	3	4	5	6
Container type	A	C	B	A	C	B
Number of containers	20	18	22	24	30	26

Table 4.2 Sample container block plans

Block Plan (Block 1)

Yard-bay number	1	2	3	4	5	6	7	8	9	10
Container type	A		B	C		C	A	B		A
Number of containers	8		15	10		15	8	12		4

Block Plan (Block 2)

Yard-bay number	1	2	3	4	5	6	7	8	9	10
Container type		B		A	C	A			C	B
Number of containers		11		10	8	14			15	10

Table 4.1 provides a sample load plan of a QC which is also the requirement of container loading sequence for the YCs. Table 4.2 is the block plans of the two container blocks which show where these containers are stacked. According to the load plan of the QC, the YCs need to pick up 20 containers of type A together at the two blocks. When all the 20 containers of type A are picked up, the YCs then will start to work for sequence 2, picking up 18 containers of type C, and so on.

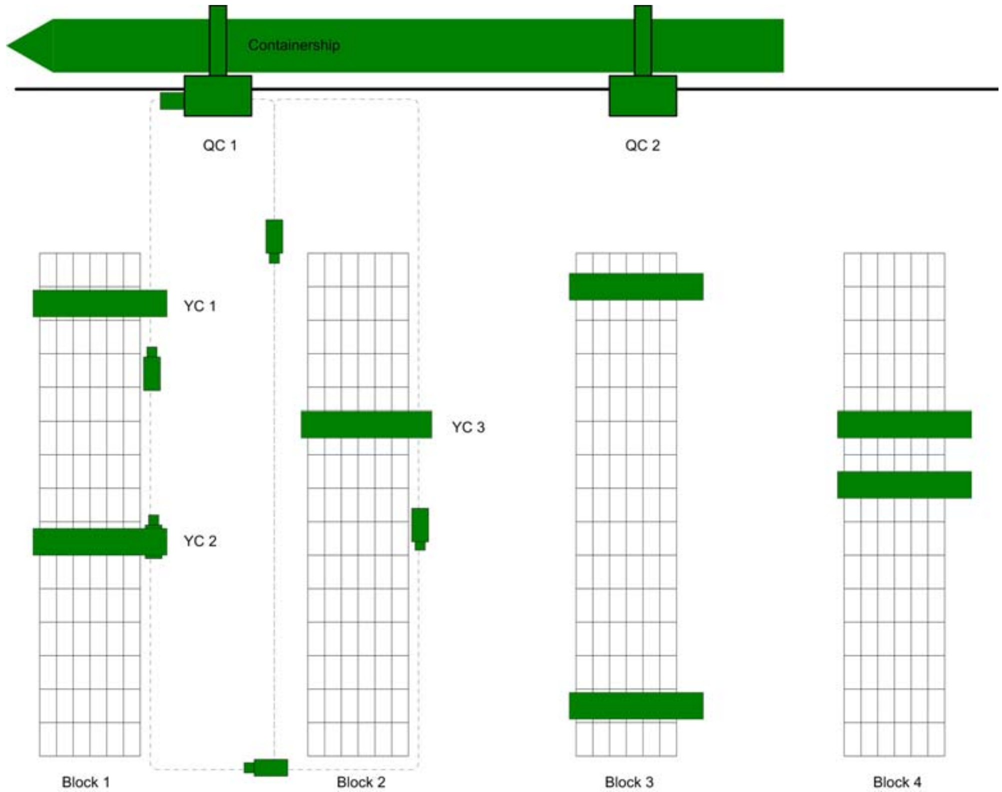


Figure 4.1 A Layout of a Container Loading System.

4.3 PROBLEM FORMULATION

To simplify the mathematical model formulation, two reasonable assumptions are introduced.

i. Only one type of container is stacked in one yard-bay, a common practice of allocating space in YC based container terminals.

ii. Despite the exact storage positions of individual containers, the loading time for all the containers is assumed to be the same.

An integer programming formulation is proposed to model the problem. A "sub-tour" (subsequence) is defined in the same way as in Chapter 3. This definition implies only when the YCs finished all the loading jobs for one sub-tour can they start to work for the posterior sub-tour.

The upper bound for the total loading time of the optimal YC scheduling is assumed to be known and this upper bound is partitioned into T time units. One time unit is defined as the time required for a yard crane traversing the distance of one yard-bay. The handling time of one container, T_H , is taken to be a multiple of this time unit.

As mentioned before, yard cranes may work at different container blocks. To facilitate the problem formulation, we join these blocks with some virtual yard-bays to get an integrated container block. The virtual yard-bays are generated in such a way that the time needed for a YC to traverse these yard-bays is the same as the time needed to travel from one container block to the other. Thus all the YCs will work on this merged block. All the *B* yard-bays in the merged block are renumbered *1* to *B* from left to right. Figure 4.2 illustrates the joining process.

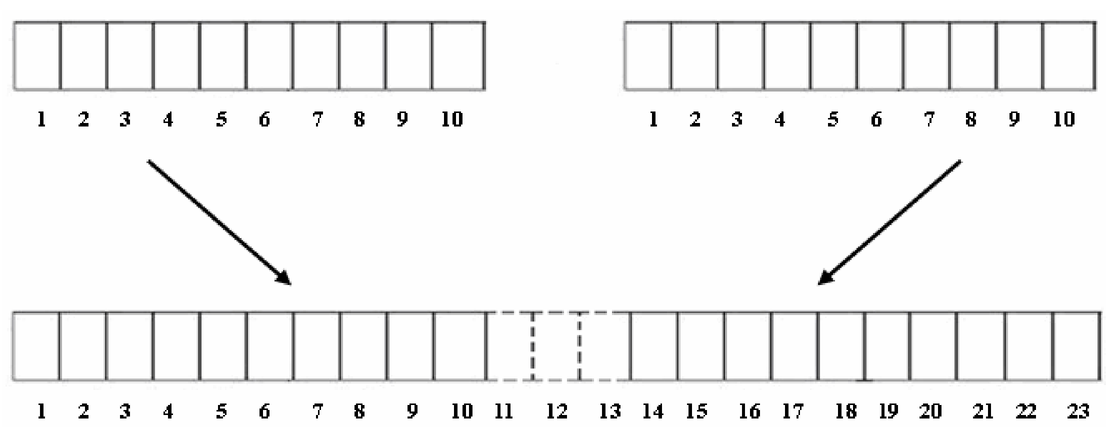


Figure 4.2 An Illustration of Joining Two Blocks

The *K* YCs are numbered *1* to *K* from left to right according to their initial location at period *0*. Since the YCs are of same size, so they cannot pass each other which implies that YC *k* can only move in the range limited by the locations of YC *k-1* and *k+1*.

The following notations are used to formulate the MYCS problem.

$$X_{i,j,k,t} = \begin{cases} 1 & \text{if Yard crane } k \text{ finishes loading one container for} \\ & \text{Sub-tour } i \text{ at Yard-bay } j \text{ at time } t \\ 0 & \text{otherwise} \quad \text{(a decision variable)} \end{cases}$$

$$Y_{j,k,t} = \begin{cases} 1 & \text{if Yard crane } k \text{ is at Yard-bay } j \text{ at time } t \\ 0 & \text{otherwise} \quad \text{(a decision variable)} \end{cases}$$

N_i the number of containers needed to pick up for Sub-tour *i*

C_j the number of containers stacked at Yard-bay *j* before the loading process starts

B (i) the set of yard-bays where the containers required by Sub-tour *i* are located

S the number of sub-tours for the whole loading process

I_k the initial position of yard crane *k*

F_k the final position of yard crane *k*

The objective function is to minimize the loading time of the containers, which can be represented by the following equation,

$$\text{Minimize } \max(tX_{i,j,k,t}) \tag{4.1}$$

Subject to

$$\sum_{i=1}^{S}\sum_{k=1}^{K}\sum_{t=1}^{T} X_{i,j,k,t} = C_j \quad j = 1,2\cdots,B \tag{4.2}$$

$$\sum_{j=1}^{B}\sum_{k=1}^{K}\sum_{t=1}^{T} X_{i,j,k,t} = N_i \quad i = 1,2\cdots,S \tag{4.3}$$

$$\sum_{j\in B(i-s)}\sum_{k=1}^{K}\sum_{t=1}^{a-1}X_{i-s,j,k,t}-N_{i-s}\le M(1-X_{i,j',k',a})$$

$$i=2,3\cdots S;\ s=1,2\cdots i-1;\ j'\in B(i);\ k'=1,2\cdots K;\ a=T_H+1,T_H+2\cdots T \tag{4.4}$$

$$\sum_{j\in B(i-s)}\sum_{k=1}^{K}\sum_{t=1}^{a-1}X_{i-s,j,k,t}-N_{i-s}\ge M(X_{i,j',k',a}-1)$$

$$i=2,3\cdots S;\ s=1,2\cdots i-1;\ j'\in B(i);\ k'=1,2\cdots K;\ a=T_H+1,T_H+2\cdots T \tag{4.5}$$

$$\sum_{i=1}^{S}\sum_{j=1}^{B}X_{i,j,k,t}\le 1 \quad k=1,2\cdots,K;\ t=1,2\cdots,T \tag{4.6}$$

$$\sum_{i=1}^{S}\sum_{j=1}^{B}\sum_{a=1}^{T_H-1}X_{i,j,k,t-a}\le M(1-X_{i,j,k,t})$$

$$i=1,2\cdots,S;\ j=1,2\cdots,B;\ k=1,2\cdots,K;\ t=1,2\cdots,T \tag{4.7}$$

$$\sum_{a=0}^{T_H}Y_{j,k,t-a}-T_H-1\ge M(X_{i,j,k,t}-1)$$

$$i=1,2\cdots,S;\ j=1,2\cdots,B;\ k=1,2\cdots,K;\ t=T_H+1,T_H+2\cdots,T \tag{4.8}$$

$$M(1-Y_{b,k,t})\ge\sum_{j=b}^{B}Y_{j,k-1,t} \quad b=1,2\cdots,B;\ k=2,3\cdots,K;\ t=1,2\cdots,T \tag{4.9}$$

$$\sum_{j=1}^{B}Y_{j,k,t}=1 \quad k=1,2,\cdots,K;\ t=1,2\cdots,T \tag{4.10}$$

$$\sum_{k=1}^{K}Y_{j,k,t}\le 1 \quad j=1,2\cdots,B;\ t=1,2\cdots,T \tag{4.11}$$

$$\sum_{j=b-1}^{b+1}Y_{j,k,t-1}\ge Y_{b,k,t} \quad b=1,2\cdots,B;\ k=1,2\cdots,K;\ t=2,3\cdots,T-1 \tag{4.12}$$

$$\sum_{j=b-1}^{b+1}Y_{j,k,t+1}\ge Y_{b,k,t} \quad b=1,2\cdots,B;\ k=1,2\cdots,K;\ t=2,3\cdots,T-1 \tag{4.13}$$

$$Y_{I_k,k,1}=1 \quad k=1,2\cdots,K \tag{4.14}$$

$$Y_{F_k,k,1} = 1 \quad k = 1,2\cdots,K \tag{4.15}$$

where M is a big positive number.

Constraints (4.2) ensure the number of containers picked up during the whole loading process at one yard-bay equals to the initial number of containers stacked in that yard-bay. Constraints (4.3) ensure the number of containers picked up during one sub-tour equals to the number required by the load plan. Constraints (4.4) and (4.5) ensure that the YCs must finish the loading jobs for all the previous sub-tours before they can start to work for the next sub-tour. Constraints (4.6) ensure the YC can at most handle one container for one period. Constraints (4.7) ensure that the YC cannot finish any loading jobs during the time interval $t - T_H - 1$ to *t-1* if it completes one loading job at period *t*. Constraints (4.8) ensure during loading one container the YC will stay at the container location throughout the operation. Constraints (4.9) ensure the movement of the YCs is free of inter-YC interference. Constraints (4.10) state that one YC can only be at one yard-bay during one period. Constraints (4.11) state that only one YC can be at one yard-bay in each period. Constraints (4.12) and (4.13) ensure that the YC can only move one yard-bay during one period. Constraints (4.14) and (4.15) state the initial and final positions of the K YCs.

4.4 HEURISTIC APPROACHES

4.4.1 A Greedy Heuristic

A greedy heuristic is proposed in this section to solve the MYCS problem. For the sake of brevity, a system of two YCs is used to illustrate this heuristic approach. The scheduling rules of this heuristic are as follows:

Rule 1

Both the two YCs will choose the containers in their nearest yard-bays, which satisfy the loading sequence requirement.

Rule 2

If the same yard-bay is identified to be the closest yard-bay to both YC *1* and YC *2* and it is also the last yard-bay of containers for the current subtour, it will be assigned to the closer YC. In the case where two YCs are of equal distance to the yard-bay, the yard-bay will be assigned to one YC arbitrarily.

Rule 3

If the same yard-bay is identified to be the closest yard-bay to both YC *1* and YC *2* and it is not the last yard-bay of containers for the current subtour, following five scenarios (see figure 4.3) are the only situations that can occur.

For the purpose of clarity, the area between the two cranes is referred as the interior area, while the extreme sides of the two cranes are referred as YC *1* and YC *2*'s exterior areas respectively illustrated in figure 4.4.

Figure 4.4 Definition of the interior and exterior areas

a) The closest yard-bay (Yard-bay *a*) is at the YC *2*'s exterior area, and there is no other yard-bay available at YC *1*'s exterior area: YC *1* will pick up containers at Yard-bay *a* and YC *2* will pick up containers at Yard-bay *b*.

b) The closest yard-bay (Yard-bay *a*) is at the YC *2*'s exterior area, and there is a yard-bay (Yard-bay *b*) available at YC *1*'s exterior area: YC *1* will pick up containers at Yard-bay *b* and YC *2* will pick up containers at Yard-bay *a*.

c) The closest yard-bay (Yard-bay *a*) is at the interior area with YC *1* being the closer crane and there is no yard-bay available at YC *1*'s exterior area: YC *1* will pick up containers at Yard-bay *a* and YC *2* will pick up containers at Yard-bay *b*.

d) The closest yard-bay (Yard-bay *a*) is at the interior area with YC *1* being the closer crane and there is a yard-bay (Yard-bay *b*) available at YC *1*'s exterior area: YC *1* will pick up containers at Yard-bay *b* and YC *2* will pick up containers at Yard-bay *a*.

e) The closest yard-bay (Yard-bay *a*) is at the interior area with YC *1* and YC *2* being of equal distance to it and there is a yard-bay (Yard-bay *b*) available at YC *2*'s exterior area: YC *1* will pick up containers at Yard-bay *a* and YC *2* will pick up containers at Yard-bay *b*.

<u>Rule 4</u>

In the case where there are two yard-bays of containers are equal distant to one YC, the YC will choose the yard-bay which is further from the other YC.

<u>Rule 5</u>

If there is no available container for an YC, it will stay still.

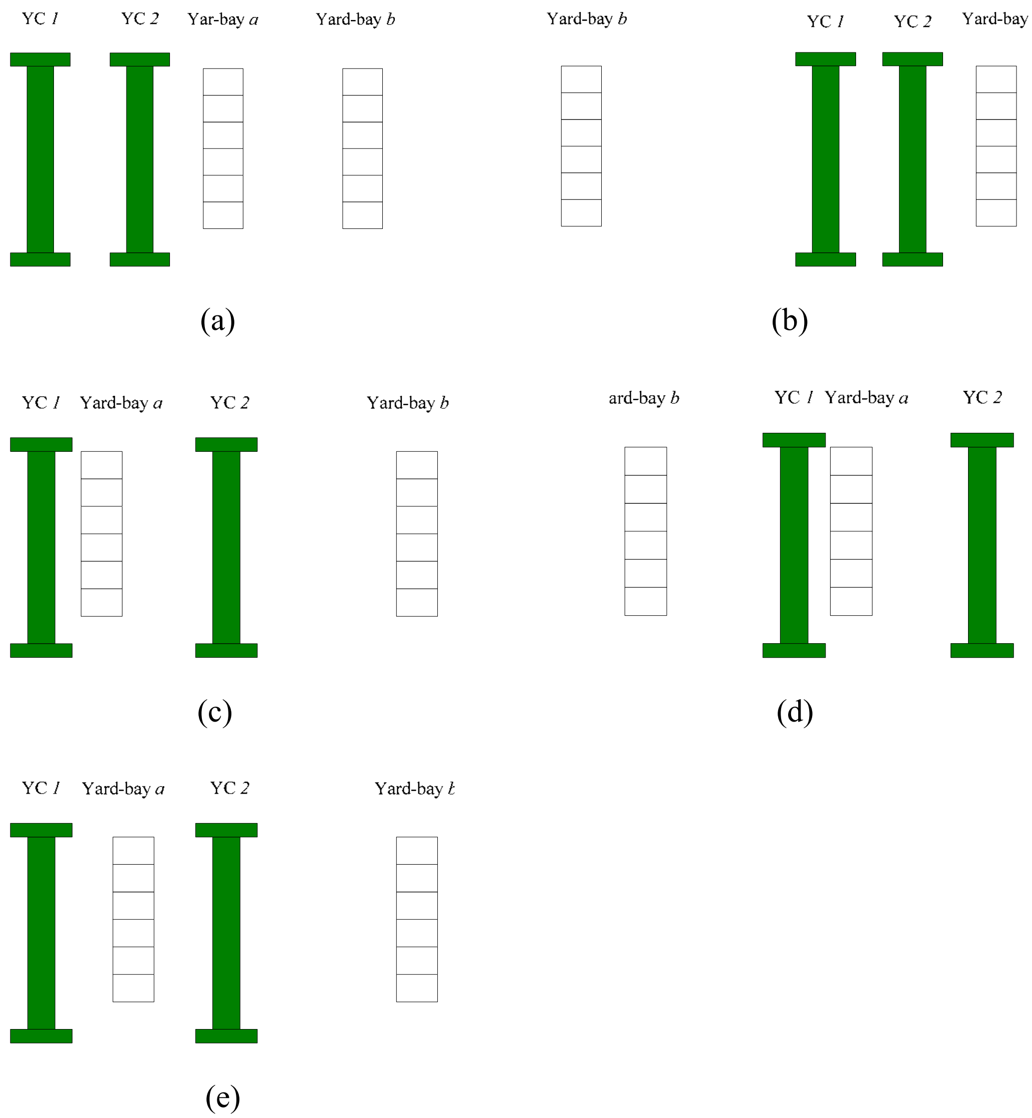


Figure 4.3 Position Relationships between YCs and Their Closest Containers

4.4.2 Simulated Annealing Algorithm

It is found through a previous research that a simulated annealing (SA) algorithm is an efficient method in solving the scheduling problem of the multiple YC system without inter-crane interference. Therefore we also apply the SA algorithm to solve the MYCS problem.

4.4.2.1 Solution representation

To use SA algorithm in solving the MYCS problem, a method of encoding the feasible solutions is first introduced.

The feasible solutions for the MYCS problem are represented by strings of integer numbers. Each string consists of several sections according to the number of sub-tours and each section includes four sectors, first two sectors indicating the visiting sequence of yard-bays of YC *1* and the number of containers picked up by it at each visit, last two sectors indicating the information of YC *2*.

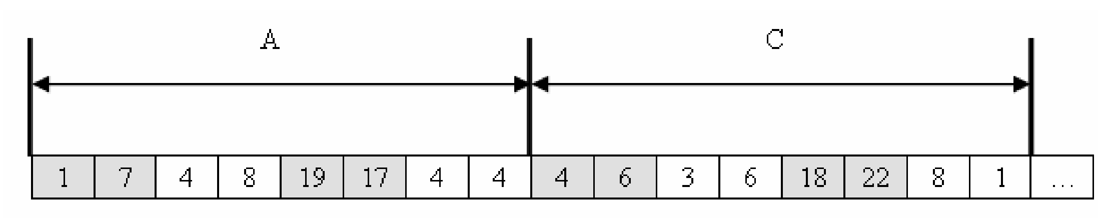


Figure 4.5 A Sample Part of a Feasible Solution.

Figure 4.5 is an example of part of the feasible solution, which contains two sections corresponding to Sub-tour *1* and Sub-tour *2* respectively. In Section *1*, the first two sectors mean that YC *1* will visit yard-bays in the sequence of 1-7 and pick up 4 and 8 containers at each visit accordingly. While the last two sectors show that YC *2* will visit yard-bays in the sequence of 19-17 and pick up 4 containers at each visit. After both the two YCs finish their work in Section *1*, they will start for work in Section *2*.

4.4.2.2 Generation mechanism of neighborhood solution.

To implement the SA algorithm on the MYCS problem, we need to generate a sequence of iterations, of which each is composed of changing the current solution in a designed way to create a neighborhood solution. The Generation mechanism of neighborhood solution employed here is the same as the method in previous chapter.

4.4.2.3 Acceptance criterion for the neighborhood solution

Once a neighborhood solution is generated, the following criterion is adopted to judge whether to accept it or not.

Let $\Delta = f(s) - f(s_0)$ (4.16)

s_0 represents the current solution and s represents the neighborhood solution generated from current solution.

$f(*)$ represents the objective function value computed from the solution $(*)$.

A random number r in $[0,1)$ is generated from a uniform distribution and if

$r \le e^{-\Delta/T_i}$, T_i represents the current temperature (4.17)

Then the neighborhood will be accepted as the current solution. If not, the current solution will remain unchanged.

4.4.2.4 Temperature updating scheme

The temperature updating scheme adopted here is the same as the scheme used in previous chapter. This scheme is shown to outperform the commonly used geometric updating scheme in a preliminary numerical experiment.

$$T_{i+1} = cT_i, \quad i = 0,1\ldots \tag{4.18}$$

In this scheme the temperature is updated by the following formula:

$$T_{i+1} = \frac{T_i}{1+\beta T_i}, \quad i = 1,\ldots,K-1 \tag{4.19}$$

Where β is the rate parameter in terms of the initial temperature, T_1 , stopping temperature, T_K and iteration number, K .

$$\beta = \frac{T_1 - T_K}{(K-1)T_1 T_K} \tag{4.20}$$

4.4.2.5 Stopping criterion

The stopping temperature, T_K , and iteration number, K are used to control the stoppage of the SA process.

4.4.3 Tabu Search Algorithm

Tabu search (TS) algorithm is a commonly used solution technique to solve combinatorial optimization problems. It had been shown to be efficient in solving many difficult optimization problems in the literatures. Therefore, we also adopted this approach to solve the MYCS problem.

The solution encoding method and the generation mechanism of neighborhood solution used in the TS algorithm are the same as methods used in SA algorithm. To construct a tabu list, a move first is defined as the position of the cutting points chosen in generating neighborhood solutions. Then a number of moves are recorded in the tabu list.

A first-in-first-out (FIFO) strategy is employed as the freeing strategy. In this procedure, once the tabu list is full each new move is written over the oldest move. The other way to free a move from the tabu list is controlled by the aspiration conditions. The aspiration conditions used here is that if a move generates a better solution than all the best solutions obtained so far, it will be accepted and freed from the tabu list. To determine a proper number of moves to be stored in the tabu list, which is also called tabu list length, a preliminary numerical experiment is conducted. Through the experiment, it is found that “three” is a proper length of the tabu list.

4.5 NUMERICAL EXPERIMENTS

4.5.1 Sensitivity Analysis of SA Parameters

To use SA to solve the MYCS problem, a sensitivity analysis of SA parameters is conducted in advance. It is found though a rudimentary experiment that 10,000 is a proper value of the initial temperature T_1. Then the other two parameters, iteration number K and stopping temperature T_K are tested by solving sample problems. The tested values of the iteration number are 500, 1000, 1500, …, 5000 and the tested values of stopping temperature are 0.1, 1 and 10.

It is found that the SA algorithm is sensitive to the random seed of the C++ program. So the combinations of parameters using 10 different random seeds are tested. Figure 4.6 illustrates the average loading time (ALT) obtained with corresponding values of parameters. Also, the best result obtained from different combination of parameter is illustrated in figure 4.7.

It is noted that both the average and best loading time achieved the smallest value when the iteration number K = 1000 and T_K = 0.1. Therefore the pair of parameter set (T_1 =10,000, T_K = 0.1, K = 1000) is chosen to compare with the designed greedy heuristic.

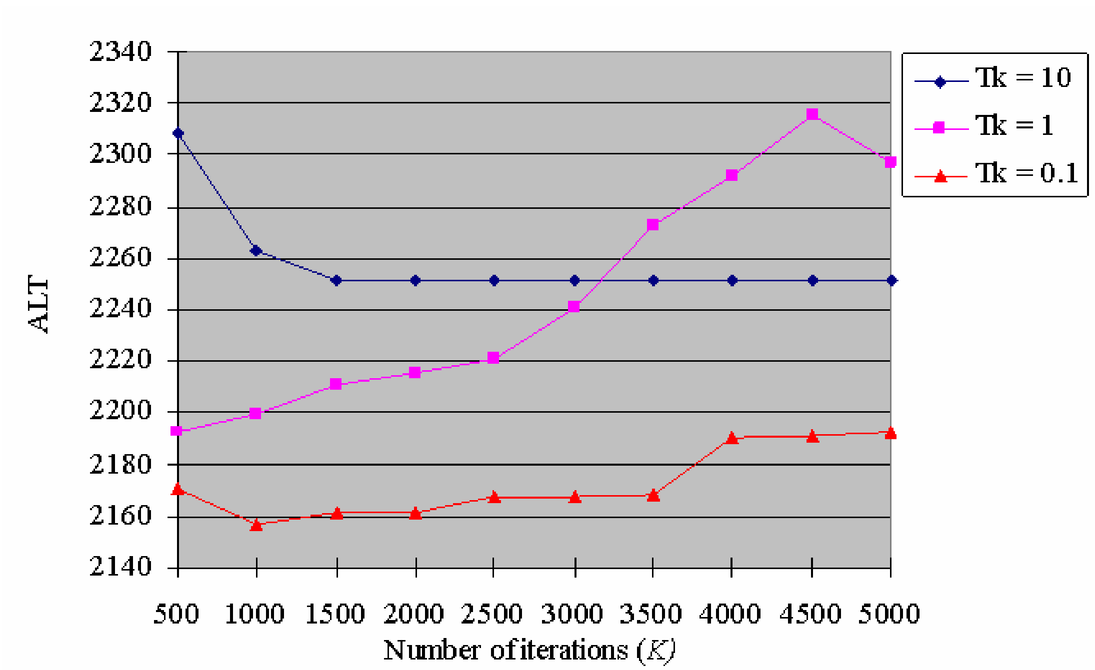


Figure 4.6 Average Loading Time for Different Values of Parameters (T_1 = 10000)

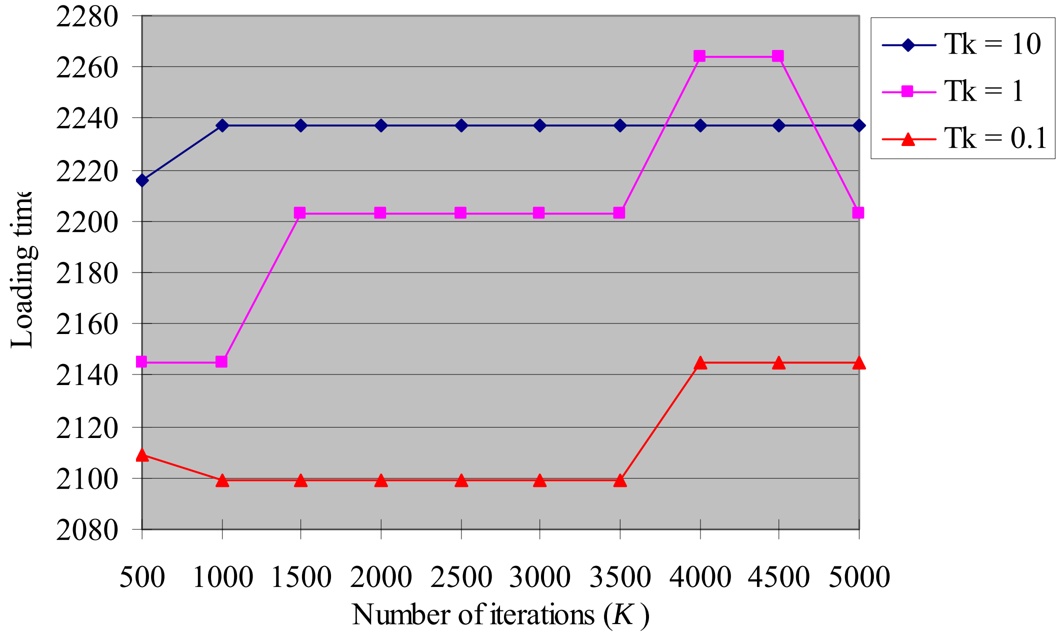


Figure 4.7 The Shortest Loading Time for Different Values of Parameters

4.5.2 Small-scale Problem Tests

Ten small-scale test problems are first randomly generated. In these problems, the multiple YC system needs to pick up 6-10 containers of 2 different types in a container block of 10 yard-bays in 3 sub-tours. The test problems are solved by CPLEX MIP algorithm of CPLEX running on a DELL PC with P IV 3.0 GHz CPU. It is noted that even for these small-scale problem, the computational time can be over 20 hours. Both the designed greedy heuristic and the SA algorithm are also used to solve these problems. The results obtained by the four solution techniques are compared with each other and are shown in figure 4.8.

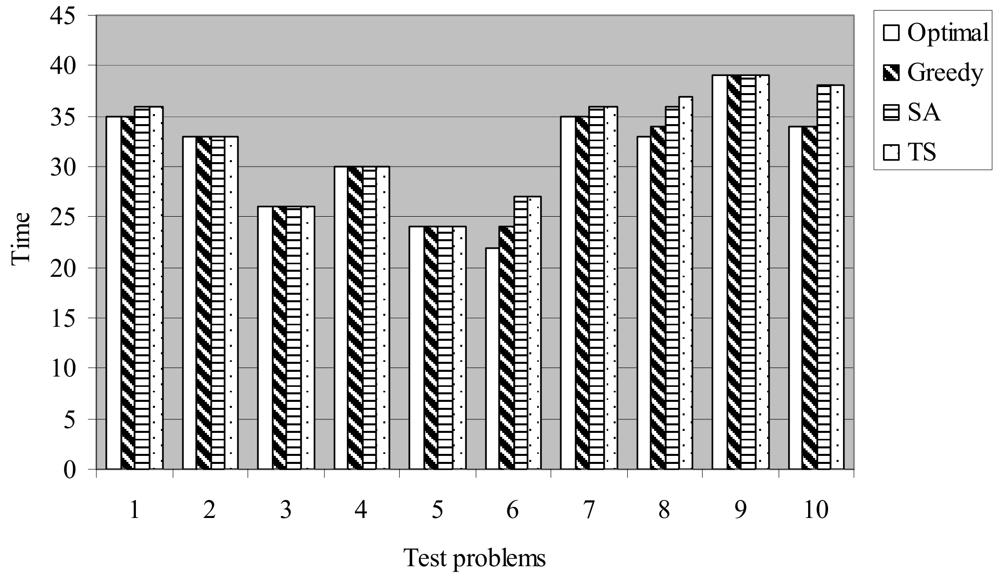


Figure 4.8 Comparisons between the Results of the Greedy Heuristic, SA and TS (Small-scale Problem).

4.5.3 Large-scale Problem Tests

Ten large-scale test problems are also generated to compare the performance of the designed greedy heuristic against the SA and TS algorithm. In these problems, 450-550 containers of 5 different types are randomly allocated in a container block of 45 yard-bays. A multiple YC system of two YCs are used to handle these containers in 10 sub-tours. It is almost impossible to use CPLEX to obtain the optimal solutions due to the excessive time. Therefore, only the results obtained from the greedy heuristic and the SA and TS algorithm are compared in figure 4.9. In solving all the ten sample problems, the proposed greedy heuristic algorithm always achieves better solutions than the SA and TS algorithm.

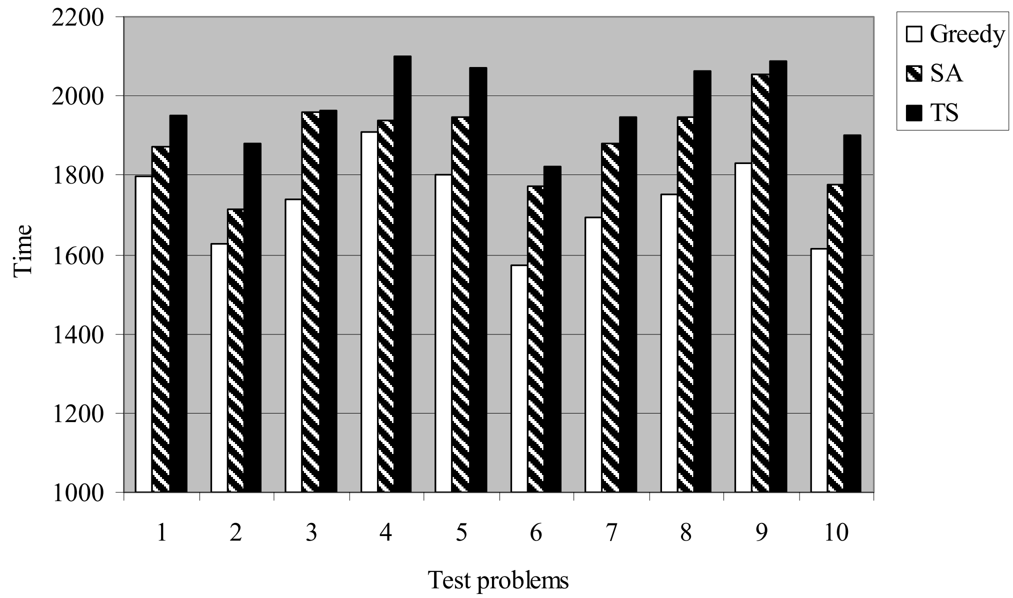


Figure 4.9 Comparisons between the Results of the Greedy Heuristic, SA and TS (Large-scale Problem).

It is noted the designed greedy heuristic outperforms both the SA and TS algorithm in solving both the small-scale and large-scale test problems. On average the results from the greedy heuristic is 8.9% better than the results from the SA algorithm and 14.2% better than the results form the TS algorithm.

4.6 SUMMARY

In this chapter, the prototype MYCS problem is investigated. In the problem both the container loading sequence constraints and the YC interference constraints are considered. An integer programming model is proposed to formulate the problem. Moreover a greedy heuristic, a simulated annealing algorithm and a tabu-search algorithm are designed to solve the proposed model. The performance of the three techniques has been tested through both small-scale and large-scale numerical examples. The result shows that the designed greedy heuristic algorithm consistently outperforms the SA and TS algorithm.

CHAPTER 5

SCHEDULING OF MULTIPLE YC SYSTEMS IN CONTAINER TERMINALS WITH BUFFER AREAS

5.1 INTRODUCTION

In the previous two chapters, the load scheduling problem of multiple YC system serving single QC in container terminals without buffer areas (MYCS problem) has been intensively studied. Since no buffer area is considered, YCs must therefore be scheduled to strictly follow the QC load schedule. In reality however, buffer areas are reserved in the stacking area for some container terminals. The containers picked up by the YCs ahead of schedule are temporarily stored in the buffer areas till they can be handled by the QCs. This special feature will help to increase the utilization of the YCs and expedite the loading operation at the stacking area. Buffer areas sometimes may not exist physically in the container terminals. Nevertheless, as long as containers are allowed to wait on the yard trucks at the wharf area, virtual buffer areas can be considered to exist in the terminals.

This chapter addresses this derived scheduling problem of multiple YC systems in container terminals with buffer areas (MYCS-B). In the problem, several YCs are used to pick up a sequence of containers for a QC. The containers picked up ahead of the schedule can be stored at buffer areas until they can be handled by the QC. An integer programming model is developed for the problem formulation. Numerical examples are conducted to compare the performances of the multiple YC systems in container terminals with and without buffer areas.

5.2 PROBLEM DESCRIPTION

Although the MYCS-B problem and MYCS problem share a lot of common aspects, the use of buffer areas affects the scheduling of YCs significantly. Figure 5.1 provides an illustration of using buffer areas in container terminals.

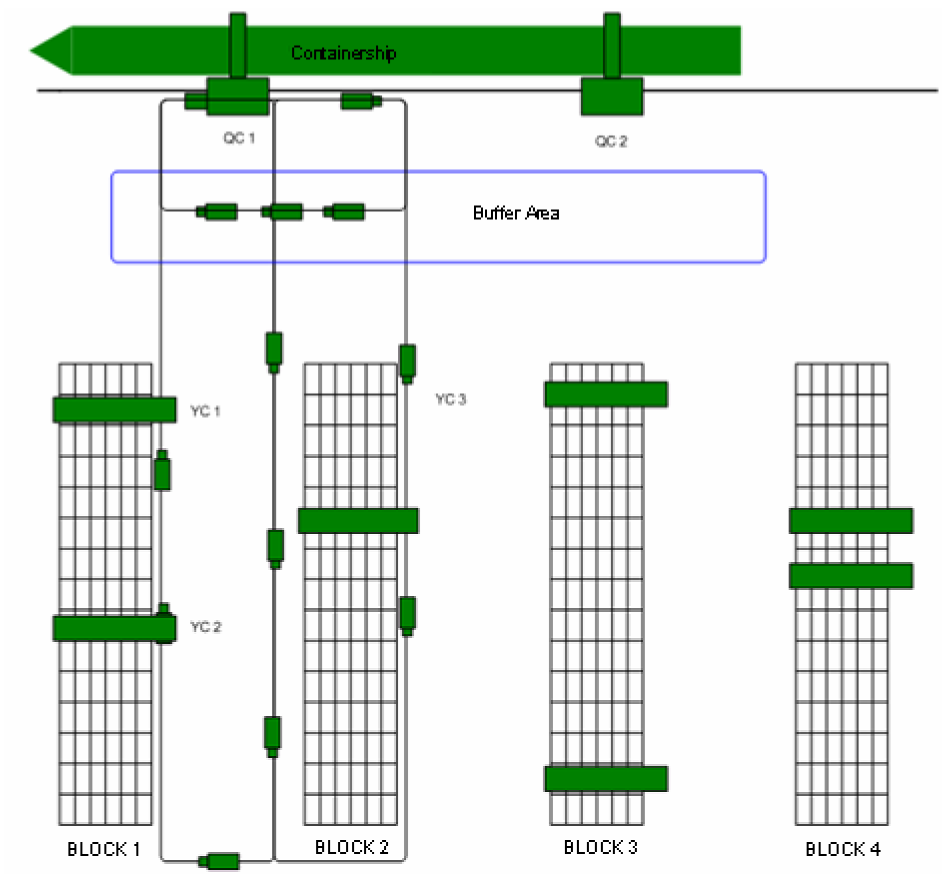


Figure 5.1 An Illustration of Using Buffer Areas in Container Terminals

In the MYCS-B problem, since there are some buffer areas available in the yard, an YC is allowed to pick up containers for the following sequences even though not all the containers required by current sequence are picked up. Those containers which are picked up ahead of schedule will be kept in the buffer area temporarily. Considering the expensive land space in container terminals, the proportion of buffer area is usually very limited. Hence, to reduce the number of containers waiting at buffer areas, in practice operation only in the situation where there is no container of current sequence available within an YC's working range, it is allowed to work for the following sequences ahead of schedule. The working range of an YC here is limited by its neighboring YCs. Figure 5.2 illustrates the working range of an YC. In the figure, YC *k-1* and YC *k+1* are working at Yard-bay *4* and *15*, respectively. The working range of YC *k* is from Yard-bay *5* to *14* in this case.

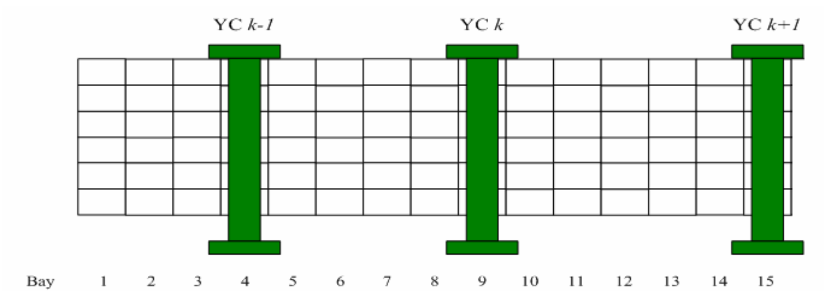


Figure 5.2 An Illustration of the Working Range of an YC

Following is an example to demonstrate the condition for an YC to work ahead of schedule. The current location of the YCs and containers is shown in Figure 5.3. Assume picking up 15 containers of Type A, is the current job of the QC and YC *1* is working at Yard-bay *12*. As shown in the figure, there is no container of Type A available in the working range of YC *2*. In such a situation, YC *2* is allowed to work for the next job of the QC.

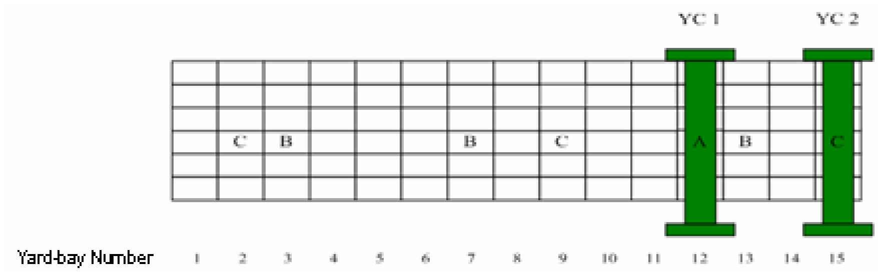


Figure 5.3 Location of the YCs and Containers

5.3 MODEL FORMULATION

To simplify the model formulation, two assumptions are first introduced.

i. Only one type of container is stacked in one bay, which is a common practice of allocating space in container terminals.

ii. Despite the exact storage positions of individual containers, the loading time for all the containers is assumed to be identical.

An integer programming model is proposed for the formulation. As aforementioned, a "sub-tour" is defined as a sequence of containers that needs to be picked up together according to the QC load schedule. The upper bound for the total loading time of the optimal YC scheduling is also assumed to be known and this upper bound is partitioned into T time units. One time unit is defined as the time required for an YC to traverse one bay. The handling time of one container, T_H , is taken to be a multiple of this time unit.

The K YCs are numbered from 1 to K from left to right according to their initial location in the block. Since the YCs are of the same size, they cannot pass each other which implies that YC k can only move in the range limited by the locations of YC $k-1$ and $k+1$.

The following notations are used in formulating the MYCS-B problem,

$$X_{i,j,k,t} = \begin{cases} 1 & \text{if YC } k \text{ finish loading one container for Sub-tour } i \text{ at Yard-bay } j \text{ at time } t \\ 0 & \text{otherwise (a decision variable)} \end{cases}$$

$$Y_{j,k,t} = \begin{cases} 1 & \text{if YC } k \text{ is at Yard-bay } j \text{ at time } t \\ 0 & \text{otherwise (a decision variable)} \end{cases}$$

N_i the number of containers needed to be picked up for Sub-tour i

C_j the number of containers stacked at Yard-bay j before the loading process start

$B\ (i)$ the set of yard-bays where the containers required by Sub-tour i are located

S the number of sub-tours for the whole loading process

φ the maximum number of sub-tours YCs can work ahead of schedule, if no containes for the current sub-tour is available in their working range

I_k the initial position of YC k

F_k the final position of YC k

The objective function is to minimize the loading time, which can be represented by equation (5.1).

$$\text{Minimize } \max(tX_{i,j,k,t}) \tag{5.1}$$

Subject to

$$\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{T} X_{i,j,k,t} = C_j \quad j = 1,2\cdots,B \tag{5.2}$$

$$\sum_{j=1}^{B}\sum_{k=1}^{K}\sum_{t=1}^{T} X_{i,j,k,t} = N_i \quad i = 1,2\cdots,S \tag{5.3}$$

$$\sum_{j\in B(i-\varphi-s)}\sum_{k=1}^{K}\sum_{t=1}^{a-1} X_{i-\varphi-s,j,k,t} - N_{i-\varphi-s} \le M(1 - X_{i,j^*,k^*,a}) \qquad i = 3,4\cdots S;\ \varphi = 1,2\cdots i-2;$$

$$s = 1, 2 \cdots i - \varphi - 1;\ j^* \in B(i);\ k^* = 1, 2 \cdots K;\ a = T_H + 1, T_H + 2 \cdots T \qquad (5.4)$$

$$\sum_{j \in B(i-\varphi-s)} \sum_{k=1}^{K} \sum_{t=1}^{a-1} X_{i-\varphi-s,j,k,t} - N_{i-\varphi-s} \geq M(X_{i,j^*,k^*,a} - 1) \qquad i = 3, 4 \cdots S;\ \varphi = 1, 2 \cdots i-2;$$

$$s = 1, 2 \cdots i - \varphi - 1;\ j^* \in B(i);\ k^* = 1, 2 \cdots K;\ a = T_H + 1, T_H + 2 \cdots T \qquad (5.5)$$

$$-M(1 - Y_{j',k-1,t}) - M(1 - Y_{j'',k+1,t}) + \sum_{j'<j<j''} C_j \leq M(1 - X_{i-s,j^*,k,t}) \qquad i = 3, 4 \cdots S;\ j', j'', j^* \in B(i);$$

$$\varphi = 1, 2 \cdots i - 2;\ s = 1, 2 \cdots \varphi \text{-} 1;\ j \in B(i - \varphi);\ k = 1, 2 \cdots K;\ t = T_H + 1, T_H + 2 \cdots T \qquad (5.6)$$

$$\sum_{i=1}^{S} \sum_{j=1}^{B} X_{i,j,k,t} \leq 1 \quad k = 1, 2 \cdots, K;\ t = 1, 2 \cdots, T \qquad (5.7)$$

$$\sum_{i=1}^{S} \sum_{j=1}^{B} \sum_{a=1}^{T_H - 1} X_{i,j,k,t-a} \leq M(1 - X_{i,j,k,t})$$

$$i = 1, 2 \cdots, S;\ j = 1, 2 \cdots, B;\ k = 1, 2 \cdots, K;\ t = 1, 2 \cdots, T \qquad (5.8)$$

$$\sum_{a=0}^{T_H} Y_{j,k,t-a} - T_H - 1 \geq M(X_{i,j,k,t} - 1)$$

$$i = 1, 2 \cdots, S;\ j = 1, 2 \cdots, B;\ k = 1, 2 \cdots, K;\ t = T_H + 1, T_H + 2 \cdots, T \qquad (5.9)$$

$$M(1 - Y_{b,k,t}) \geq \sum_{j=b}^{B} Y_{j,k-1,t} \quad b = 1, 2 \cdots, B;\ k = 2, 3 \cdots, K;\ t = 1, 2 \cdots, T \qquad (5.10)$$

$$\sum_{j=1}^{B} Y_{j,k,t} = 1 \quad k = 1, 2, \cdots, K;\ t = 1, 2 \cdots, T \qquad (5.11)$$

$$\sum_{k=1}^{K} Y_{j,k,t} \leq 1 \quad j = 1, 2 \cdots, B;\ t = 1, 2 \cdots, T \qquad (5.12)$$

$$\sum_{j=b-1}^{b+1} Y_{j,k,t\pm1} \geq Y_{b,k,t} \quad b = 1, 2 \cdots, B;\ k = 1, 2 \cdots, K;\ t = 1, 2 \cdots, T \qquad (5.13)$$

$$Y_{I_k,k,1} = 1 \quad k = 1, 2 \cdots, K \qquad (5.14)$$

$$Y_{F_k,k,1} = 1 \quad k = 1, 2 \cdots, K \qquad (5.15)$$

where M is a big positive number.

Constraints (5.2) ensure the number of containers picked up during the whole loading process at one bay equals to the initial number of containers stacked at that bay. Constraints (5.3) ensure the number of containers picked up during one sub-tour equals to the number which is required by the QC load schedule. Constraints (5.4) and (5.5) ensure that before the YCs can start to work on Sub-tour *i*, they must first finish loading all sub-tours before Sub-tour $i - \varphi$. In the event that no containers of the current sub-tour are within an YC's working range, Constraints (5.6) allow the YC to move on to a maximum of φ more sub-tour. Constraints (5.7) ensure the YC can at most finish handling one container for one period. Constraints (5.8) ensure that the YC cannot finish any handling jobs during the time interval $t - T_H - 1$ to *t-1* if it completes one handling job at period *t*. Constraints (5.9) ensure during the loading one container the YC will stay at the container location throughout the operation. Constraints (5.10) ensure the movement of the YCs is free of inter-YC interference. Constraints (5.11) state that one YC can only be at one bay during one period. Constraints (5.12) state that only one YC can be at one bay in each period. Constraints (5.13) ensure that the YC can at most move one yard-bay during one period. Constraints (5.14) and (5.15) state the initial and final positions of the K YCs.

As mentioned, φ is the maximum number of sub-tours YCs can work ahead of schedule, in the case no containers for the current sub-tour are available in their working range. For example,

$\varphi = 1$, YCs are allowed to work for the next sub-tour if no containers for the current sub-tour are available in their working range;

$\varphi = 2$, YCs are allowed to work for any of the next two sub-tours if no containers for the current sub-tour are available in their working range;

One boundary value of φ is *0*. In this case, YCs are not allowed to work for following sub-tours even there is no containers for the current sub-tour available in their working range. Therefore, the MYCS-B problem becomes the previous studied MYCS problem. The other boundary value of φ is *i-1*. In this case, YCs are free to work for any following sub-tours if there is no container for the current sub-tour available in their working range.

In general, with a larger number of φ, YCs will have more freedom in the loading process, which will shorten the loading time and improve the utilization of YCs. On the other hand, a larger number of φ will also lead to a greater number of containers picked up ahead of schedule, which requires larger buffer areas. To deal with this conflict, the terminal operators need to determine a proper value of φ, with which a higher utilization of YCs is achieved with an appropriate size of buffer areas. Based on empirical experience, *1* is a prevailing value of φ used by the terminal operators. In line with the practice operation, a heuristic algorithm for $\varphi = 1$ is introduced in the following section. This algorithm can also be extended for other values of φ with simple revisions.

5.4 A SCHEDULING HEURISTIC

It is well-known that the single YC scheduling problem is an NP-complete problem. Needless to say, the MYCS-B problem is also an NP-complete problem which makes exact algorithms not practical in solving the large scale cases. Hence, a scheduling heuristic based on greedy principle is proposed for the solution of the MYCS-B problem. For the sake of brevity, a system of two YCs is used to illustrate this heuristic approach. It should be noted the scheduling heuristic is implemented on a container by container basis, which means that once an YC finishes picking up one container, it needs to identify which yard-bay to work at next.

The scheduling rules of the proposed heuristic are as follows:

Situation 1: Only one yard-bay which contains containers for the current sub-tour remains in the block.

<u>Rule 1</u>

The yard-bay, therefore, will be the nearest yard-bay to both YCs. This yard-bay will be assigned to the closer YC. The other YC will then work at the nearest bay which contains containers for the next sub-tour and the picked up containers will be carried to the buffer area by the yard trucks. In the case where two YCs are of equal distance to the yard-bay, the yard-bay will be assigned to one YC arbitrarily.

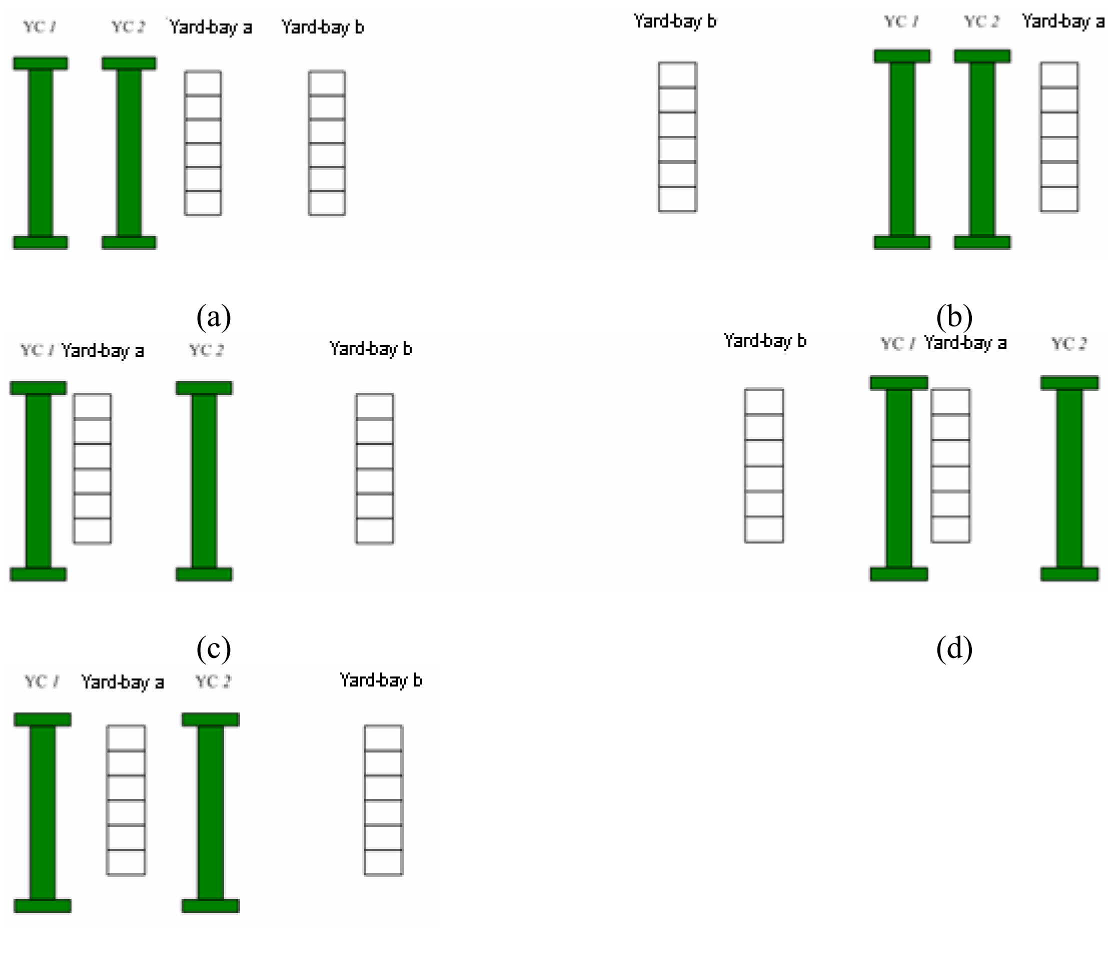
Situation 2: More than one yard-bay which contains containers for the current sub-tour remains in the block.

<u>Rule 2</u>

Both of the two YCs will pick up containers at their nearest yard-bays which contains containers for the current sub-tour.

<u>Rule 3</u>

If the same yard-bay is identified to be the closest one to both YC *1* and YC *2* and it is not the last yard-bay of containers for the current sub-tour, following five scenarios (Figure 5.4) are the only permitted situations. (Both Yard-bay *a* and *b* are the yard-bays that contain containers for the current sub-tour)



(a) (b) (c) (d) (e)

Figure 5.4 Spatial Relationships between YCs and Their Closest Containers

For the purpose of clarity, the interior area and the exterior area of the YCs are defined in the same way as in Chapter 4.

a) The closest yard-bay (Yard-bay *a*) is at the YC *2*'s exterior area, and there is no other yard-bay available at YC *1*'s exterior area: Yard-bay *a* will be assigned to YC *1* as the next yard-bay to work at and Yard-bay *b* will be assigned to YC *2*.

b) The closest yard-bay (Yard-bay *a*) is at the YC *2*'s exterior area, and there is a yard-bay (Bard-bay *b*) available at YC *1*'s exterior area: Yard-bay *b* will be assigned to YC *1* as the next yard-bay to work at and Yard-bay *a* will be assigned to YC *2*

c) The closest yard-bay (Yard-bay *a*) is at the interior area with YC *1* being the closer crane and there is no bay available at YC *1*'s exterior area: Yard-bay *a* will be assigned to YC *1* as the next yard-bay to work at and Yard-bay *b* will be assigned to YC *2*.

d) The closest yard-bay (Yard-bay *a*) is at the interior area with YC *1* being the closer crane and there is a Yard-bay (Yard-bay *b*) available at YC *1*'s exterior area: Yard-bay *b* will be assigned to YC *1* as the next yard-bay to work at and Yard-bay *a* will be assigned to YC *2*

e) The closest yard-bay (Yard-bay *a*) is at the interior area with YC *1* and YC *2* being of equal distance to it and there is a yard-bay (Yard-bay *b*) available at YC *2*'s exterior area: Yard-bay *a* will be assigned to YC *1* as the next yard-bay to work at and Yard-bay *b* will be assigned to YC *2*.

Rule 4

In the case where there are two yard-bays of containers of equal distances to one YC, the YC will choose the yard-bay which is further from the other YC.

Rule 5

If there is no available container for an YC, it will stay still.

5. 5 NUMERICAL EXPERIMENTS

To measure the performance of the multiple YC systems with and without buffer areas, ten test problems are generated as follows:

1) Generate the QC load schedule:
   a) The total number of containers for each problem is randomly chosen in the range of 300 to 450.
   b) The containers are randomly classified into five types, namely A, B, C, D and E.
   c) Each type is then further divided into 2 or 3 groups.
   d) The QC load schedule is finally generated by joining these groups in a random sequence.

2) Allocate the containers required by the QC in the stacking area: Containers are randomly allocated in a container block, which consists of 25 yard-bays, subjected to

the constraint that only one type of container can be stacked in a yard-bay.

Both the multiple YC systems, consisting of two YCs, with and without a buffer area are tested on the generated sample problems. In the previous chapter a greedy heuristic and a SA algorithm were developed to solve the MYCS problem and the results show that the greedy heuristic algorithm consistently outperforms the SA algorithm. Hence the greedy heuristic algorithm is tested on the generated problems and the results are compared with the results of the MYCS-B. Figure 5.5 shows that by adopting the proposed scheduling heuristic, the result of MYCS-B outperforms the result of MYCS using the greedy heuristic algorithm in all the numerical experiments.

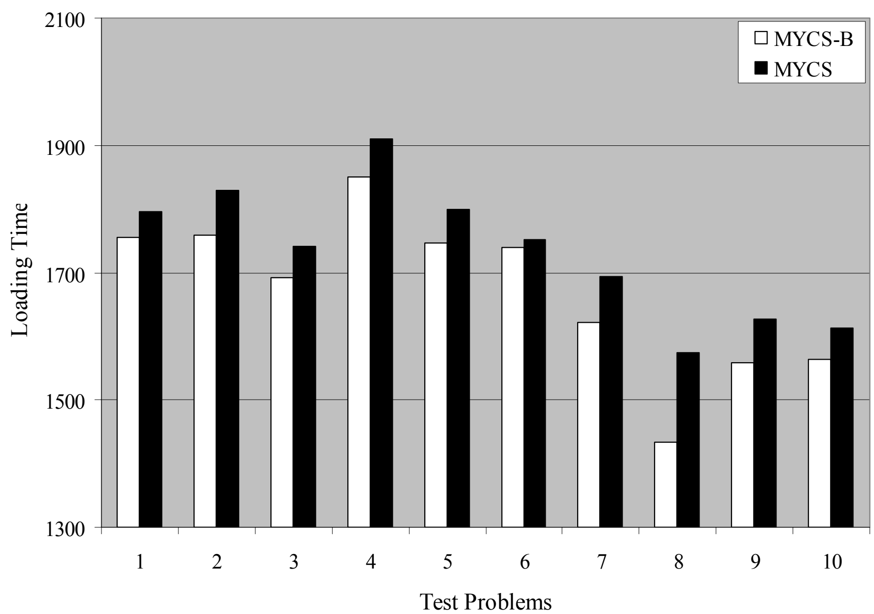


Figure 5.5 Comparisons between the Loading Time of MYCS-B and MYCS

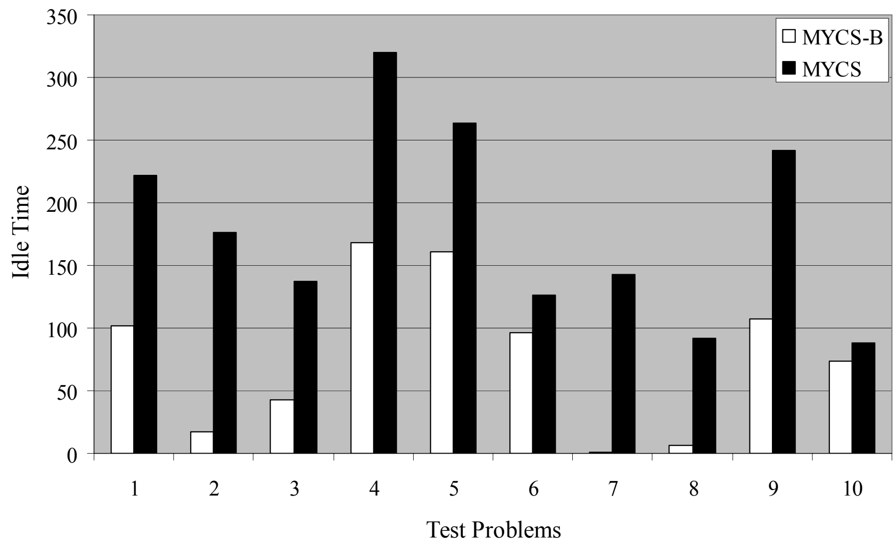


Figure 5.6 Comparisons between the Idle Time of MYCS-B and MYCS

During the loading process, there may not be any appropriate container available for an YC for a certain period of time. Hence the YC will stay still during that time. In the study, this period of time is called the idle time of the YC. The numerical results (in Figure 5.6) show that with the proposed scheduling heuristic, the total idle time of the YCs of the MYCS-B is significantly reduced compared to the idle time of the MYCS. This can be explained that adding buffer areas in a container terminal actually increases the degree of freedom of the YC operations by allowing YCs to work ahead of the QC load schedule. This reduces the probability of an YC to be idle in the case that no containers for the current sub-tour are available in its working range.

The utilization rate of YCs is defined as the following equation (16),

$$\text{Utilization rate} = \frac{\text{Loading time - Total idle time}}{\text{Loading time}} \times 100\% \tag{16}$$

Table 5.1 shows the utilization rates of the YCs of both MYCS-B and MYCS. On average the utilization rate of YCs of MYCS-B is increased by 6% compared to the utilization rate of YCs of MYCS.

Table 5.1 Utilization rate of YCs of MYCS-B and MYCS %

	1	2	3	4	5	6	7	8	9	10
MYCS	87.6	90.4	92.1	83.2	85.3	92.8	91.6	94.2	85.1	94.5
MYCS-B	94.2	99.0	97.5	90.9	90.8	94.5	99.9	99.6	93.1	95.3

5.6 SUMMARY

Reserving buffer areas in container terminals where containers picked up ahead of the schedule can be kept temporarily will help to increase the efficiency of YC operations. In this chapter, an integer programming model has been developed to formulate the MYCS-B scheduling problem. Moreover, a scheduling heuristic has been designed to solve the proposed problem. Numerical experiments show that adopting the designed scheduling heuristic the MYCS-B is capable of achieving a higher productivity in terms of needed loading time than MYCS, which will contribute to improve the overall efficiency of container terminal operation.

CHAPTER 6

DEPLOYMENT STRATEGIES OF DOUBLE RAIL MOUNTED GANRY CRANE SYSTEMS IN YARD TRUCK BASED CONTAINER TERMINALS

6.1 INTRODUCTION

Double rail mounted gantry crane (DRMG) system is an emerging container handling equipment technology that was recently introduced in Europe. The system consists of two rail mounted gantry cranes of different height and width. Intuitively, this special feature will help to increase the productivity of the two cranes since they can pass each other during their movement along a container block. Following figure 6.1 is the DRMG system in real operations.

Figure 6.1 A DRMG System in Operation (Steenken et al., 2004)

This chapter focuses on providing an efficient operation strategy for the DRMG systems to load outbound containers in the yard truck based container terminals. An integer programming model is developed to formulate the problem. A greedy heuristic algorithm and a simulated annealing (SA) algorithm are designed to solve the proposed problem. Computational experiments show that the greedy heuristic outperforms the SA algorithm. Since the greedy algorithm performs well and is easy to implement, it has a high potential to be used in scheduling DRMG systems in real operation.

6.2 USING DRMG SYSTEMS IN YARD TRUCK BASED CONTAINER TERMINALS

Figure 6.2 is the front view of a DRMG system. Although the DRMG system has been put into practice in the Port of Hamburg and it may have a significant influence on the future development of container terminals. Up to now, to the best of the author's knowledge, the only report in literature regarding the operation of DRMG is conducted by Kim et al. (2002). In that paper, the authors conducted a simulation study on the operation rules of DRMG in an Automated Guided Vehicles (AGV) based container terminal.

However, most container terminals are still using yard trucks as the prime movers to transport containers between stacking yards and berths. Unlike in Europe, the labor cost in these terminals is not that high, therefore building a fully automated container terminal might not be cost efficient to them. In these terminals, using DRMG system with traditional yard trucks could be a promising approach to enhance the operational efficiency and enjoy the benefit of low labor costs. Hence, the operation problem of DMRG system in traditional yard truck based container terminals is highly desired.

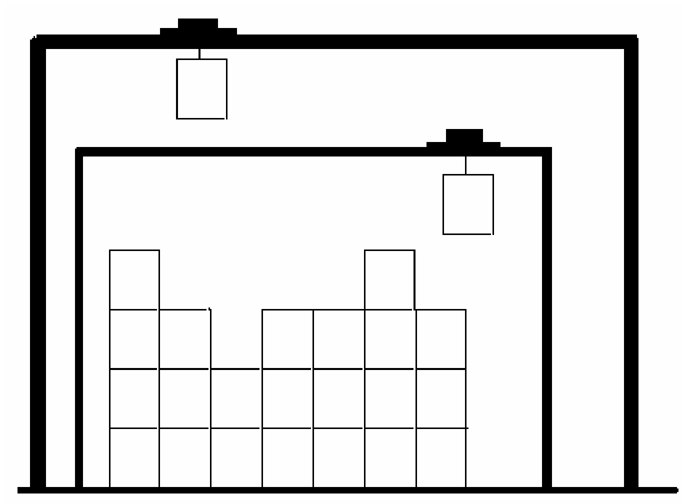


Figure 6.2 Front View of a DRMG System

Figure 6.3 shows a yard truck based container handling system using DRMGs. In the figure, DRMG 1 loads the outbound containers in container block 1 to yard trucks for transportation to QC 1. At the same time, DRMG 2 unloads the inbound containers discharged by QC 2 from yard trucks for storage in block 2. As aforementioned, the operations on the inbound containers is relatively simple, therefore this chapter focuses on the loading operation of outbound containers, which is to determine the optimal work schedule of the two RMGs of DRMG 1 to minimize the container loading time.

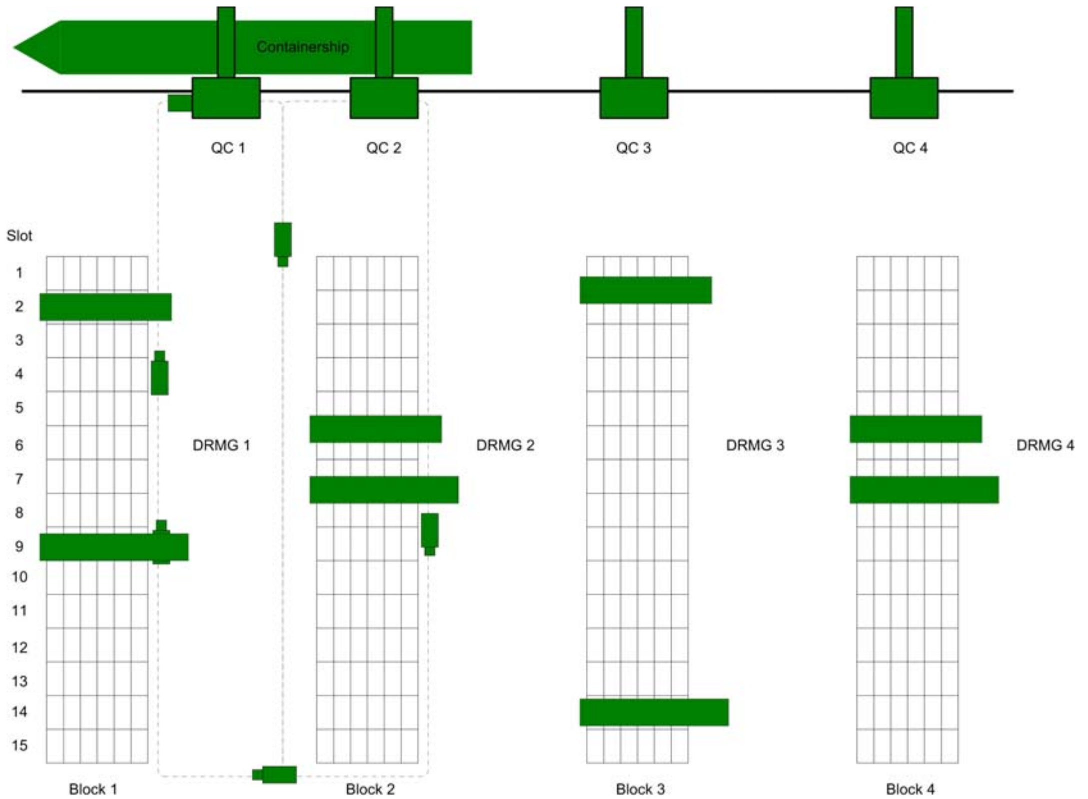


Figure 6.3 A Yard Truck Based Container Handling System Using DRMGs

The two decision factors in the problem are the yard-bay visiting sequences of the two RMGs of the DRMG system and the number of containers picked up at each yard-bay visit. Since the loading jobs are distributed among the two RMGs, making their working schedule dependent on one another, the schedules of the two RMGs need to be coordinated to minimize the overall loading time. For the sake of presentation, synchronizing the transport and loading activities of two RMGs of the DRMG system is called DRMG scheduling problem (DRS) in this study.

Although the two RMGs of a DRMG are of different size and can pass each other, there still may be some interference between the operations of them. The following depicts the interferences that may occur.

1) The two RMGs cannot work at the same yard-bay at the same time.
2) The small RMG cannot pass the large RMG when it's loading a container.

6.3 MATHEMATICAL FORMULATION

To simplify the mathematical model for the DRS problem, two reasonable assumptions are made.

i. There is only one type of container sacked in one yard-bay, the common practice in allocating space in the stack area of container terminals.
ii. The loading time for all the containers is assumed to be the same despite the exact storage positions of individual containers.

Similar modeling method is used to formulate the DRS problem. A "sub-tour" (subsequence) is also defined as a sequence of containers that needs to be picked up together, which is according to load plan of the QC. The upper bound for the total loading time of the optimal DRMG scheduling is assumed to be known and this upper bound is partitioned into T time unit. One time unit is defined as the time required for a RMG to travel the distance of a single yard-bay. The time required to handle a single container, T_H, is taken to be a multiple of this time unit. The B yard-bays in the block are numbered *1* to B from top to down.

The following notations are used to formulate the DRS problem.

$$X_{i,j,t} = \begin{cases} 1 & \text{if the large RMG finishes loading one container} \\ & \text{for Sub-tour } i \text{ at Yard-bay } j \text{ at time } t \\ 0 & \text{otherwise} \quad \text{(a decision variable)} \end{cases}$$

$$Z_{i,j,t} = \begin{cases} 1 & \text{if the small RMG finishes loading one container} \\ & \text{for Sub-tour } i \text{ at Yard-bay } j \text{ at time } t \\ 0 & \text{otherwise} \quad \text{(a decision variable)} \end{cases}$$

$$Y_{j,t} = \begin{cases} 1 & \text{if the large RMG is at Yard-bay } j \text{ at time } t \\ 0 & \text{otherwise} \quad \text{(a decision variable)} \end{cases}$$

$$W_{j,t} = \begin{cases} 1 & \text{if the small RMG is at Yard-bay } j \text{ at time } t \\ 0 & \text{otherwise} \quad \text{(a decision variable)} \end{cases}$$

N_i the number of containers needed to pick up for Sub-tour i

C_j the number of containers stacked in Yard-bay j before the loading process starts

B(i) the set of yard-bays where the containers required by Sub-tour i are located

n the number of sub-tours for the whole loading process

The objective of the DRS problem is to minimize the loading time of the containers, which can be represented by the following equation,

$$\text{Minimize } \max(tX_{i,j,t},\ tZ_{i,j,t}) \tag{6.1}$$

Subject to

$$\sum_{i=1}^{n}\sum_{t=1}^{T} X_{i,j,t} + Z_{i,j,t} = C_j \quad j = 1, 2, \cdots, B \tag{6.2}$$

$$\sum_{j=1}^{B}\sum_{t=1}^{T} X_{i,j,t} + Z_{i,j,t} = N_i \quad i = 1, 2, \cdots, n \tag{6.3}$$

$$\sum_{j \in B(i-s)} \sum_{t=1}^{a-1} (X_{i-s,j,t} + Z_{i-s,j,t}) - N_{i-s} \le M(1 - X_{i,j',a})$$

$$s = 1, 2, \cdots i-1;\ j' \in B[i];\ i = 2, 3, \cdots n,\ a = 2, 3 \cdots T; \tag{6.4}$$

$$\sum_{j\in B(i-s)}\sum_{t=1}^{a-1}(X_{i-s,j,t}+Z_{i-s,j,t})-N_{i-s}\geq M(X_{i,j',a}-1)$$

$$s=1,2,\cdots i-1;\ j'\in B[i];\ i=2,3,\cdots n,\ a=2,3\cdots T; \quad (6.5)$$

$$\sum_{j\in B(i-s)}\sum_{t=1}^{a-1}(X_{i-s,j,t}+Z_{i-s,j,t})-N_{i-s}\leq M(1-Z_{i,j',a})$$

$$s=1,2,\cdots i-1;\ j'\in B[i];\ i=2,3,\cdots n,\ a=2,3\cdots T; \quad (6.6)$$

$$\sum_{j\in B(i-s)}\sum_{t=1}^{a-1}(X_{i-s,j,t}+Z_{i-s,j,t})-N_{i-s}\geq M(Z_{i,j',a}-1)$$

$$s=1,2,\cdots i-1;\ j'\in B[i];\ i=2,3,\cdots n,\ a=2,3\cdots T; \quad (6.7)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{B}X_{i,j,t}\leq 1 \quad t=1,2,\cdots,T \quad (6.8)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{B}Z_{i,j,t}\leq 1 \quad t=1,2,\cdots,T \quad (6.9)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{B}\sum_{a=1}^{T_H-1}X_{i,j,t-a}\leq M(1-X_{i,j,t}) \quad i=1,2,\cdots,n;\ j=1,2,\cdots,B;\ t=1,2,\cdots,T \quad (6.10)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{B}\sum_{a=1}^{T_H-1}Z_{i,j,t-a}\leq M(1-Z_{i,j,t}) \quad i=1,2,\cdots,n;\ j=1,2,\cdots,B;\ t=1,2,\cdots,T \quad (6.11)$$

$$\sum_{a=0}^{T_H}Y_{j,t-a}-T_H-1\geq M(X_{i,j,t}-1) \quad i=1,2,\cdots,n;\ j=1,2,\cdots,B;\ t=T_H+1,T_H+2\cdots,T \quad (6.12)$$

$$\sum_{a=0}^{T_H}W_{j,t-a}-T_H-1\geq M(Z_{i,j,t}-1) \quad i=1,2,\cdots,n;\ j=1,2,\cdots,B;\ t=T_H+1,T_H+2\cdots,T \quad (6.13)$$

$$M(1-X_{i,j,a})\geq\sum_{t=a-T_H}^{a+T_H}Z_{i,j,t} \quad i=1,2,\cdots,n;\ j=1,2,\cdots,B;\ a=T_H,T_H+1,\cdots,T \quad (6.14)$$

$$M(1-Z_{i,j,a})\geq\sum_{t=a-T_H}^{a+T_H}X_{i,j,t} \quad i=1,2,\cdots,n;\ j=1,2,\cdots,B;\ a=T_H,T_H+1,\cdots,T \quad (6.15)$$

$$M(1-X_{i,j,a}) \geq \sum_{t=a-T_H}^{a} W_{j,t} \quad i=1,2,\cdots,n;\ j=1,2,\cdots,B;\ a=T_H,T_H+1,\cdots,T \tag{6.16}$$

$$\sum_{j=1}^{B} Y_{j,t} = 1 \quad t=1,2,\cdots,T \tag{6.17}$$

$$\sum_{j=1}^{B} W_{j,t} = 1 \quad t=1,2,\cdots,T \tag{6.18}$$

$$\sum_{j=b-1}^{b+1} Y_{j,t+1} \geq Y_{b,t} \quad b=1,2,\cdots,B;\ t=1,2,\cdots,T \tag{6.19}$$

$$\sum_{j=b-1}^{b+1} Y_{j,t-1} \geq Y_{b,t} \quad b=1,2,\cdots,B;\ t=1,2,\cdots,T \tag{6.20}$$

$$\sum_{j=b-1}^{b+1} W_{j,t+1} \geq W_{b,t} \quad b=1,2,\cdots,B;\ t=1,2,\cdots,T \tag{6.21}$$

$$\sum_{j=b-1}^{b+1} W_{j,t-1} \geq W_{b,t} \quad b=1,2,\cdots,B;\ t=1,2,\cdots,T \tag{6.22}$$

where M is a big positive number.

Constraints (6.2) ensure the number of containers picked up during the whole loading process at one yard-bay equals to the initial number of containers stacked in that yard-bay. Constraints (6.3) ensure the number of containers picked up during one sub-tour equals to the number required by the load plan. Constraints (6.4) to (6.7) state that the RMGs must finish the loading jobs for all the previous sub-tours before they can start to work for the next sub-tour. Constraints (6.8) and (6.9) ensure the RMG can finish loading at most one container at one time period. Constraints 6.(10) and (6.11) state that the RMG cannot finish any handling jobs during the time interval $t-T_H-1$ to *t-1* if it completes one loading job at period *t*. Constraints (6.12) and (6.13) ensure that during the loading process of one container the RMG must stay at the container location throughout the operation. Constraints (6.14) and (6.15) state that the two RMGs cannot load containers at the same yard-bay, at the same time. Constraints (6.16) ensure that the small RMG will not pass the large RMG when it is loading a container. Constraints (6.17) and (6.18) state that one RMG can only be at one yard-bay during one period. Constraints (6.19) to (6.22) ensure that the RMG can only move one yard-bay during one period.

6.4 SCHEDULING HEURISTICS

6.4.1 A Greedy Heuristic

It is well-known that the single YC scheduling problem is an NP-complete problem. Needless to say, the DRS problem is also an NP-complete problem which makes exact algorithm not practical to solve the large scale cases. Hence, heuristic algorithms are required to solve the DRS problem efficiently. A greedy heuristic is proposed in this section to solve the DRMG scheduling problem. The scheduling rules of this heuristic are as follows:

Rule 1

Both the large RMG and small RMG will choose the containers at their closest yard-bays. The containers also need to satisfy the loading sequence requirement and will not cause aforementioned interference.

Rule 2

If the two RMGs choose the container at the same yard-bay, the container will be assigned to the small RMG and the large RMG then will choose the container at its second closest yard-bay.

Rule 3

If there is no available container for a RMG, it will stay still.

Rule 4

In the case where two containers are of equal distances to one RMG, it will choose the container which is further from the other RMG.

6.4.2 Simulated Annealing Algorithm

Simulated annealing (SA) algorithm is also applied to solve the proposed DRS problem and the performance of the SA algorithm is compared with the greedy heuristic. The SA algorithm is implemented in the same way as the one in the precious chapter. For the sake of brevity, the details of the algorithm are not discussed here.

6.5 NUMERICAL EXPERIMENTS

6.5.1 Sensitivity Analysis of SA Parameters

To use SA to solve the DRS problem, a sensitivity analysis of SA parameters is conducted in advance. It is found though a rudimentary experiment that 10,000 is a proper value of the initial temperature T_1. Then the other two parameters, iteration number K and stopping temperature T_K are tested by solving sample problems. The tested values of the iteration number are 500, 1000, 1500, …, 5000 and the tested values of stopping temperature are 0.1, 1 and 10.

It is found that the SA algorithm is sensitive to the random seed of the C++ program. Hence we test the combination of parameters using different random seed and calculate the average of their performance. Figure 6.4 illustrates the average loading time obtained with corresponding values of parameters.

Also, the best result obtained from different combination of parameter is illustrated in figure 6.5. It is noted that for the designed SA algorithm, the objective function achieves the smallest value when the iteration number K = 1000 and T_K = 0.1. Therefore the pair of parameter set (T_1 =10,000, T_K = 0.1, K = 1000) is chosen to compare with the greedy heuristic.

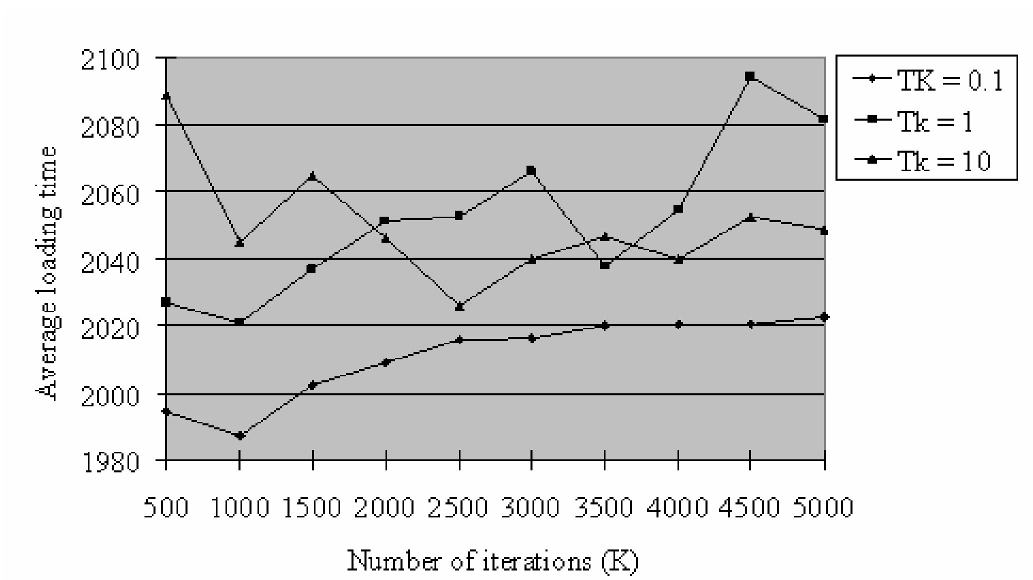


Figure 6.4 Average Loading Time for Different Values of Parameters (T_1 = 10,000)

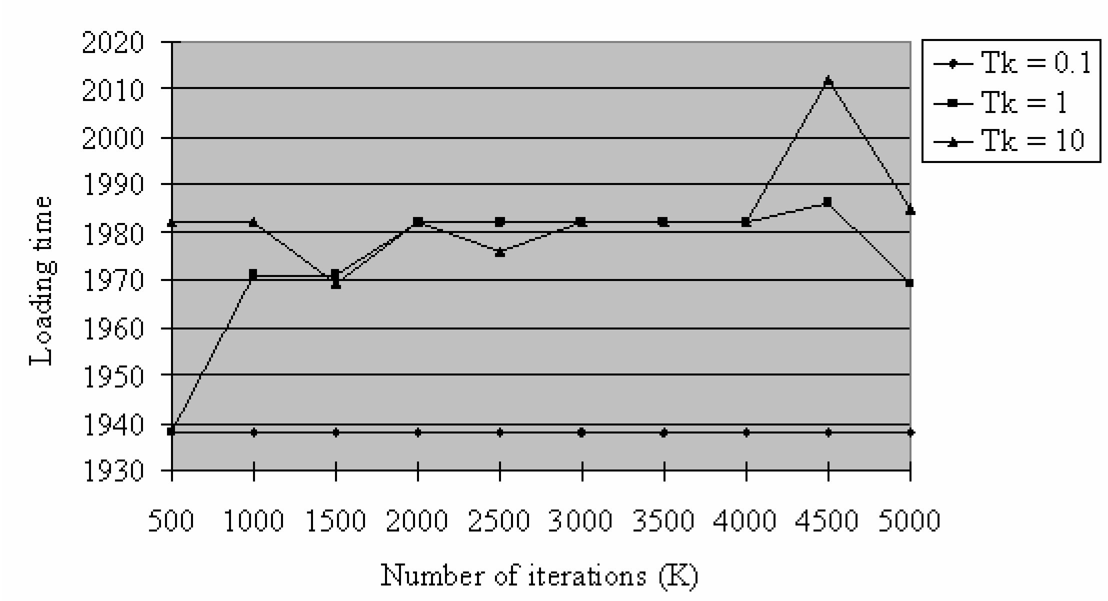


Figure 6.5 The Shortest Loading Time for Different Values of Parameters (T_1 = 10,000)

6.5.2 Small-scale Problem Tests

Ten small-scale test problems are first randomly generated. In these problems, the DRMG system needs to pick up 6-10 containers of 2 different types in a container block of 10 yard-bays in 3 sub-tours. The test problems are solved by *CPLEX MIP algorithm* of CPLEX running on a DELL PC with P IV 3.0 GHz CPU. It is noted that even for these small-scale problem, the computational time can be over 10 hours. The designed greedy heuristic and the SA algorithm are also used to solve these problems. The results obtained by the three solution techniques are compared with each other and are shown in figure 6.6. In general, most of the results from the greedy heuristic and the SA algorithm are equal or close to the optimal solution.

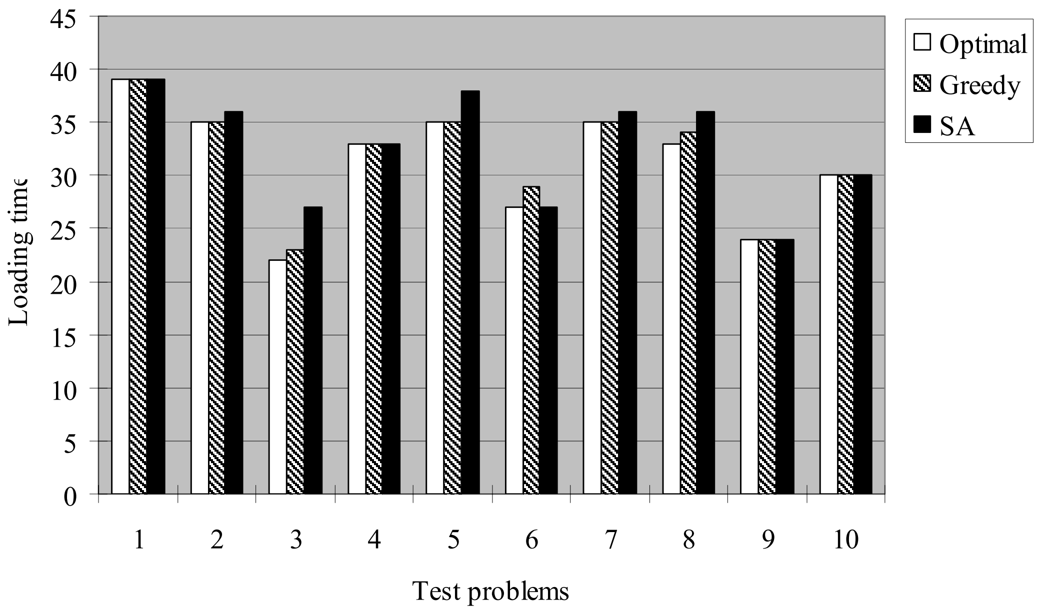


Figure 6.6 Comparison between The Results of CPLEX, Greedy Heuristic and SA (small-scale problems)

6.5.3 Large-scale Problem Tests

Ten large-scale test problems are also generated to compare the performance of the designed greedy heuristic against the SA algorithm. In these problems, 450-550 containers of 5 different types are randomly allocated in a container block of 45 yard-bays. The DRMG system needs to handle these containers in 10 sub-tours. It is almost impossible to use CPLEX to obtain the optimal solutions due to the excessive time. Therefore, only the results obtained from the greedy heuristic and the SA algorithm are shown in figure 6.7. It is noted the designed greedy heuristic outperforms the SA algorithm in solving all the ten problems. On average, the result from the greedy heuristic is 7.5% better than that from the SA algorithm.

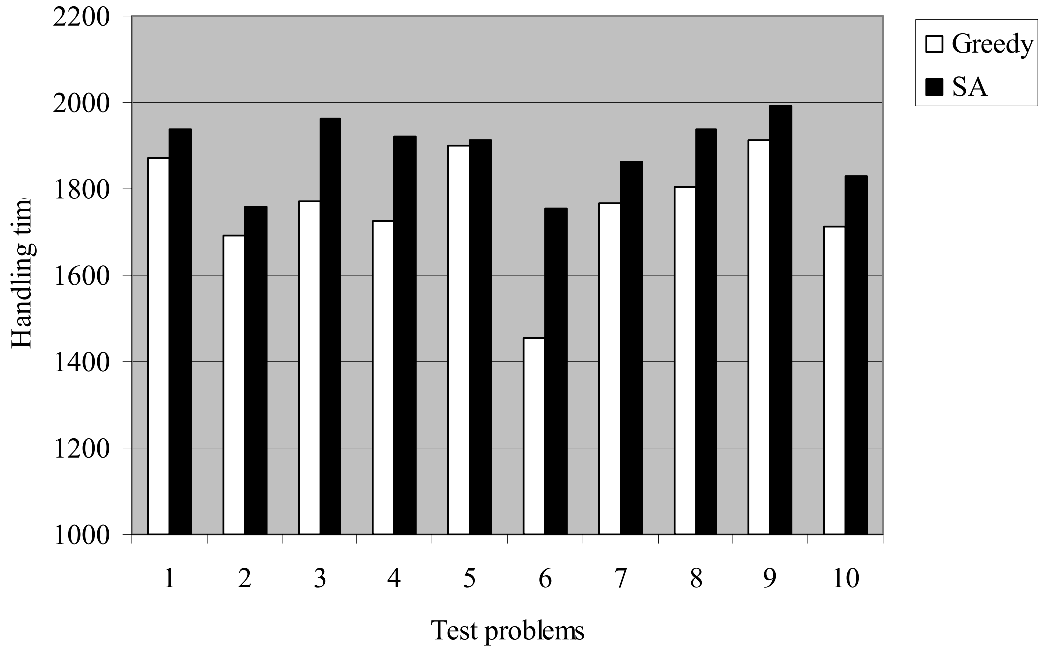


Figure 6.7 Comparison between The Results of The Greedy Heuristic and SA (large-scale problems)

6.6 SUMMARY

DRMG is an emerging container handling technology which recently came into use in Hamburg recently. In this chapter, the scheduling problem of the DRMG system used in loading outbound containers has been studied. An integer programming model is developed to formulate the problem. A greedy heuristic and a SA algorithm, therefore, is designed to solve the problem. Both small-scale and large-scale test problems are generated to evaluate the performance of the designed greedy heuristics. The small-scale problem tests show that the results of both the algorithms are close to the optimal solution. The large-scale problem tests show that the greedy heuristic outperforms a simulated annealing algorithm, which has been shown to perform well in solving similar scheduling problems. Since the greedy algorithm performs well and is easy to implement, it could be a promising operation strategy of DRMG systems in traditional yard truck based container terminals.

CHAPTER 7

SIMULTANEOUS LOAD SCHEDULING OF QUAY CRANE AND YARD CRANE IN PORT CONTAINER TERMINALS

7.1 INTRODUCTION

The two main types of equipment in port container terminals are QCs and YCs. The scheduling problems of both types of equipment are important issues in port terminal operations and will significantly affect the overall efficiency of terminal operations. The scheduling problem of multiple YC systems alone has been intensively studied in the past chapters. Nevertheless due to the fact that the YC scheduling problem is closely related to QC scheduling problem, it will be meaningful to consider the two problems at the same time.

This chapter proposes a load scheduling method which takes into account both the QC scheduling problem and the yard scheduling problem. A QC load schedule and its corresponding YC load schedule are constructed simultaneously so that a holistic consideration of the loading process is achieved. A mathematical model is developed to formulate the simultaneous load scheduling problem of QC and YC. A genetic algorithm is designed for the problem solution. The best performing parameters of the algorithm are found through numerical experiments presented.

7.2 SIMULTANEOUS SCHEDULING OF QUAY CRANE AND YARD CRANE

Loading outbound containers and discharging inbound containers are the two primary operations in port container terminals. In loading outbound containers, YCs will pick up the desired containers from container blocks and load them onto the yard trucks waiting aside. These yard trucks will then transport the containers to QCs, which will finally load the containers onto the containerships. The converse is true for the discharging of inbound containers.

In real operation, the terminal operators usually will receive the information of the ship's contents from the ship operator. The information includes the layout of the onboard containers, the list of containers needed to be discharged as well as the containers needed to be uploaded at the terminal. Based on the information, the terminal operators will conduct QC scheduling to determine the number of QCs to be assigned to the ship and the sequence of ship-bay that each QC will serve. After constructing the QC scheduling, the terminal operator then can develop YC scheduling which is to determine the job sequences of the YCs to serve the QC operations.

Since both the QC scheduling and YC scheduling are key issues in container terminals, several studies have been conducted to acknowledge the great importance of the scheduling problem of QC and YC in determining the overall efficiency of container port operations. However, despite the fact that the QC scheduling problem and YC scheduling problem are closely related with each other, there has not been any attempt to study these two problems simultaneously in the literature.

7.2.1 QC Load Scheduling Problem

Containers to be loaded in one terminal normally will be stacked in several ship-bays on the vessel. In practice, QCs will load the containers on a ship-bay by ship-bay basis. Once a QC completes all the loading jobs in one ship-bay, it will move to another ship-bay and perform the loading jobs there. Therefore, the goal of the QC scheduling problem is to determine the sequence of ship-bay that each QC will serve so that the loading time is minimized. In practice, it is common to divide a vessel into several working areas and each area is served by one QC. As a preliminary study on the simultaneous scheduling of QC and YC, only the load schedule of one QC is investigated in this chapter. Figure 7.1 shows a plan view of a containership. The shadowed ship-bays in the figure are the ship-bays where the containers will be stacked and hence are the ship-bays where the QC needs to work at. It can be easily seen that the QC scheduling problem here is a typical Traveling Salesman Problem (TSP). Hence a network model can be developed to formulate the QC scheduling problem. The feasible solutions of the QC scheduling problem can be represented by cycles on Figure 7.2, where the numbers on the arrows indicate the distance between each pair of nodes.

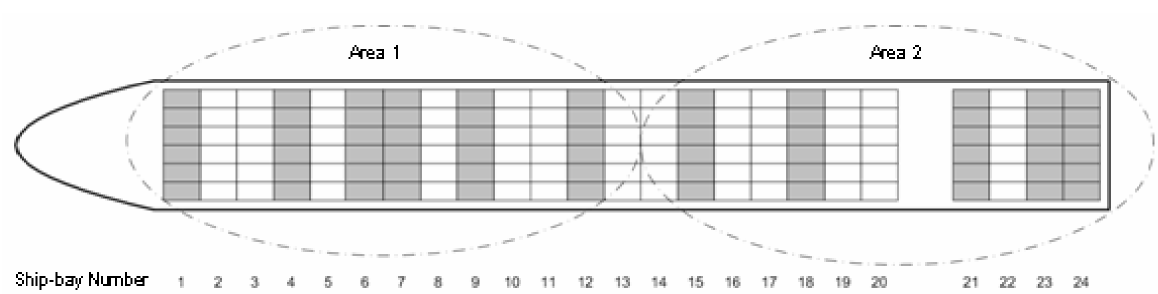


Figure 7.1 A Plan View of a Containership

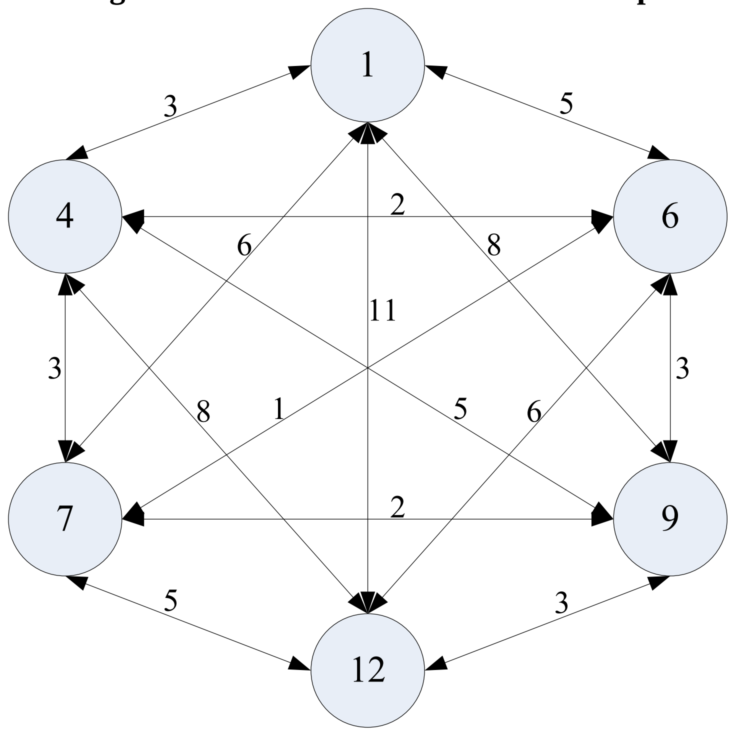


Figure 7.2 A Sample Network of the QC Scheduling

7.2.2 YC Load Scheduling Problem

Once the QC load schedule is determined, YCs, which are assigned to the QC, need to load the containers from the stack area in the exact same order as specified by the QC load schedule. As aforementioned, outbound containers are usually stored in a series of separated yard-bays in the container blocks, thus the YCs need to traverse the container blocks to fetch the required containers. The YC load scheduling problem here is to determine the sequence of yard-bays for each YC to visit and the number of containers to be picked up at each visiting yard-bay. In line with the practical operation, two YCs are used to serve one QC in this study.

The following example illustrates the YC load scheduling problem. The QC load schedule is given and shown in Table 7.1. Container type may refer to the destination or other attributes of the containers. In this study, to simplify the problem formulation, only one type of containers is assumed to be stored in one ship-bay. In the case where two or even more types of containers are stored in one ship-bay, the ship-bay can be split into several virtual ship-bays sharing the same position according to the number of container types. The container block map which shows the distribution of the required containers in the yard is shown in Table 7.2.

Table 7.1 A Sample QC Load Schedule

Sequence	1	2	3	4	5	6
Ship-bay number	1	4	7	9	6	12
Container type	A	C	B	A	C	B
No. of containers	20	18	22	19	22	25

Table 7.2 The Distribution of Containers in the Yard

Yard-bay number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Container type	A		B	C		C	A	B	C	A		B	A	C	B
No. of containers	8		15	10		15	8	12	8	13		11	10	7	9

Two YCs, namely, YC 1 and YC 2 are assumed to work in the container block for the loading operation of the quay carne. According to the load schedule of the QC, the two YCs need to pick up 20 containers of Type A together first. One possible schedule of the two YCs could be YC 1: 1(6) – 7(5); YC 2: 10(4) – 13(5) (YC 1 first visits Yard-bay 1 and pick up 6 containers there then visits Yard-bay 7 and pick up 5 containers. Meanwhile YC 2 will visit Yard-bay 10 and 13 and pick up 4 and 5 containers, respectively). Alternative schedules could be YC 1: 1(5) – 7(4); YC 2: 13(8) – 10(3) and so on. It should be noted that the YCs are of the same size and cannot pass each other. Therefore some of the schedules may not be feasible due to the non-interference constraint of the YCs. After all the 20 containers of Type A are picked up, the YCs then can start to work for sequence 2, picking up 18 containers of Type C, and so on.

It is obvious that different QC load schedules will lead to different YC schedules and subsequently lead to different loading times. Therefore it is meaningful to study the QC and the YC scheduling problems at the same time in order to synchronize the QC and YC operations. One important issue for achieving a simultaneous scheduling of QC and YC is to ensure that the containers picked up by the YCs are in the same order as the QC required, which is in fact the linkage between QC and YC operations.

7.3 MATHEMATICAL PROGRAMMING FORMULATION

To formulate the mathematical programming model for the proposed problem, following assumptions are first introduced.

i. Once the QC starts to work at a ship-bay, it will finish loading all the containers for the ship-bay before it moves to other ship-bays. This implies that the QC will work at one ship-bay exactly one time.

ii. There is only one type of container stacked in one yard-bay, which is the common practice in allocating space in the stack area of container terminals.

iii. The time required for an YC to load a container is assumed to be the same for all the containers despite the exact storage positions of individual containers.

iv. The transportation time of containers from the YCs to the QC is assumed to be the same as the average transportation time.

If the detailed transportation process of containers is considered, the problem will become much more complicated. Therefore as a preliminary research on the simultaneous scheduling of QC and YC, this study will mainly focus on synchronizing the loading sequence of QC and YC.

The upper bound for the total loading time of the optimal YC scheduling is also assumed to be known and this upper bound is partitioned into T time units. One time unit is defined as the time required for an YC traversing the distance of one yard-bay. The time required for a QC to traverse the distance of one ship-bay is also assumed to one time unit. The loading time of one container, T_H, is taken to be a multiple of this time unit. The K YCs are numbered from *1* to K from left to right according to their initial location at Period *0*. Since the YCs are of same size, so they cannot pass each other which implies that YC k can only move in the range limited by the locations of YC k-*1* and k+*1*.

To formulate the problem, a "sub-tour" is defined as a sequence of containers that needs to be picked up together by the YCs, which is according to the load schedule of QC. For example, the containers to be loaded by the QC to ship-bay i is defined as Sub-tour i. The following notations are used to formulate the problem.

$$W_{i,j} = \begin{cases} 1 & \text{if the QC loads containers at Ship-bay } j \text{ immmediately after loading container} \\ & \text{at Ship-bay } i \\ 0 & \text{otherwise (a decision variable)} \end{cases}$$

$$X_{i,l,k,t} = \begin{cases} 1 & \text{if YC } k \text{ finishes loading one container for Sub-tour } i \text{ at Yard-bay } l \text{ at time } t \\ 0 & \text{otherwise (a decision variable)} \end{cases}$$

$$Y_{l,k,t} = \begin{cases} 1 & \text{if YC } k \text{ is at Yard-bay } l \text{ at time } t \\ 0 & \text{otherwise (a decision variable)} \end{cases}$$

Ω the set of ship-bays where the QC needs to work at

$D_{i,j}$ the distance between Ship-bay i and Ship-bay j

N_i the number of containers need to be loaded for Sub-tour i

C_j the number of containers stacked at Yard-bay j before the loading process start

$B(i)$ the set of yard-bays where the containers required by Sub-tour i are located

S the number of sub-tours for the whole loading process

Ψ the set of sub-tours

I_{QC} the initial position of the QC

I_k the initial position of YC k

F_k the final position of YC k

α_1 the weight of the travel time of the QC

α_2 the weight of the loading time of the YCs

F auxiliary variable

The objective function is to minimize the summation of the weighted QC loading time and YC loading time. The QC loading time can be separated into two parts: (1) the handling time for the QC to load containers, and (2) the travel time for the QC to traverse along the track. Since the total number of containers to be loaded is known, the handling time of the QC becomes a constant, which makes the QC loading time only depending on the travel time. Therefore only the QC travel time needs to be considered. The objective function then can be represented by the following equation, in which the first term represents the weighted QC travel time and the second term represents the weighted YC loading time.

$$\text{Minimize } \alpha_1 \sum_{i\in\Omega, j\in\Omega} D_{i,j} W_{i,j} + \alpha_2 F \tag{7.1}$$

Subject to

$$\sum_{j\in\Omega} W_{I_{QC},j} = 1 \tag{7.2}$$

$$\sum_{i\in\Omega} W_{i,I_{QC}} = 1 \tag{7.3}$$

$$\sum_{i\in\Omega} W_{i,j} = 1 \quad j\in\Omega \tag{7.4}$$

$$\sum_{j\in\Omega} W_{i,j} = 1 \quad i\in\Omega \tag{7.5}$$

$$\sum_{i\in\Phi}\sum_{j\in\Phi} W_{i,j} \le |\Phi| - 1 \qquad \Phi\subseteq\Omega \tag{7.6}$$

$$F \ge tX_{i,l,k,t} \qquad i=1,2\cdots,S;\ l=1,2\cdots,B;\ k=1,2\cdots,K;\ t=1,2\cdots,T \tag{7.7}$$

$$\sum_{l\in B(i)}\sum_{k=1}^{K}\sum_{t=1}^{a-1} X_{i,l,k,t} - N_i \le M(1-X_{j,l',k',a}) + M(1-W_{i,j})$$

$$i,j\in\Psi;\ l'\in B(j);\ k'=1,2\cdots,K;\ a=T_H+1,T_H+2\cdots,T \tag{7.8}$$

$$\sum_{l\in B(i)}\sum_{k=1}^{K}\sum_{t=1}^{a-1} X_{i,l,k,t} - N_i \ge M(X_{j,l',k',a}-1) + M(W_{i,j}-1)$$

$$i,j\in\Psi;\ l'\in B(j);\ k'=1,2\cdots,K;\ a=T_H+1,T_H+2\cdots,T \tag{7.9}$$

$$\sum_{i\in\Psi}\sum_{k=1}^{K}\sum_{t=1}^{T} X_{i,l,k,t} = C_l \quad l=1,2\cdots,B \tag{7.10}$$

$$\sum_{l=1}^{B}\sum_{k=1}^{K}\sum_{t=1}^{T} X_{i,l,k,t} = N_i \quad i\in\Psi \tag{7.11}$$

$$\sum_{i\in\Psi}\sum_{j=1}^{B} X_{i,j,k,t} \le 1 \quad k=1,2\cdots,K;\ t=1,2\cdots,T \tag{7.12}$$

$$\sum_{i\in\Psi}\sum_{l=1}^{B}\sum_{a=1}^{T_H-1} X_{i,l,k,t-a} \le M(1-X_{i,l,k,t}) \quad k=1,2\cdots,K;\ t=1,2\cdots,T \tag{7.13}$$

$$\sum_{a=0}^{T_H} Y_{l,k,t-a} - T_H - 1 \ge M(X_{i,l,k,t}-1)$$

$$i \in \Psi;\ l = 1,2\cdots,B;\ k = 1,2\cdots,K;\ t = T_H + 1, T_H + 2\cdots,T \tag{7.14}$$

$$M(1 - Y_{b,k,t}) \geq \sum_{l=b}^{B} Y_{l,k-1,t} \quad b = 1,2\cdots,B;\ k = 2,3\cdots,K;\ t = 1,2\cdots,T \tag{7.15}$$

$$\sum_{l=1}^{B} Y_{l,k,t} = 1 \quad k = 1,2\cdots,K;\ t = 1,2\cdots,T \tag{7.16}$$

$$\sum_{k=1}^{K} Y_{l,k,t} \leq 1 \quad l = 1,2\cdots,B;\ t = 1,2\cdots,T \tag{7.17}$$

$$\sum_{l=b-1}^{b+1} Y_{l,k,t+1} \geq Y_{b,k,t} \quad b = 1,2\cdots,B;\ k = 1,2\cdots,K;\ t = 1,2\cdots,T \tag{7.18}$$

$$\sum_{l=b-1}^{b+1} Y_{l,k,t-1} \geq Y_{b,k,t} \quad b = 1,2\cdots,B;\ k = 1,2\cdots,K;\ t = 1,2\cdots,T \tag{7.19}$$

$$Y_{I_k,k,1} = 1 \quad k = 1,2\cdots,K \tag{7.20}$$

$$Y_{F_k,k,1} = 1 \quad k = 1,2\cdots,K \tag{7.21}$$

, where M is a big positive number.

Constraints (7.2) to (7.6) are the flow conservation constraints for the QC movement. A feasible solution of the QC is represented by a cycle on the corresponding network: Constraints (7.2) are the outflow constraints at the initial node. Constraints (7.3) are the inflow constraints at the final node. Constraints (7.4) and (7.5) are the inflow and outflow constraints for the other nodes respectively. Constraints (7.6) are to ensure the connectivity of the solutions, which eliminate the isolated cycles from the solution set. Constraints (7.7) state the definition of YC loading time, which is actually the makespan of the YC loading process. Constraints (7.8) and (7.9) ensure that the containers loaded by the YCs are in the same order as the QC required. Constraints (7.10) ensure the number of containers picked up the YCs during the whole loading process at one yard-bay equals to the initial number of containers stacked at that yard-bay. Constraints (7.11) ensure the number of containers picked up the YCs during one sub-tour equals to the number which is required by the YC load schedule. Constraints (7.12) ensure an YC can at most finish loading one container for one period. Constraints (7.13) ensure that an YC cannot finish any loading jobs during the time interval $t-T_H-1$ to *t-1* if it completes one loading job at period *t*. Constraints (7.14) ensure during handling one container the YC will stay at the container location throughout the operation. Constraints (7.15) ensure the movement of the YCs is free of inter-YC interference. Constraints (7.16) state that one YC can only be at one yard-bay during one period. Constraints (7.17) state that only one YC can be at one yard-bay in each period. Constraints (7.18) and (7.19) ensure that the YCs can at most move one yard-bay during one period. Constraints (7.20) and (7.21) state the initial and final positions of the *K* YCs.

7.4 SOLUTION TECHNIQUES

Since the simple YC scheduling problem has been proven to be NP-complete. It is needless to say that the more complicated simultaneous QC and YC scheduling problem is also an NP-Complete problem, which makes exact algorithm impractical in solving the large-scale problems. Therefore the genetic algorithm is first adopted to solve the proposed problem. A problem-oriented QC and YC scheduling heuristic is also developed for the problem solution. For the sake of brevity, a system of two YCs is used to illustrate these heuristic approaches.

7.4.1 A Genetic Algorithm

Genetic algorithm (GA) is the most commonly used meta-heuristic algorithm for solving intractable optimization problem. The solution representation, fitness evaluation, crossover operator and mutation operator of the GA are illustrated as follows.

7.4.1.1 Solution representation

To implement the GA algorithm, a method of encoding the feasible solutions is first introduced. The feasible solutions of the problem are represented by strings of integer numbers in this study. Each string consists of three sections. The first section is the load schedule of the QC and the last two sections are the corresponding load schedules of YC *1* and YC *2*, respectively. Figure 7.3 is a sample of the first section of a solution string, which indicates the QC will work at ship-bays in the sequence of 1→4→7→9→6→12. To satisfy the load schedule of the QC, the YCs need to load sub-tours in the same order. Figures 7.4 and 7.5 show parts of the second and third sections, respectively. Each section can be further divided into several segments according to the number of sub-tours. The shaded integers in one segment indicate the visiting sequence of yard-bays of an YC and the followed integers indicate the number of containers to be loaded by the YCs at these yard-bays. For example, for Sub-tour *1*, YC *1* will visit yard-bay 1→7 and load 6 and 5 containers at each visit accordingly. Meanwhile YC *2* will visit yard-bay 10→13 and load 4 and 5 containers respectively.

1	4	7	9	6	12

Figure 7.3 First Section of a Sample Solution String

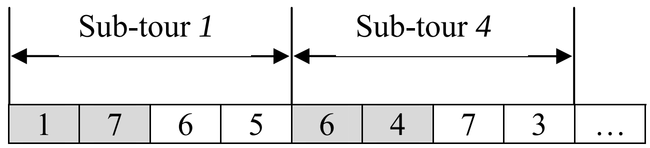


Figure 7.4 Parts of the Second Section of a Sample Solution String

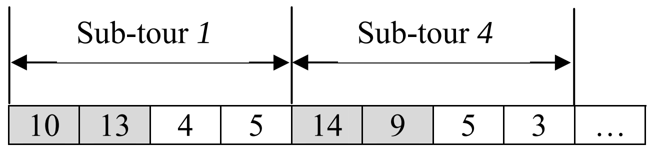


Figure 7.5 Parts of the Third Section of a Sample Solution String

7.4.1.2 Fitness evaluation

Objective function (1) is used to evaluate the fitness function of the solutions. All the solution strings of a population are sorted from small to large according to their objective function values, and then the reciprocals of their ranks (*r*) are used to calculate the relative fitness value of the strings with equation (7.21).

$$r_i' = \frac{1/r_i}{\sum_i 1/r_i} \tag{7.21}$$

7.4.1.3 Crossover Operator

The position-based crossover operator is adopted in this study. The crossover operator is first executed on the first section of a solution string. The procedure of the operation is as following:

Step 1: Choose two cut points randomly.

Step 2: Copy the integers of Parent *1* between these two points to Offspring *1* according to their initial position in Parent *1*.

Step 3: Delete the integers which have been copied in Step *2* from Parent *2*.

Step 4: Place the rest integers of Parent *2* to the empty positions of Offspring *1* according to their initial sequence in Parent *2*.

The following Figure 7.6 provides an example of the aforementioned crossover process. Once the first section of Offspring *1* is generated, the second and third sections of the offspring are generated by reordering the segments of the second and third sections of Parent *1* according to the first section of the offspring. Offspring *2* can be generated in the same way by switching the roll of Parent *1* and Parent *2*.

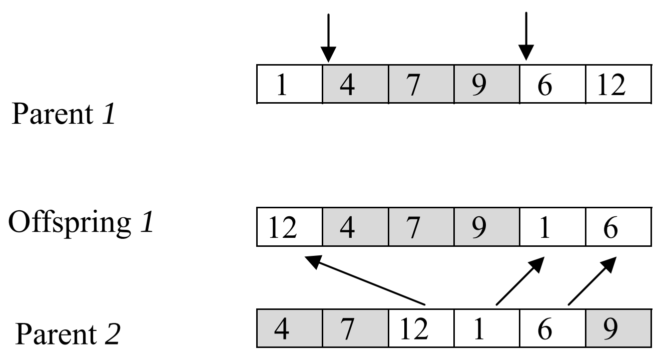


Figure 7.6 An Illustration of the Crossover Operation

7.4.1.4 Mutation Operator

The procedure of the mutation operator is presented in the following:

Step 1: A cut point is randomly chosen in the first section.

Step 2: The integers of the first section after the cut point is reordered.

Step 3: The segments of the second and third section are rearranged according to the updated first section.

7.4.1.5 Selection Method

A selection method states how to choose new population from original population and offspring. The selection approach adopted here is based on enlarged sampling space. Figure 7.7 illustrates the procedure of this selection operation

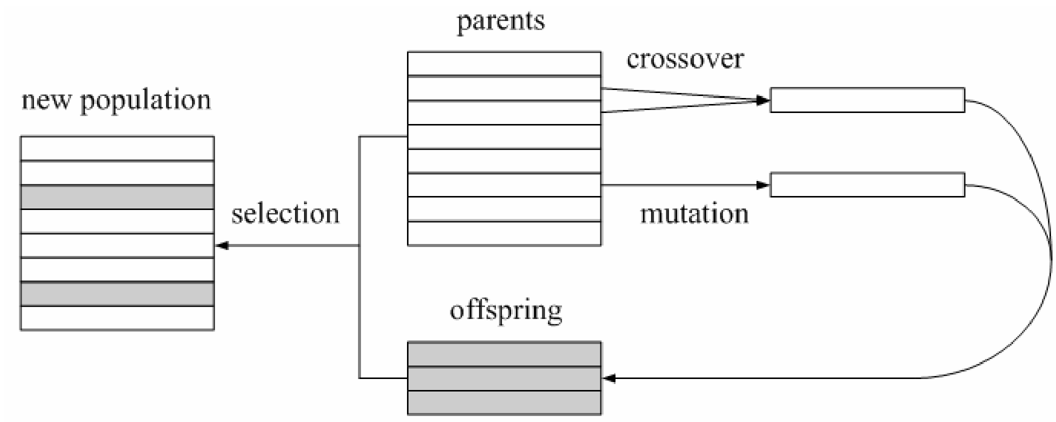


Figure 7.7 An Illustrates of the Selection Method

7.4.2 QC and YC Scheduling Heuristic

Besides the aforementioned genetic algorithm, a problem-oriented QC and YC scheduling heuristic (QYSH) is also developed to solve the proposed problem. The QC scheduling of QYSH is obtained by enumerating all the possible QC schedules. Based on each generated QC schedule, the scheduling of YC is then conducted using the following rules.

Rule 1

Both the two YCs will choose the containers in their nearest yard-bays, which satisfy the loading sequence requirement.

Rule 2

If the same yard-bay is identified to be the closest yard-bay to both YC *1* and YC *2* and it is also the last yard-bay of containers for the current sub-tour, it will be assigned to the closer YC. In the case where two YCs are of equal distance to the yard-bay, the yard-bay will be assigned to one YC arbitrarily.

<u>Rule 3</u>

If the same yard-bay is identified to be the closest yard-bay to both YC *1* and YC *2* and it is not the last yard-bay of containers for the current sub-tour, following figure 7.8 illustrates the nine scenarios that can occur and the scheduling rules of the YCs in these scenarios.

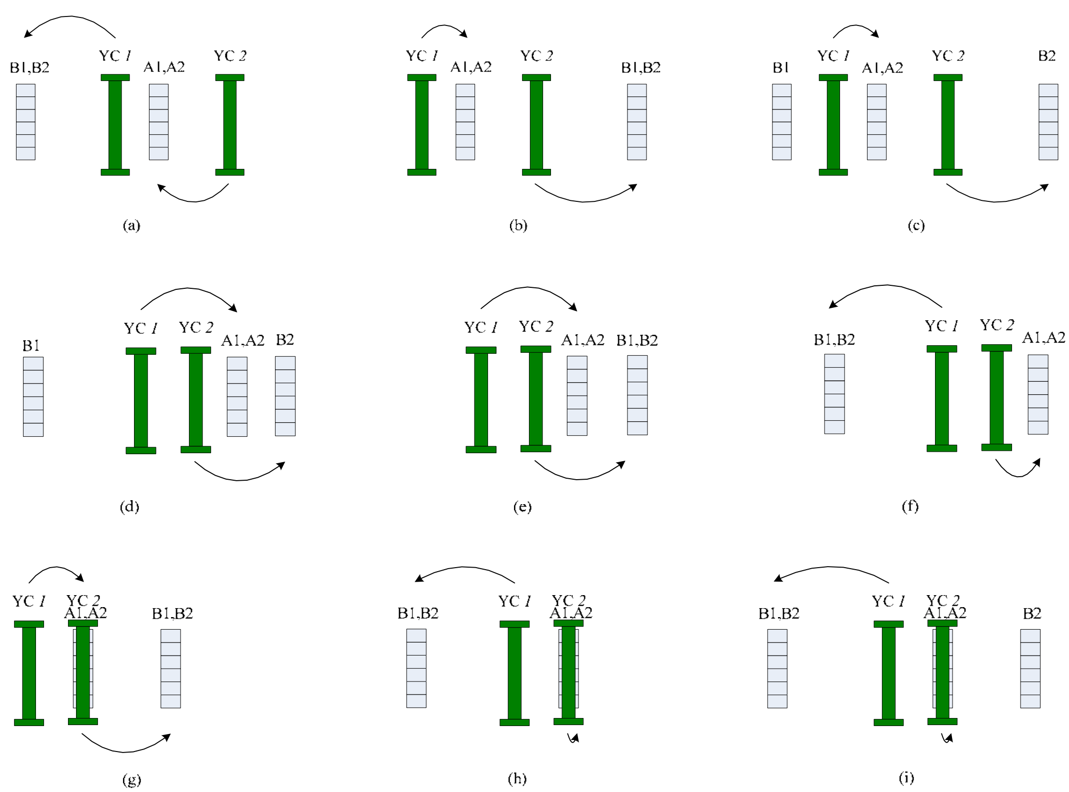


Figure 7.8 An Illustration of the Rules to Choose Yard-bays

In the above figure, A1 and B1 are the closest and second closest yard-bays to YC *1* respectively. A2 and B2 are the closest and second closest yard-bays to YC *2* respectively. The arrows indicate which yard-bays the YCs will choose. E.g. in (a), YC *1* will choose yard-bay B1 (B2) and YC *2* will choose A1 (A2).

Rule 4

In the case where there are two yard-bays of containers are equal distant to one YC, the YC will choose the yard-bay which is further from the other YC.

Rule 5

If there is no available container for an YC, it will stay still.

7.5 NUMERICAL EXPERIMENTS

Numerical experiments are conducted to evaluate the performance of the two proposed solution methods. The relationship between the weights (α_1 and α_2) and their corresponding QC travel time and YC loading time are also investigated through the experiments.

7.5.1 Experiment Design

Ten sample problems are generated as follows:

1) Generate the ship plan:
   a) Randomly choose the total number of containers to be loaded from 300 to 450.
   b) For each sample problem, the containers are randomly classified into five types, namely A, B, C, D and E.
   c) Each type is then further divided into 2 or 3 groups.
   d) Each group is randomly assigned to one ship-bay.

2) Allocate the containers in the stack area: Containers are randomly allocated in the container block, which consists of 45 yard-bays subject to the constraint that only one type of container can be stacked in one yard-bay.

7.5.2 Sensitivity Analysis of the Parameters of the GA Algorithm

It is well known that the performance of GA is sensitive to the parameters used. Thus computer codes programmed by C++ language are executed to find the best combination of GA parameters. It was found through a primary test that 200 is a proper value of population size and 500 generations are sufficient to make the average objective value converge to a stabilized value. The tested values of the crossover rate (p_c) were 0.2, 0.4, 0.6 and 0.8. The tested values of mutation rate (p_m) were 0.1, 0.3, 0.5 and 0.7, subject to the constraint that the sum of p_c and p_m is not greater than one. In the case that the sum of p_c and p_m is less than one, new solution strings will be generated to fill up the vacancies in the next generation.

It is noted that the results of the GA algorithm were sensitive to the random seed generated. To avoid this bias, both the average objective function values and the best objective function values were recorded over ten runs in order to find the best combination of parameters. Tables 7.3 and 7.4 illustrate the average objective function values and the best objective function values obtained by different combinations of p_c and p_m. According to the obtained results, $p_c = 0.4$ and $p_m = 0.5$ was chosen as the best performing combination of parameters. Figure 7.9 shows the trends of objective function value, QC travel time and YC loading time in one experiment using the selected parameters. All these values converged and stabilized within 500 iterations.

Table 7.3 The Average Objective Function Value for Different Values of Parameters

Pm \ Pc	0.2	0.4	0.6	0.8
0.1	1928.1	1937	1928.4	1930.1
0.3	1932.7	1930.9	1928.9	
0.5	1925.3	**1921.8**		
0.7	1928.3			

$(\alpha_1 = \alpha_2 = 1)$

Table 7.4 The Best Objective Function Value for Different Values of Parameters

Pm \ Pc	0.2	0.4	0.6	0.8
0.1	1907	1919	1907	1922
0.3	1898	1909	1916	
0.5	1920	**1890**		
0.7	1919			

$(\alpha_1 = \alpha_2 = 1)$

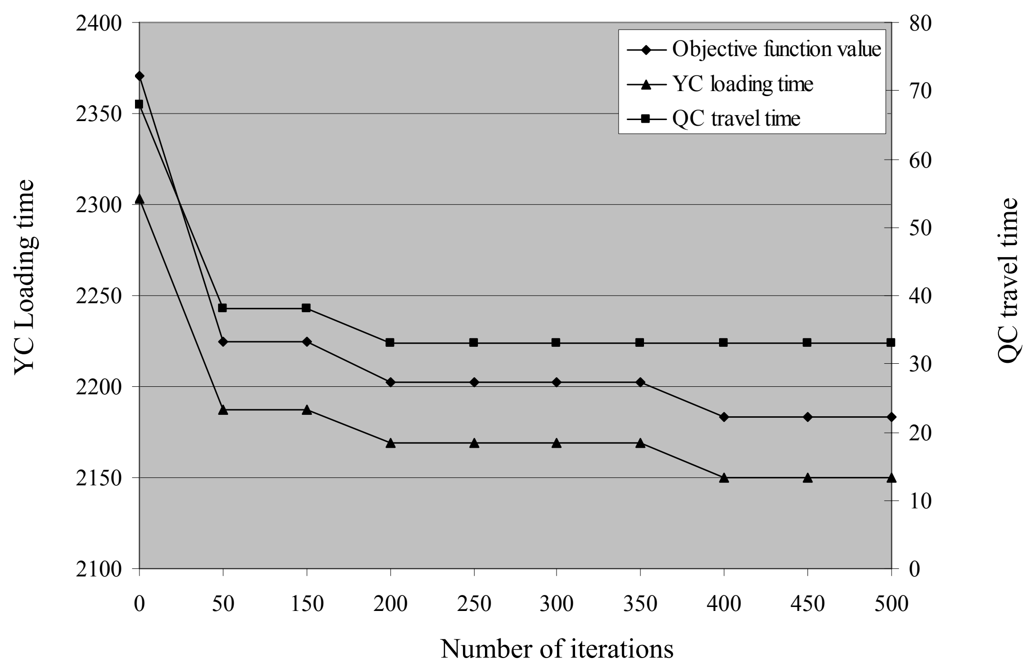


Figure 7.9 Trends of Objective Function Value, QC Travel Time and YC Loading Time

7.5.3 Performance Comparison between GA and QYSH

Since p_c = 0.4 and p_m = 0.5 were chosen as the best performing combination of GA parameters, this set of parameters is also used to solve other generated test problems. The aforementioned QYSH method is also coded into computer programs and executed to obtain the problem solution. The comparison of the results obtained by these two methods is shown in figure 7.10. The results show that the QYSH method consistently outperforms the GA. On average the objective function value obtained by QYSH is 14.9% lower than that obtained by GA. This suggests that the designed QYSH method could be a promising approach to conduct the simultaneous QC and YC scheduling.

$(\alpha_1 = \alpha_2 = 1)$

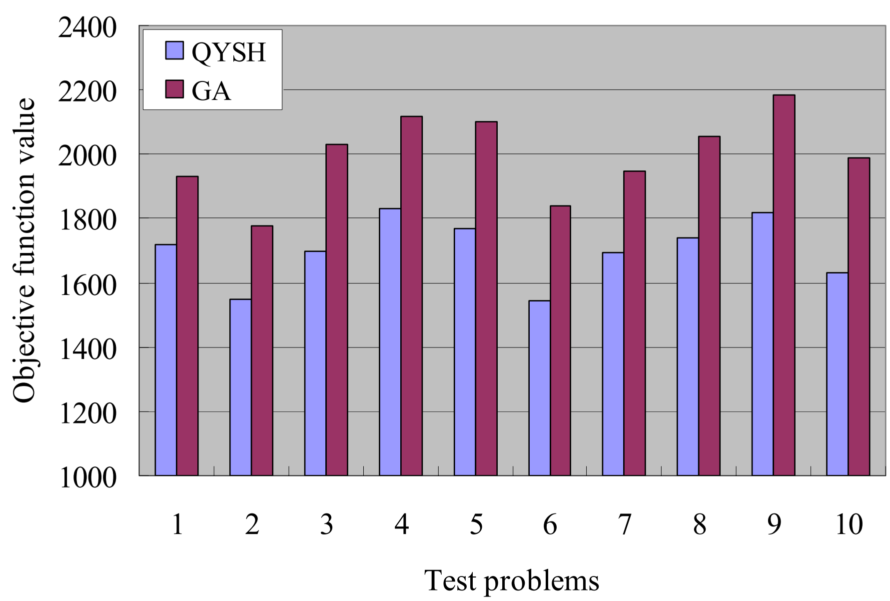


Figure 7.10 Comparison between QYSH and GA

7.5.4 QC travel time and YC loading time

Table 7.5 is the YC loading time and the QC travel time obtained by QYSH with different values of the weights, α_1 and α_2, which actually represent the importance of the QC operation and YC operation, respectively. The results showed that, consistent with intuition, in general the QC travel time decreased and the YC loading time increased with higher value of α_1 while the QC travel time increased and YC loading time decreased with higher value of α_2. It was also noted that in some test problems, despite the changes of the weight, the QC travel time and YC loading time remain consistent. It is possible to speculate that this is due to the inherent characteristics of input information of the problems.

Table 7.5 Relationship between the weights (α_1 and α_2) and QC travel time and YC loading time

(YC: time unit; QC: hold)

α_1	α_2	Problem	1	2	3	4	5	6	7	8	9	10
10	1	QC	24	28	19	28	29	17	18	29	20	25
		YC	1720	1540	1692	1881	1741	1600	1712	1745	1810	1620
1	1	QC	39	38	37	57	29	42	34	40	47	36
		YC	1678	1509	1660	1775	1741	1501	1659	1698	1770	1594
1	10	QC	46	60	62	104	71	48	52	100	47	67
		YC	1675	1504	1649	1746	1714	1499	1641	1683	1770	1547

7.6 SUMMARY

The operations of QC and YC are two key components of the container terminal operations. Although the two operations are closely related to each other, to the authors' best knowledge, these two problems have not been simultaneously considered in one model in the literature. As the first study on simultaneous scheduling of QC and YC, this chapter has developed an integer programming model to formulate the combined scheduling problem and also proposed the GA and the QYSH methods to solve the problem. The results obtained through numerical experiments showed that the problem-oriented QYSH method significantly outperforms the GA and could be applied in real operation to conduct simultaneous QC and YC scheduling. The influence of the weights on the corresponding QC travel time and YC loading time was also investigated. The results showed that a higher weight will generally leads to a shorter operation time.

CHAPTER 8

CONCLUSIONS

8.1 CONCLUSIONS

The goal of this thesis was to provide efficient strategies for operating multiple YC systems in port container terminals. First a simplified MYCS problem, TYCS problem, was examined, followed by a study on the typical MYCS problem. Subsequently, problems derived from the MYCS problem, MYCS-B problem and DRS problem, were investigated. Finally a simultaneous scheduling problem of QC and YC was studied. All these problems are formulated by mathematical models and successively solved by designed solution techniques.

In the first part of the thesis, the proposed TYCS problem was studied. It is the first attempt in the literature to investigate the scheduling problem of multiple YC system under container loading sequence constraints. The problem was formulated by a mathematical model. A SA algorithm was developed to solve the proposed problem. A series of numerical experiments were designed to test the performance of the SA algorithm. To evaluate the algorithm, the computational results obtained from the algorithm are compared against the estimated lower bounds. The result showed that the proposed SA algorithm is an efficient approach in solving the TYCS problem.

In the second part of the thesis, the typical MYCS problem was formulated by an integer programming model. It is a innovative work on the MYCS problem. Both the precedence constraints and the interference constraints are considered in the problem formulation. Three different heuristic algorithms were developed to solve the proposed problem. The results showed that all the algorithms performed well in solving small scale problems and the greedy heuristic algorithm consistently outperforms the other two algorithms in solving large scale problems. The reason why meta-heuristic algorithms failed in achieving better solutions compared to a simple greedy heuristic algorithm probably lies in the complexity of the problem itself. Due to the complicated non-interference constraints and loading sequence requirement constraints, the capabilities of the meta-heuristics to generate feasible solutions are significantly restricted. Therefore the solution space that the algorithms can explore is limited and as a result the quality of the solution of the solutions is reduced.

The third part of the thesis treated the MYCS-B problem. It is also an original work on the scheduling problem of multiple YCs in container terminals with buffer areas. The problem was also formulated as an integer programming model. A similar greedy heuristic algorithm was applied to the problem. The results showed that the multiple YC system in the terminals with buffer areas outperformed that in the terminal without buffer areas and the idle time of the YCs was significantly reduced by using buffer areas. This is because adding buffer areas in a container terminal increases the degree of freedom of YC operations by allowing them to work ahead of the QC load schedule. This prevents an YC from idling in the case that no containers for the current sub-tour are available in its working range.

The fourth part of the thesis investigated the DRS problem. DRMG system is a new container handling technology which is able to avoid the inter-crane interference problem in YC operations. Although currently the DRMG systems are only used in the AGV based container terminals, it is promising to deploy this technology in the yard truck based terminals in the near future. Therefore a mathematical model is developed to formulate the scheduling problem of DRMG system in yard truck based terminals. An operational strategy of the DRMG system was also proposed and it was shown to outperform the SA algorithm through computational experiments.

The last part of the thesis focused on the simultaneous scheduling problem of QC and YC. This is a novel study on the combined QC scheduling and YC scheduling problem. The problem was also formulated by an integer programming model and solved by a genetic algorithm. The results showed that, consistent with intuition, in general the YC operation time increased and the QC travel distance decreased with lower weight of YC operation while the YC operation time decreased and the QC travel distance increased with lower weight of QC operation.

It should be noted that the multiple YC scheduling were restricted to the loading process in import-export terminals only in this study. For the discharging process, it is the common practice that an inbound container is usually stacked at a designated empty space next to the inbound container which arrives before it. Therefore the YC operations are performed quickly and relatively simply. However for the loading process, since the outbound containers are usually scattered in the container blocks in the stack area and the containers picked up by YCs must satisfy the job sequences of the QCs, the scheduling problem of YCs, therefore, becomes much more complicated and needs intensive study. Hence only the load scheduling of YCs are studied in this thesis.

8.2 RESEARCH CONTRIBUTIONS

The main contributions of this study can be described as follows:

i. A comprehensive literature review on the scheduling of YC has been made and the details of the operations in port container terminals have been documented. This can serve as a stepping-stone for future research in the field of container terminals operations.

ii. The modeling approach used in this thesis can shed light on the mathematical formulation of other problems which share similar characteristic with the YC scheduling problem, especially the method proposed to formulate the interference and precedence constraints.

iii. In this thesis, several solution techniques are developed to solve the YC scheduling problem. The results of the study on MYCS problem indicates that compared to the widely used meta-heuristic algorithms, the relatively simple greedy heuristics algorithm is a more effective solution technique for solving the scheduling problem of the multiple YC system. Therefore it can be adopted by the container terminal operators to improve the efficiency of their operations.

iv. The proposed solution methods have been coded into computer programs. These source codes of the algorithms can be adopted as the core component of the future software development.

v. The influence of using buffer area in container terminals has also been examined in the study. By allocating some buffer areas inside the terminal, the productivity of YCs could be enhanced and the loading time at the stack side can be shortened at the expense of using more land space and more yard trucks. This result can be used by the terminal operators as a reference when deciding whether to use buffer areas in their terminals.

vi. The deployment strategy of the DRMG system in yard truck based container terminals was also investigated. The use of DRMG system in traditional yard truck based container terminals can eliminate the interference of YCs. The operational strategy of the DRMG system proposed outperformed the SA algorithm through numerical experiments. The result of this research can help to guide the future deployment of DRMGs in yard truck based container terminals.

vii. A simultaneous scheduling of QC and YC was also successfully accomplished in the study. Being the first study of its kind, this study can be used to improve the overall performance of QCs and YCs. It can also work as one component of the wholly integrated container terminal operating system which is to be developed in the future research.

8.3 RECOMMENDATIONS FOR FUTURE RESEARCH

i. In the proposed multiple YC scheduling problem, all the YCs are assumed to work for a single QC. It will be interesting to study the more complicated situation where multiple YCs are used to serve multiple QCs. The result of such a study could help to further increase the efficiency of both QCs and YCs.

ii. In this thesis, the containers in the same slot of one container block are treated equally despite their exact positions in the slot. A more detailed study which takes into account the exact location of containers in determining the YC schedule could help to ameliorate the results obtained from the current research.

iii. An integrated container terminal operation system which takes into account all the import aspects of the terminal operations will help the operators to eventually achieve a state-of-the-art operation. Two important components of the terminal operation, QC scheduling and YC scheduling have been studied simultaneously in the thesis. Future research can integrate the other components such as berth allocation and yard storage with the existing work presented in the thesis.

REFERENCES

Bish, E. K. A Multiple-crane-constrained Scheduling Problem in a Container Terminal. *European Journal of Operational Research,* 144, 83-107. 2003.

Daganzo, C.F. The Crane Scheduling Problem. *Transportation Research Part B,* 23, 159-175. 1989.

Glover, F. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operation Research,* 13, 533-549. 1986.

Glover, F. Tabu Search-part I. *ORSA Journal on Computing*, 1, 190-206. 1989.

Glover, F. Tabu Search-part II, *ORSA Journal on Computing*, 2, 14-32. 1990.

Hansen, P. The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming. In *Proceedings of Conference on Numerical Methods in Combinatorial Optimisation,* 1986, Capri, Italy.

Holland, J. Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press. 1975.

Kim, K.H. and K.Y. Kim. An Optimal Routing Algorithm for a Transfer Crane in Port

Container Terminals. *Transportation Science* 33(1), 17-33. 1999.

Kim, K.H. and K.Y. Kim. Routing Straddle Carriers for the Loading Operation of Containers Using a Beam Search Algorithm. *Computers & industrial Engineering,* 36, 109-136. 1999.

Kim, K.H., K.M. Lee and H. Hwang. Sequencing Delivery and Receiving Operations for Yard Cranes in Port Container Terminals, *International Journal of Production Economics,* 84, 283-292. 2003.

Kim, K.Y. and K.H. Kim. Heuristic Algorithms for Routing Yard-Side Equipment for Minimizing Loading Times in Container Terminals. *Naval Research Logistics,* 50, 498-514. 2003.

Kim, K.Y., S.H. Jung and K. H. Kim. Load Scheduling for Two Quay Cranes in Port Container Terminals. In *Proceeding of Intelligent Manufacturing & Logistics Systems*, August 2005, Japan, 1-11.

Kim, K.H., S.J. Wang, Y.M. Park, C.H. Yang and J.W. Bae. A Simulation Study on Operation Rules for Automated Container Yards. In *Proceedings of 7th Annual International Conference*, October 2002, Busan, Korea, 250-253.

Kim, K.H. and Y.M. Park. A Crane Scheduling Method for Port Container Terminals. *European Journal of Operation Research*, 156, 752-768. 2004.

Kirkpatrick, S., C.D. Gelatt, Jr. and M.P. Vecchi. Optimization by Simulated Annealing. *Science*, 220 (4598), 671-680. 1983.

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

Linn, R., J.Y. Liu, Y.W., Wan, C. Zhang and K.G.. Murty. Rubber Tired Gantry Crane Deployment for Container Yard Operation. *Computers & Industrial Engineering*, 45, 429-442. 2003.

Lundy, M. and A. Mees. Convergence of An Annealing Algorithm. *Mathematical Programming*, 34, 111–124. 1986.

Narasimhan, A. and U.S. Palekar. Analysis and Algorithm for the Yard Crane Routing Problem in Container Port Operation. *Transportation Science*, 36(1), 63-78. 2002.

Ng, W.C. Crane Scheduling in Container Yards with Inter-crane Interference. *European Journal of Operational Research*, 164, 64-78. 2005.

Park, Y.M. and K.H. Kim. A scheduling method for Berth and Quay cranes. *OR Spectrum*, 25, 1-23. 2003.

Pham, D.T. and D. Karaboga. Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks. U.K.: Springer. 2000.

Steenken, D., S. Voss and R. Stahlbock. Container Terminal Logistics and Operations Research – A Classification and Literature Review. *OR Spectrum,* 26(1), 3-49. 2004.

Taha, H. A. Operations Research: An Introduction. New Jersey, U.S.A.: Prentice Hall. 1997.

APPENDIX: Recent Research Accomplishments

[1] Lee D.H., Z. Cao and Q. Meng. Scheduling of Two-Transtainer Systems for Loading Outbound Containers in Port Container Terminals with Simulated Annealing Algorithm. *International Journal of Production Economics*, In press. 2006.

[2] Lee D.H., Z. Cao and Q. Meng. Simultaneous Scheduling of Quay Crane and Yard Crane in Port Container Terminals. *OR Spectrum*, Submitted, 2006

[3] Cao Z., D.H. Lee and Q. Meng. Load Scheduling for Two-Transtainer Systems in Port Container Terminals. *Journal of the Institution of Engineers, Singapore,* Accepted. 2006.

[4] Cao Z., D.H. Lee and Q. Meng. Scheduling of Multiple Yard Crane System for Loading Outbound Containers in Port Container Terminals. *Computers & Industrial Engineering*, Submitted, 2005.

[5] Cao Z., D.H. Lee and Q. Meng. Deployment Strategies of Double Rail Mounted Gantry Crane Systems for Loading Outbound Containers in Container Terminals. *International Journal of Production Economics*, Submitted, 2005.

[6] Cao Z., D.H. Lee and Q. Meng. Load Scheduling of Multiple Yard Crane Systems in Container Terminals with Buffer Areas. *OR Spectrum*, submitted, 2006

[7] Cao Z., D.H. Lee and Q. Meng. Scheduling of Multiple Yard Crane System with Container Loading Sequence Consideration. *Proceeding of 85th Transportation Research Board Annual Meeting, 2006,* Washington D.C., U.S.A.

[8] Cao Z., D.H. Lee and Q. Meng. Scheduling Two-Transtainer Systems for the Loading Operation of Containers Using a Revised Genetic Algorithm. *Proceeding of 85th Transportation Research Board Annual Meeting,* 2006, Washington D.C., U.S.A.

[9] Cao Z., D.H. Lee and Q. Meng. Scheduling of Two-Transtainer Systems in Port Container Terminals. *Proceeding of the First International Conference on Transportation Logistics, 2005,* Singapore.

[10] Cao Z., D.H. Lee and Q. Meng. Load Scheduling of Multiple Yard Crane Systems in Container Terminals with Buffer Areas. *Proceeding of the 36th International Conference on Computers & Industrial Engineering*, 2006, Taiwan.

[11] Wang H.Q., D. H. Lee, Z. Cao and L.X. Miao. Quay Crane Scheduling with Non-crossing and Safety Distance Constraints in Port Container Terminals. *Proceeding of the 36th International Conference on Computers & Industrial Engineering*, 2006, Taiwan.