Abstract of thesis entitled

Efficient Algorithms for Designing Yard Storage Templates for Export Containers

Submitted by

Li Ming Kun

For the Degree of Doctor of Philosophy

at the University of Hong Kong

October 2010

Effective storage space planning in the yards of many container terminals is important and complex especially when container stacking is of high density. The technique of yard template design, which normally specifies a static set of storage locations in a group of yard blocks for export containers of various vessel services, is particularly useful for achieving even workloads within each yard block to improve the landside productivity of such container terminals.

This research attempts to solve the problem of designing the optimal yard template to balance the workload of the yard blocks for a set of vessel services. The problem has received relatively little attention in the literature. Hence, the research firstly analyses the design of the optimal yard template for vessel services with a cyclical calling pattern. It is shown that the problem can be easily formulated as an integer program. However, results of computational experiments show that, even for some small-scale problems, the determination of a feasible solution requires a prohibitively long computational time. Therefore, the template design problem is decomposed into two sub-problems - Cluster Allocation Problem (CAP), the problem of determining the size of clusters assigned to each export yard block, and Cluster

Configuration Problem (CCP), the problem of determining the location and the shape of each cluster assigned to each yard block in the solution of CAP. These two sub-problems are then analyzed in detail to develop a two-phase solution algorithm so that CAP and CCP can be solved sequentially.

To solve CAP, an aggregation-disaggregation approach is adopted. Vessel services calling at the terminal at the same period are aggregated as a service group and the problem of determining the size of clusters assigned to different service groups in different yard blocks is formulated as an integer program. The integer program is then transformed into a problem similar to a transportation problem which can easily be solved. A method is developed to disaggregate the solution of the aggregated problem into the number of slots assigned to each service in each yard block.

With the solution of CAP found in phase 1, CCP is formulated as an integer program which can be interpreted as fitting clusters of different sizes into a number of rectangular yard blocks. CCP is shown to be identical to the well-known two-dimensional bin-packing problem except that the shape of each cluster is a decision variable. A heuristic is developed for solving CCP. Results of computational experiments show that the two-phase algorithm is indeed effective and efficient, with over 90% of the solutions found being the optimal ones.

The analysis is then extended to consider the yard planning problem for vessel services with an irregular calling pattern. The Scattered Stacking Strategy is proposed to guide the storage of export containers. The integer program and the two-phase algorithm developed earlier are then modified to solve the problem. Results of extensive computational experiments clearly show that the modified two-phase algorithm is an effective means for solving real life optimal yard template design problems.

(498 words)

Efficient Algorithms for Designing Yard Storage Templates for Export Containers

by

Li Ming Kun

M. M.; B. Sc.

Fudan University

A thesis submitted for fulfillment of the requirements for

the Degree of Doctor of Philosophy

at The University of Hong Kong

October 2010

DECLARATION

I declare that the thesis represents my own work, except where due acknowledgement is made, and that it has not been previously included in a thesis, dissertation or report submitted to this University or any other institution for a degree, diploma or other qualifications.

Li Ming Kun

October 2010

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my supervisors, Prof. K. L. Mak and Dr. W. C. Ng, for their constant guidance, support and help throughout the period of my Ph.D study. Without their supervision, I would never be devoted into the profound research fields for container terminal operations and complete this study on storage yard planning. Specifically, I would like to thank Prof. Mak for providing me good advices and Dr. Ng for leading me to an academic career.

I also wish to thank Ms. Christina Yeung, Ms. Kate Lee, Ms. Winne Tse and the technical staff members of the Department for their wonderful help and technical support during the last four years. My Cantonese has been greatly improved through my conversations with them.

Special thanks are made to the University of Hong Kong for giving me the valuable opportunity to learn in a good environment and to live in a unique university culture. I am fortune to make friends from both my residential hall- Morrison hall and my colleagues. Many of them are always willing to offer their helping hands.

Finally, I would like to express the greatest gratitude to my parents, my sister and my girlfriend who provide invaluable love and care supporting me at every time period. The thesis is dedicated to my dear family. Thanks for all.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Container Traffic

With the growth of the world economy, containerization has experienced significant expansion since 1960s. Many Asian countries and areas have recently become the world production and commercial centre that consequently promotes the container transport between Asia and the rest of the world. To compete in the booming market, a large amount of investment has been put into sea port construction in many Asian cities that strongly increases the handling capacities for containers in these terminals. According to the yearly reports of Hong Kong Census and Statistics Department in 2001 and 2010, half of the world top 10 container ports in terms of throughput were located in Asia in 2000. Only 9 years later, all the top 9 positions in the list were captured by Asian sea ports. The statistics also indicates that China has become a dominant player in the current sea freight market. Although most of its container ports were established after 1978 with even few specialized container terminals within the first 8 years after the opening up of its economy, the development of container terminals in China is incredible with around 30% yearly growth in terms of throughput since 1985 except in 2009, when the container service business experienced a huge decline due to the world-wide financial crisis. Table 1.1 shows the growth of the throughput in the top 5 container ports in Mainland China.

	1985	1990	1995	2000	2005	2006	2007	2008	2009
China	503	1560	6630	23480	67678	83543	101969	115782	110702
Shanghai	202	456	1527	5612	16469	19784	23853	25788	22551
Shenzhen	0	33	284	3994	14763	16793	18805	19733	16493
Guangzhou	47	110	515	1431	4209	6073	8388	10312	10107
Ningbo	1	22	160	902	4720	6507	8595	10166	9582
Qingdao	33	135	603	2120	5746	7039	8645	9197	9396

Table 1.1 Throughput of Mainland China's top 5 container ports ('000 TEU) *Source*: www.chineseshipping.com.cn

With the powerful import and export engine of Mainland China, further increase of container transport is still expected in Chinese sea ports as well as other important and busy regional container ports such as Hong Kong and Singapore. In order to tackle the large amount of shipping services, port owners must pursue sustained investments and effective methods for improving the handling capacity and efficiency on containers. However, it is unlikely to expand the handling capacity continuously with relatively scarce terminal resources such as the storage space for stacking containers. In many Asian sea ports, land is a particularly limited resource that pushes operators to utilize the space in the yard, aisle and berth well for facilitating the handling process on containers. Operational studies on these subjects, thus, exhibit much usefulness in real practice on dealing with heavy workload of containers with limited resources in these terminals.

1.2 Container Terminal Operations

Container terminals are basically acting as hubs connecting shipping lines with other container transportation networks. Although container terminals could differ in some features or functions, they principally contain three functional areas including the quay, yard and gateway. Figure 1.1 shows an overview of the three major areas in Shanghai Waigaoqiao Container Terminal.

Figure 1.1 Overview of Shanghai Waigaoqiao international terminal

There are three operational areas in a container terminal. Figure 1.2 shows the typical layout of a container terminal. The quay is the operational area in a container terminal for berthing ships and quay cranes in container berths load/unload containers onto/from ships. The storage yard is an area for temporary storage of import (unloading) and export (loading) containers for further handling by yard cranes for trucks. The yard is also a functional area for storing empty containers or performing value-added logistic services on containers. In many container terminals, yard area is a physical extension of the quay and this kind of layout can reduce the transportation time of containers between berths and the yard. Gate is another major interface area with the outside world in a container terminal. It links the terminal with the outside road transportation networks and also performs documentation control for containers.

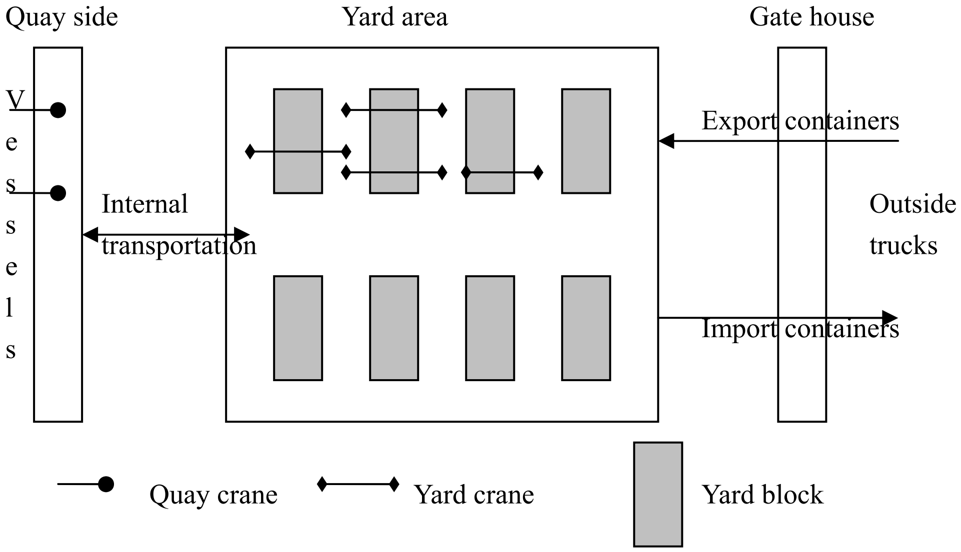


Figure 1.2 Layout of a container terminal

Container operations in a terminal can be analyzed based on the types of major handling processes. Loading and unloading are the two most common processes in a terminal, in which containers are sequentially handled by major terminal resources according to predetermined plans. Upon a berthing request submitted by a shipping line, a terminal has to decide on the berth to be allocated to the vessel. At the allocated berth, quay cranes are employed and positioned for handling containers for the vessel. Once the vessel is berthed, the unloading process is normally performed first to unload import containers from the vessel. Figure 1.3 shows the flow of the typical unloading process. Containers are picked up from the vessel by the spreaders of quay cranes and transferred to internal vehicles which transport the containers from the berth to predetermined locations in yards for temporary storage. The internal delivery of containers may be performed by various equipment such as straddles carriers, trucks and automated guided vehicles (AGVs) while the transfer operations in yards for stacking containers are mainly performed by yard cranes. In a terminal, quay cranes, internal vehicles and yard cranes are the most important equipment for container handling.

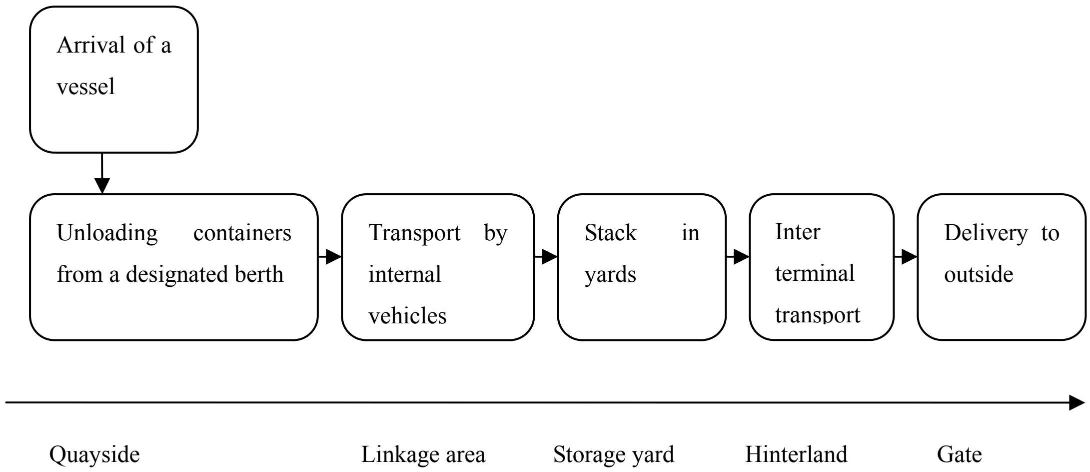


Figure 1.3 Flow of the unloading process

It is worthwhile to note that the flow of the loading process of export containers is basically in reverse order of that of the unloading process. In a loading operation, export containers stored in the yard are mounted onto trucks by yard cranes and they are then off-loaded at the quay and loaded onto a vessel by quay cranes.

Managing the two container-handling processes mentioned above requires excellent planning and control of complicated terminal operations. Each process workflow, in fact, involves a number of concurrent operations in different work areas which need effective scheduling and on-site co-ordinations. Due to unique characteristics of the container handling operations by different terminal resources, different strategies should be employed for different container handling operations in different processes. It is also noted from the terminal layout and operations that storage in a container yard is an important buffer linking various container handling operations.

A container yard is basically a functional area for temporarily storing import and export containers. It is normally divided into a number of yard blocks. A typical block contains 40 to 60 slots that each slot may have 6 to 8 rows. Figure 1.4 shows a layout of a typical yard block. It is worthwhile to note that the term “bay” used in some references cited in this thesis has the same meaning of “slot”.

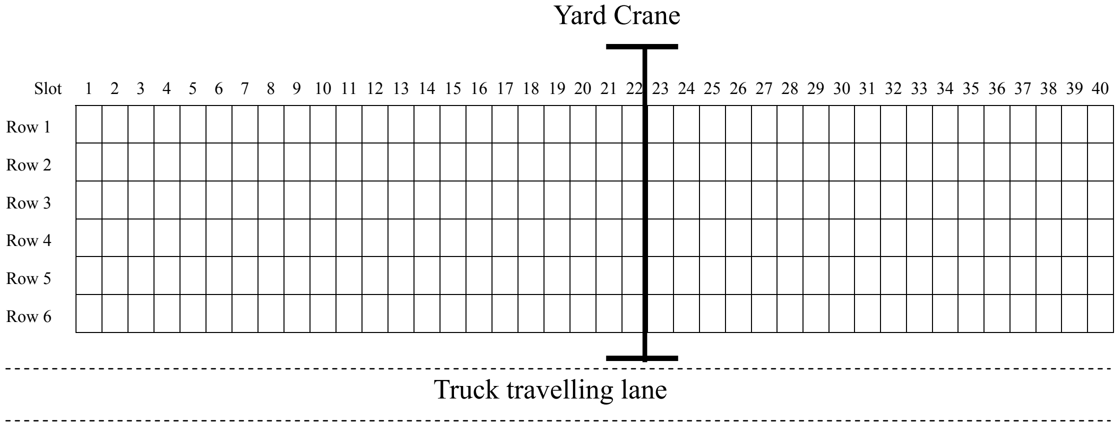


Figure 1.4 Layout of a typical yard block

A small rectangle in the figure denotes a ground slot and it has a stacking capacity of 5 to 7 tiers of standardized size containers (Twenty-foot Equivalent Unit) in a yard block. The storage location in a yard block can be specified in terms of slot number, row number and tier number. The most commonly employed yard cranes in terminals are rail mounted gantry cranes (RMGs) and rubber tired gantries (RTGs). In order to improve crane handling efficiency, operators would infrequently move yard cranes. It is a common practice that yard storage space is allocated in terms of slots since a crane often stays at a slot and containers are picked up by the spreaders of the crane and transferred to internal vehicles which transport the containers between berths and yards. The approach adopted in the thesis follows the practice used in the terminal industry. Many modern yard cranes nowadays can perform high density stacking to increase yard storage capacity. However, the high stacking may result in inefficient yard operations due to frequent re-handling of containers, especially when the stacking sequence of containers deviates from the desired sequence.

1.3 Planning Container Terminals

It has been discussed in the previous section that container operations in a terminal involve various handling equipment which need to be scheduled effectively to smooth the container flow. To co-ordinate huge number of activities in handling containers, excellent planning and real-time control are indispensable. Hierarchical planning approach is often adopted to manage this kind of complicated operations. This approach advocates the decomposition of a complex planning and control problem into a number of levels. A three-level hierarchical framework for classifying decision making problems involved in the planning and control of a container terminal has been proposed by a number of researchers (Vis and de Koster, 2003, Günther and Kim, 2006). Planning problems in the first level usually determine the strategic factors such as the terminal layout, hardware and software used in a terminal covering long-term planning horizon. The planning problems at the strategic level are normally tackled in the initial stage of planning for a new terminal and the decisions are made based on operating costs and technical feasibility. Once the strategic planning problems are solved, terminal planners can make tactical or operational planning for operations covering medium-term planning horizon. In the last level, the control for real-time operations is usually considered.

The daily operation in a container terminal normally covers a series of planning problems. Before a vessel berths at a terminal, the export containers for the vessel will arrive at the terminal a few days in advance. To smooth the flow of containers from yard blocks to the berth allocated to the vessel, the problem of determining the optimal storage location for these export containers needs to be solved before the actual vessel arrival time. Without knowing the berth allocated, it is difficult to find the optimal storage locations. Thus, most terminals adopt a policy called home-berth policy. Under this policy, each incoming vessel of regular services is assigned to berth within a predetermined stretch of berth. In general, the stretch of berth is a few times longer than the vessel length. The problem of determining the home berth for a service is normally done on a quarterly basis, roughly the same frequency that shipping lines make changes to their schedules of vessel services. For vessel services with cyclic calling pattern, a yard planning policy similar to home-berth policy is used to determine the yard blocks for storing the incoming export containers of the vessel services. Thus, there are "home yard blocks" for storing these export containers and the "home yard blocks" are specified in a yard template. The problem of designing a yard template is a complicated planning problem. However, most terminal planners still use their own experience and judgment to manually design the yard template.

The operational planning of terminal operations can be best described following the sequence of major terminal activities. As mentioned previously, planning of terminal activities is triggered by vessel arrivals. Upon the berthing request of an incoming vessel, a terminal needs to determine the berth to be allocated to the vessel. In order to optimize the terminal performance, terminal operators need to find an effective berth allocation and this often requires a complex tradeoff between the services offered to different vessels.

The successive planning problems for the vessel are to determine the stowage plan which specifies the exact storage location of the export container to be loaded onto the vessel, to find the sequence for loading containers onto and unloading containers from the vessel, and to compute the number of quay cranes to be deployed to the vessel and assign different parts of loading/unloading sequence to the quay cranes.

The planning problems for the yard operations are to determine the exact storage location for each loading/unloading container, to schedule yard cranes to handle containers for internal vehicles or external trucks, and to schedule internal vehicles to transport containers between quayside and the yard. Figure 1.5 shows the relationship of these major operational planning problems.

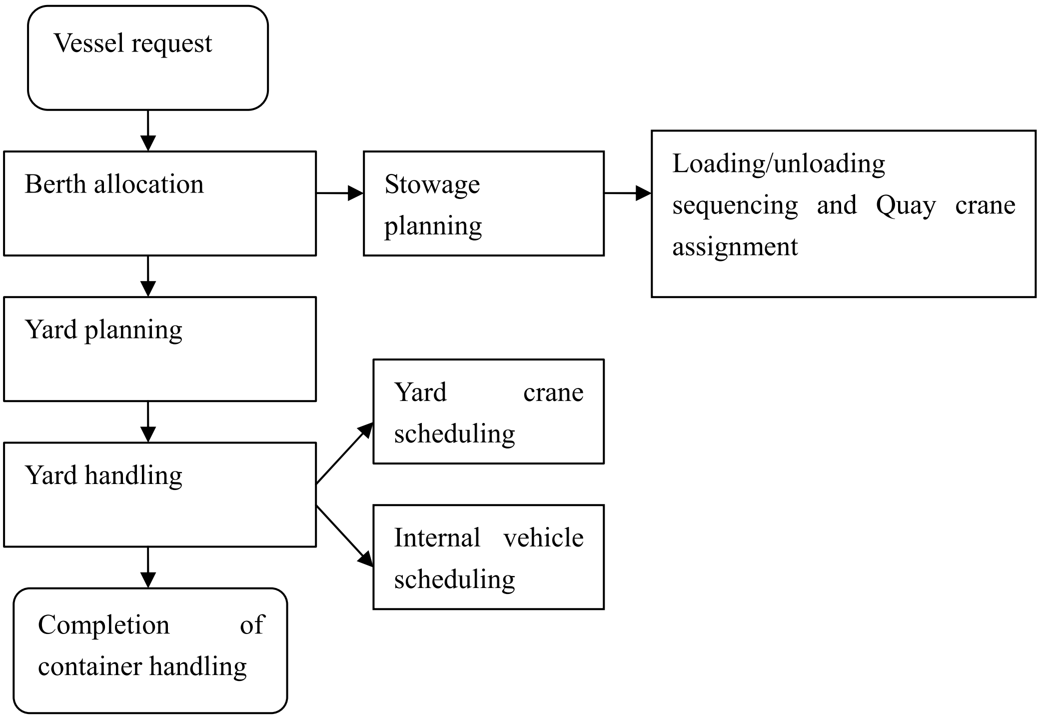


Figure 1.5 Relationship of major operational planning problems in a terminal

Although each container handling workflow involves various decision making problems, the quay side productivity is, normally, a basic and primary measure for evaluating the performance of a container terminal. As a consequence, quay side planning, quay crane operations and other synchronized activities for handling containers attract most of previous research works. However, the handling efficiency at the quay depends on a number of operational activities in practice as listed before. It is widely recognized that the loading or unloading process for exports and imports cannot be analyzed effectively by studying only one or a few research subjects separately (Günther and Kim, 2006). The landside operations, in fact, serve as important nodes in the entire handling process in that the operational efficiency in yard may be critical in a terminal.

In land-scarce container terminals in Asia, the productivity of landside operations for expediting other handling operations is even more critical to the overall performance of a terminal. Thus, the landside is often regarded as the heart of the terminal operations. An efficient yard operation requires well-planned container storage locations and various factors should be considered in the preparation of such a plan, including utilization of the storage space, travelling distances between storage locations and berths, facilitation of different workflows and the efficiency of retrieving containers from any specific area. Indeed, planning of container storage locations has, recently, become an important focus for research on terminal operations.

Yard is recognized for fulfilling a basic function in a container terminal - temporary storage of containers (Hee and Wijbrands, 1988). However, recent studies indicate that the effect of storage yard as a buffer for controlling the entire workflow is more recognized by terminal planners. In storing a container in yard, the selection of stacking location in terms of tier, slot and block is an important factor for smoothing operations especially in loading services when the productivity depends much on the accessibility of containers and the transport distance from the stacking point to the berth. In land-scarce container terminals, the effect of buffering in the stacking procedure is more prominent. In view of the limited storage space, multi-level stacking of containers is very common in many sea ports. Containers in some Asian sea ports may even be stacked up to 8 tiers to maximize the spatial utilization of the yard. Figure 1.6 shows a typical stacking pattern of containers in a land-scarce container terminal in Hong Kong. Consequences of the high density stacking policy in yard are reshuffles, re-handles and possible congestions which need more exquisite planning on container stacking for maintaining the space utilization as well as the handling efficiency. Yard planning has indeed been a key issue in operating a container terminal more productively.

Figure 1.6 An overview of storage yard in Hong Kong Modern Container Terminal

Source: www.modernterminals.com

To facilitate the work for different container flows, yard blocks are normally partitioned into various functional areas. Export and import containers are stored into different functional areas with export blocks or import blocks only. More specialized functional area may be divided such as an area specified for storing empty containers in Shanghai Waigaoqiao Container Terminal. Planning techniques and operational methods are differentiated in different yard areas. Normally, import containers are stored in blocks without further differentiations as some containers could be directly delivered to customers after they are unloaded from vessels, whereas export containers are usually grouped by their attributes, vessel and destination and are specified certain storage places in blocks. Storing and loading export containers involves more concurrent or successive operations on the containers which need precise and detailed planning on a storage template specifying home vessels and destinations for expediting the handling jobs. In order to achieve the most effective operations in yard area, this research focuses on designing suitable strategies on storing and handling export containers. Furthermore, the storage strategies in this yard planning are designed and deployed for static and dynamic situations, respectively.

Note that the yard planning for both scenarios is on the basis of a common practice that terminal planners are accessible for the schedules of arrival of vessels and export containers in advance since many vessel services are deployed by big shipping lines with regular running patterns. Export containers are normally arrived at the terminal several days in advance for the calling of the vessel such that an average dwelling time of 3-5 days is needed in the yard storage plan. The static scenario assumes a regular vessel calling pattern with deterministic schedules, whereas the dynamic scenario handles another practice that ships or containers do not arrive at the port on schedule in view of uncertain circumstances. Different stacking methods for containers are suggested for respective planning problems. When the information on the arrival of containers and ships is sufficient and accurate, the yard planning for the first scenario implements a container stacking strategy in clusters, i.e. a contiguous number of slots in blocks for export containers is reserved for each one vessel. By this method, containers to be loaded on the same vessel will be put close to each other with further classifications to facilitate the handling process. To avoid loading containers onto a wrong service, yard planners do not allow different clusters to share the same slot on the same day. Thus, once a slot is assigned to a particular service, it cannot be used to store the containers of other services with loading date later than the loading date of that service. On the other hand, for dealing with fluctuated vessel calling patterns as supposed in the second research problem, Scattered Stacking Strategy (Steenken et al., 2004) is introduced that yard area is assigned to a berthing place instead of to a specific ship first. When there are containers arrived to the terminal, operators find out the berthing place assigned to its vessel and allocate an appropriate stacking place within the storage area for the berth.

In consideration of the principal objective for increasing the turnover in a container terminal, this research on yard planning decides precise storage plans for export containers with further efficient loading operations expected and avoiding possible bottlenecks in yard blocks in collaborative handling on containers by yard cranes, trucks and other devices.

1.4 Research objectives

It is shown from the above background introduction that yard planning is indeed a resource allocation problem in terminals which normally aims at improving the yard productivity by implementing effective stacking methods for containers. Yard optimizations can be conducted from various aspects or hierarchies such as deciding capacity of container handling equipment, scheduling yard cranes and vehicles, developing containers stacking policies and optimizing storage utilization in yard blocks. Above all the operative problems, planning the workload distribution in yard blocks is principally regarded high priority in a hierarchical analysis for yard optimizations. It is a common practice and has been widely used in some research work (Ku et al., 2010, Lee et al., 2006) that the workload in yards is measured by slot. From the point of view of hierarchical analysis for the yard optimizations, this yard planning of balancing workload in terms of slot is a basis for many other successive planning works such as monitoring the container stacking at different tiers in a slot, optimizing the equipment allocation, yard-to-ship transport and work collaboration. To achieve the ultimate objective in a container terminal - optimizing using of all terminal resources for maximizing the productivity of handling containers for vessel services, this research for yard planning is carried out for completing the following objectives.

- First, this research aims at improving the landside productivity by optimizing allocation of workload among yard blocks in equilibrium to reduce possible congestions of vehicles and yard cranes during the loading process. It is worth noting in the thesis that the workload of any one yard block is measured by the number of slots storing containers for vessels which will be loaded in the block when the vessels call the terminal.

- Second, by effective planning methods, the limited yard space can be efficiently and economically allocated to vessel services for storing export containers under various conditions in land-scarce container terminals. More specifically, the research sets up a prime objective of balancing the workload among yard block and completes further yard template design within each yard block with the exact stacking slot determined for each incoming export container.

- Third, different stacking strategies for containers are discussed to designate appropriate methods for allocating yard storage space to vessel services under different circumstances and facilitating container handling in the whole work process.

- Last, a general methodology could be generated from the research for conducting the yard planning with optimization models and efficient algorithms to approach optimal or good feasible solutions for both theoretical studies and realistic applications.

1.5 Summary of Contributions

It has been discussed in Section 1.3 that many yard operations have exhibited critical impact on the efficiency of a terminal. In practice, various operational problems in yard optimizations need to be studied. On the other hand, there exist limited research works on effective terminal yard planning in the literature that produce no mature framework for conducting this work. To fill the gap, this research defines concepts of a yard template, outlines boundaries of the planning problem and develops methods for approaching good solutions for the storage yard planning. Specifically, this research on yard storage planning in controlling the whole work process on export containers exhibits the following characteristics that differentiate it from previous works.

1) From the point of view of hierarchical analysis and optimizing global yard operations, the research work constructs the prime objective for achieving the workload equilibrium among yard blocks. Since the research is conducted for land-scarce container terminals which have increasing pressure on storing more containers in limited yard area, the proposed objective is particularly effective in improving the overall productivity by reducing the possibility of traffic block and interference in handling containers.

2) It proposes a method for conducting the precise space allocation, i.e. determining the exact position of slots for incoming export containers, based on the optimization objective proposed in the yard. In other words, efficient yard template design is conducted to help terminal operators making detailed storage allocation and stacking plan for these containers.

3) The yard planning problem is solved by a two-phase solution method with hierarchical breakdowns of the original formulations. The sub-problem in each phase, with respective objective of achieving workload equilibrium and efficient use of storage space in the yard, is solved by efficient algorithms in short computational time. The solutions may be more feasible for practical operations in land-scarce terminals.

4) The methodology of solving the yard planning problem under various operational conditions is useful for improving the landside productivity as well as coordinating all land related operations. It explores possible applications and extension of the optimization models to an integrated decision making process with berth allocations,

quay crane operations, internal transport and yard operations.

In brief, the research optimizes the storage space design for export containers to strike a workload balance among yard blocks for efficient handling under various and changeable conditions.

1.6 Framework of the thesis

The framework of the thesis is shown in Figure 1.7. Following the introduction of the research work, Chapter 2 reviews related works on terminal planning and operation from both the quayside and the landside to gain some insight from previous scheduling or planning methods for this yard planning. The main focus of the literature review is put on various optimization methods for different yard planning problems in a container terminal. Research works closely related to landside operations are also discussed in separate sections, which include berth allocations, terminal device planning and the process simulation. With respect to the literature, methods for effective container yard planning are still scarce. To supply more effective ideas for this study, this chapter discusses some useful methods from other research subjects such as flexible manufacturing systems and bin packing as well.

The research for yard planning is carried out for vessel services with a cyclical vessel calling pattern and non-cyclical vessel calling patterns, respectively. Thus, the main body of the thesis is organized into three chapters. As shown in Figure 1.7, Chapter 3 and 4 discuss formulations and solution methods for the yard planning problem under a deterministic situation. Chapter 5 discusses the problem with irregular vessel calling patterns, thereafter.

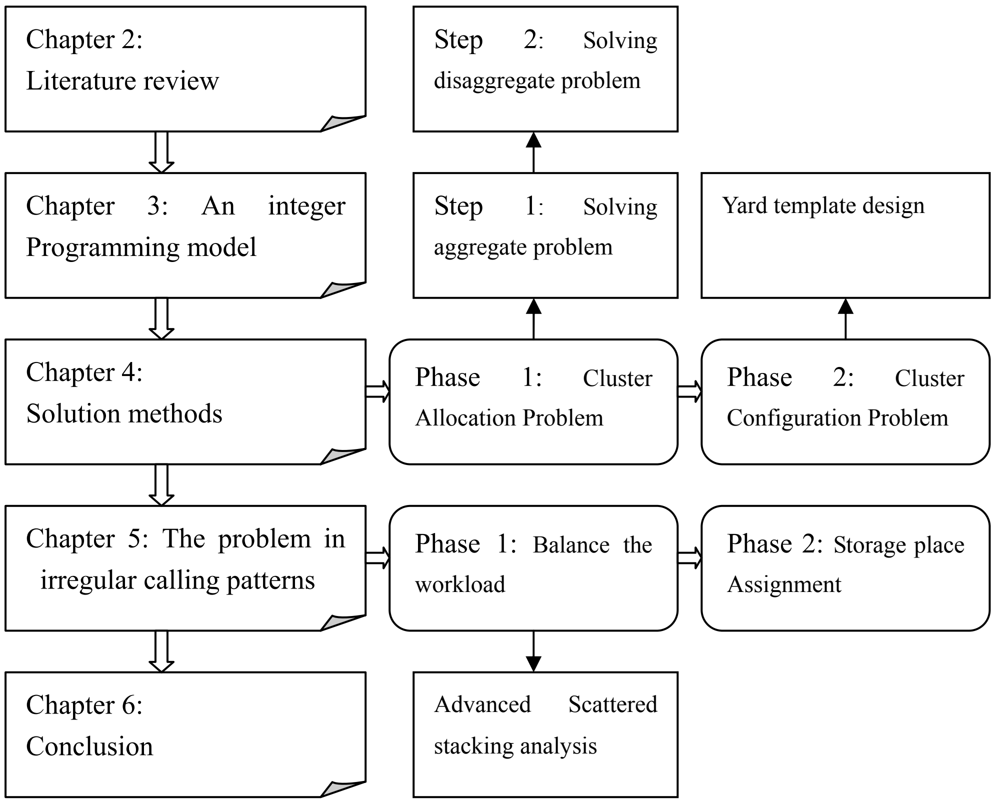


Figure 1.7 The framework of the thesis

In Chapter 3, the yard planning problem for vessel services in a regular calling pattern is defined. A 0-1 integer programming model, with constraints generalized from real practice, is proposed to determine the exact position of slots in yard blocks for storing export containers. The objective of the model is to achieve a minimum sum of workload imbalances to expedite container handling in the yard as discussed in the previous sections. Computational experiments are then carried out for solving this model by a commercial software. However, the result is not satisfactory as it is time-consuming for even getting feasible solutions for a problem of common size.

To efficiently solve the problem, a two-phase solution method is introduced and discussed in Chapter 4. In Phase 1, the integer programming model of determining the exact position for export containers is simplified into a model of finding the number of slots associated with handling jobs in each block achieving the workload balance. The result of the alternative integer program is used for constructing the yard template within each yard block in the successive planning problem of Phase 2. It is noted from the simplified model of Phase 1 that its objective is similar to that of the original planning problem and hence optimal solutions are expected in solving the problem of the phase. Further decomposition on formulations of Phase 1 is then suggested. Vessel services are, first, aggregated by their container loading dates. Solutions are found for these services groups and then are disaggregated to vessels afterwards. Approximation methods for transportation models are adopted in the solution procedure. When solutions for slots allocation are determined, this chapter then proceeds to the template design problem in Phase 2. However, solving the problem in Phase 2 is still unlikely to be completed by current commercial software with branch and bound algorithms. Results of computational experiments reveal that it may take even longer time solving the problem with CPLEX. An effective template designed heuristic is, thus, critical for finding feasible solutions and then to construct a solution for the original yard planning problem.

On the basis of the solution methods for the yard planning problem with a cyclic vessel calling pattern, this research proceeds to discuss the planning problem for irregular vessel services. Chapter 5, then, extends the model to the yard planning problem with flexible schedules of arrival of vessels and containers and applies a similar two-phase solving idea to the respective solution procedure. Due to different operational situations, Scattered Stacking Strategy for containers is introduced in the yard planning. It is quite straightforward for designing the yard template based on this container stacking strategy and thus the main focus of the research work in Chapter 5 could be put on methods for efficient handling from berths to yards.

Last, Chapter 6 summarizes the research work for yard planning in this thesis and presents ideas for further research works in yard.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Effective operations of container terminals depend on a series of handling activities from the quay side to the storage yard. Normally, the overall performance of a container terminal is measured by the throughput over a period of time. Hence, improving the terminal productivity, especially the quay side productivity, is always regarded as the core in optimizing terminal operations. However, it is a common practice in terminal operations that improving the daily terminal productivity often trades off some satisfaction of ship owners and is constrained by many resource limits. Since the construction and maintenance of port hardware and equipment are pretty expensive, there is a trade-off between the investment on terminal resources and the level of service offered to container vessels. Resource planning in container terminals is, then, regarded as an essential way for achieving various objectives which may sometimes be in conflicts. As early as 1990s, Daganzo (1990) already focused on the amount of investment in a container terminal when designing methods for operations. According to his idea, operational models should properly allocate the terminal resources in different handling functions. Specifically, berths and storage space in yard are noted as two major resources in a container terminal. Analyzing the decision making processes in these two specific areas are essentially important especially when some objectives of the optimizations are in conflicts (Legato and Mazza, 2001).

Berths are the most important resource which directly constrains the efficiency of handling containers for customers, i.e. vessels. Minimizing the ship waiting time at the terminal is obviously an important criterion from the terminal customers' perspective. However, cost-effective management of container terminals entails an efficient utilization of berths for all incoming vessels which may occasionally sacrifice the service level for an individual vessel. Berth allocation, which needs to consider the requirements of all stakeholders, appears to be a challenging job for terminal planners. To reflect the varying perspectives for operations, the total costs for waiting and handling at berths were usually set as the main objective in many previous research works (Imai et al., 2001, Hansen et al., 2008). In some research works (Imai et al., 1997, Li et al., 1998), two research objectives - the minimization of the service time and the minimization of the dissatisfaction level of ships or the make span of the ship berthing schedule were considered.

Planning for the yard space has recently become another focus with a similar dilemma that operators must tackle the steadily increasing space requirement for storing containers with limited space of yard area in many land-scarce container terminals. Yard planning, especially the storage space allocation in yards, needs to be explicitly considered and studied. The main process of operations in a container terminal can be classified into several work flows among which improving the handling efficiency is still regarded as the primary objective in yard optimizations (Steenken et al., 2004). To achieve the objective, there are a few researchers who have worked on developing effective methods for improving the handling process in yard. Zhang (2000) proposed an idea of balancing the total workload in each yard block for import and export containers. The author generalized the ideas for dispatching resources in a terminal including the storage space in yard. However, as far as the research works conducted for operations optimization in a terminal, methods for yard planning and yard template design are still scarce.

It is noted that planning methods and decision making methodologies can be widely applied in operations in container terminals from strategic planning and construction (Friedman, 1992, Cullinane et al., 2004, Bao and Luo, 2005, Anderson et al., 2008) to operational tactics and decision making (Dekker et al., 2006, Liu et al., 2006, Dai et al., 2008). These methods and methodologies are important inputs to the major research work on yard planning and terminal operations reported in the literature.

This chapter focuses on reviewing research work related to quayside operations and landside operations. In Section 2.2, methods on berth allocation, berth handling and other related quayside operation optimization problems are discussed. The research on yard operations such as resource planning and equipment dispatching and scheduling are summarized Section 2.3. Section 2.4 reviews studies done in other research areas which are relevant to the research presented in this thesis.

2.2 Optimization problems at the quayside

Since the quayside productivity is generally considered to be essential to port operators, various research studies have been conducted on improving operations at the quayside with different objectives and constraints. The main criteria for evaluating quay operations include:

(1) Minimizing the ship waiting time and operational cost for ship handling,
(2) Optimizing utilization of important resources like berths and quay cranes in the quay,
(3) Minimizing the travelling distance of containers from the yard area to quay.

The optimization problems studied can be classified into 3 major types: berth allocation, berth handling and planning equipment to support quayside operations.

2.2.1 Berth allocation

There are two major types of berth allocation problems, namely, Berth Allocation Problems for Discrete locations (BAPD) and Berth Allocation Problems for Continuous locations (BAPC).

According to the construction and operational strategies, the berths can be regarded as discrete or continuous resources for vessel handling works. In discrete berth systems, the quay is partitioned into a number of berthing places for incoming vessels and each handling job can be completed with the determined berth and equipment. Many research studies (Lai and Shih, 1992, Brown et al., 1994, Imai et al., 1997, Imai et al., 2001, Imai et al., 2003, Cordeau et al., 2005, Hansen et al., 2008) have treated berths as discrete entities. They proposed different algorithms for optimizing the quay utilization and minimizing the total cost or service time for all incoming vessels in a period of time.

In the early study of Imai et al. (1997), the concept of BAPD was adopted for studying berth allocation in commercial ports. The berth allocation problem considered an objective of minimizing both vessel service time and ship dissatisfaction, i.e. the service order of the ship. It was formulated as a three dimensional assignment problem which was then reduced to a two-dimensional assignment problem. With some idealistic assumptions, the problem was easily solved by the Hungarian method (Papadimitriou and Steiglitz, 1982). The problem was extended to a dynamic berth assignment for public berth systems in the study of Imai et al. (2001). As additional constraints were added to the previously proposed formulations, the modified problem could not be solved in polynomial-bounded time. A sub-gradient solution method was introduced that solutions were found within short solving time for some numerical examples generalized from real applications. Further studies were made on the multi-user container terminals by Imai et al. (2003) that priorities for vessel were considered. Since the problem was difficult to solve with a non-linear formulation, a genetic algorithm was introduced to find solutions for scheduling berth services. Based on the previous works and research methods, Hansen et al. (2008) proposed some variable neighborhood search algorithms for solving the BAPD (Berth Allocation Problems for Discrete locations). They run the program on both the neighborhood search algorithms and a genetic search algorithm. Results were compared which showed that an improvement of cost saving in berth operations can be achieved with their methods.

Although discrete allocation of the quay is a common practice in berth planning, dynamic allocation of the berthing place for a vessel at a long quay is another useful option implemented in some busy container ports. Since the berthing plan may always be updated for a number of incoming vessels to be handled simultaneously, to improve the quay productivity, specific and same quay locations are replaced by a dynamic allocation of the continuous quay for vessels. In the work of Imai et al. (2005), a model was built to solve berth allocation problems with continuous locations in a multi-user container terminal. With more flexible planning strategies, the idea of allocating berths to vessel services as continuous locations shown more efficiency in berth operations. The problem was solved by an algorithm in two stages that the number of partitioned berths was identified in the first stage and then the ships were relocated in a scheduling space. More research works on continuous allocations of berth locations or multiple using of one berth were given in the studies of Park and Kim (2002), Kim and Moon (2003) and Guan and Cheung (2004).

The same BAPC was studied by Park and Kim (2002) and Kim and Moon (2003) to minimize the costs of ship delay and the costs of ship handling when ships were moored at non-optimal locations in ports. The former introduced a simulated annealing method and the latter proposed a sub-gradient optimization method for solutions. Their works were more useful when the costs setting in the objective function was associated with quay locations for the ships. In the study of Guan and Cheung (2004), the objective of the BAPC was minimizing the total weighted completion time of ship services.

The berth allocation problems also differed in the assumption of vessel arrival times. Given a determined set of ships with known vessel arrival times, the problem was studied as static berth planning that all the vessels arrived the port at the beginning of the planning horizon (Imai et al., 1997). Since the assumption of static berth planning is unrealistic for many cases, Imai et al. (2001, 2003, 2005, 2006, 2007) extended the model on dynamic berth allocation problems which considered the dynamic ship arriving when other works were in progress.

Most containers berths are owned and operated privately by container terminal operators, for example local operators like Hong Kong International Terminals (HIT) and Modern Terminals Limited (MTL). Unlike the private berth system which is widely used in sea ports, the public berth system emphasize the idea of concentrating handling jobs to some public berths to improve the cost-effectiveness of the berths. Nishimura et al. (2001) studied the public berth systems in several Japan sea ports. A Lagrangian relaxation-based heuristic was proposed to find effective solutions for allocating berths of a public berth system. The same berth system was studied by Imai et al. (2001). Although the public berth was unlikely applicable to most sea ports, their research studies still provide some valuable insights for berth future studies on berth allocation problem.

In a terminal, the waiting time of vessels is normally considered as a key performance indicator in berth allocation. Indeed, some research studies considered the case that arriving vessels cannot afford to wait. Brown et al. (1997) studied the case of berthing submarines where parallel mooring and shifting of submarines were permitted. To handle the special case of berthing submarines, an integer programming model was developed with an objective of maximizing of the benefits of berthing. To reduce the number of requests for revising an approved berthing plan due to early or late arrival of submarines, the authors introduced some persistence incentives in the optimization model that decreased the uncertainty of management effectively. Due to the unique characteristics of berth planning in naval ports, their work is unlikely to be applicable to berth allocation in commercial ports. Lim (1998) considered the objective of minimizing the total berth length for a given set of vessels. The berth allocation problem was transformed into a graph and for a given number of cranes deployed to each vessel, the problem was reduced to a special case of the stock-cutting problem. Due to the NP-hard nature of the stock-cutting problem, an efficient heuristic was developed to find good feasible solutions for the network.

In the literature, some research studies considered solving the problem of berth allocation and quay crane scheduling in multi-user container terminals simultaneously (Park and Kim, 2003, Imai et al., 2008, Giallombardo et al., 2008), whereas the combined problem is often computationally intractable. More commonly, the quay crane scheduling and other important berth handling problems are studied respectively.

2.2.2 Berth handling

It is noted that the efficiency of berth handling equipment is critical for terminal operations, planning and scheduling for the equipment especially the quay cranes has become a major research topic in terminal planning and control. It is well known that upgrading equipment and their associated planning control software in a terminal are very costly. Many terminal operators often rely on more effective research methods for planning or operating the resources to improve their operational efficiency. Quay crane operations are regarded in the research literature as the major research area for improving quay side productivity. Two decades ago, Daganzo (1989) studied the quay crane scheduling problem. His work considered the simple static case that a crane can work only on one hold at a time. With the objective of minimizing the aggregate cost of ship delay, an exact optimization method and a heuristic were developed for solving small size problems.

Park and Kim (2002, 2003) studied the scheduling problem of berth resources. They decomposed the problem into two phases. The Phase 1 problem was to determine the berthing time and position for each arriving vessel while Phase 2 determined the crane dispatching and scheduling based on the solutions of Phase 1. Park and Kim (2002) proposed a sub-gradient optimization method to find a near optimal solution for Phase 1 problem whereas Park and Kim (2003) developed a dynamic programming solution method for solving the Phase 2 problem. It is noted that in the former study, a static ship operation time was assumed and the performance of the proposed algorithms was evaluated by the solutions obtained by the commercial software LINDO. In the latter study, the researchers used the solutions obtained by a complete enumeration approach and a heuristic to benchmark solution procedure. The integrated problem of crane scheduling and berth allocation was also studied by Lim (1998). In his study, the complexity of the berth scheduling problem was analyzed and it was shown to be a special case of the stock-cutting problem.

Kim and Park (2004) developed a mix-integer programming model for the quay crane scheduling problem. They classified the loading and unloading operations at one bay into multiple tasks. Since the problem was unlikely to be solved in a reasonable amount of time, they proposed an effective method to reduce the size of the solution set. A search algorithm with a greedy randomized adaptive search procedure was developed for solving the problem. The performance of the algorithm was evaluated by the solution obtained from a branch and bound method for test problems with different sizes.

Liang et al. (2009) studied the quay crane assignment problem together with the planning problem for determining berth positions and timing in a Shanghai container terminal. Their objective was to minimize the sum of handling time, waiting time and delay time for vessels. A genetic-algorithm based heuristic was developed for solving the problem.

Ng and Mak (2006) focused on quay crane scheduling and proposed methods for scheduling the movement of quay cranes during their entire handling operation. The problem of scheduling quay cranes working on a common linear rail was analyzed and the objective was to determine the work schedule of each quay crane to minimize the ship's service time at port. In their study, the multi-crane scheduling problem was reduced to several sub-problems with multiple partitioned work zones for a ship. A simple rule was proposed to find the optimal solution of each sub-problem and an effective heuristic was developed on the basis of the optimal solution of sub-problems to determine an effective solution for the quay crane scheduling problem.

Most of the aforementioned research works on quay crane scheduling and handling considered the operating strategy of single cycle which is commonly adopted in container terminals. Under the strategy, quay crane will only handle a discharging or loading container in a handling cycle of moving back and forth once between a vessel and the quay side. The problem in this case is often formulated as integer programming models and solved by genetic algorithms or other meta-heuristics. In recently years, a new operating strategy, double cycle, has been proposed. Under this strategy, a quay crane will handle two containers, a discharging container and a loading container, in a handling cycle. With this strategy, a vacant crane move is converted into a productive one and the quay crane productivity is greatly increased. A number of researchers have studied the problem of planning quay crane operation under the strategy of double cycle (Goodchild and Daganzo, 2005, 2007). These research studies analyzed the performance of two quay crane operating strategies and discussed the necessary operational changes in loading plan and sequence, transportation and storage plan to support the new handling strategy. The studies concluded that implementing the double cycle strategy required a higher level of planning skills and a much more complex organization of activities in order to achieve a more efficient handling.

2.2.3 Other related quayside operation optimization problems

It is noted that quay crane productivity does not only depend on the quay crane efficiency, but also on the container transportation between quayside and landside to support quay cranes' discharging and loading activities. Steenken et al (2004) pointed out that disturbance and congestions in transporting containers as the major reason for reducing the crane productivity from 50-60 boxes per hour to 20-30 boxes per hour in many sea ports. Thus, transportation optimization to support the quayside operations has attracted the attention of many researchers. In the research study by Bish (2003), the quay crane scheduling and vehicle dispatching was modeled as a scheduling problem with multiple constrained resources with the objective of achieving the minimum service time for a set of ships. The problem was transformed to a transshipment problem and the problem was then solved by a transshipment method-based heuristic.

The main equipment for transporting containers internally in terminals include trucks, automatic guided vehicles (AGVs) and straddle carries. AGV is a popular transportation equipment in an automated container terminal operated in high-labour cost countries. Thus, the scheduling and routing of AGV in an automated container terminal has attracted considerable attention of researchers.

Hoshino et al. (2006) studied some major problems, like agent cooperation, container storage scheduling and container transportation planning, associated with a AGV transportation system. The study developed several management models to determine the transportation and storage order to AGVs and optimize the scheduling with consideration of the sequence of orders. Random container storage and container storage scheduling were analyzed to evaluate the cost effectiveness of the AGV system. Later, Hoshino et al. (2007) improved the design of the methodology developed in their previous study to build a more robust and reliable transportation system dealing with dynamic transportation demands. A case study was used to study the effectiveness of the methodology for detecting and resolving bottlenecks in the system.

Positioning AGVs in a loop layout is another major focus in planning and control of a AGV system in an automated seaport container terminal in a number of studies (Egbelu, 1993, Kim, 1995). The common problem objective is to improve the response time of vehicles. In the study of Egbelu (1993), the problem of determining home positions for idle AGVs was addressed. Several single loop home selection models were built and a few solution methodologies were proposed to select the home position with minimal unproductive travel time of vehicles. Kim (1995) further analyzed the problem with two strategies- static positioning and dynamic positioning. In static positioning strategy, a single-parking location was assigned to all idle vehicles. Dynamic positioning strategy considered the case that a location was immediately allocated to an idle vehicle in need.

The positioning problem for AGVs was further analyzed for both unidirectional and bidirectional flow systems by Gademann and van de Velde (2000). Dynamic programming algorithm was proposed for minimizing any regular function of the response times, i.e. travel time from the nearest waiting location of an idle AGV to a pickup point. Researchers have found that many algorithms reported in the literature for scheduling AGVs are applicable to real world operations. A variety of policies and strategies for scheduling, routing and dispatching AGVs were summarized in the survey articles (Qiu et al., 2002, Le-Anh and De Koster, 2006, Grunow et al., 2007, Stahlbock and Voß, 2008).

It is worth noting that transporting and stacking containers in yard is the other essential part of container handling operations in a terminal yard. Researchers have spent considerable effort on studying the problem of dispatching and scheduling yard vehicles. Kim and Kim (1999b, 1999c) studied the routing problem of straddle carriers for transporting loading containers from a yard to berths to minimize the total travelling distance. Dynamic programming algorithm and efficient beaming search algorithm were developed to determine the optimal route for straddle carriers in yard.

To achieve efficient discharging and loading of containers, one needs to study the ship planning problem which is tightly coupled with yard planning and yard handling. Before the arrival of a vessel, a terminal planner needs to determine the number of quay cranes to be deployed to the vessel, the amount of workload of each quay crane and the stowage plan. As ship planning is closely related to crane productivity and yard configuration, the planning problem was also considered as a major part of a larger planning problem (Imai et al., 2006).

Stowage planning, an important aspect of ship planning, determines the number of containers to be stacked in a vessel, their exact bay positions and the sequence for discharging and loading. In the literature, minimizing the number of container shuffles to be performed in discharging and loading processes and maximizing utilizations of ship space were the two common objectives (Avriel et al., 1998, Avriel et al., 2000). Since the ship stowage plan must consider a number of constraints and requirements such as the stability of a vessel, the weight of discharging and loading containers and the special requirement for storing containers, the stowage planning problem is often very complicated. In the early studies on this problem, the stowage planning was normally analyzed with certain assumptions to simplify constraints. Avriel and Penn (1993) modeled the vessel stowage problem without considering the stability constraints. With the assumption that the number of containers for stowage was known in advance, a 0-1 binary programming model was proposed. Optimal solutions could easily be found with the GAMS optimization software for small and medium scale problems. An effective heuristic based on a transportation matrix derived from the model was developed for solving large scale problem. In other research studies, various heuristics were developed to plan the ship stowage especially for the automated ship stowage. The Tabu search meta-heuristic was proposed to automatically generate solutions for ship container stowage (Wilson and Roach, 1999). The stowage planning was divided into 2 stages. In strategic planning stage, the loading containers were assigned to different stacking blocks within the ship while in the tactical planning stage, the exact stowage position within the blocks for each container was determined. A traditional branch and bound algorithm and advanced Tabu search algorithms were both developed for solving the problem and near-optimal solutions were found.

It is worthwhile noting that many research studies regarded the landside operations as the most critical component of efficient handling operation in a terminal. However, they were not considered essential in improving the port throughput such that yard productivity did not attract enough attention previously. Consequently, the effective planning strategies and methods for yard space and equipment were scarce to provide sufficient supports for a cost-effective port operation especially when more and more yard operations showed important effects on the overall port performance.

2.3 Optimization Problems in yard

Yard area basically serves as a temporary storage buffer between quayside operations and external transportation equipment. Apart from the temporary storage function, it also serves a buffer, together with proper optimization of yard storage, which can smooth the container flow to and from the quayside and thus increasing the productivity of a terminal.

To optimize the yard operations, one needs to consider:

(1) Effective utilization of the land and handling equipment,

(2) Minimization of the unproductive movements of containers in yard,

(3) Optimizing the plan for the yard storage location for containers to shorten the travelling distance and avoid bottlenecks causing serious traffic jams.

2.3.1 Yard handling

Yard handling is an important area of yard optimization. Effective scheduling of container handling equipment in yard is critical to efficient terminal operation. In early studies by Castillo and Daganzo (1993) and Kim and Kim (1998), the optimization of the allocation of required storage space and the handling equipment for either export or import containers under different stacking policies was the main focus. Various mathematical programming models and cost-effective analysis were conducted to determine the optimal handling policy for different situations.

Recently, researchers have focused more on optimizing the container handling equipment in yard. Ng and Mak (2005b) studied the problem of scheduling yard cranes for handling jobs in yard with heavy workload. A branch and bound algorithm was proposed to solve the problem with the objective of minimizing the sum of job waiting times. Results of extensive computational experiments showed that the algorithm could solve the problem optimally for most realistic problems. They also proposed an effective heuristic on scheduling yard cranes for a given set of handling jobs with different ready times in another study (Ng and Mak, 2005a). The performance of the heuristic was evaluated by test problems generated from real-life terminal data. Results showed that effective solutions could be found by the proposed heuristic.

Ng (2005) studied the problem of scheduling multiple yard cranes in a container terminal. Due to sharing of the same travelling lane, inter-crane interference might happen if yard crane schedules were not properly designed. Scheduling yard cranes under such a situation was NP-complete. A dynamic programming-based heuristic was proposed for solving the scheduling problem. To evaluate the heuristic, another algorithm was developed to present effective lower bounds. Results of computational experiments showed that the heuristic could find effective yard crane schedules.

Lee et al. (2007) studied the problem of scheduling two transtainers serving one quay crane and two yard blocks in loading operation of export containers. To minimize the total loading time at the stack area, a mathematical model was proposed and a simulated annealing algorithm was developed for solving the problem. The algorithm was found to be effective in determining good feasible solutions with an average 10.03% deviation from the optimal ones.

Zhang et al (2002) studied the dynamic crane deployment problem. They assumed that the workload of daily yard cranes could be estimated. The study proposed a mixed integer programming model and a lagrangean relaxation solution method to find the optimal times and routes for the total delayed workload in yard. Petering and Murty (2009) analyzed the relations between block length and yard crane deployment. A dynamic discrete event simulation model was built for the analysis. Simulation results showed that a block length between 56 and 72 slots could yield the highest quay crane handling rate. Some researchers (Kim et al., 2003) studied the problem with the objective of improving the service level for external trucks. Assuming that trucks arrived continuously, a simulation study was carried out for evaluating the decision model.

2.3.2 Yard planning

As shown in the ship loading process, optimization of yard operations can greatly smooth the internal container flow between yard blocks and berths and improve the efficiency of terminal operations (Steenken et al., 2001). A common objective for terminal operators is to shorten the vessel service time at the quay so as to increase the productivity of the terminal. It necessitates the smooth flow of containers to and from the quayside. Many handling jobs for containers must be carried out to a pre-determined plan which considers vessel stability, port of discharge and stacking locations in the yards, etc. The efficiency of handling containers at the quayside is greatly dependent on a smoothly-operated workflow for stacking and retrieving containers in yards. Thus, it has been recognized that optimization in yards is critical to the improvement of the efficiency of entire terminal operation (Ku et al. 2010, Lee et al. 2006, Ng et al. 2010). In yard planning, storage space capacity constraints of yard and storage space utilization of yard are also important considerations in some land-scarce sea ports.

As there are two major types of resources in yard - the storage space in yard blocks and container handling equipment such as yard cranes, yard optimization problems can be classified into two types, namely, yard planning and yard equipment allocation and scheduling. As discussed previously, a good yard plan not only achieves a better utilization of storage place, but also facilitates the loading with the optimal stacking of containers. In an early study (Taleb-Ibrahimi et al., 1993), internal buffer areas in yard blocks were suggested to temporarily store containers before they were moved to a 'permanent' stacking location. The 'permanent' location referred to a place that the container would be stacked until being loaded into vessels. The study also showed the usefulness of the storage strategy in improving handling efficiency for a yard with low density of storage and enough space for adaption. However, the strategy requires frequent re-marshaling of containers. Kim and Bae (1998), thus, focused on studying re-marshaling of export containers in ports to minimize the number of re-marshaling moves. The research paper proposed the idea of decomposing the original problem into three sub-problems: the problem of bay matching which matched a bay configuration of a specific bay with the stacking plan in the target yard block, the problem of re-marshaling planning which determined the number of containers for re-marshaling and the problem of task sequencing which minimized the completion time of re-marshaling operations. Dynamic programming models were proposed for solving the first two sub-problems and a transportation model was developed for determining near-optimal solutions for the final sub-problem. However, in view of the high cost of re-marshaling containers in yard and the additional storage space requirement, the problem of finding the exact 'permanent' storage locations, which do not require any re-marshaling, has attracted more attention of both academic researchers and industry practitioners.

To determine the optimal storage location, both the storage space utilization and handling efficiency need to be considered. Since utilization of storage space usually depends on the investment in container handling equipment and the service standard of container handling time, a trade-off analysis between the closely related factors needs to be conducted. It is well known that the stacking density of container yard in Asian terminals is much higher than that in seaports in Europe or North America. Managing unproductive moves and reshuffles of containers in yard operations shows much more importance in Asian sea ports (Kim, 1997, Chen, 1999). In order to correctly arrange the stacking for import containers achieving the minimum number of container re-handles, Kim and Kim (1999a) formulated several mathematical models for analyzing constant, cyclic or dynamic arrived import containers. A basic strategy for stacking these containers was proposed and under this strategy, newly arrived import containers should not be put on top of those which have been in stack. In Kim and Kim (2002), they further extended their previous study to determine the optimal size of storage space and the number of transfer cranes for import containers. They found that costs of operating these containers were the major criteria for selecting the temporary storing place in yards. Two different cases - when costs of port owners were considered only and when costs of port owners and customers were both considered, were studied and a deterministic model and a stochastic model were proposed accordingly. Although their work contributed to improving the handling efficiency in yard for import containers, the results indeed shows limited impact on real life operations when most import containers are delivered to customers directly.

It is noted that the handling efficiency in yard is still a key objective which is subject to the space constraints etc. Several researchers (Zhang et al., 2003) studied the problem of balancing the workloads in yard when making space allocation plans for inbound, outbound and transit containers so as to avoid the possible transportation bottlenecks for container handling jobs. They decomposed the problem into two levels: to determine the workload balance in the first level and to decide the stacking place according to traveling distance in the second level. By adopting a rolling-horizon approach, solutions could be found which significantly improved operations in yard. An important aspect of their research work is that it is the first study proposing the idea of balancing workloads among yard blocks by scheduling equal number of containers to be handled in each yard block. However, such a measurement seems too aggregate to determine the timing of container handling operations for individual vessel loading operations and thus, bottlenecks may still occur even the workload is well balanced with respect to their measurement. In real life yard operations, the timings for handling loading containers for different services are different and the workload should be differentiated by vessel services so that the chance of having bottlenecks can be reduced.

Bazzazi et al. (2009) studied an extension of the above study and developed a genetic algorithm based method for solving the problem. The algorithm was demonstrated to be efficient for an extended storage space allocation problem. Han et al. (2008) considered the yard planning problem in a transshipment hub with heavy workloads. They introduced a high–low workload balancing protocol to reduce the traffic congestion for both loading and discharging activities. Kim and Park (2003) applied a dynamic space allocation method for planning export containers. They classified the handling system into a direct transfer system and an indirect transfer system. They proposed a sub-gradient optimization method and used numerical experiments to evaluate the performance of the method with different values of parameters.

Stacking strategies for containers, which are closely related to storage location planning, are another other focus of the research on yard operations. Dekker et al. (2006) gave a detailed discussion about container stacking policies. They proposed a number of effective methods for stacking container at an automated container terminal. Results of extensive simulation experiments showed that their methods could facilitate container stacking, stowage planning and quay crane handling. The storage location of export containers is an important input to determining a preferred berthing location for an incoming vessel and other major decision making process in a terminal. Moorthy and Teo (2007) studied the berth template design problem – the problem of determining the desired berth location for each vessel with cyclic calling schedule design. They suggested the berth template should be used together with a yard template which specifies the storage location of containers to be loaded into those vessels to improve the performance of a terminal. The research study points out an important direction for planning container yard, namely yard template approach. Indeed, yard planners in bug terminals are using this approach to determine the storage locations of export containers. However, due to the complexity of the yard template design problem, the planners have to rely on their experience, rules and judgment to design the yard template. Their manual process is tedious and time-consuming. Moreover, the myopic nature of their planning methodology often produces yard templates resulting in severe bottlenecks. Therefore, further research studies should be carried out to develop effective and efficient algorithm for designing yard templates.

2.4 Research related to yard operations

To gain some useful insights for developing effective methods for solving yard planning problem, a review of research studies conducted on analyzing systems similar to terminal systems is needed.

It is noted that the operation in a Flexible Manufacturing System (FMS) is similar to landside operations, for example, the workload balance is an important objective when optimizing both the FMS and the landside operations (Arbib et al., 1991). Thus, it is worthwhile to review some related studies on the scheduling of a FMS. A well designed FMS possesses two types of flexibilities. The production flexibility refers to the ability of producing different product types and different operational orders with minimal set up effort while the routing flexibility refers to the ability of using and scheduling multiple machines for the same production operation of a part/component. A typical FMS comprises of two major subsystems – a machining subsystem and a AGV subsystem. The machining subsystem, being responsible for the machining of parts, usually includes several versatile numerical control machines. The AGV subsystem is responsible for the transportation of parts between any two different locations in the FMS. Effective methods for precise planning and control are needed for efficient operations of a FMS.

The job planning and loading problems in a FMS can be formulated into a number of mathematical programs. In the study of Stecke (1983), five production planning problems were considered for optimizing the operations of a FMS and these problems were the part type selection problem, the machine grouping problem, the production ratio problem, the resource allocation problem and the loading problem. The machine grouping problem and the loading problem were formulated as nonlinear programs. These nonlinear programs were then linearized into MIP formulations so as reduce the computational effort required. Effective methods were developed to solve the problems.

The loading problem considered in Stecke (1983) was further studied by Berrada and Stecke (1986). Their study focused on balancing the workload of each machine with the assumption that each operation was assigned to one machine only. They then discussed the solution approach of solving the nonlinear integer programming model proposed for the loading problem. The solution method was designed to solve a sequence of sub-problems and a fixed maximum workload was set to each sub-problem. The solution of each sub-problem was used to reduce the size of the interval between the lower bound and the upper bound used in their branch and bound solution method. A detailed numerical example was used to illustrate the solution method and the result of extensive computational experiments showed that the proposed algorithm was indeed effective. Stecke (1986) proposed the idea of a hierarchical approach for solving the machine grouping and loading problems simultaneously. A number of models with different levels of details were built to represent the problems. An aggregate model and several detailed models were developed for both problems. The study showed that the models were useful for solving realistic problems in a FMS.

The scheduling and tool allocation problems in FMS were also widely studied by a number of researchers. Kusiak (1986) proposed a framework to classify the planning problems of a FMS into a number of operational and planning problems. Based on the framework of this classification, relevant studies on each problem were reviewed and the proposed technique and methods were categorized. Their survey, thus, was a useful reference for the application of operational planning methods for managing a FMS. The review also provides useful insight for further research studies on FMS planning. Many of the methods considered in the review were designed for real time control of FMS (Maimon and Gershwin, 1988, Sabuncuoglu and Hommertzheim, 1992).

Maimon and Gershwin (1988) discussed a real time loading and routing problem of different machines working for the same operation in a FMS with occasional machine failures. They extended the research method proposed by Kimemia and Gershwin (1983) which was only effective under certain idealistic assumptions for identical machines. A real time algorithm with failure responsible control was proposed for the operational planning problem which was partitioned into two levels. In level 1, the approximate solution for assigning a set of machine to an operation was proposed. The solution of level 1 was used to determine the short-term loading rates of parts with various routes. The online dispatching algorithms for the scheduling problem were also analyzed with different rules and criteria (Sabuncuoglu and Hommertzheim, 1992). Through extensive simulation experiments, the most effective criterion for machine scheduling was determined in term of the mean flow-time and tardiness.

Ozpeynirci and Azizoglu (2009) proposed a Lagrangean relaxation solution method to solve the capacity allocation problem of a FMS. After analyzing the constraints related to the operational time and tool capacity for a machine, they found that an efficient allocation should assign as many as operations to a machine with the minimum cost of tools. An effective algorithm was developed to solve the allocation problem. Results of computational experiments showed that their algorithm was able to find near-optimal solutions.

It is noted that the resource allocation and parts scheduling in a FMS exhibits many similarities with some other resource planning problems in container terminals such as yard planning and operations. The operation methods, therefore, provide very valuable insights for designing effective heuristics for terminal operations. A more detailed discussion will be presented in the next chapter to discuss how these insights can be used in designing the objective function for the yard planning problem.

2.5 Conclusion

It is shown from the review of related research studies on container terminals that effective port operations are highly dependent on the application of a series of optimization models and solution algorithms designed for planning and controlling the operations of quayside, container handling equipment and yards. It is well known in terminal industry that optimization of yard operations, especially planning of container storage location in yard, is critical not only to yard operations but also the quayside operations. However, the yard operations optimization has not attracted enough attentions of researchers on terminal operations.

Minimizing the traffic congestions and reshuffling of containers in yard, which contributes to improving the landside operations efficiency, to support efficient quayside operations has been considered as a main objective in some of the aforementioned research studies on yard optimizations. However, the yard planning problem has not been studied thoroughly and the solution methods proposed cannot easily be modified to solve some important but not yet studied yard optimization problems. As discussed in Steenken et al. (2004), many factors in yard optimization for minimizing the number of re-shuffles and maximizing the storage utilization need to be considered, including container distribution in blocks, allocation of gantry cranes and other equipment, workload balance and transportation distances of containers. In other words, there are still a number of aspects of yard optimization not fully considered in the research literature. In view of the critical importance of yard operations in improving the performance of operations in land-scarce container terminals, research studies should be carried out to investigate the key factors leading to the formation of bottlenecks in yard and to develop effective methods to resolve the bottleneck. Thus, this research focuses on studying ways to prevent of formation of bottlenecks through workload balancing among yard blocks. In the literature, there are only a few research studies which have considered the workload distribution in designing the storage space allocation in yard for incoming containers to facilitate container handling (Zhang et al., 2003, Lee et al., 2006, Han et al., 2008). These papers, however, still do not sufficiently address the particular needs of a land-scarce terminal which must effectively allocate the limited storage space for different groups of containers to achieve high efficiency of loading operations in a terminal. Methods for yard template design with specific requirements for container stacking are particularly scarce in previous research studies. This research aims at proposing some feasible ideas for effectively determining the yard template for export containers and achieving even workload distribution to support efficient yard operations.

CHAPTER 3

YARD PLANNING FOR REGULAR VESSEL SERVICES

3.1 Introduction

To reduce the occurrence of operational bottlenecks in yard blocks, it is imperative for terminal planners to achieve a well balanced workload among yard blocks. In a terminal, different strategies are used for balancing the workload of different type of yard blocks. As mentioned in Chapter 1, it is difficult to predict the workload of yard blocks storing import containers due to the random nature of the retrieval time of import containers. Thus, terminal planners normally focus on balancing the workload of yard blocks storing export containers. Shipping lines usually operate a number of vessel services to call several container ports along different sea routes. The vessel services can be classified into two types – services with cyclical calling pattern and services with non-cyclical calling pattern. Large shipping lines often run regular vessel services to call major container ports along major sea routes. The estimated arrival time and the throughput of these services with cyclical calling pattern can be predicted with high accuracy whereas those of services with non-cyclical calling pattern are difficult to predict. Therefore, terminals often partition export yard blocks into two sections, one for storing export containers of services with cyclical calling pattern and the other for storing export containers of services with non-cyclical calling pattern.

Since the loading operations of export containers for a vessel are carried out within a short period of time, the operations often create bottlenecks in yard operations. It is a common practice in the terminal industry that the workload of a yard block on a particular day is measured in terms of the number of slots storing export containers to be loaded onto vessels on that day. Chapter 1 has discussed some practical stacking strategies for export containers in a container yard for coping with regular as well as irregular vessel calling schedules.

This chapter focuses on analyzing the yard planning problem for vessel services with cyclical vessel calling patterns. These vessel services are typically run by big shipping lines like Maersk Line and China Ocean Shipping. These shipping lines normally design and run a long-term static weekly vessel calling schedule for transporting seaborne cargo between big seaports. Port operators, thus, are able to forecast the workflow for storing and handling export containers and devise effective yard plans for such vessel services according to the detailed schedules of vessel services and container throughput of the services. Effective yard plan can thus be designed using a mathematical programming approach for determining the storage location of export containers to be loaded onto these services.

3.2 Storage space allocation

Typically, different stacking methods are used in different terminals to optimize their operations based on their unique operational characteristics, cost and availability of terminal resources. In mega container terminals with limited yard space, cluster stacking strategy is often adopted for allocating storage space and determining storage locations in different yard blocks for the loading containers (export containers) of each vessel service. Such a strategy is effective in facilitating the transportation and loading activities for export containers in real life operations. The main idea of cluster stacking is that a continuous stretch of slots is allocated to store the loading containers of a vessel service in a yard block. The continuous stretch of slots is called a cluster.

To avoid too high a concentration of loading activities in a container yard, it is a common practice in the terminal industry that at most one cluster is allocated for a particular service in a yard block.

Figure 3.1 shows a typical yard plan of a yard block for a weekly cycle. In the figure, there are three clusters with different loading periods (dates) allocated to the first 9 slots of a yard block. The slots occupied by each cluster are denoted by the shaded rectangle with the corresponding cluster number. It is noted from Figure 3.1 that a slot can be only assigned to at most one cluster at a period and only slots contiguous to each other can be assigned to the same cluster. Table 3.1 gives the slots assigned to the three clusters over the weekly cycle. Since each cluster can only store the loading containers for only one vessel service, the cluster number can also be used to uniquely identify the vessel service. To simplify the notation used in this thesis, the same scheme is used to number clusters and vessel services. It is clear from Figure 3.1 that vessel services 1, 2 and 3 call the terminal at period 3, 5 and 6, respectively. Suppose that clusters 1, 2 and 3 are the only clusters in the yard block. It is obvious from the figure that the workload of the block at period 3, which equals the number of slots with loading activity at that period (slots 2 to 7), is 6. Similarly, the workload at period 5 is 6 slots (slots 4 to 9) and the workload at period 6 is slots (slots 1 to 4). For cluster 1, the lower and upper range limits of the cluster at period 3 are slots 2 and 7, respectively. Since export containers for a vessel service arrive at the terminal continuously before their loading time, the size of a cluster grows steadily before their loading operation starts. Thus, it is noted from Figure 3.1 that once a cluster starts its buildup, it gradually grows bigger over time until its loading period. For example, cluster 2 starts its buildup at period 1 and occupies slot 8 and grows over time to occupy more slots until its loading operation at period 5. Thus, as time is approaching the loading period of a cluster, the lower range limit of the cluster is non-increasing as well as its upper range limit. It is noteworthy that there are some vacant slots at certain time periods, for example, slot 2 at period 4 is vacant. The number of vacant slots in a yard plan depends on the number, the position and shape of clusters assigned to a yard block.

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

Figure 3.1 A typical yard plan

Date	Slots assigned to cluster 1	Slots assigned to cluster 2	Slots assigned to cluster 3
Period 1	0	Slot 8	0
Period 2	Slots 2-3	Slots 8-9	0
Period 3	Slots 2-7	Slots 8-9	0
Period 4	0	Slots 4-9	Slot 3
Period 5	0	Slots 4-9	Slots 1-3
Period 6	0	0	Slots 1-4
Period 7	0	0	0

Table 3.1 Assignment of slots to clusters

For yard blocks dedicated to store export containers of vessel services with cyclical calling pattern, yard planners often use a yard template, a yard plan which specifies the position and shape of each cluster assigned to each service, to assign a storage location to an export container. Once determined, the yard template need not be updated until there are substantial changes in service calling time or service throughput. Currently, yard planners use their own experience, judgment to design the yard template. In view of the complexity of determining the number of services assigned to yard blocks, the position and the shape of various clusters in the yard template, it is unlikely that the current manual method is effective.

3.3 Balancing Workload in Yard Blocks

As mentioned in Section 3.1, a common objective for the yard planning problem is to determine the location for storing export containers in a container yard to balance the workload of all yard blocks. The idea of balancing workload has been discussed in many research studies. As discussed in Chapter 2, the operations of a FMS show many similarities with those of a terminal. The research studies on balancing workload of a FMS can provide some useful insights of resolving the workload balancing problem studied in this chapter. Balancing the workload of loading machines in a FMS is a very common objective for improving the productivity (Stecke and Solberg, 1981, Stecke and Morin, 1985), reducing the work-in-process inventory (Shanthikumar and Stecke, 1986) and minimizing the amount of part transfer (Arbib et al., 1991).

In a study on production planning problem of a FMS, Stecke (1983) proposed a number of objectives to be considered for optimizing the performance of the FMS. A primary objective is to balance the workload of machines in a system of pooled machines. The following two expressions were proposed to measure the workload imbalance of M machines:

1) $\sum_{l=1}^{M-1} \sum_{k=l+1}^{M} \left|\omega_l - \omega_k\right|^{\gamma}, \ \gamma > 0,$

2) $\beta - \alpha, \ 0 \le \alpha \le \omega_l \le \beta \quad l = 1,\dots,M$

where ω_l and ω_k denote the respective workloads of machine l and k and γ is a positive constant. The first expression is the sum of weighted differences between the workloads of each pair of machines while the second expression is the maximum difference in workload among all the machines.

It is a common practice in the terminal industry that the yard planners attempt to minimize the maximum differences of workloads among yard blocks. The same objective is also used in the study by Zhang et al. (2003). Using a workload-balance objective similar to expression (2), they studied the problem of allocating storage space in a container yard to in each time period to balance the workload of yard blocks storing both loading and unloading containers. The major decisions of the problem are to determine $D_{i,t}$, the total number of containers which will be unloaded from vessels at period *t* to block *i*, $L_{i,t}$, the total number of containers which will be loaded onto vessels at period *t* from block *i*, $G_{i,t}$, the total number of containers brought in by external trucks to be stored in block *i* at period *t,* and $P_{i,t}$, the total number of containers to be picked up by external trucks in block *i* at period *t*. Thus, the workload of block *i* at period *t* generated from handling operation for vessels was equal to $D_{i,t} + L_{i,t}$ and the total workload of block *i* at period *t* $D_{i,t} + L_{i,t} + G_{i,t} + P_{i,t}$. Their study proposed workload balancing objective:

$$\sum_{t=1}^{T} \{ \; w_1[\max_i(D_{i,t} + L_{i,t}) - \min_i(D_{i,t} + L_{i,t})] +$$

$$w_2[\max_i(D_{i,t} + L_{i,t} + G_{i,t} + P_{i,t}) - \min_i(D_{i,t} + L_{i,t} + G_{i,t} + P_{i,t})]\}$$

where w_1 and w_2 were the weights for expressing importance of the two types of workloads. The objective measures the sum of weighted maximum differences of workloads among yard blocks for the entire planning horizon. In most terminals, a higher priority is given to the workload generated from handling operation for vessels.

It is noted from the aforementioned research studies that the sum of weighted maximum differences of the workload among all the equipment is a common measure of workload imbalance.

When planning the yard to balance the workload for a given planning horizon, terminals often treat the balance of workload at each time period as equally important. For planning the yard section storing export containers only, the objective considered in Zhang's study can be simplified as $\sum_{t=1}^{T}\left\{\max_i(L_{i,t})-\min_i(L_{i,t})\right\}$. Since $L_{i,t}$ has a very high correlation with the number of loading slots in block *i* at period *t*, the number of loading slots is often used as a measure of workload.

3.4 An integer programming model

In this section, the problem of determining the location and the size of each yard cluster in a planning cycle for a number of vessel services with known storage requirements is formulated as an integer program. Suppose that there are *I* blocks in the yard, each contains *K* slots for storing export containers from *J* vessel services which may call the terminal at different time in the planning cycle of *T* time periods.

The following information is assumed to be known at the beginning of the yard planning.

Ω_j the time period that service *j* calls for a loading job,

$R_{j,t}$ the space requirement of service *j* at period *t*.

It is clear from the definition that R_{j,Ω_j} is the largest number of slots required by service *j* and $R_{j,t} \le R_{j,t+1}$ for $t = 1,2,...,\Omega_j - 1, \Omega_j + 1,..., T-1$ since storage space requirement increases when the day is approaching the loading period. As the vessel services are operated with a cycle length of *T* periods, a parameter is needed to denote the case that period *T* is the loading period of a vessel service so as to state the correct relationship between $R_{j,t}$ and $R_{j,t+1}$. Define

$$\rho_j = \begin{cases} 0, & \text{if } R_{j,T} > R_{j,1} \text{ (this case will only occur when } \Omega_j = T\text{)} \\ 1, & \text{otherwise} \end{cases}.$$

The following variables are needed for formulating the problem.

$$X^i_{k,j,t} = \begin{cases} 1, & \text{if slot } k \text{ of block } i \text{ is assigned to service } j \text{ at period } t \\ 0, & \text{otherwise} \end{cases},$$

$$Y^i_{k,j,t} = \begin{cases} 1, & \text{if slot } k \text{ of block } i \text{ is the upper range limit of the cluster assigned} \\ & \text{to service } j \text{ at period } t \\ 0, & \text{otherwise} \end{cases},$$

α_t the smallest number of slots with loading jobs among all blocks at period t,

β_t the largest number of slots with loading jobs among all blocks at period t.

Denote the set $\{1,2,...,I\}$ by **I**, the set $\{1,2,...,K\}$ by **K**, the set $\{1,2,...,T\}$ by **T**, the set $\{1,2,...,J\}$ by **J**, the set of $X^i_{k,j,t}$ by **X** and the set of $Y^i_{k,j,t}$ by **Y**. Define $X^i_{K+1,j,t}$ to be 0 for $i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T}$.

The yard planning problem, then, can be stated as the following integer programming problem.

Problem P1

Minimize $\sum_{t=1}^{T} (\beta_t - \alpha_t)$

Subject to

$$X^i_{k,j,t-1} \le X^i_{k,j,t} \qquad k \in \mathbf{K}, i \in \mathbf{I}, j \in \mathbf{J}, t = 2,3,\ldots,\Omega_j,\Omega_j+2,\ldots,T \tag{3.1}$$

$$\rho_j \cdot X^i_{k,j,T} \le X^i_{k,j,1} \qquad k \in \mathbf{K}, i \in \mathbf{I}, j \in \mathbf{J} \tag{3.2}$$

$$\sum_{j=1}^{J} X^i_{k,j,t} \le 1 \qquad k \in \mathbf{K}, i \in \mathbf{I}, t \in \mathbf{T} \tag{3.3}$$

$$\sum_{i=1}^{I} \sum_{k=1}^{K} X^i_{k,j,t} = R_{j,t} \qquad j \in \mathbf{J}, t \in \mathbf{T} \tag{3.4}$$

$$\sum_{j=1}^{J}\sum_{k=1}^{K} X_{k,j,t}^{i} \le K \qquad i \in \mathbf{I}, t \in \mathbf{T} \tag{3.5}$$

$$\sum_{k=1}^{K} Y_{k,j,t}^{i} \le 1 \qquad i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T} \tag{3.6}$$

$$X_{k,j,t}^{i} - X_{k+1,j,t}^{i} \le Y_{k,j,t}^{i} \qquad k \in \mathbf{K}, i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T} \tag{3.7}$$

$$\sum_{j \in \mathbf{J}, \Omega_j = t}\ \sum_{k=1}^{K} X_{k,j,t}^{i} \ge \alpha_t \qquad i \in \mathbf{I}, t \in \mathbf{T} \tag{3.8}$$

$$\sum_{j \in \mathbf{J}, \Omega_j = t}\ \sum_{k=1}^{K} X_{k,j,t}^{i} \le \beta_t \qquad i \in \mathbf{I}, t \in \mathbf{T} \tag{3.9}$$

The objective of Problem P1 is to balance the workload of the export blocks by minimizing the sum of maximum differences of workload among the yard block over the entire planning horizon**.** Constraints (3.1) and (3.2) ensure that as time is approaching the loading period of service *j*'s cluster, the respective lower and upper range limits of the cluster are non-increasing and non-decreasing. Constraints (3.3) state that each slot of each export block can only be assigned to at most one vessel service at any period. Constraints (3.4) ensure that the total number of slots assigned to vessel service *j* at period *t* equal to vessel *j*'s storage space requirement at that period. Constraints (3.5) state that the total number of slots assigned to clusters in block *i* cannot exceed the storage capacity of the block. Constraints (3.6) ensure that at most one cluster is assigned to vessel service *j* in block *i* as there is at most one slot in the block being the upper range limit of the cluster at service's *j* loading period. Constraints (3.7) give the relationship between variables in **X** and **Y**. Consider the two mutually exclusive cases implied by Constraints (3.6). For the case that $Y_{k,j,t}^{i} = 0$ for all *k*, it can be shown by sequentially replacing $Y_{k,j,t}^{i}$ by 0 in Constraints (3.7) from $k = K$ to $k = 1$ that $X_{k,j,t}^{i}$ must be 0 for $k \in \{1, 2, .., K\}$; thus, there is no cluster assigned to service *j* in block *i*. For the case that there exists only one *k*′, $k' \in \{1, 2, .., K\}$, such that $Y_{k',j,t}^{i} = 1$, it can be shown by sequentially replacing $Y_{k,j,t}^{i}$

by 0 in Constraints (3.7) from $k = K$ to $k = k'+1$ that $X^i_{k,j,t}$ must be 0 for all $k \in \{k'+1, 2, .., K\}$ and $X^i_{k,j,t}$ can be equal to 1 for $k \in \{1, 2, .., k'\}$. If there exists a k'', $k'' \in \{1, 2, .., k'-1\}$, such that $X^i_{k'',j,t}=1$, it can be shown that by sequentially replacing $Y^i_{k,j,t}$ by 0 in Constraints (3.7) from $k = k''$ to $k = k'-2$ that $X^i_{k,j,t}$ must be equal to 1 for $k \in \{k'', k''+1, .., k'-1\}$. Therefore, Constraints (3.6) and (3.7) together ensure that the slots in block i assigned to vessel j must be located at adjacent positions from each other. Constraints (3.8) and (3.9) give the relationship between **X** and β_t and that between **X** and α_t, respectively.

3.5 Feasibility Conditions

This section studies the conditions under which for a given set of parameter values Problem P1 has feasible solutions. A necessary feasibility condition and a sufficient feasibility condition are derived.

3.5.1 Necessary Feasibility Condition

It is obvious from the definition of $R_{j,t}$ and K that the total demand of storage space for all vessel services at period t equals $\sum_{j=1}^{J} R_{j,t}$ and the total storage capacity of all the blocks is $I \cdot K$. Therefore, for Problem P1 to be feasible, a simple necessary condition is $\sum_{j=1}^{J} R_{j,t} \le I \cdot K$.

3.5.2 Sufficient Feasibility Condition

Consider a hypothetical case that all the yard blocks in the export yard are now merged into one mega yard block with infinite storage capacity. Let Z be the largest slot number of the slot occupied by services in the mega yard block. Define $X_{k+1,j,t}$ to be 0 for $j \in \mathbf{J}, t \in \mathbf{T}$ when $k = \sum_{j=1}^{J} R_{j,\Omega_j}$. The problem of constructing a yard template for the mega yard block can be formulated as:

Problem FP

Minimize Z

Subject to

$$X_{k,j,t-1} \le X_{k,j,t} \qquad k = 1,\ldots,\sum_{j=1}^{J} R_{j,\Omega_j}, j \in \mathbf{J}, t = 2,3,\ldots,\Omega_j,\Omega_j + 2,\ldots,T \tag{3.10}$$

$$\rho_j \cdot X_{k,j,T} \le X_{k,j,1} \qquad k = 1,\ldots,\sum_{j=1}^{J} R_{j,\Omega_j}, j \in \mathbf{J} \tag{3.11}$$

$$\sum_{j=1}^{J} X_{k,j,t} \le 1 \qquad k = 1,\ldots,\sum_{j=1}^{J} R_{j,\Omega_j}, t \in \mathbf{T} \tag{3.12}$$

$$\sum_{k=1}^{R_{1,\Omega_1}+\ldots+R_{J,\Omega_J}} X_{k,j,t} = R_{j,t} \qquad j \in \mathbf{J}, t \in \mathbf{T} \tag{3.13}$$

$$\sum_{k=1}^{R_{1,\Omega_1}+\ldots+R_{J,\Omega_J}} Y_{k,j,t} \le 1 \qquad j \in \mathbf{J}, t \in \mathbf{T} \tag{3.14}$$

$$X_{k,j,t} - X_{k+1,j,t} \le Y_{k,j,t} \qquad k = 1,\ldots,\sum_{j=1}^{J} R_{j,\Omega_j}, j \in \mathbf{J}, t \in \mathbf{T} \tag{3.15}$$

$$k \cdot X_{k,j,t} \le Z \qquad k = 1,\ldots,\sum_{j=1}^{J} R_{j,\Omega_j}, j \in \mathbf{J}, t \in \mathbf{T} \tag{3.16}$$

The objective of this problem is to construct a yard block template for all vessel services with the minimum number of slots used. The constraints of Problem FP are similar to those of Problem P1.

It is obvious that Problem FP is always feasible. It is interesting to note that the feasible solutions of Problem FP have a property that is useful for deriving a sufficient condition for Problem P1 to have feasible solutions. This property is summarized in the lemma below.

Lemma 3.1

Suppose that $\{\widetilde{X}, \widetilde{Y}, \widetilde{Z}\}$ is a feasible solution of Problem FP. A feasible solution can always be constructed for Problem P1 if $\widetilde{Z} \le I \cdot K$.

Proof:

$\{\widetilde{X}, \widetilde{Y}, \widetilde{Z}\}$ is a feasible solution of Problem FP and hence, it satisfies Constraints (3.10)- (3.16).

It can be derived from Constraints (3.16) and Expression $\widetilde{Z} \le I \cdot K$ that

$$k \cdot \widetilde{X}_{k,j,t} \le \widetilde{Z} \le I \cdot K \quad \text{for} \quad k = 1,\ldots,\sum_{j=1}^{J} R_{j,\Omega_j}, j \in \mathbf{J}, t \in \mathbf{T}.$$

Let $\max\{k \cdot \widetilde{X}_{k,j,t}\} = \hat{K}$ for $k = 1,\ldots,\sum_{j=1}^{J} R_{j,\Omega_j}, j \in \mathbf{J}, t \in \mathbf{T}$.

The storage space in this mega yard block is sequentially partitioned into I yard blocks with identical size of K slots. For yard block i, $\{\overline{X}, \overline{Y}\}$ is constructed as follows:

Set $\overline{X}^{i}_{k,j,t} = \widetilde{X}_{(i-1)\cdot K+k,j,t}$ and $\overline{Y}^{i}_{k,j,t} = \widetilde{Y}_{(i-1)\cdot K+k,j,t}$ for $i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T}$ and k satisfying the expression $1 \le (i-1)\cdot K + k \le \min\{\sum_{j=1}^{J} R_{j,\Omega_j}, I \cdot K\}$.

If $\sum_{j=1}^{J} R_{j,\Omega_j} < I \cdot K$, set $\overline{X}^{i}_{k,j,t} = 0$, $\overline{Y}^{i}_{k,j,t} = 0$ for $i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T}$ and k satisfying the expression $\sum_{j=1}^{J} R_{j,\Omega_j} \le (i-1)\cdot K + k \le I \cdot K$.

For $i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T}$, if $\widetilde{X}_{i\cdot K,j,t} > 0$, set $\overline{Y}^{i}_{K,j,t} = 1$.

Set $\overline{X}^{i}_{K+1,j,t}=0$ for $i\in\mathbf{I}, j\in\mathbf{J}, t\in\mathbf{T}$.

The above procedure constructs $\{\overline{X},\overline{Y}\}$ has been constructed for *I* yard blocks on the basis of the solution $\{\widetilde{X},\widetilde{Y},\widetilde{Z}\}$ for problem FP. Next, it is proved in the following that $\{\overline{X},\overline{Y}\}$ satisfies all constraints of Problem P1.

It is obvious from the construction of solution $\{\overline{X},\overline{Y}\}$ that $\overline{X}^{i}_{k,j,t} = \widetilde{X}_{(i-1)\cdot K+k,j,t}$ for the case that $1\le (i-1)\cdot K+k\le \min\{\sum_{j=1}^{J} R_{j,\Omega_j}, I\cdot K\}$. For this case, it can be shown from Constraints (3.10) that $\{\overline{X},\overline{Y}\}$ must satisfy Constraints (3.1) when $1\le (i-1)\cdot K+k\le \hat{K}$ for all $i\in\mathbf{I}, j\in\mathbf{J}, t\in\mathbf{T}$. It is clear from the procedure for constructing $\overline{X}^{i}_{k,j,t}$ that $\overline{X}^{i}_{k,j,t}=0$ when $\hat{K}\le (i-1)\cdot K+k\le I\cdot K$ for all $i\in\mathbf{I}, j\in\mathbf{J}, t\in\mathbf{T}$. Obviously, it satisfies Constraints (3.1) as well. Similarly, it can be shown from the same procedure that the solution satisfies Constraints (3.2) and (3.3). It can be shown from Constraints (3.13) that

$$\sum_{k=1}^{R_{1,\Omega_1}+\ldots+R_{J,\Omega_J}} \widetilde{X}_{k,j,t} = \sum_{k=1}^{\hat{K}} \widetilde{X}_{k,j,t} + \sum_{k=\hat{K}}^{R_{1,\Omega_1}+\ldots+R_{J,\Omega_J}} \widetilde{X}_{k,j,t} = \sum_{k=1}^{\hat{K}} \widetilde{X}_{k,j,t} = R_{j,t}.$$

Hence, it is clear the construction procedure of $\{\overline{X},\overline{Y}\}$ that $\sum_{i=1}^{I}\sum_{k=1}^{K} \overline{X}^{i}_{k,j,t} = \sum_{k=1}^{\hat{K}} \widetilde{X}_{k,j,t} = R_{j,t}$. Thus, $\{\overline{X},\overline{Y}\}$ satisfies Constraints (3.4).

It can be shown from $\sum_{j=1}^{J} \overline{X}^{i}_{k,j,t} \le 1$ that $\sum_{j=1}^{J}\sum_{k=1}^{K} \overline{X}^{i}_{k,j,t} = \sum_{k=1}^{K}\sum_{j=1}^{J} \overline{X}^{i}_{k,j,t} \le K$ which are Constraints (3.5).

It can be shown from Constraints (3.14) that $\sum_{k=1}^{R_{1,\Omega_1}+\ldots+R_{J,\Omega_J}} \widetilde{Y}_{k,j,t} \le 1$, namely, there is at most one *k*, $k=1,\ldots,\sum_{j=1}^{J} R_{j,\Omega_j}$ satisfying $\widetilde{Y}^{i}_{k,j,t}=1$ in the mega yard block. However, during constructing solutions for yard block i, $i \in \mathbf{I}$, the value of $\overline{Y}^{i}_{K,j,t}$ may be set to 1 when $\widetilde{X}_{i \cdot K,j,t} > 0$. Next, it is proved in the following that it is impossible for slot k, $k \neq K$ satisfying $\overline{Y}^{i}_{k,j,t} = 1$ if $\overline{Y}^{i}_{K,j,t} = 1$.

If there exists $\overline{Y}^{i'}_{k',j,t} = 1$ for slot k', $k' < K$ in yard block i', it is obvious that $\widetilde{Y}_{(i'-1)\cdot K+k',j,t} = 1$, i.e., the slot is the upper range limit of cluster j. Then, it indicates that $\widetilde{X}_{i' \cdot K,j,t} = 0$ and $\overline{Y}^{i'}_{K,j,t} = 0$. Therefore, it is impossible that two $\overline{Y}^{i}_{k,j,t}$ with different values of k in the same yard block both take up the value 1 at the same time period t for the same service j. Thus, Constraints (3.6) are satisfied.

Substituting $\overline{X}^{i}_{k,j,t}$ and $\overline{Y}^{i}_{k,j,t}$ into Constraints (3.7), it can be shown from $\overline{X}^{i}_{k,j,t} - \overline{X}^{i}_{k+1,j,t} = \widetilde{X}_{(i-1)\cdot K+k,j,t} - \widetilde{X}_{(i-1)\cdot K+k+1,j,t} \leq \overline{Y}^{i}_{k,j,t}$ when $\overline{Y}^{i}_{k,j,t} = 0$ and $\overline{X}^{i}_{k,j,t} - \overline{X}^{i}_{k+1,j,t} = 1 \leq \overline{Y}^{i}_{k,j,t}$ when $\overline{Y}^{i}_{k,j,t} = 1$. Hence, $\{\overline{X}, \overline{Y}\}$ satisfies Constraints (3.7).

It is obvious that $\{\overline{X}, \overline{Y}\}$ satisfies Constraints (3.8) and (3.9).

Therefore, $\{\overline{X}, \overline{Y}\}$ is a feasible solution of problem P1 given the condition that $\widetilde{Z} \leq I \cdot K$.

It follows from the proof of the above lemma that once a feasible solution of FP satisfying the sufficient feasibility condition is determined, it can be used for constructing a feasible solution for P1.

3.6 A simple example

A simple experiment is designed to evaluate the possibility of finding optimal solutions for the problem using a commercial optimization software CPLEX 7.1. Some small size test problems with only 3 yard blocks serving for export containers from 10 vessel services are tested. Given a storage capacity of 40 slots in each yard block and a planning cycle of 7 time periods, the loading date Ω_j for each service j was randomly generated from an uniform distribution [1, 7] and the largest space requirement R_{j,Ω_j} was randomly generated from an uniform distribution [10, 30]. CPLEX was run on a desktop computer with 2.4 GHz CPU. For most test problems, CPLEX could find the optimal within 30 minutes. However, when the total storage requirement is close to the capacity of yard blocks as shown in Table 3.2, it took 2 hours for finding a feasible solution. Thus, effective algorithms are needed for solving this problem quickly with satisfactory solutions.

Slots requirement							
service	period						
	1	2	3	4	5	6	7
1	1	2	2	2	2	13	15
2	1	2	3	4	5	6	17
3	0	1	10	10	10	11	20
4	2	4	7	9	10	15	16
5	4	5	7	9	10	13	16
6	11	15	16	2	4	7	9
7	11	23	24	4	5	7	8
8	2	8	9	20	0	0	0
9	0	5	15	21	0	0	0
10	7	9	12	20	0	5	6

Table 3.2 A yard planning problem with tight capacity constraint

3.7 Summary of the Chapter

This chapter has discussed in detailed the yard planning problem for vessel services with cyclical calling pattern. The yard planning problem has been formulated as an integer program. A computational experiment has been carried out to evaluate the feasibility of solving the yard planning problem using a popular optimization software. The computational results have shown that it is not practical to solve the problem by the software and an effective solution algorithm needs to be developed.

CHAPTER 4

EFFICIENT ALGORITHMS FOR SOLVING THE YARD PLANNING PROBLEM

4.1 Introduction

As shown in the last section of Chapter 3, it is very time consuming to find the optimal solution of small-scale test problems. In view of the scale of real-life yard planning problems, it is unlikely that Problem P1 can be solved optimally to yield the optimal yard plan within an acceptable period of time. It is noted from the results of computational experiment conducted in Chapter 3 that even finding a feasible solution to Problem P1 can take a long time.

This chapter aims at developing an efficient and effective algorithm for finding good feasible solutions to Problem P1. Problem P1 is a complicated problem of fitting a variable number of yard clusters, each of which has variable size and shape, into each yard block of fixed storage capacity such that the total storage space assigned to a vessel meets its storage space requirement at each period. It is difficult to determine the size, shape and location of each cluster at the same time to fit into a yard block since such a solution strategy involves a large number of possible combinations. Diagram 4.1 below shows all the possible shapes of a cluster of size defined by the following yard-slot allocation that 1 slot is allocated on day 4, 3 slots on day 2 and 4 slots on day 7.

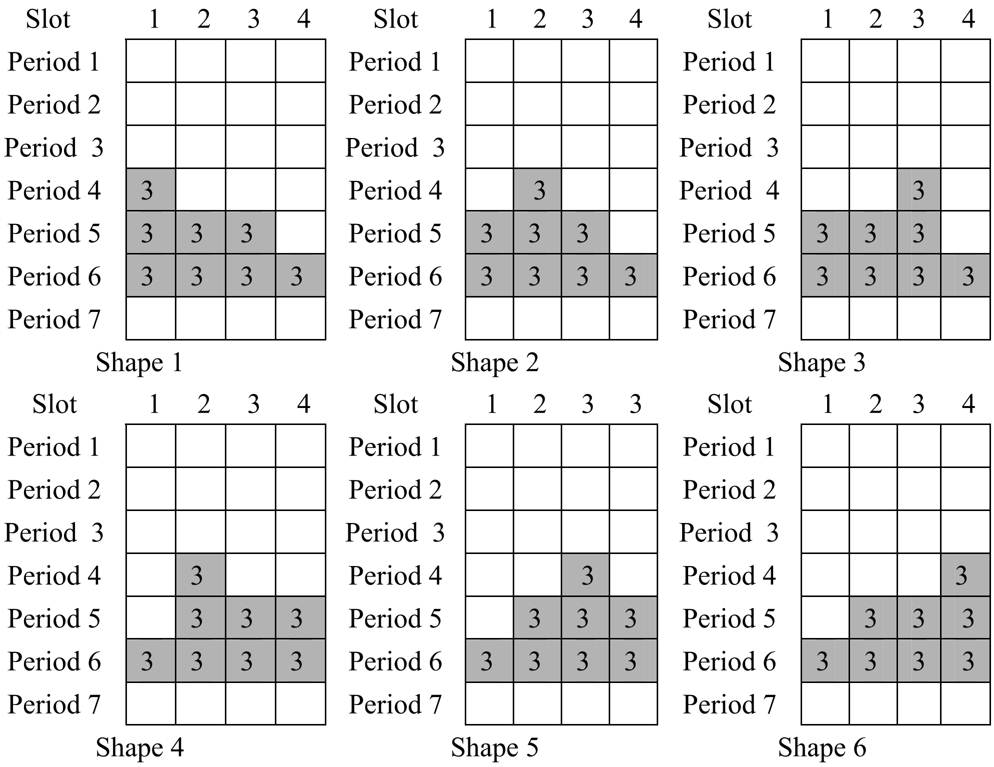


Figure 4.1 Six possible shapes of a cluster

To reduce the computational effort spent on examining large number combinations of variables associated with yard clusters, a two-phase approach is proposed to solve Problem P1. The approach decomposes Problem P1 into two smaller interrelated sub-problems – Cluster Allocation Problem (CAP) and Cluster Configuration Problem (CCP). CAP, the sub-problem considered in Phase 1, is the problem of determining the number of clusters, the size of each cluster and the vessel service served by each cluster in each yard block while CCP, the sub-problem tackled in Phase 2, is the problem of determining the shape and location of each cluster based on the solution found by the first phase. Details of the two sub-problems are presented below.

4.2 Cluster Allocation Problem in Phase 1

To simplify the problem, the constraints that define the shape of a cluster in

Problem P1 are relaxed. With the relaxation of such constraints, a modified version of Problem P1 is formulated below to find the number of clusters and the size of these clusters in each yard block such that the total storage space assigned to a vessel meets its storage space requirement at each period. The following variables are needed to state the problem:

$U^i_{j,t}$ the number of slots assigned to the cluster of service *j* in block *i* at period *t*;

α_t the smallest workload of all the yard blocks at period *t*;

β_t the largest workload of all the yard blocks at period *t*.

CAP:

Minimize $\sum_{t=1}^{T}(\beta_t - \alpha_t)$

Subject to

$$U^i_{j,t-1} \le U^i_{j,t} \qquad i \in \mathbf{I}, j \in \mathbf{J}, t = 2,3,\ldots,\Omega_j, \Omega_j + 2,\ldots,T \qquad (4.1)$$

$$\rho_j U^i_{j,T} \le U^i_{j,1} \qquad i \in \mathbf{I}, j \in \mathbf{J} \qquad (4.2)$$

$$\sum_{i=1}^{I} U^i_{j,t} = R_{j,t} \qquad j \in \mathbf{J}, t \in \mathbf{T} \qquad (4.3)$$

$$\sum_{j=1}^{J} U^i_{j,t} \le K \qquad i \in \mathbf{I}, t \in \mathbf{T} \qquad (4.4)$$

$$\sum_{j \in J \& \Omega_j = t} U^i_{j,t} \ge \alpha_t \qquad i \in \mathbf{I}, t \in \mathbf{T} \qquad (4.5)$$

$$\sum_{j \in J \& \Omega_j = t} U^i_{j,t} \le \beta_t \qquad i \in \mathbf{I}, t \in \mathbf{T} \qquad (4.6)$$

The objective of the problem is to minimize the total difference of workload among blocks at loading days. Constraints (4.1) and (4.2) ensure that the number of slots assigned to one cluster increases as time approaches the loading day of the cluster. Constraints (4.3) state that the total number of slots assigned to a vessel service in all the yard blocks should equal to the space requirement of the service. Constraints (4.4) ensure that the number of slots assigned to various services in block i at period t must not be greater than the total number of slots in the block. Constraints (4.5) and (4.6) state the relationship between decision variables $U_{j,t}^{i}$, α_t and β_t.

An experiment is carried out to work on the numerical example stated in Table 3.2 by ILOG CPLEX, the typical solution for CAP is shown in Table 4.1- 4.3. It can be shown from these tables that the storage space requirement for ten services has been decomposed to three blocks that each block may only provide space for storing export contains of several services. In block 1, storage space will be reserved for storing export containers for 7 vessel services, i.e. $\{2,3,4,6,7,8,9\}$ as shown in Table 4.1.

$U_{j,t}^{1}$							
	t						
j	1	2	3	4	5	6	7
2	1	2	2	2	2	2	2
3	0	0	5	5	5	5	10
4	2	4	7	9	10	15	16
6	5	6	6	0	0	3	5
7	7	7	7	4	5	6	7
8	2	8	9	19	0	0	0
9	0	1	1	1	0	0	0

Table 4.1 Solution of $U_{j,t}^{i}$ in block 1

In block 2, storage space will be assigned to services $\{1,3,5,6,7,8,10\}$ to meet their storage space requirements in Table 4.2.

$U^2_{j,t}$							
	t						
j	1	2	3	4	5	6	7
1	1	2	2	2	2	2	2
3	0	1	5	5	5	6	10
5	4	5	7	9	10	13	16
6	6	9	10	2	4	4	4
7	3	3	3	0	0	0	0
8	0	0	0	1	0	0	0
10	7	9	12	20	0	5	6

Table 4.2 Solution of $U^i_{j,t}$ in block 2

In block 3, only 4 services are assigned storage space by the solution which are {1,2,7,9}.

$U^3_{j,t}$							
	t						
j	1	2	3	4	5	6	7
1	0	0	0	0	0	11	13
2	0	0	1	2	3	4	15
7	1	13	14	0	0	1	1
9	0	4	14	20	0	0	0

Table 4.3 Solution of $U^i_{j,t}$ in block 3

Although the exact position of storage space for any one service in the service cycle has not been determined yet, this solution of CAP has determined the number of clusters in each block and their space requirements such that it provides useful data for clusters' configuration. Moreover, it is shown from the three tables that

$\alpha_1 = \alpha_2 = \alpha_5 = \alpha_6 = 0$, $\alpha_3 = 13$, $\alpha_4 = 20$, $\alpha_7 = 28$

and

$\beta_1 = \beta_2 = \beta_5 = \beta_6 = 0$, $\beta_3 = 14$, $\beta_4 = 21$, $\beta_7 = 28$

such that the solution has achieved the workload balance for loading days with the minimum objective value 2.

4.3 Cluster Configuration Problem in Phase 2

Once CAP discussed in the previous section is solved, the partial the solution of the problem, $U^i_{j,t}$, can be used as an input to the problem of determining the exact shape and location of each cluster in each yard block. It is clear from the definition of $U^i_{j,t}$ that with known $U^i_{j,t}$ one can determine the number of slots assigned to vessel service j in yard block i at period t. With these slot assignments determined in Phase 1, Problem P1 can be decomposed into I smaller independent sub-problems, with sub-problem i corresponding to yard block i.

It is clear from the typical solution for CAP in section 4.2 that the space requirement for ten services in Table 3.2 has been decomposed to three blocks as shown in Table 4.1- 4.3. Therefore, solving Problem P1 in Table 3.2 has been transformed into solving 3 independent sub-problems such as $U^1_{j,t}$, in Table 4.1 is the input of the first sub-problem for determining the exact shape and locations of clusters {2,3,4,6,7,8,9} in Block 1.

Define z_i to be the biggest slot number used for defining the shape and location of all the clusters in block i. The larger is the value of z_i, the larger the number of slots used for fitting all the clusters found in Phase 1 into block i. With known $U^i_{j,t}$, Constraints (3.4), (3.7) and (3.8) of Problem P1 can be modified and Cluster Configuration Problem for block i, CCP_i, is stated below.

CCP_i:

Minimize z_i

Subject to

$$X^i_{k,j,t-1} \le X^i_{k,j,t} \qquad k \in \mathbf{K}, j \in \mathbf{J}, t = 2,3,\ldots,\Omega_j, \Omega_j + 2,\ldots,T \qquad (4.7)$$

$$\rho_j \cdot X^i_{k,j,T} \le X^i_{k,j,1} \qquad k \in \mathbf{K}, j \in \mathbf{J} \qquad (4.8)$$

$$\sum_{j=1}^{J} X^i_{k,j,t} \le 1 \qquad k \in \mathbf{K}, t \in \mathbf{T} \qquad (4.9)$$

$$\sum_{k=1}^{K} X^i_{k,j,t} = U^i_{j,t} \qquad j \in \mathbf{J}, t \in \mathbf{T} \qquad (4.10)$$

$$\sum_{k=1}^{K} Y^i_{k,j,t} \le 1 \qquad j \in \mathbf{J}, t \in \mathbf{T} \qquad (4.11)$$

$$X^i_{k,j,t} - X^i_{k+1,j,t} \le Y^i_{k,j,t} \qquad k \in \mathbf{K}, j \in \mathbf{J}, t \in \mathbf{T} \qquad (4.12)$$

$$k \cdot X^i_{k,j,t} \le z_i \qquad k \in \mathbf{K}, j \in \mathbf{J}, t \in \mathbf{T} \qquad (4.13)$$

The objective of the problem is to minimize the number of slots assigned to all the clusters in block *i*. The constraints of CCP$_i$ are the same as the constraints of Problem P1 associated with block *i* except that Constraints (3.4) are replaced by (4.10) and Constraints (3.7) and (3.8) replaced by (4.13).

4.4 Properties of Cluster Allocation Problem

In this section, the properties of CAP and CCP$_i$ are analyzed and their two important properties are discussed in detail. These properties, given in the following two lemmas, can be used to design a solution algorithm for solving Problem P1.

Lemma 4.1

The optimal solution of Problem P1 is a feasible solution of CAP.

Proof:

Suppose that $\{\hat{X}, \hat{Y}, \hat{\alpha}, \hat{\beta}\}$ is the optimal solution of Problem P1. Hence, the optimal objective function value of Problem P1 = $\sum_{t=1}^{T}(\hat{\beta}_t - \hat{\alpha}_t)$. It is clear from the definition of $U^i_{j,t}$ that for $i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T}$,

$$U^i_{j,t} = \sum_{k=1}^{K} X^i_{k,j,t} \; . \tag{4.14}$$

It follows from Expressions (4.14) and Constraints (3.4) that $\sum_{i=1}^{I}\sum_{k=1}^{K} \hat{X}^i_{k,j,t} = \sum_{i=1}^{I} U^i_{j,t} = R_{j,t}$. Thus, $\{\hat{X}, \hat{Y}, \hat{\alpha}, \hat{\beta}\}$ satisfies Constraints (4.3).

By replacing $\sum_{k=1}^{K} \hat{X}^i_{k,j,t}$ with $U^i_{j,t}$ in Constraints (3.5), (3.8) and (3.9), it is clear that $\{\hat{X}, \hat{Y}, \hat{\alpha}, \hat{\beta}\}$ satisfies Constraints (4.4), (4.5) and (4.6).

Since $\{\hat{X}, \hat{Y}, \hat{\alpha}, \hat{\beta}\}$ satisfies Constraints (3.1) and (3.2), it is obvious that $\hat{X}^i_{k,j,t-1} \le \hat{X}^i_{k,j,t}$ for $i \in \mathbf{I}, k \in \mathbf{K}, j \in \mathbf{J}, t = 2,3,...,\Omega_j, \Omega_{j+2}...,T$ and $\rho_j \sum_{k=1}^{K} \hat{X}^i_{k,j,T} \le \sum_{k=1}^{K} \hat{X}^i_{k,j,1}$. Hence, it follows from Expressions (4.14) that $U^i_{j,t-1} = \sum_{k=1}^{K} \hat{X}^i_{k,j,t-1} \le \sum_{k=1}^{K} \hat{X}^i_{k,j,t} = U^i_{j,t}$ and $\rho_j U^i_{j,T} = \rho_j \sum_{k=1}^{K} \hat{X}^i_{k,j,T} \le \sum_{k=1}^{K} \hat{X}^i_{k,j,1} = U^i_{j,1}$. Thus, $\{\hat{X}, \hat{Y}, \hat{\alpha}, \hat{\beta}\}$ satisfies Constraints (4.1) and (4.2).

Therefore, the optimal solution of Problem P1 satisfies all the constraints of CAP. Hence, the lemma is true. ■

Since the objective functions of Problem P1 and CAP are the same, it is clear from Lemma 4.1 that the following corollary is true.

Corollary 4.1

The optimal objective function value of Problem P1 is greater than or equal to the optimal objective function value of CAP.

Another important property of CAP and CCP$_i$ is given in the following lemma.

Lemma 4.2

Suppose that $\{\hat{U},\hat{\alpha},\hat{\beta}\}$ is the optimal solution of CAP, $\{\tilde{X}^i,\tilde{Y}^i,\tilde{z}_i\}$ is a feasible solution of CCP$_i$ where $\tilde{X}^i$ is the set of $\tilde{X}^i_{k,j,t}$ and $\tilde{Y}^i$ is the set of $\tilde{Y}^i_{k,j,t}$, $k\in\mathbf{K}, j\in\mathbf{J}, t\in\mathbf{T}$. Let $\tilde{X}=\{\tilde{X}^1,...,\tilde{X}^I\}$ and $\tilde{Y}=\{\tilde{Y}^1,...,\tilde{Y}^I\}$. $\{\tilde{X},\tilde{Y},\hat{\alpha},\hat{\beta}\}$, the set formed by combining subsets of $\{\hat{U},\hat{\alpha},\hat{\beta}\}$ and $\{\tilde{X},\tilde{Y},\tilde{z}_i\}$, is the optimal solution of Problem P1.

Proof:

It is clear from Corollary 4.1 that the optimal objective value of CAP is not larger than that of Problem P1. Thus, the lemma can be proved by showing that $\{\tilde{X},\tilde{Y},\hat{\alpha},\hat{\beta}\}$ is a feasible solution of Problem P1.

Since $\{\tilde{X}^i,\tilde{Y}^i,\tilde{z}_i\}$ is a feasible solution of CCP$_i$, $\{\tilde{X},\tilde{Y}\}$ satisfies Constraints (4.7)-(4.9), (4.11) and (4.12). Hence, $\{\tilde{X},\tilde{Y}\}$ satisfies Constraints (3.1)-(3.3), (3.6) and (3.7) for $k\in\mathbf{K}, i\in\mathbf{I}, j\in\mathbf{J}, t\in\mathbf{T}$.

As $\{\hat{U},\hat{\alpha},\hat{\beta}\}$ is an optimal solution of CAP, $\{\hat{U},\hat{\alpha},\hat{\beta}\}$ satisfies Constraints (4.3)-(4.6). Combining Constraints (4.3) and (4.10), we have $\sum_{i=1}^{I}U^i_{j,t}=\sum_{i=1}^{I}\sum_{k=1}^{K}X^i_{k,j,t}=R_{j,t}$ which is Constraints (3.4). Hence, $\{\tilde{X},\tilde{Y},\hat{\alpha},\hat{\beta}\}$ satisfies Constraints (3.4). It follows from Constraints (4.4) and (4.10) that $\{\tilde{X},\tilde{Y},\hat{\alpha},\hat{\beta}\}$ satisfies Constraints (3.5). Similarly, Constraints (3.8) and (3.9) can be derived by combining Constraints (4.5) and (4.6) with Constraints (4.10), respectively.

Thus, $\{\tilde{X}, \tilde{Y}, \hat{\alpha}, \hat{\beta}\}$ satisfies Constraints (3.8) and (3.9). Therefore, $\{\tilde{X}, \tilde{Y}, \hat{\alpha}, \hat{\beta}\}$ satisfies all the constraints of Problem P1 and is a feasible solution of Problem P1. Hence, the lemma is true. ■

On the basis of the properties analyzed above, the next two sections propose effective algorithms for solving CAP and CPP$_i$.

4.5 Algorithm for solving Cluster Allocation Problem

It is noted from Chapter 3 and Section 4.2 that the integer programming model of CAP is simpler than that of Problem P1. However, results of computational experiments show that finding the optimal solution to CAP of realistic sizes by the commercial optimization software CPLEX can still take a long time for certain test problems. It is noted that when the size of the problem increases, the corresponding computational effort increases dramatically. Thus, an efficient algorithm is needed to solve CAP.

This section proposes an aggregation-disaggregation approach to solve CAP. In the aggregation step, vessel services calling at the same period for loading are grouped together to form a service group, the integer programming model of CAP is modified to find the optimal allocation of space to each service group in each yard block at each period and an optimal algorithm is developed for solving the allocation problem. With the optimal space allocation of service groups found, the disaggregation step then disaggregates the optimal space allocation to determine the optimal solution of CAP.

4.5.1 Finding optimal space allocation for service groups

The first step of the aggregation-disaggregation approach is to group vessel services calling at period π_m, $\pi_m \in \mathbf{T}$, together to form service group *m*. It is clear that the total number of service groups formed, *M*, is less than or equal to *T*. Let $\gamma_{m,t}$ be the storage space requirement of group *m* at period *t* and $G^i_{m,t}$ be the number of slots allocated to group *m* in block *i* at period *t*. It follows from the definition of $\gamma_{m,t}$,and $G^i_{m,t}$ that

$$\gamma_{m,t} = \sum_{j \in \mathbf{J} \& \Omega_j = \pi_m} R_{j,t} \quad m \in \mathbf{M}, t \in \mathbf{T} \tag{4.15}$$

$$G^i_{m,t} = \sum_{j \in \mathbf{J} \& \Omega_j = \pi_m} U^i_{j,t} \quad i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T} \tag{4.16}$$

It is clear from Expressions (4.15) that $G^i_{m,t}$ has the property of $U^i_{j,t}$ implied by Expressions (4.1) and (4.2), i.e., the number of slots allocated to service group *m* is non-decreasing until its loading period. Replacing $U^i_{j,t}$ by $G^i_{m,t}$ in Expressions (4.1) and (4.2) results in

$$G^i_{m,t-1} \le G^i_{m,t} \quad i \in \mathbf{I}, m \in \mathbf{M},\ t = 2,3,...,\pi_m, \pi_m + 2,...,T \qquad (4.17),$$

$$\rho_m G^i_{m,T} \le G^i_{m,1} \quad i \in \mathbf{I}, m \in \mathbf{M} \qquad (4.18).$$

It follows from Expressions (4.16) that for $m \in \mathbf{M}, t \in \mathbf{T}$,

$$\sum_{i=1}^{I} G^i_{m,t} = \sum_{i=1}^{I} \sum_{j \in J \& \Omega_j = \pi_m} U^i_{j,t} = \sum_{j \in J \& \Omega_j = \pi_m} R_{j,t} ,$$

and for $i \in \mathbf{I}, t \in \mathbf{T}$,

$$\sum_{m=1}^{M} G^i_{m,t} = \sum_{m=1}^{M} \sum_{j \in J \& \Omega_j = \pi_m} U^i_{j,t} = \sum_{j=1}^{J} U^i_{j,t} \le K .$$

Hence,

$$\sum_{i=1}^{I} G^i_{m,t} = \gamma_{m,t} \quad m \in \mathbf{M}, t \in \mathbf{T} \qquad (4.19),$$

$$\sum_{m=1}^{M} G^i_{m,t} \le K \quad i \in \mathbf{I}, t \in \mathbf{T} \qquad (4.20).$$

It is obvious from Expressions (4.5), (4.6), (4.16) that

$$G^{i}_{m,\pi_m} \geq \alpha_{\pi_m} \quad i \in \mathbf{I}, m \in \mathbf{M} \qquad (4.21),$$

$$G^{i}_{m,\pi_m} \leq \beta_{\pi_m} \quad i \in \mathbf{I}, m \in \mathbf{M} \qquad (4.22).$$

Hence, the problem of finding the optimal space allocation to service groups, SGP, can be stated as follows:

SGP

Minimize $\sum_{m=1}^{M}(\beta_{\pi_m} - \alpha_{\pi_m})$ subject to Expressions (4.17)-(4.22).

Compared with CAP, the above model is much simpler as there is only a small number of groups. However, there are still thousands of integer variables in the model and CPLEX may take a long time to find the optimal solution. Thus, an efficient algorithm is proposed to solve SGP.

To develop an efficient solution algorithm, the properties of SGP need to be analyzed in detail. It is noted from the objective function of SGP that in the optimal solution of SGP, the minimum deviation of workload among all the yard blocks at any period is zero if the number of loading slots can be evenly allocated to each block, i.e., if the number of loading slots is divisible by the number of yard blocks; otherwise, the minimum deviation is 1. On the basis of this property, an optimality condition for SGP is given in the lemma below.

Lemma 4.3

Suppose that $\{\widetilde{\mathbf{G}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}}\}$ is a feasible solution of SGP where $\widetilde{\mathbf{G}}$ = the set of $\widetilde{G}^{i}_{m,t}$, $\widetilde{\boldsymbol{\alpha}}$= the set of $\widetilde{\alpha}_{\pi_m}$ and $\widetilde{\boldsymbol{\beta}}$= the set of $\widetilde{\beta}_{\pi_m}$. If $\{\widetilde{\mathbf{G}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}}\}$ satisfies the expression that $\widetilde{\beta}_{\pi_m} - \widetilde{\alpha}_{\pi_m} \leq 1$ for $m \in \mathbf{M}$, it is the optimal solution of SGP.

Proof:

The lemma is proved by contradiction. Suppose there exists a feasible but non-optimal solution of SGP, $\{\widetilde{\mathbf{G}},\widetilde{\boldsymbol{\alpha}},\widetilde{\boldsymbol{\beta}}\}$, that satisfies $\widetilde{\beta}_{\pi_m}-\widetilde{\alpha}_{\pi_m}\le 1$ for $m\in\mathbf{M}$. Let $\widetilde{D}$ be the objective function value of solution $\{\widetilde{\mathbf{G}},\widetilde{\boldsymbol{\alpha}},\widetilde{\boldsymbol{\beta}}\}$ and $\widetilde{d}_m$ be $\widetilde{\beta}_{\pi_m}-\widetilde{\alpha}_{\pi_m}$. Hence,

$$\widetilde{D}=\sum_{m=1}^{M}\widetilde{d}_m\ . \tag{4.23}$$

Denote the optimal objective function value of SGP by $\hat{D}$ and $\hat{\beta}_{\pi_m}-\hat{\alpha}_{\pi_m}$ by $\hat{d}_m$. Thus,

$$\hat{D}=\sum_{m=1}^{M}\hat{d}_m\ . \tag{4.24}$$

In view of Expressions (4.22) and (4.23) and the fact that $\widetilde{D}>\hat{D}$, there must exist m', $m'\in\mathbf{M}$, such that $\widetilde{d}_{m'}>\hat{d}_{m'}$.

Since $\widetilde{\beta}_{\pi_m}-\widetilde{\alpha}_{\pi_m}\le 1$ for $m\in\mathbf{M}$, it is clear from Expressions (4.21) and (4.22) that $\widetilde{d}_{m'}\le 1$ and $\widetilde{d}_{m'}$ is nonnegative integer. Thus, $\widetilde{d}_{m'}$=1 and $\hat{d}_{m'}$=0. It is obvious that $\hat{d}_{m'}$=0 when all the yard blocks have the same workload at loading period of group $\pi_{m'}$. Hence, $\gamma_{m,\pi_{m'}}$ must be divisible by *I* and each block has a workload of $\gamma_{m,\pi_{m'}}/I$. For $\widetilde{d}_{m'}$=1, there must exist $\widetilde{i},\widetilde{i}\in\mathbf{I}$ such that $\widetilde{G}^{\widetilde{i}}_{m',\pi_{m'}}\ne\gamma_{m',\pi_{m'}}/I$. Without loss of generality, suppose that $\widetilde{G}^{\widetilde{i}}_{m',\pi_{m'}}<\gamma_{m',\pi_{m'}}/I$. Since $\gamma_{m',\pi_{m'}}/I$ is integer and $\{\widetilde{\mathbf{G}},\widetilde{\boldsymbol{\alpha}},\widetilde{\boldsymbol{\beta}}\}$ satisfies $\widetilde{\beta}_{\pi_m}-\widetilde{\alpha}_{\pi_m}\le 1$, it is clear that $\widetilde{G}^{\widetilde{i}}_{m',\pi_{m'}}\le\gamma_{m',\pi_{m'}}/I$ -1, $\widetilde{i}\in\mathbf{I}$ and $\widetilde{G}^{i}_{m',\pi_{m'}}\le\gamma_{m',\pi_{m'}}/I$, $i\in\mathbf{I},i\ne\widetilde{i}$. Therefore,

$$\sum_{i=1}^{I}\widetilde{G}^{i}_{m',\pi_{m'}}=\widetilde{G}^{\widetilde{i}}_{m',\pi_{m'}}+\sum_{i\in I\&i\ne\widetilde{i}}\widetilde{G}^{i}_{m',\pi_{m'}}<\gamma_{m',\pi_{m'}}/I+\sum_{i\in I\&i\ne\widetilde{i}}\widetilde{G}^{i}_{m',\pi_{m'}}<\gamma_{m',\pi_{m'}}\ .$$

The above expression violates Expressions (4.19) and thus, $\{\widetilde{\mathbf{G}},\widetilde{\boldsymbol{\alpha}},\widetilde{\boldsymbol{\beta}}\}$ is not a feasible solution of SGP. Hence, both $\tilde{d}_{m'}$=1 and $\hat{d}_{m'}$=0 cannot be true at the same time. Thus, for all $m, m \in M$, it is clear that $\tilde{d}_m \le \hat{d}_m$; hence, $\tilde{D} = \sum_{m=1}^{M} \tilde{d}_m \le \hat{D} = \sum_{m=1}^{M} \hat{d}_m$. Therefore, $\tilde{D}$ is the optimal objective function value of SGP and the lemma is true. ■

The optimality condition given in the above lemma can be stated in a way involving variables G^i_{m,π_m} , $i \in \mathbf{I}$ and $m \in \mathbf{M}$, only. Let $\lfloor \gamma_{m,t} / I \rfloor$ be the largest integer number which is smaller than $\gamma_{m,t} / I$ and $\lceil \gamma_{m,t} / I \rceil$ be the smallest integer number which is greater than $\gamma_{m,t} / I$. The optimality condition given in Lemma 4.3 can be restated as follows:

Lemma 4.4

Suppose that $\{\tilde{\mathbf{G}}, \tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}\}$ is a feasible solution of problem SGP. If $\tilde{\mathbf{G}}$ satisfies the expression that

$$\lfloor \gamma_{m,\pi_m} / I \rfloor \le \tilde{G}^i_{m,\pi_m} \le \lceil \gamma_{m,\pi_m} / I \rceil \qquad (4.25)$$

for all $i \in \mathbf{I}$ and $m \in \mathbf{M}$, it is an optimal solution of SGP.

Proof:

Since $\{\tilde{\mathbf{G}}, \tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}\}$ satisfies all the constraints of SGP and the expression that $\lfloor \gamma_{m,\pi_m} / I \rfloor \le \tilde{G}^i_{m,t} \le \lceil \gamma_{m,\pi_m} / I \rceil$, it can be shown that for $i \in \mathbf{I}$ and $m \in \mathbf{M}$,

$$\lfloor \gamma_{m,\pi_m} / I \rfloor \le \tilde{\alpha}_{\pi_m} = \min_{i \in I} \{\tilde{G}^i_{m,\pi_m}\} \le \tilde{G}^i_{m,\pi_m} \le \max_{i \in I} \{\tilde{G}^i_{m,\pi_m}\} = \tilde{\beta}_{\pi_m} \le \lceil \gamma_{m,\pi_m} / I \rceil.$$

It is obvious from the definition of $\lfloor \gamma_{m,t} / I \rfloor$ and $\lceil \gamma_{m,t} / I \rceil$ that $\lceil \gamma_{m,t} / I \rceil - \lfloor \gamma_{m,t} / I \rfloor \le 1$. Hence, it can be shown from the above that

$$\tilde{\beta}_{\pi_m} - \tilde{\alpha}_{\pi_m} \le \tilde{\beta}_{\pi_m} - \lfloor \gamma_{m,\pi_m} / I \rfloor \le \lceil \gamma_{m,\pi_m} / I \rceil - \lfloor \gamma_{m,\pi_m} / I \rfloor \le 1 .$$

It follows from Lemma 4.3 that $\{\tilde{\mathbf{G}}, \tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}\}$ is the optimal solution of SGP if it satisfies the expression that $\tilde{\beta}_{\pi_m} - \tilde{\alpha}_{\pi_m} \leq 1$. Thus, the lemma is proved. ■

The optimality condition given in the above lemma can be used to modify the integer programming model of SGP so that an efficient algorithm can be developed to find the optimal solution of SGP. Consider Expressions (4.18) and (4.19). Since $\sum_{m=1}^{M} \gamma_{m,t} = \sum_{m=1}^{M}\sum_{i=1}^{I} G_{m,t}^{i} = \sum_{i=1}^{I}\sum_{m=1}^{M} G_{m,t}^{i} \leq \sum_{i=1}^{I} K \leq I \cdot K$, it is clear that no feasible solutions exist when $I \cdot K < \sum_{m=1}^{M} \gamma_{m,t}$ for any $t \in T$. Thus, in the subsequent analysis, one only needs to consider the case that $I \cdot K \geq \sum_{m=1}^{M} \gamma_{m,t}$ for all $t \in \mathbf{T}$.

When analyzing the integer programming model of SGP in a greater detail, it is noted that $\{\hat{\mathbf{G}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}\}$, the optimal solution of SGP, consists of three partial optimal solutions, namely, $\hat{\mathbf{G}}$, $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$. For a given $\hat{\mathbf{G}}$, it is straightforward to determine the other two partial optimal solutions using Expressions (4.21) and (4.22); hence, these two expressions need not be considered when finding the optimal solution of SGP if there exists a condition that the partial optimal solution $\hat{\mathbf{G}}$ must satisfy. Such an optimality condition on $\hat{\mathbf{G}}$ is actually given in Lemma 4.4. Thus, with the optimality condition given in Lemma 4.4, the problem of finding $\hat{\mathbf{G}}$ is to determine $\mathbf{G}$, where $\mathbf{G}$ = the set of $G_{m,t}^{i}$, that satisfies Expressions (4.17)-(4.20) and (4.25).

Since Expressions (4.17), (4.18) and (4.25) are simple upper bound and lower bound constraints on $\mathbf{G}$, these expressions can be combined and rewritten as follows: for $i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}$

$$G_{m,t}^{i} \leq \lceil \gamma_{m,\pi_m} / I \rceil \tag{4.26}$$

$$G^i_{m,t} \geq \max\{\theta_{m,t} \cdot G^i_{m,t-1}, \mu_{m,t} \cdot \lfloor \gamma_{m,\pi_m} / I \rfloor\} \qquad (4.27)$$

where

$$\theta_{m,t} = \begin{cases} 0 & \text{if } \pi_m = t-1 \\ 1 & \text{otherwise} \end{cases},$$

$$\mu_{m,t} = \begin{cases} 0 & \text{if } \pi_m \neq t \\ 1 & \text{otherwise} \end{cases}$$

and

$t-1$ is defined as T when $t=1$,

$G^i_{m,0}$ is defined as $G^i_{m,T}$.

After combining Expressions (4.17), (4.18) and (4.25) into Expressions (4.26) and (4.27), the problem of finding $\hat{\mathbf{G}}$ becomes determining $\mathbf{G}$ that satisfies Expressions (4.19), (4.20), (4.26) and (4.27). Hence, as stated in the lemma below, any $\mathbf{G}$ satisfying Expressions (4.19), (4.20), (4.26) and (4.27) is a partial optimal solution of SGP.

Lemma 4.5

If $\mathbf{G}$ satisfies Expressions (4.19), (4.20), (4.26) and (4.27), $\mathbf{G}$ is a partial optimal solution of SGP.

Proof:

Suppose that $\mathbf{G}$ satisfies Expressions (4.19), (4.20), (4.26) and (4.27). It follows from $\mathbf{G}$ satisfies Expressions (4.27) that

$G^i_{m,t} \geq \theta_{m,t} \cdot G^i_{m,t-1}$ for $t \neq \pi_m$

and

$G^i_{\pi,\pi_m} \geq \max\{\theta_{m,\pi_m} \cdot G^i_{m,\pi_m-1}, \lfloor \gamma_{m,\pi_m} / I \rfloor\} \geq \theta_{m,\pi_m} \cdot G^i_{m,\pi_m-1}$.

According to the definition of $\theta_{m,t}$ and $G^i_{m,0}$, the above expressions can be expressed as

$G^i_{m,t} \geq G^i_{m,t-1}$ for $i \in I, m \in M, t = 1,2,...,\pi_m, \pi_m + 2,...,T$ and

$G^i_{m,1} \geq G^i_{m,T}$ for $i \in \mathbf{I}, m \in \mathbf{M}$ if T is not the loading period for group m.

Hence, the above two expressions give the identical set of constraints defined by Expressions (4.17) and (4.18).

It is clear that there always exist α_{π_m} and β_{π_m} such that $\alpha_{\pi_m} = \min_{i \in \mathbf{I}} \{G^i_{m,\pi_m}\} \leq G^i_{m,\pi_m} \leq \max_{i \in \mathbf{I}} \{G^i_{m,\pi_m}\} = \beta_{\pi_m}$ for $i \in \mathbf{I}, m \in \mathbf{M}$. Thus, $\mathbf{G}$ is partial feasible solution of SGP.

Since $\mathbf{G}$ satisfies Expressions (4.26) and (4.27), it can easily be deduced from Lemma 4.4 that $\mathbf{G}$ is a partial optimal solution of SGP. ■

It is clear that Expressions (4.19) and (4.20) are the simple supply and demand constraints of a typical transportation problem when the number of slots available in yard block i is modeled as the availability of source i at period t and the number of slots required by service group m at period t is modeled as the requirement of demand centre m at period t. It is noted that the problem of finding $\mathbf{G}$ that satisfies Expressions (4.19), (4.20), (4.26) and (4.27) is the same as finding feasible solutions of some typical transportation problems with bounded variables (Murty 1983) except that the lower bounds given in Expressions (4.27) are interdependent.

If Expressions (4.27) is ignored, the problem of finding feasible $\mathbf{G}$ can be decomposed into T separate transportation problems with bounded variables which can easily be solved by Murty's method. This research proposes an algorithm that finds feasible solutions to transportation problems with bounded variables for each t while at the same time ensure the feasible solution found for each of transportation problems satisfies Expressions (4.27). For period t, the value of $G^i_{m,t}$ will be determined by the method (Murty 1983) where $G^i_{m,t}$ is a feasible solution of the transportation problem associated with period t. Without loss of generality, it is assumed that the algorithm starts with $t = 1$. The algorithm initialize $G^i_{m,0}$ to 0 in the first iteration since no space is allocated yet. The main idea of the algorithm is outlined below.

First, find a feasible solution by the method (Murty 1983) from the following expressions which are obtained by setting t = 1 in Expressions (4.19), (4.20), (4.26) and (4.27):

$$\sum_{i=1}^{I} G_{m,1}^{i} = \gamma_{m,1} \quad m \in \mathbf{M},$$

$$\sum_{m=1}^{M} G_{m,1}^{i} \le K \quad i \in \mathbf{I},$$

$$G_{m,1}^{i} \le \lceil \gamma_{m,\pi_m} / I \rceil,$$

and

$$G_{m,1}^{i} \ge \mu_{m,1} \cdot \lfloor \gamma_{m,\pi_m} / I \rfloor.$$

After finding the feasible $G_{m,1}^{i}$, the value of. $G_{m,1}^{i}$ is a lower bound for determining $G_{m,2}^{i}$, i.e. $G_{m,2}^{i} \ge \max\{\theta_{m,2} \cdot G_{m,1}^{i}, \mu_{m,2} \cdot \lfloor \gamma_{m,\pi_m} / I \rfloor\}$, and then $G_{m,2}^{i}$ is a lower bound for $G_{m,3}^{i}$. Thus, the above procedure is repeated to find $G_{m,t}^{i}$ by setting t = 2, 3, ..., T in Expressions (4.19), (4.20), (4.26) and (4.27).

It is noted from the discussion that feasible **G** should be found within T steps. However, finding $G_{m,t}^{i}$ sequentially may lead some infeasible cases and those cases often happen when storage space allocation to blocks is uneven. Consider the case that storage space is not evenly allocated to $G_{m,1}^{i}$ in different blocks as illustrated by the following example.

$\gamma_{m,t}$							
m	t						
	1	2	3	4	5	6	7
1	78	78	78	0	0	58	60
2	14	14	14	15	9	14	14
3	21	24	0	0	0	5	6

Table 4.4 Space requirements of 3 groups

In the example there are 3 yard blocks serving 3 vessel service groups with storage space requirement shown in Table 4.4 and $\pi_1 = 3$, $\pi_2 = 4$ and $\pi_3 = 2$. $G^i_{m,1}$ found by Murty's algorithm is given in Table 4.5. It can be deduced from Expressions (4.26) and (4.27) that each block must be allocated 8 slots for group 3, i.e. $G^i_{3,2} = 8$ for all $i \in \mathbf{I}$. However, for the case that i = 1 and t = 2, the allocation violates Expressions (4.20) and (4.27) as $K - \sum_{m=1}^{M} \max\{\theta_{m,t} \cdot G^i_{m,t-1}, \mu_{m,t} \cdot \lfloor \gamma_{m,\pi_m} / I \rfloor\} = 40 - (28 + 5 + 8) < 0$.

$G^i_{m,t}$									
t	i =1			i =2			i =3		
	m=1	m=2	m=3	m =1	m=2	m=3	m=1	m=2	m=3
1	28	5	7	25	5	7	25	4	7
2	28	5	8	25	5	8	25	4	8

Table 4.5 An infeasible case of solutions

The example shows an infeasible case caused by uneven storage space allocation among yard blocks as the value of $G^1_{1,1}$ is much greater than that of $G^2_{1,1}$ and $G^3_{1,1}$. By testing more numerical examples, it is noted that there is a higher possibility of having such infeasible cases when storage space is not evenly allocated to blocks. To reduce the likelihood of having infeasible cases, the following expressions are added to the transportation problems, for $i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}$

$$G^i_{m,t} \leq \lceil \gamma_{m,t} / I \rceil \tag{4.28}$$

$$G^i_{m,t} \geq \max\{\theta_{m,t} \cdot G^i_{m,t-1}, \lfloor \gamma_{m,t} / I \rfloor\} \tag{4.29}$$.

It is noted that $G^i_{m,t} \leq \lceil \gamma_{m,t} / I \rceil \leq \lceil \gamma_{m,\pi_m} / I \rceil$ and $G^i_{m,t} \geq \max\{\theta_{m,t} \cdot G^i_{m,t-1}, \lfloor \gamma_{m,t} / I \rfloor\}$ $\geq \max\{\theta_{m,t} \cdot G^i_{m,t-1}, \mu_{m,t} \cdot \lfloor \gamma_{m,\pi_m} / I \rfloor\}$. Hence, if **G** satisfies Expressions (4.28) and (4.29), it also satisfies Expressions (4.26) and (4.27). Moreover, the lower bound and upper bound given by Expressions (4.28) and (4.29) achieve a better balance of workload among blocks for $t \in \mathbf{T}$ which satisfy the optimality condition (4.25). Thus, the problem of finding **G** that satisfies Expressions (4.19), (4.20), (4.26) and (4.27) becomes the problem of finding **G** that satisfies Expressions (4.19),(4.20),(4.28) and (4.29).

It is noted from Expressions (4.28) and (4.29) that for $i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}$, $\lfloor \gamma_{m,t} / I \rfloor \leq G^i_{m,t} \leq \lceil \gamma_{m,t} / I \rceil$. Since $\lceil \gamma_{m,t} / I \rceil - \lfloor \gamma_{m,t} / I \rfloor \leq 1$, it can be shown that $\max_m\{G^i_{m,t}\} - \min_m\{G^i_{m,t}\} \leq 1$. Thus, $G^i_{m,t}$ that satisfies Expressions (4.19), (4.20), (4.28) and (4.29) heads to a better balance of workload. To design an effective and efficient algorithm for finding **G**, some modifications to the transportation problems need to be made and the details are discussed below.

Let $H^i_{m,t}$ be the difference between $G^i_{m,t}$ and $\lfloor \gamma_{m,t} / I \rfloor$ i.e.,

$$H^i_{m,t} = G^i_{m,t} - \lfloor \gamma_{m,t} / I \rfloor \quad i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}. \tag{4.30}$$

It follows from Expressions (4.29) and (4.30) that $H^i_{m,t} \geq 0$.

Rearranging the terms in Expressions (4.30) gives

$$G^i_{m,t} = H^i_{m,t} + \lfloor \gamma_{m,t} / I \rfloor \quad i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}.$$

It can be shown by replacing $G^i_{m,t}$ with $H^i_{m,t} + \lfloor \gamma_{m,t} / I \rfloor$ in Expressions (4.19) and

(4.20) that for $t \in \mathbf{T}$,

$$\sum_{i=1}^{I} H_{m,t}^{i} = \gamma_{m,t} - \lfloor \gamma_{m,t}/I \rfloor \cdot I \quad m \in \mathbf{M}, \tag{4.31}$$

and

$$\sum_{m=1}^{M} H_{m,t}^{i} \le K - \sum_{m=1}^{M} \lfloor \gamma_{m,t}/I \rfloor \quad i \in \mathbf{I}. \tag{4.32}$$

It can be deduced from Expressions (4.28) and (4.30) that

$$H_{m,t}^{i} \le \lceil \gamma_{m,t}/I \rceil - \lfloor \gamma_{m,t}/I \rfloor \le 1 \quad i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}. \tag{4.33}$$

Thus, $H_{m,t}^{i}$ $i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}$ are 0-1 integer variables.

To manage the relations of $H_{m,t}^{i}$, the properties in Expressions (4.29) need to be considered in more detail. It follows from Expressions (4.29) by replacing $G_{m,t}^{i}$ with $H_{m,t}^{i} + \lfloor \gamma_{m,t}/I \rfloor$ and $G_{m,t-1}^{i}$ with $H_{m,t-1}^{i} + \lfloor \gamma_{m,t-1}/I \rfloor$ that for $i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}$,

$$H_{m,t}^{i} + \lfloor \gamma_{m,t}/I \rfloor \ge \max\{\theta_{m,t} \cdot (H_{m,t-1}^{i} + \lfloor \gamma_{m,t-1}/I \rfloor), \lfloor \gamma_{m,t}/I \rfloor\}. \tag{4.34}$$

It is clear from Expressions (4.15) and the definition of $R_{j,t}$ that $\gamma_{m,t} \ge \theta_{m,t} \cdot \gamma_{m,t-1}$ such that

$$\lfloor \gamma_{m,t}/I \rfloor \ge \theta_{m,t} \cdot \lfloor \gamma_{m,t-1}/I \rfloor \quad \text{for} \quad m \in \mathbf{M}, t \in \mathbf{T}. \tag{4.35}$$

It can be shown from Expressions (4.34) and (4.35) that $H_{m,t}^{i} \ge H_{m,t-1}^{i}$ if $\lfloor \gamma_{m,t}/I \rfloor = \lfloor \gamma_{m,t-1}/I \rfloor$ and *t*-1 is not the loading period of group *m*.

Define $\eta_{m,t} = 1$ when $\lfloor \gamma_{m,t}/I \rfloor = \lfloor \gamma_{m,t-1}/I \rfloor$ and *t*-1 is not the loading period of group *m*; otherwise, $\eta_{m,t} = 0$. Then for $i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}$,

$$H_{m,t}^{i} \ge \eta_{m,t} \cdot H_{m,t-1}^{i} \tag{4.36}$$

Where $H_{m,0}^{i}$ is defined as $H_{m,T}^{i}$.

The problem of finding $\mathbf{G}$ that satisfies Expressions (4.19), (4.20), (4.26) and (4.27) is transformed as the problem of finding $\mathbf{G}$ and $\mathbf{H}$, where $\mathbf{H}$ = the set of $H_{m,t}^{i}$, that satisfies Expressions (4.30)-(4.33) and (4.36). The lemma below gives a property of $\mathbf{G}$ and $\mathbf{H}$ which is critical to the development of the algorithm proposed in this section for solving SGP.

Lemma 4.6

If $\{\mathbf{G},\mathbf{H}\}$ satisfies Expressions (4.30)-(4.33) and (4.36) for $t \in \mathbf{T}$, $\mathbf{G}$ is a partial optimal solution of SGP.

Proof:

Suppose that $\{\mathbf{G},\mathbf{H}\}$ satisfies Expressions (4.30)-(4.33) and (4.36) $t \in \mathbf{T}$. It follows from Expressions (4.30) that

$$H^i_{m,t} = G^i_{m,t} - \lfloor \gamma_{m,t}/I \rfloor \quad i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}.$$

It can be shown from the above expression and Expressions (4.31) that for $m \in \mathbf{M}, t \in \mathbf{T}$,

$$\sum_{i=1}^{I} G^i_{m,t} - \sum_{i=1}^{I} \lfloor \gamma_{m,t}/I \rfloor = \gamma_{m,t} - \sum_{i=1}^{I} \lfloor \gamma_{m,t}/I \rfloor.$$

Hence, $\sum_{i=1}^{I} G^i_{m,t} = \gamma_{m,t} \quad m \in \mathbf{M}, t \in \mathbf{T}$; hence, $\mathbf{G}$ satisfies Expressions (4.19).

It can be shown from Expressions (4.30) and (4.32) that for $i \in \mathbf{I}, t \in \mathbf{T}$

$$\sum_{m=1}^{M} G^i_{m,t} - \sum_{m=1}^{M} \lfloor \gamma_{m,t}/I \rfloor \le K - \sum_{m=1}^{M} \lfloor \gamma_{m,t}/I \rfloor.$$

Thus, $\mathbf{G}$ satisfies Expressions (4.20).

Next, it is clear from Expressions (4.30) and (4.33) that for $i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}$ $G^i_{m,t} \le \lceil \gamma_{m,t}/I \rceil \le \lceil \gamma_{m,\pi_m}/I \rceil$. Hence, $\mathbf{G}$ satisfies Expressions (4.25).

Since $H^i_{m,t}$ is non-negative, it is obvious that $G^i_{m,t} = H^i_{m,t} + \lfloor \gamma_{m,t}/I \rfloor \ge \lfloor \gamma_{m,t}/I \rfloor$.

According to the definition of $\eta_{m,t}$, it follows from Expressions (4.36) that $H^i_{m,t} \ge H^i_{m,t-1}$ when $\lfloor \gamma_{m,t}/I \rfloor = \lfloor \gamma_{m,t-1}/I \rfloor$ and t-1 is not the loading day of group m.

It is clear that Expressions (4.35) can always be satisfied such that

$G^i_{m,t} = H^i_{m,t} + \lfloor \gamma_{m,t}/I \rfloor \geq \theta_{m,t} \cdot (H^i_{m,t-1} + \lfloor \gamma_{m,t-1}/I \rfloor)$ when $\lfloor \gamma_{m,t}/I \rfloor = \lfloor \gamma_{m,t-1}/I \rfloor$.

Considering that $H^i_{m,t}$ are 0-1 integer variables, the expression is satisfied that

$$G^i_{m,t} = H^i_{m,t} + \lfloor \gamma_{m,t}/I \rfloor$$

$$\geq \max\{\theta_{m,t} \cdot (H^i_{m,t-1} + \lfloor \gamma_{m,t-1}/I \rfloor), \lfloor \gamma_{m,t}/I \rfloor\} = \max\{\theta_{m,t} \cdot G^i_{m,t-1}, \lfloor \gamma_{m,t}/I \rfloor\}.$$

Noted that $G^i_{m,t} \geq \max\{\theta_{m,t} \cdot G^i_{m,t-1}, \lfloor \gamma_{m,t}/I \rfloor\} \geq \max\{\theta_{m,t} \cdot G^i_{m,t-1}, \mu_{m,t} \cdot \lfloor \gamma_{m,\pi_m}/I \rfloor\}$, **G** also satisfies Expressions (4.27). Therefore, **G** satisfies Expressions (4.19), (4.20), (4.26) and (4.27) and is a partial optimal solution of SGP by Lemma 4.5. ■

On the basis of the above analysis, an algorithm can be given for finding **G** to Expressions (4.30)-(4.33) and (4.36) that Expressions (4.31) and (4.32) can be considered as the respective demand and supply constraints of a transportation problem while Expressions (4.33) and (4.36) give the lower bound and upper bound for variable $H^i_{m,t}$. Thus, for given $H^i_{m,t-1}$, finding $H^i_{m,t}$ that satisfies Expressions (4.30)-(4.33) and (4.36) is equivalent to finding a feasible solution to a transportation problem with bounded 0-1 integer variables.

Another consideration should be put on the start point of *t* for the solution procedure. A common sense of space allocation in a cycle with space limits is that the time period with the largest workload should be considered first. Normally if one can satisfy the largest space requirement with even workloads, the space capacity should also satisfy space requirements of other periods. Suppose that t_0 is the day with the maximum storage space requirement that $\sum_{m=1}^{M} \gamma_{m,t_0} = \max_{t \in T}(\sum_{m=1}^{M} \gamma_{m,t})$. The algorithm starts the computation from time period t_0. For period *t*, $t = t_0, t_0+1, ..., T, 1, ..., t_0 - 1$, the value of $H^i_{m,t}$ is determined by the method (Murty, 1983). Since the objective of the yard planning problem is to balance workload, the initial value of H^i_{m,t_0-1} is set to

0. This value will be updated in the last iteration step $t = t_0 - 1$.

Then, the details of the algorithm are shown as follows:

Algorithm GP

Step 0 Set $t = t_0$. Initialize the value of H^i_{m,t_0-1} to 0 for $i \in \mathbf{I}, m \in \mathbf{M}$.

Step 1 Find a feasible solution to Expressions (4.31)-(4.33) and (4.36) by the method (Murty 1983) and determine $G^i_{m,t}$ using Expressions (4.30).

Step 2 If $t = t_0 - 1$, terminate the algorithm;

Otherwise, set t= (t+1) mod T, go to Step1.

4.5.2 Finding optimal space allocation for services

This section discusses the method developed in this thesis to find the optimal $U^i_{j,t}$ for CAP based on the solution of SGP, $G^i_{m,t}$, found by Algorithm GP presented in the previous section. The proposed method aims at disaggregating the allocated slots in $G^i_{m,t}$ into slots assigned to each vessel service within each group that needs to analyze the properties of SGP and CAP first as follows.

Lemma 4.7

The optimal solution of CAP is a feasible solution of SGP.

Proof:

Suppose that $\{\hat{U}, \hat{\alpha}, \hat{\beta}\}$ is the optimal solution of Problem CAP. It is clear that $\{\hat{U}, \hat{\alpha}, \hat{\beta}\}$ satisfies all constraints of SGP by repeating the procedure of defining SGP.

Hence, the lemma is true. ■

As the objective functions of SGP and CAP are, in effect, identical, it is clear from Lemma 4.7 that the following corollary is true.

Corollary 4.2

The optimal objective function value of Problem CAP is greater than or equal to the optimal objective function value of SGP.

On the basis of the above properties, once an optimal **G** of SGP is found, the problem of solving CAP can be transformed into the problem of finding **U,** where **U** is the set of $U^i_{j,t}$, that satisfies Expressions (4.1)-(4.6) and (4.16).

It can be shown from the formulation of SGP and CAP that the partial solution **U** satisfies Expressions (4.4) – (4.6) if {**G**,**U**} satisfies Expressions (4.16) and **G** satisfies Expressions (4.20) – (4.22). Thus, for a given **G**, an optimal solution of SGP, it follows from Lemma 4.7 and Corollary 4.2, finding the optimal of CAP can be simplified as finding **U** that satisfies Expressions (4.1)-(4.3) and (4.16). The rest of the section discusses the algorithm proposed for finding the optimal **U.**

With Expressions (4.3) considered as the demand constraints and Expressions (4.16) as the supply constraints, it is noted that the problem of finding a feasible solution to Expressions (4.1)-(4.3) and (4.16) can be modeled as the problem of finding a feasible solution to a transportation problem except that the variables are interrelated. It is shown from Expressions (4.3) and (4.16) that $\sum_{j\in\mathbf{J}\&\Omega_j=\pi_m} R_{j,t} = \sum_{j\in\mathbf{J}\&\Omega_j=\pi_m}\sum_{i=1}^{I} U^i_{j,t} = \sum_{i=1}^{I}\sum_{j\in\mathbf{J}\&\Omega_j=\pi_m} U^i_{j,t} = \sum_{i=1}^{I} G^i_{m,t} \quad m\in\mathbf{M}, t\in\mathbf{T}$; thus, the demand and supply are balanced. Since $U^i_{j,t}$ are interrelated for $t\in\mathbf{T}$, a variable is needed to address this complexity. Let $A^i_{j,t}$ be the increment of slots allocated to service *j* in block *i* at period *t*, i.e., for $i\in\mathbf{I}, j\in\mathbf{J}$,

$$A^i_{j,t} = \begin{cases} U^i_{j,t} - U^i_{j,t-1} & t = 1,2,...,\Omega_j, \Omega_j + 2,...,T \\ U^i_{j,t} & t = \Omega_j + 1 \end{cases} \tag{4.37}$$

where $A^i_{j,t} \geq 0$ and $U^i_{j,0} = U^i_{j,T}$.

It follows from Expressions (4.3)， (4.16) and (4.37) that for $j \in \mathbf{J}$,

$$\sum_{i=1}^{I} A^i_{j,t} = \begin{cases} R_{j,t} - R_{j,t-1}, & t = 1,2,...\Omega_j, \Omega_j + 2,...,T \\ R_{j,t}, & t = \Omega_j + 1 \end{cases} \tag{4.38}$$

and for $i \in \mathbf{I}, m \in \mathbf{M}$

$$\sum_{j \in \mathrm{J} \& \Omega_j = \pi_m} A^i_{j,t} = \begin{cases} G^i_{m,t} - G^i_{m,t-1}, & t = 1,2,...,\Omega_j, \Omega_j + 2,...,T \\ G^i_{m,t}, & t = \Omega_j + 1 \end{cases} \tag{4.39}$$

where $R_{j,0}$ is defined as $R_{j,T}$.

With variable A defined above, the problem of finding **U** that satisfies Expressions (4.1)-(4.3) and (4.16) is transformed to the problem of finding {**U,A**}, where **A** is the set of $A^i_{j,t}$, that satisfies Expressions (4.16) and Expressions (4.37)-(4.39). A property of **G**, **U** and **A** critical to design the algorithm for solving CAP is discussed in the following lemma.

Lemma 4.8

Suppose that **G** is a feasible solution of SGP and {**U,A**} satisfies (4.37) – (4.39), then **U** is a partial feasible solution of CAP.

Proof:

G is a feasible solution of SGP, then **G** satisfies Expressions (4.17)-(4.22). Suppose that {**U,A**} satisfies Expressions (4.37) – (4.39).

For $t = \Omega_j + 1$, it is shown from Expressions (4.37) – (4.39) that $\sum_{i=1}^{I} U^i_{j,t} = R_{j,t}$ and $\sum_{j \in \mathrm{J} \& \Omega_j = \pi_m} U^i_{j,t} = G^i_{m,t}$.

For t =1,2,… Ω_j, Ω_j +2,…,*T*, it follows from Expressions (4.37) – (4.39) that

$\sum_{i=1}^{I} U^i_{j,t} - \sum_{i=1}^{I} U^i_{j,t-1} = R_{j,t} - R_{j,t-1}$ and $\sum_{j\in J\&\Omega_j=\pi_m} U^i_{j,t} - \sum_{j\in J\&\Omega_j=\pi_m} U^i_{j,t-1} = G^i_{m,t} - G^i_{m,t-1}$.

Since $\sum_{i=1}^{I} U^i_{j,\Omega_j+1} = R_{j,\Omega_j+1}$, it follows from $\sum_{i=1}^{I} U^i_{j,t} - \sum_{i=1}^{I} U^i_{j,t-1} = R_{j,t} - R_{j,t-1}$ that

$\sum_{i=1}^{I} U^i_{j,\Omega_j+2} = R_{j,\Omega_j+2}$ when $t = \Omega_j + 2$.

Repeating the above argument, it can easily be shown that the expression $\sum_{i=1}^{I} U^i_{j,t} =$

$R_{j,t}$ is also satisfied for $t = \Omega_j + 3, \ldots, T, 1, 2, \ldots \Omega_j$. Hence, **U** satisfies Expressions (4.3).

Similarly, it can be proved that **U** satisfies Expressions (4.16) as $\sum_{j\in J\&\Omega_j=\pi_m} U^i_{j,t} = G^i_{m,t}$ for $i \in \mathbf{I}, m \in \mathbf{M}, t \in \mathbf{T}$.

It can be shown from Expressions (4.16) and (4.20) that for $i \in \mathbf{I}, t \in \mathbf{T}$, $\sum_{m=1}^{M} G^i_{m,t} = \sum_{m=1}^{M} \sum_{j\in J\&\Omega_j=\pi_m} U^i_{j,t} = \sum_{j=1}^{J} U^i_{j,t} \le K$. Hence, Expressions (4.4) are satisfied.

It follows from the definition of $A^i_{j,t}$ and the fact that $U^i_{j,t} - U^i_{j,t-1} = A^i_{j,t} \ge 0$ that

$$U^i_{j,t-1} \le U^i_{j,t} \text{ for } i \in \mathbf{I}, j \in \mathbf{J}, t = 2,3,\ldots,\Omega_j, \Omega_j + 2, \ldots, T$$

and

$$\rho_j U^i_{j,T} \le U^i_{j,1} \text{ for } i \in \mathbf{I}, j \in \mathbf{J}.$$

Therefore, Expressions (4.1) and (4.2) are satisfied.

Obviously, Expressions (4.5) and (4.6) can always be satisfied by **U** since $\alpha_t = \min_{i\in\mathbf{I}}\{\sum_{j\in\mathbf{J},\Omega_j=t} \tilde{U}^i_{j,t}\} \le \sum_{j\in\mathbf{J},\Omega_j=t} \tilde{U}^i_{j,t} \le \max_{i\in\mathbf{I}}\{\sum_{j\in\mathbf{J},\Omega_j=t} \tilde{U}^i_{j,t}\} = \beta_t$.

Therefore, all constraints of CAP are satisfied by **U**. Hence, the lemma is true. ■

It follows from Lemma 4.8 that one can always find **U** feasible to CAP using Expressions (4.37) based on **A** feasible to Expressions (4.38) and (4.39). It is noted that Expressions (4.38) and (4.39) can be considered as the respective demand and supply constraints of a transportation problem. The problem of finding **U** that satisfies Expressions (4.1) to (4.3) and (4.16) can be simplified as constructing **U** recursively using Expression (4.37) based on a feasible solution of the transportation problem involving variable **A.** There are many methods for finding a feasible solution to the transportation problem. Among those methods, Triangularity rule (Dantzig 1997) is effective for finding solutions on the models without transportation fee which is applicable to the problem. The details of the algorithm are given as follows:

Algorithm DP

Step 0 Set $m = 1$.

Step 1 Initialize t to1 when $\pi_m = T$; Otherwise, initialize $t = \pi_m + 1$.

Step 2 Find a feasible solution to Expressions (4.38) and (4.39) by Triangularity rule (Dantzig, 1997).

When $t = \pi_m + 1$, determine $U^i_{j,t}$ using Expressions $U^i_{j,t} = A^i_{j,t}$;

Otherwise, determine $U^i_{j,t}$ using Expressions $U^i_{j,t} = U^i_{j,t-1} + A^i_{j,t}$ for $i \in \mathbf{I}, j \in \mathbf{J} \,\&\, \Omega_j = \pi_m$.

Step 3 If $m = M$ and $t = \pi_M$, terminate the algorithm;

Otherwise, set $t = (t+1) \bmod T$, go to Step 2 when $t \neq \pi_m$,

set $m= m+1$, go to Step 1 when $t = \pi_m$ and $m<M$.

Algorithm DP presents an effective way for finding **U** that satisfies Expressions (4.1), (4.2), (4.3) and (4.16). It is clear that the algorithm must start from period $\pi_m + 1$ for each group $m, m \in \mathbf{M}$ such that the value of $U^i_{j,t}$, $t =$ $\pi_m + 1$ can be directly determined by expression $U^i_{j,t} = A^i_{j,t}$. Once U^i_{j,π_m+1} is determined, all $U^i_{j,t}$ can easily be determined using Expressions (4.37). Repeating the steps with $t = \pi_m + 2,\ldots,T,1,2,\ldots, \pi_m$, the solution procedure finds solution of $A^i_{j,t}$ by the

Triangularity rule introduced by Dantzig and then determines $U^i_{j,t}$ by Expressions (4.37). Thus, Algorithm DP can easily find feasible $\{\mathbf{U},\mathbf{A}\}$ to Expressions (4.37), (4.38) and (4.39). As shown in Lemma 4.7, Algorithm DP can feasible solutions to CAP based on **G** feasible to SGP.

Algorithm DP gives an effective procedure for solving the disaggregation problem while in Section 4.5.1, Algorithm GP is effective procedure for solving SGP. It is noted that the solutions found by Algorithm GP and DP have an interesting property and this property is presented in the lemma below.

Lemma 4.9

If $\{\mathbf{G},\mathbf{H}\}$ is the optimal solution found by Algorithm GP and $\{\mathbf{U},\mathbf{A}\}$ is a feasible solution found by Algorithm DP based on $\{\mathbf{G},\mathbf{H}\}$, then **U** is a partial optimal solution of CAP.

Proof:

According to the solution procedure of algorithm GP, $\{\mathbf{G},\mathbf{H}\}$ must satisfy Expressions (4.30)-(4.33) and (4.36). It follows from Lemma 4.6 that if $\{\mathbf{G},\mathbf{H}\}$ satisfies Expressions (4.30)-(4.33) and (4.36) for $t \in \mathbf{T}$, $\mathbf{G}$ is a partial optimal solution of SGP. Thus, **G** satisfies Expressions (4.17)-(4.22) of SGP.

Since $\{\mathbf{U},\mathbf{A}\}$ is a feasible solution found by Algorithm DP based on $\{\mathbf{G},\mathbf{H}\}$, Expressions (4.16), (4.37) – (4.39) are satisfied by the solution according to the solution procedure. Thus, it follows from Lemma 4.7 that **U** is a partial feasible solution of CAP.

It is clear from Corollary 4.2 that the objective function value of CAP corresponding to **U** found by Algorithm DP is the optimal one as the partial optimal solution of SGP, $\mathbf{G}$, has been found by Algorithm GP.

Hence, solution **U** also achieves the optimal objective value of CAP subsequent to **G**. Therefore, **U** is a partial optimal solution of CAP. The lemma is proved. ■

4.6 Numerical Example of solving CAP

In this section, the numerical example stated in Section 3.6 will be tested by the aggregation-disaggregation method to show the effectiveness of finding optimal solutions for CAP, i.e. to determine the space allocation of clusters for Phase 1. The problem of Phase 2 will be worked in the next section when further studies can be put on this example with the solution method for CCP*i* of Phase 2 introduced.

As stated in Section 3.6, the numerical example gives a yard planning problem that there are 3 blocks in the yard, each block contains 40 slots, serving for ten vessel services with known time schedule in a weekly cycle. The space requirement for the problem is given in Table 3.2.

It is shown from the aggregation-disaggregation method that the solution procedure of CAP for the example will be given with two parts. First, the solution procedure for service groups is stated below.

4.6.1 The solution procedure for service groups

It is noted from the aggregation method that services need to be grouped by the time period with loading jobs such that in the first step of aggregation the ten services in Table 3.2 will be put into three groups specified by the loading period: the first five vessel services are put into group 1 for having loading work on the same period (Period 7), services 6-7 are put into group 2 with loading work on period 3 and services 8-10 are in group 3. Table 4.6 shows the services assignment to each group.

Group	Services assigned to the group	Period with loading work
1	Services 1-5	Period 7
2	Services 6-7	Period 3
3	Services 8-10	Period 4

Table 4.6 Services aggregation in each group

It follows a new space requirement pattern for the three groups by summing up all space requirements of services in each group as shown in Table 4.7.

$\gamma_{m,t}$							
m	T						
	1	2	3	4	5	6	7
1	8	14	20	34	37	58	84
2	22	38	40	6	9	14	17
3	9	22	36	61	0	5	6

Table 4.7 Space requirements of groups during each time period

It is clear that solving such a service group problem with only three groups in the model is much simpler than solving the CAP of the example. It can be shown from Table 4.6 that the SGP satisfies the feasible condition $I \cdot K \geq \sum_{m=1}^{M} \gamma_{m,t}$ for all $t \in \mathbf{T}$, i.e. $I \cdot K = 120 > \max_{t \in T}(\sum_{m=1}^{M} \gamma_{m,t}) = \sum_{m=1}^{M} \gamma_{m,7} = 107$ and $t_0 = 7$, such that the computation procedure in Algorithm GP can be started.

Algorithm GP

Step 0 Set $t = t_0 = 7$. Initialize the value of H^i_{m,t_0-1} to 0 for $i \in \mathbf{I}, m \in \mathbf{M}$.

Step 1 The following expressions are obtained by setting t =7 and inputting the value of $\gamma_{m,7}$ into Expressions (4.31)-(4.33) and (4.36) that for $t \in \mathbf{T}$,

$$\sum_{i=1}^{I} H_{m,7}^{i} = \gamma_{m,7} - \lfloor \gamma_{m,7}/I \rfloor \cdot I \quad m \in \mathbf{M}\,,$$

$$\sum_{m=1}^{M} H_{m,7}^{i} \le K - \sum_{m=1}^{M} \lfloor \gamma_{m,7}/I \rfloor \quad i \in \mathbf{I}$$

and

$$0 \le H_{m,7}^{i} \le 1 \quad i \in \mathbf{I}, m \in \mathbf{M}\,.$$

Find a feasible solution to the above expressions by the method (Murty 1983) that the result is shown in Table 4.8.

$H_{m,7}^{i}$					
i	m				Supplies
	1	2	3	4	
1	0	1	0	4	5
2	0	1	0	4	5
3	0	0	0	5	5
Demands	0	2	0	13	

Table 4.8 Determine the value of $H_{m,7}^{i}$

It follows from Expressions (4.30) by setting t =7 that the value of $G_{m,7}^{i}$ is determined as below.

$G_{m,7}^{i}$			
i	m		
	1	2	3
1	28	6	2
2	28	6	2
3	28	5	2

Table 4.9 Determine the value of $G_{m,7}^{i}$

Step 2 Since $t \ne t_0 - 1 = 6$, set t= (t+1) mod T = 1, and go to Step1.

The computation of finding feasible solutions to Expressions (4.31)-(4.33) and (4.36) and then finding **G** by Expressions (4.30) will be repeated by setting a value of t within the calculation of each step until all the value of $H^i_{m,t}$ and $G^i_{m,t}$ are determined.

Below are the results of the service group problem.

$H^i_{m,t}$									
t	i =1			i =2			i =3		
	m=1	m=2	m=3	m =1	m=2	m=3	m=1	m=2	m=3
1	1	1	0	1	0	0	0	0	0
2	1	1	1	1	1	0	0	0	0
3	1	1	0	1	0	0	0	0	0
4	1	0	1	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0
6	1	1	1	0	0	2	0	0	0
7	0	1	0	0	1	0	0	0	0

Table 4.10 Results of $H^i_{m,t}$

Followed is the result of $G^i_{m,t}$ determined by Expressions (4.30).

$G^i_{m,t}$									
t	i =1			i =2			i =3		
	m=1	m=2	m=3	m =1	m=2	m=3	m=1	m=2	m=3
1	3	8	3	3	7	3	2	7	3
2	5	13	8	5	13	7	4	12	7
3	10	14	12	10	13	12	9	13	12
4	12	2	21	11	2	20	11	2	20
5	13	3	0	12	3	0	12	3	0
6	20	5	2	19	5	2	19	4	1
7	28	6	2	28	6	2	28	5	2

Table 4.11 Solution of $G^i_{m,t}$ for SGP

4.6.2 Disaggregate the space allocation to services

When the value of $G^i_{m,t}$, which is slots assigned to each group during each time period among blocks, has been determined by Algorithm GP, it follows from the disaggregation method that the space assigned to groups shown in Table 4.11 will be disaggregated to vessel services by Algorithm DP. The solution procedure for services is proceeded within each group as shown below.

Algorithm DP

Step 0 Initialize m to1.

Step 1 Since $\pi_1 = 7$, set t=1 .

Step 2 The following transportation table can be got by setting t=1 into Expressions (4.38) and (4.39) where the supply is equal to $G^i_{1,t}$ and demand is equal to $R_{j,t}$. The solution is found by the method (Dantzig 1997) as shown below.

$A^i_{j,1}$						
i	j					Supplies
	1	2	3	4	5	
1	1	1	0	1	0	3
2	0	0	0	1	2	3
3	0	0	0	0	2	2
Demands	1	1	0	2	4	

Table 4.12 Slots assigned to services in group 1 for $t = 1$

It is clear that the value of $U^i_{j,1}$ equals to $A^i_{j,1}$ since t = (π_1+1) mod T.

Step 3 Noted that $t \neq \pi_1$, set t = 2.

Go to Step 2. And then for t = 2, a transportation table can be constructed for $A^i_{j,t}$ to Expressions (4.38) and (4.39) with the supply equal to $G^i_{m,t} - G^i_{m,t-1}$

and demand equal to $R_{j,t} - R_{j,t-1}$. The value of $A^i_{j,t}$ is determined from the given transportation table by the same method.

$A^i_{j,2}$						
i	j					Supplies
	1	2	3	4	5	
1	1	1	0	0	0	2
2	0	0	1	1	0	2
3	0	0	0	1	1	2
Demands	1	1	1	2	1	

Table 4.13 Results of $A^i_{j,2}$

It is shown in Table 4.14 that the value of $U^i_{j,t}$ is determined by expression $U^i_{j,t} = A^i_{j,t} + U^i_{j,t-1} \quad i \in \mathbf{I}, j \in \mathbf{J} \,\&\, \Omega_j = \pi_m$.

$U^i_{j,2}$					
i	j				
	1	2	3	4	5
1	2	2	0	1	0
2	0	0	1	2	2
3	0	0	0	1	3

Table 4.14 Slots occupied by services in group 1 for $t = 2$

The allocation procedure will be repeated for $A^i_{j,t}$ and $U^i_{j,t}$, $t = \pi_m + 3, \ldots, T$ in Group 1 . Then proceed the same procedure for Group 2 and 3 afterwards.

When the space allocation for services within each group *m*, *m* =1,2,3 has been completed by Algorithm DP, the value of $U^i_{j,t}$ for all $i \in \mathbf{I}, j \in \mathbf{J}$ is determined. Show the result of **U** within each yard block *i*, the space allocation for services can be given in Table 4.15, 4.16 and 4.17. It can be shown from Table 4.15 that in block 1 storage space will be reserved for storing export containers for 9 vessel services, i.e.{1,2,3,4,6,7,8,9,10}.

$U^1_{j,t}$							
j	t						
	1	2	3	4	5	6	7
1	1	2	2	2	2	9	11
2	1	2	3	4	5	5	11
3	0	0	4	4	4	4	4
4	1	1	1	2	2	2	2
6	8	12	13	2	3	5	6
7	0	1	1	0	0	0	0
8	1	6	7	16	0	0	0
9	0	0	3	3	0	0	0
10	2	2	2	2	0	2	2

Table 4.15 Slots occupied by services in block 1 during each time period

In block 2 there will be storage space assigned for all the ten services shown as below.

$U^2_{j,t}$							
j	t						
	1	2	3	4	5	6	7
1	0	0	0	0	0	4	4
2	0	0	0	0	0	1	6
3	0	1	6	6	6	7	11
4	1	2	2	3	4	5	5
5	2	2	2	2	2	2	2
6	3	3	3	0	1	2	3
7	4	10	10	2	2	3	3
8	0	1	1	3	0	0	0
9	0	3	5	11	0	0	0
10	3	3	6	6	0	2	2

Table 4.16 Slots occupied by services in block 2 during each time period

Only seven services, i.e. {3,4,5,7,8,9,10}, needs storage space in block 3 shown as Table 4.17. It is noted from the solution of CAP in these three tables that this space allocation has achieved the workload balance when vessel services call the terminal. Hence, the optimal objective value for CAP is 2 slots showing that $\max_i \sum_{j \in J \& \Omega_j = \pi_1} U^i_{j,\pi_1} - \min_i \sum_{j \in J \& \Omega_j = \pi_1} U^i_{j,\pi_1} = 0$, $\max_i \sum_{j \in J \& \Omega_j = \pi_2} U^i_{j,\pi_2} - \min_i \sum_{j \in J \& \Omega_j = \pi_2} U^i_{j,\pi_2} = 1$ and $\max_i \sum_{j \in J \& \Omega_j = \pi_3} U^i_{j,\pi_3} - \min_i \sum_{j \in J \& \Omega_j = \pi_3} U^i_{j,\pi_3} = 1$. Once the solution of $U^i_{j,t}$ is determined, Problem P1 can be decomposed into 3 independent sub-problems denoted by CCPi that is corresponding to yard block i , i=1,2,3. The exact position for storing export contains of services in three blocks of this yard planning problem will be determined in next section.

$U^3_{j,t}$							
	t						
j	1	2	3	4	5	6	7
3	0	0	0	0	0	0	5
4	0	1	4	4	4	8	9
5	2	3	5	7	8	11	14
7	7	12	13	2	3	4	5
8	1	1	1	1	0	0	0
9	0	2	7	7	0	0	0
10	2	4	4	12	0	1	2

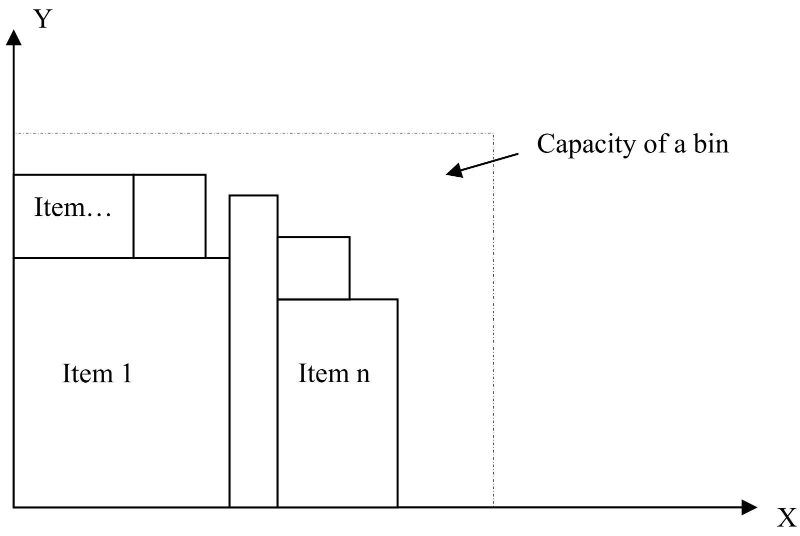
Table 4.17 Slots occupied by services in block 3 during each time period

4.7 Heuristic for Solving Cluster Configuration Problem

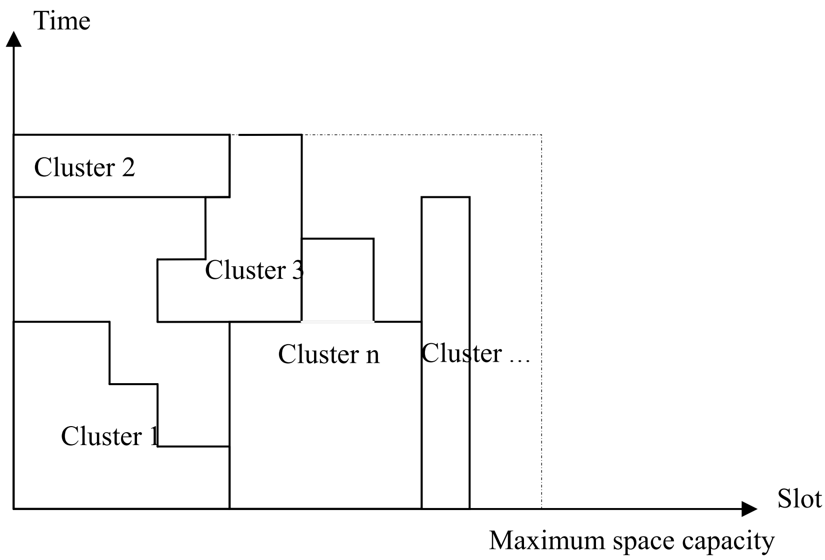
Once the partial solution of Cluster Allocation Problem, $U^i_{j,t}$, is determined by Algorithm DP, the exact shape and location of each cluster in each yard block can be determined by solving CCP$_i$ with the value of $U^i_{j,t}$ set to the solution found by Algorithm DP. It is shown from Lemma 4.2 that feasible solutions of CCP$_i$ are sufficient for constructing a solution for problem P1. However, in view of the large number of combinations of cluster shapes in a yard template, it is very time-consuming to design a yard template for block *i*, i.e., a feasible solution of CCP*i*. The numerical example in Section 4.6 was solved by ILOG CPLEX. The computational result shows that even for such a small-scale yard planning problem, CPLEX failed to find any feasible solution after running for 1 hour. Thus, an efficient solution method is needed for finding feasible solutions of the problem.

4.7.1 The Bin Packing Problem and solution methods

It is noted that the cluster design problem is, in fact, a resource allocation problem, i.e., to optimally allocate the limited yard block space to vessel services. It shows some similarities with the Bin Packing problem which is to assign items (rectangles) to a bin without exceeding the capacity of the bin and achieving the minimum number of bins used. A typical two-dimensional Bin Packing problem can be illustrated in Figure 4.2(a). If the clusters are regarded as items of irregular shape, the template design problem can be considered as packing the special items (clusters) into two-dimensional bins (templates) with X axis – Slot and Y axis - Time as shown in Figure 4.2(b). Specifically, it is worth noting that in the cluster-fitting problem, the exact shape of each cluster could be unknown. Moreover, since the arrival time of containers for each service is determined in the schedule, i.e., positions of items (clusters) at Time dimension are fixed, the objective of the special Bin Packing is to minimize the total slots used in the templates. Therefore, in order to solve the template design problem efficiently, effective algorithms are required. Algorithms for solving the Bin Packing problems are reviewed first.



(a) A two-dimensional Bin Packing problem



(b) Packing clusters in a yard block

Figure 4.2 The special Bin Packing in a yard template

A well known heuristic for the Bin Packing problem is the First Fit Decreasing heuristic developed by Johnson et al. (1974). The heuristic sorts all the items in non-increasing order of their size and then fit the items into the bins with the order, i.e., in each time one item will be fitted into the lowest indexed bin which can provide enough size for the item. Baker (1985) and Simchi-Levi (1994) analyzed the worst case of the First Fit Decreasing heuristic that shown the usefulness of the method. It is noted that many other heuristics have been proposed for different kinds of bin packing problems. As discussed in Berkey and Wang (1987), five algorithms-the Finite next-fit, Finite first-fit, Finite best-strip, Finite bottom-left and Hybrid first-fit algorithms were presented for solving two-dimensional bin packing problems. The method of each algorithm was demonstrated by computational experiments among which the worst case for the problem was analyzed. Lodi et al. (2002) gave a survey on algorithms for two-dimensional Bin Packing problem. They classified the algorithms into two families, one-phase algorithms which directly packed items into bins and two-phase algorithms which packed items into a single strip in the first phase and then constructed a packing order for them into finite bins. Computational experiments were carried out for testing different size problems that the polynomial-time approximation schemes were discussed for different algorithms.

It is known that the Bin Packing problem is NP-hard (Garey, 1979). To solve the problem efficiently, most heuristics need to follow some similar principles that

1) A suitable fitting order for all items is needed in an algorithm to avoid large computations for all possible combinations of items in the bins,
2) Space utilization is normally a major consideration for fitting an item into the bins in each step.

The generalized principles for solving Bin Packing problems, in fact, imply useful ideas for solving the cluster-fitting problem in yard blocks, i.e. to place all items (clusters) into the bin (the template) such that the width of the packing (number of slots) is minimized. However, previous heuristic programs are unlikely to be directly applied to the cluster-fitting problem in view of the obvious differences exhibited with the Bin Packing problem. A significant difference is the irregular shape of any one cluster in the cluster-fitting problem that results in much more combinations of clusters than those of packing regular rectangular items in a bin. As shown in previous results of computational experiments, the problem cannot be solved by ILOG CPLEX efficiently.

Thus, to design an effective heuristic for the problem, it is critical to set simple rules for generating a fitting order as well as determining exact shape and position for clusters in the template.

4.7.2 A Heuristic for Constructing Yard Clusters

Since determining the shape and location of each yard cluster is a critical step in designing a yard template, the properties of a yard cluster have to be analyzed first. According to the properties of a yard cluster, the size of a yard cluster is non-decreasing with time until the vessel calling period. Fitting clusters of various sizes and shape together in a yard block, in general, will lead to some vacant storage space between them. Different ways of fitting yard clusters into a yard block results in different amounts of vacant storage space. To find good feasible solutions, one may need to consider a large number of possible combinations of clusters' shape and locations when fitting the clusters into a yard block. This section proposes an effective yard template heuristic to quickly decide the cluster-fitting order and determine the shape and location of each yard cluster so that the number of slots used in each yard block in the planning horizon is as few as possible.

The basic idea of the yard template heuristic is to construct a space-saving yard template by fitting clusters one by one into yard block *i*. Define T_1 the peak period that $\sum_{j=1}^{J} U_{j,T_1}^{i} = \max_{t \in \mathbf{T}} \{\sum_{j=1}^{J} U_{j,t}^{i}\}$, i.e. the period with largest number of slots required. It is obvious that the number of slots used in a template will not be smaller than the space requirement at the peak period, i.e. the following property is true.

Property 4.1

Suppose that $\hat{z}_i$ is the optimal objective value of CCP*i*, then $\hat{z}_i \geq \sum_{j=1}^{J} U^i_{j,T_1}$.

It is shown from Property 4.1 that in a yard template, the less the slots are left vacant at the peak period, the less the total number of slots is used. Thus, the heuristic first determines block i's peak period with the largest $\sum_j U^i_{j,t}$. Then, it determines the location of the cluster to be fitted relative to the clusters already fitted. Next, using a simple rule, it selects the cluster that can utilize the storage space available at the peak period most effectively as the one to be fitted next. Finally, the heuristic determines the exact shape of the selected cluster when it is fitted into the template.

A simple numerical example is used to illustrate the heuristic. Consider a yard planning problem with only one yard block serving 4 vessel services, the space requirement in the block is given in Table 4.18.

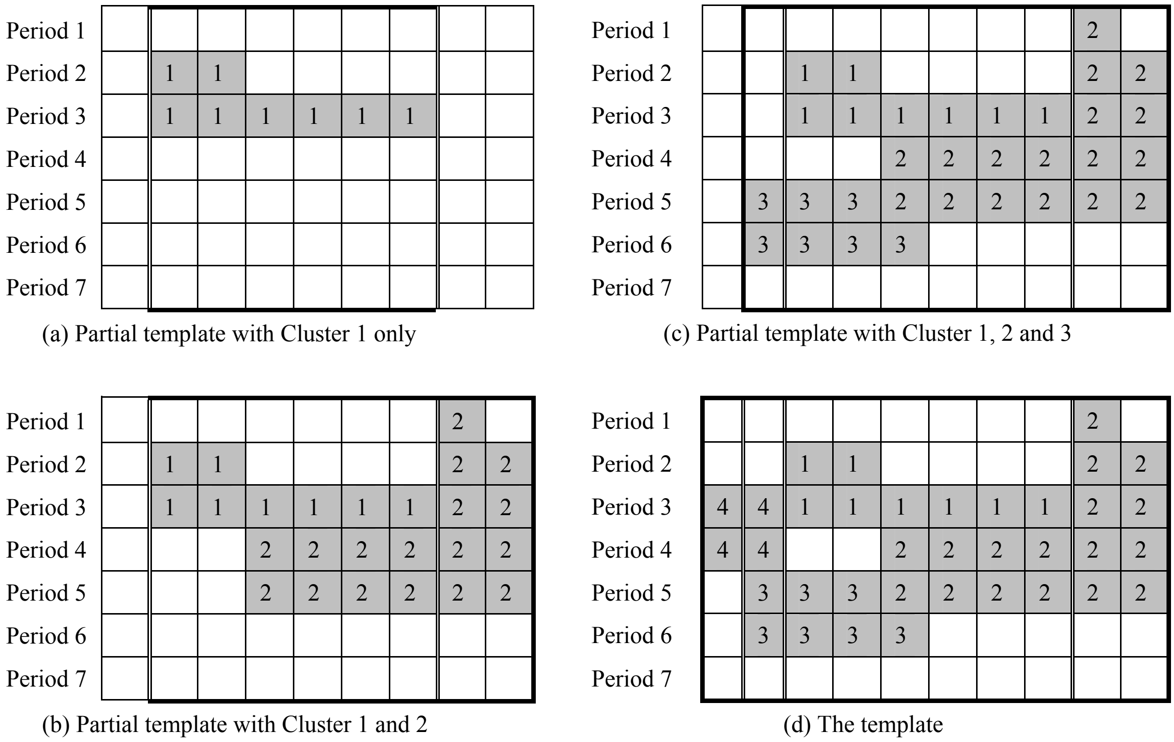
$U^i_{j,t}$							
	t						
j	1	2	3	4	5	6	7
1	0	2	6	0	0	0	0
2	1	2	2	6	6	0	0
3	0	0	0	0	3	4	0
4	0	0	2	2	0	0	0

Table 4.18 Space requirement for the four-service problem

In the example, Services 1, 2, 3 and 4 are loaded at periods 3, 5, 6 and 4, respectively (i.e., $\Omega_1 = 3$, $\Omega_2 = 5$, $\Omega_3 = 6$ and $\Omega_4 = 4$) with workloads $U^i_{1,3} = 6$, $U^i_{2,5} = 6$, $U^i_{3,6} = 4$ and $U^i_{4,4} = 2$. It is obvious from Table 4.18 that $\sum_j U^i_{j,1}$ = 1, $\sum_j U^i_{j,2}$ = 4, $\sum_j U^i_{j,3}$ = 10, $\sum_j U^i_{j,4}$ = 8, $\sum_j U^i_{j,5}$ = 9, $\sum_j U^i_{j,6}$ = 4, $\sum_j U^i_{j,7}$ = 0 and the peak period is period 3. It is clear that a total of 18 slots are needed if the slots assigned to a service (cluster) cannot be used by the other services. The yard template heuristic proposed in this section attempts to maximize the sharing of slots among the clusters so as to minimize the number of slots required in the yard template.

Suppose that Cluster 1 (the cluster of Service 1) has been selected and arbitrarily inserted into a yard template (the later part of this section will discuss how to select the first cluster). It is worthwhile to note that the exact location of the cluster in the block has not been fixed yet. With Cluster 1 in the yard template, one can find the rectangle with the smallest width that can enclose the cluster, as shown in Figure 4.3(a).

Note that there may exist various shapes for Cluster 1. It is generated from many heuristics of Bin Packing that if items are placed as close to one of the ends of the bin, the remaining space may provide larger size for fitting other items. Similarly, if Cluster 1 is placed at a position of the leftmost or rightmost slots of the rectangle enclosed at each time period, the remaining slots could be more easily for placing other clusters. The time position is fixed for Cluster 1 as stated before in that the loading date is determined. Therefore, without losing generality, all the slot allocations for Cluster 1 are supposed starting from the leftmost slot in the heuristic.



(a) Partial template with Cluster 1 only

(c) Partial template with Cluster 1, 2 and 3

(b) Partial template with Cluster 1 and 2

(d) The template

Figure 4.3 Yard template design with Rule 1

It is noted that in Figure 4.3(a), there are altogether 42 cells in the rectangle, with 34 of them being vacant. The larger number of the vacant slots is allocated to other clusters, the more tightly the clusters can be packed together and the smaller the total number of slots is needed for all the clusters .As each cell in the figure represents the resource of a slot of yard space available for a period, the space resource availability can be measured in terms of slot-period; thus, in the figure, the space resource available in the rectangle for allocating to other clusters are 34 slot-periods. Theoretically, all the vacant slot-periods are allocable for other clusters. However, in view of the complexity of considering all possible combinations of clusters for using the vacancies, the heuristic proposes a simple and effective method that in each cluster-fitting, one cluster will only be inserted into the partial template from either the left or the right edge of the rectangle. Where to place the cluster into the template depends on which way can likely lead to a more tightly packed yard template, i.e. a higher utilization of space resource in the rectangle.

Therefore, by the method, only continuous vacant slots during each time period to the right or left edge of the rectangle will be considered in the next cluster-fitting. Define the number of right-vacant (left-vacant) slots at period t, V_t (L_t), to be the number of vacant slots inside the rectangle contiguous to the right (left) edge of the rectangle at that period. It is clear form Figure 4.3(a) and the definition of V_t and L_t that V_1 = 6, V_2 = 4, V_3 = 0, V_4 = V_5 = V_6 = V_7 = 6, L_1 = 6, L_2 = 0, L_3 = 0, L_4 = L_5 = L_6 = L_7 = 6, $\sum_{t=1}^{7} V_t$ =34 and $\sum_{t=1}^{7} L_t$ =30. Since $\sum_{t=1}^{7} V_t > \sum_{t=1}^{7} L_t$, it is more likely that placing the next cluster from the right edge of rectangle can lead to more tightly packed yard template.

When the way of fitting a cluster is decided, the next step of the heuristic is to determine which cluster should be inserted into the partial yard template. As the peak period is the period with the largest space requirement, to minimize the total number of slots required by the yard template, one should try to fit the clusters into the template in such a way that the number of vacant slots in the peak period is minimized. Thus, the following fitting rule for selecting the next cluster to be inserted into a partial template is proposed:

Fitting Rule: maximizing the allocation of vacant slots to the next cluster at the peak period.

However, for the partial yard template in Figure 4.3(a), there is no vacant slot inside the rectangle at the peak period (Period 3). In fact, there exist some cases in which the above fitting rule cannot determine a cluster to be fitted into the template. Under such cases, an additional rule is needed to determine which of the remaining clusters should be inserted next.

Additional Rule 1: maximizing the number of vacant slot-periods allocated to the cluster.

For a given partial yard template with the smallest rectangle identified for enclosing all the clusters already inserted, Rule 1 determines the maximum number of vacant slot-periods inside the rectangle that can be allocated to each remaining cluster and then selects the one with the largest number of vacant slot-periods allocated to it. It can be determined from the given partial yard template in Figure 4.3(a) and $U^i_{j,t}$ given in Table 4.18 that the maximum number of vacant slot-periods that can be allocated to Clusters 2, 3 and 4 are 8, 7 and 0, respectively. Thus, Cluster 2 is the next cluster to be inserted. The partial yard template for the two clusters achieving the largest number of slots shared is shown in Figure 4.3(b).

The packing method is repeated for the remaining two clusters. Since $V_1^i = 1$, $V_2^i = V_3^i = V_4^i = V_5^i = 0$ and $V_6^i = V_7^i = 8$ whereas $L_1^i = 6$, $L_2^i = L_3^i = 0$, $L_4^i = L_5^i = 2$ and $L_6^i = L_7^i = 8$ in the new partial template of Figure 4.3(b), inserting one of the remaining clusters into the left-vacant slot-periods will likely lead to a more tight yard template as $\sum_{t=1}^{T} V_t^i = 17 \le \sum_{t=1}^{T} L_t^i = 26$. Still there are no vacant slot-periods at the peak period (period 3). Following the same cluster-fitting procedure with additional rule 1, Cluster 3 which can use 5 vacant slot-periods is selected as shown in Figure 4.3(c). The complete yard template for 4 services can be finalized with cluster-fitting order [1, 2, 3, 4] by the solution method that totally 10 slots are needed for the template.

For constructing the yard template for the set of clusters specified in the solution of CAP, a fitting order has been set in the construction steps discussed above for each of the clusters in the set except for the first one. An idea is, then, proposed for the first cluster-selection that among all the yard templates being constructed, the template with the smallest number of slot-periods vacated is selected since it will likely generate a more tightly packed partial template when other clusters are fitted into the rectangle. In view of the difficulty for theoretical analysis, the method is testified by an experiment that each cluster in the set is designed as the first one into the template to get an ultimate objective value of slots used. After testing hundreds of numerical examples, the results demonstrate that such a method of fitting the one with the smallest number of vacant slot-periods first into the template uses much less storage space eventually. Therefore, details of the heuristic are given below.

Yard Template Construction Heuristic

Step 1 Determine the peak period, i.e., the period with the largest $\sum_j U^i_{j,t}$.

Step 2 Insert the first cluster of the set of clusters specified in the solution of CAP into the yard template.

Step 3 Find the smallest rectangle that can enclose all the clusters already inserted into the yard template. Determine V_t and L_t. If $\sum_{t=1}^{T} V_t > \sum_{t=1}^{T} L_t$, the next cluster is placed at the right edge of the rectangle; otherwise, the next cluster is placed at the left edge.

Step 4 Select the cluster that can use the maximum number of vacant slot-periods inside the rectangle at the peak period. If there is more than one such cluster, apply Rules 1 to select the cluster. Insert the selected cluster into the yard template according to Step 3.

4.7.3 Further discussion about the Heuristic

It is shown from the yard template design heuristic that the time-consuming computation for considering all possible combinations is now replaced by generating a straightforward order for all clusters to be fitted into the template by a simple rule. In order to evaluate the performance of the solution method comprehensively, it is worthwhile to discuss some other rule for determining the cluster fitting order.

In this research, an alternative additional rule is suggested to determine the next cluster that

Additional Rule 2: maximizing the ratio of the maximum number of allocated vacant slot-periods to the cluster's total slot-period requirement.

Note that Rule 1 focuses on the absolute number of sharing of slot-periods of clusters; the alternative additional rule presents another idea of considering the relative space utilization in selecting clusters. If Rule 2 is conducted for selecting the cluster with the largest ratio, the maximum number of vacant slot-periods inside the rectangle that can be allocated to each remaining cluster will be computed first and then the ratio of the number to the cluster j's total slot period requirement (i.e. $\sum_t U^i_{j,t}$) for all remaining cluster j can be determined.

The idea is illustrated by the same simple example with space requirements in Table 4.20. Given a similar condition that Cluster 1 (the cluster of Service 1) has been determined in the yard template, the respective total slot-period requirements of Clusters 2, 3 and 4 are 17, 7 and 4 and the respective ratios for Clusters 2, 3 and 4 are 0.47, 1.00 and 0. Hence, Cluster 3 is selected as the next cluster. Following the same procedure, the yard template can be completed with cluster-fitting order [1, 3, 2, 4] that a total of 11 slots are used for all services as shown in Figure 4.4.

Period 1						
Period 2	1	1				
Period 3	1	1	1	1	1	1
Period 4						
Period 5						
Period 6						
Period 7						

(a) Partial template with Cluster 1

Period 1						
Period 2	1	1				
Period 3	1	1	1	1	1	1
Period 4						
Period 5	3	3	3			
Period 6	3	3	3	3		
Period 7						

(b) Partial template with Cluster 1,3

Period 1							2		
Period 2	1	1					2	2	
Period 3	1	1	1	1	1	1	2	2	
Period 4				2	2	2	2	2	2
Period 5	3	3	3	2	2	2	2	2	2
Period 6	3	3	3	3					
Period 7									

(c) Partial template with Cluster 1, 2 and 3

Period 1							2				
Period 2	1	1					2	2			
Period 3	1	1	1	1	1	1	2	2		4	4
Period 4				2	2	2	2	2	2	4	4
Period 5	3	3	3	2	2	2	2	2	2		
Period 6	3	3	3	3							
Period 7											

(d) The template

Figure 4.4 Yard template design with Rule 2

It is clear from Figure 4.3 and 4.4 that different templates are generated by applying different additional rules. For the above simple example, Rule 1 generates a better template as few slots are used. However, it does not ensure that the additional rule 1 is definitely better than the other rule. Consider the same example, if a few numbers of space requirements are changed as shown in Table 4.19, the solution procedure with additional Rule 2 will generate a template using the same number of slots as that of Rule 1.

$U^i_{j,t}$							
	t						
j	1	2	3	4	5	6	7
1	0	2	6	0	0	0	0
2	1	2	2	6	6	0	0
3	0	0	0	0	3	4	0
4	0	0	2	0	0	0	0

Table 4.19 Space requirement for the four-service problem

The new templates generated by the procedure with Rule 1 and Rule 2 are shown in Figure 4.5 and Figure 4.6, respectively. It is obvious that all the templates, although the cluster-fitting order and shape is much different, use the same number of slots (10 slots).

Period 1									2	
Period 2			1	1					2	2
Period 3	4	4	1	1	1	1	1	1	2	2
Period 4					2	2	2	2	2	2
Period 5		3	3	3	2	2	2	2	2	2
Period 6		3	3	3	3					
Period 7										

Figure 4.5 The template generated with additional Rule 1

Period 1							2			
Period 2	1	1					2	2		
Period 3	1	1	1	1	1	1	2	2	4	4
Period 4				2	2	2	2	2	2	
Period 5	3	3	3	2	2	2	2	2	2	
Period 6	3	3	3	3						
Period 7										

Figure 4.6 The template generated with additional Rule 2

The above example demonstrates another rule for generating a different fitting order for clusters which may achieve better templates for some numerical cases. It, in fact, implies a way for improving the heuristic in future studies. However, to make the fitting order clear and straightforward, each rule may only emphasis on some partial information for the yard space optimization. The results of using different rules vary with data setting of examples. It is still hard to know whether a selection of fitting order achieves better templates finally in view of the difficulty for theoretically analyzing the template design process. In this study, an experiment is carried out to test the performance of the solution procedure with different rules for cluster selection. The result of the experiment reveals that the solution idea with additional Rule 1 will likely achieve a smaller deviation of results with the lower bounds, i.e. less storage space is needed for the template. Therefore, in this study, Rule 1 is adopted as the additional rule for cluster selection in the heuristic. Details of experiments will be presented in the later part of the section.

4.7.4 A lower bound

To evaluate the performance of the heuristic presented in the previous section, a lower bound for the optimal objective function value of CCP*i* is proposed in this section. The following parameters are needed to state the lower bound:

T_2 the period immediately before the peak period which is equal to

$$\begin{cases} T_1 - 1, & if\ T_1 = 2,...7 \\ 7, & if\ T_1 = 1 \end{cases},$$

$\bar{j}_1$ the service such that $U^i_{\bar{j}_1,T_1} - U^i_{\bar{j}_1,T_2} = \max_{j\in\mathbf{J}\&U^i_{j,T_2}\neq 0}\{U^i_{j,T_1} - U^i_{j,T_2}\}$,

$\bar{j}_2$ the service such that $U^i_{\bar{j}_2,T_1} - U^i_{\bar{j}_2,T_2} = \max_{j\in\mathbf{J}\&j\neq\bar{j}_1\&U^i_{j,T_2}\neq 0}\{U^i_{j,T_1} - U^i_{j,T_2}\}$.

Lemma 4.10

Suppose that $\hat{z}_i$ is the optimal objective value of CCP*i*.

$$\sum_{j=1}^{J} U^i_{j,T_1} + \max\{0, \max_{j\in\mathbf{J}\&\Omega_j=T_2}\{U^i_{j,T_2} - U^i_{j,T_1}\} - \sum_{j\in\mathbf{J}\&U^i_{j,T_2}=0} U^i_{j,T_1} - (U^i_{\bar{j}_1,T_1} - U^i_{\bar{j}_1,T_2}) - (U^i_{\bar{j}_2,T_1} - U^i_{\bar{j}_2,T_2})\}$$

is a lower bound on $\hat{z}_i$.

Proof:

Suppose that $\hat{z}_i$ is the optimal objective value of CCP*i*. It is clear from Constraints (4.13) that $max\{k\cdot X^i_{k,j,t}\} = \hat{z}_i$ for $k\in\mathbf{K}, j\in\mathbf{J}, t\in\mathbf{T}$. Hence, $X^i_{k,j,t}=0$ for $k>\hat{z}_i$ and

$$\sum_{k=1}^{K}\sum_{j=1}^{J} X^i_{k,j,t} = \sum_{k=1}^{\hat{K}}\sum_{j=1}^{J} X^i_{k,j,t} \le \hat{z}_i. \quad \text{(A)}$$

Since $\sum_{j=1}^{J} U^i_{j,T_1}$ is the total sum of slots required by all services in block *i* at period T_1, it is obvious from Constraints(4.10) and Expression (A) that $\hat{z}_i$ satisfies

$$\hat{z}_i \ge \sum_{j=1}^{J} U^i_{j,T_1}. \quad \text{(B)}$$

It is proved below that $\max_{j\in\mathbf{J}\&\Omega_j=T_2}\{U^i_{j,T_2} - U^i_{j,T_1}\} - \sum_{j\in\mathbf{J}\&U^i_{j,T_2}=0} U^i_{j,T_1} - (U^i_{\bar{j}_1,T_1} - U^i_{\bar{j}_1,T_2}) - (U^i_{\bar{j}_2,T_1} - U^i_{\bar{j}_2,T_2})$ is a lower bound for $\hat{z}_i - \sum_{j=1}^{J} U^i_{j,T_1}$.

It is clear from Expression (B) that the number of vacant slots at period T_1

equal $\hat{z}_i - \sum_{j=1}^{J} U^i_{j,T_1}$. A lower bound for $\hat{z}_i - \sum_{j=1}^{J} U^i_{j,T_1}$ can be derived based on the slot assignment at periods T_1 and T_2.

For a service, say service j, with loading at period T_2, the slot requirement of the service at period T_1 is in general smaller than that at period T_2 and the number of slots vacated at period T_1 is equal to $U^i_{j,T_2} - U^i_{j,T_1}$. Among all the services with loading at period T_2, $\max_{j \in \mathbf{J} \& \Omega_j = T_2} \{U^i_{j,T_2} - U^i_{j,T_1}\}$ denote the maximum difference between their slot requirements at periods T_1 and T_2. The slots vacated after the loading of these services can be assigned to the clusters of vessels with loading at periods other than T_2. An upper bound on the number of vacated slots that can be assigned is derived below by considering the following two cases:

Case 1. For services without any slot requirement at period T_2, an upper bound on the number of vacated slots that can be assigned to these services at period T_1 is $\sum_{j \in \mathbf{J} \& U^i_{j,T_2} = 0} U^i_{j,T_1}$.

Case 2. For services with positive slot requirement at period T_2, the clusters of two of such services can be placed next to the cluster corresponding to $\max_{j \in \mathbf{J} \& \Omega_j = T_2} \{U^i_{j,T_2} - U^i_{j,T_1}\}$ as shown in Figure 4.7.

	Service $\bar{j}_2$	Service with loading at period T_2						Service $\bar{j}_1$	
Period T_2	+	*	*	*	*	*	*	#	#
Period T_1	+	+					#	#	#

Figure 4.7 Placing two clusters next to the cluster corresponding to $\max_{j \in \mathbf{J} \& \Omega_j = T_2} \{U^i_{j,T_2} - U^i_{j,T_1}\}$

It is clear from the definition of $\bar{j}_1$ and $\bar{j}_2$ that services $\bar{j}_1$ and $\bar{j}_2$ are the two services that can make use of the most vacated slots. It is obvious that an upper bound on the number of the vacated slots that can be assigned to the two services at period T_1 is $(U^i_{\bar{j}_1,T_1} - U^i_{\bar{j}_1,T_2}) + (U^i_{\bar{j}_2,T_1} - U^i_{\bar{j}_2,T_2})$.

In view of the above two cases, an upper bound on the number of the vacated slots that can be assigned to all the services at period T_1 is $\sum_{j \in \mathbf{J} \& U^i_{j,T_2}=0} U^i_{j,T_1} + (U^i_{\bar{j}_1,T_1} - U^i_{\bar{j}_1,T_2}) + (U^i_{\bar{j}_2,T_1} - U^i_{\bar{j}_2,T_2})$. Therefore, a lower bound on the number of vacant slots at period T_1 is $\max\{0, \max_{j \in \mathbf{J} \& \Omega_j = T_2} \{U^i_{j,T_2} - U^i_{j,T_1}\} - \sum_{j \in \mathbf{J} \& U^i_{j,T_2}=0} U^i_{j,T_1} - (U^i_{\bar{j}_1,T_1} - U^i_{\bar{j}_1,T_2}) - (U^i_{\bar{j}_2,T_1} - U^i_{\bar{j}_2,T_2})\}$. Hence, the lemma is proved.

To assess quality of the lower bound given in Lemma 4.10, an experiment is designed to test some small size problems with only 5 vessel services. The yard design heuristic stated in Section 4.7.2 is used to construct a yard block template for each problem with 5 clusters corresponding to the vessel services. To provide a comprehensive test, the computational experiment randomly generates test problems of three levels of slot requirements at the peak period, with 500 test problems for each level. In levels 1, 2 and 3, the number of the slot requirement at the peak period is not greater than 20, 30 and 40, respectively.

The results of the experiment are summarized in Table 4.20 that LB represents lower bound and OV denotes the objective value determined by the heuristic. It is shown from the result summary that the average deviation between the lower bound and the objective value generated by the heuristic is small than 4% for all the three levels. Moreover, the lower bound equals the objective value, which means the optimal objective value achieved, for about 50% of the test problems and the average deviation of the lower bound from the objective value is 3.33%. Thus, the lower bound is indeed an effective one.

Group	1	2	3
Percentage LB=OV	70%	49%	51%
Average Deviation	2.57%	3.72%	3.69%
Maximum Deviation	18.18%	18.75%	21.87%

Table 4.20 A result summary of the performance of lower bounds

Since there are some cases with large deviations (> 15%), a further analysis is conducted on the lower bound and ten test problems used in the previous experiment are solved by ILOG CPLEX. It is noted that even for such small-scale test problems, CPLEX is still not able to find a feasible solution for half of them within 2 hours. For the test problem that can be solved by CPLEX, the lower bounds given by ILOG are the same as those given by the method in Lemma 4.10. The further comparison from this aspect is given in Table 4.21.

Example	1	2	3	4	5	6	7	8	9	10
Lower bound	39	42	28	36	41	28	32	27	30	23
LB by ILOG	39	40	28	36	40	29	33	29	31	23
Solution by the heuristic	40	45	33	42	43	30	36	30	34	24
Optimal Value	40		29			29	33			24
Solved By ILOG	Y	N	Y	N	N	Y	Y	N	N	Y

Table 4.21 A comparison of results

On the basis of the above experiments, the lower bound given by Lemma 4.9 is indeed effective.

4.8 Numerical example of solving CCP*i*

It is shown in Section 4.6 that optimal **U** has been determined by Algorithm GP and DP for the numerical example with space requirement shown in Table 3.2. The result was summarized in Table 4.15, 4.16 and 4.17 for each yard block, respectively. The exact yard template, then, will be constructed by the solution procedure of the yard template construction heuristic in Section 4.7.2. The algorithm is coded in C++ and the program runs on a desktop computer with 2.4 GHz CPU. Results of the experiment show that all the templates can be determined by the heuristic within 1 second. The yard template for the three yard blocks are shown in Figure 4.8, 4.9 and 4.10.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
period 1	10	10	1		8										6	6	6	6	6	6	6	6			2													4		
period 2	10	10	1	1	8	8	8	8	8	8		6	6	6	6	6	6	6	6	6	6	6	6		2	2					7							4		
period 3	10	10	1	1	8	8	8	8	8	8	8	6	6	6	6	6	6	6	6	6	6	6	6	6	2	2	2	9	9	9	7			3	3	3	3	4		
period 4	10	10	1	1	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	6	6		2	2	2	2	9	9	9				3	3	3	3	4	4	
period 5			1	1																6	6	6	2	2	2	2	2							3	3	3	3	4	4	
period 6	10	10	1	1	1	1	1	1	1	1	1							6	6	6	6	6	2	2	2	2	2							3	3	3	3	4	4	
period 7	10	10	1	1	1	1	1	1	1	1	1	1	1				6	6	6	6	6	6	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	4	4	

Figure 4.8 Yard template in block 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
period 1	5	5										7	7	7	7			4		10	10	10												6	6	6				
period 2	5	5	9	9	9			7	7	7	7	7	7	7	7	7	7	4	4	10	10	10				3						8		6	6	6				
period 3	5	5	9	9	9	9	9	7	7	7	7	7	7	7	7	7	7	4	4	10	10	10	10	10	10	3	3	3	3	3	3	8		6	6	6				
period 4	5	5	9	9	9	9	9	9	9	9	9	9	9	7	7		4	4	4	10	10	10	10	10	10	3	3	3	3	3	3	8	8	8						
period 5	5	5												7	7	4	4	4	4							3	3	3	3	3	3			6						
period 6	5	5	1	1	1	1	2						7	7	7	4	4	4	4	4	10	10			3	3	3	3	3	3	3			6	6					
period 7	5	5	1	1	1	1	2	2	2	2	2	2	7	7	7	4	4	4	4	4	10	10	3	3	3	3	3	3	3	3	3	3	3	6	6	6				

Figure 4.9 Yard template in block 2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
period 1	8	5	5										7	7	7	7	7	7	7	10	10																			
period 2	8	5	5	5				7	7	7	7	7	7	7	7	7	7	7	7	10	10	10	10						4				9	9						
period 3	8	5	5	5	5	5	7	7	7	7	7	7	7	7	7	7	7	7	7	10	10	10	10						4	4	4	4	9	9	9	9	9	9	9	
period 4	8	5	5	5	5	5	5	5							7	7	10	10	10	10	10	10	10	10	10	10	10	10	4	4	4	4	9	9	9	9	9	9	9	
period 5	5	5	5	5	5	5	5	5							7	7	7												4	4	4	4								
period 6	5	5	5	5	5	5	5	5	5	5	5				7	7	7	7		10					4	4	4	4	4	4	4	4								
period 7	5	5	5	5	5	5	5	5	5	5	5	5	5	5	7	7	7	7	7	10	10			4	4	4	4	4	4	4	4	4	3	3	3	3	3			

Figure 4.10 Yard template in block 3

It is obvious from the figures that the yard template for blocks 1, 2 and 3 require 39, 36 and 39 slots, respectively and hence, the yard template gives a feasible solution for CCP*i*. It follows from Lemma 4.2, the heuristic actually finds an optimal solution for the example.

4.9 Performance Evaluation

In this section, an extensive computational experiment was conducted to evaluate the performance of the two-phase algorithm developed in this chapter to solve the yard planning problem of a typical export yard section of 10 yard blocks of 40 slots each dedicated to vessels with weekly calling pattern. In the experiment, 500 test problems were randomly generated for each of three different levels of service calling frequencies, namely, 5, 10 and 15 vessel services. Each test problem generated satisfied the necessary condition stated in Section 3.5.1. For each vessel service, the space requirement $R_{j,t}$ was randomly generated with the largest space requirement R_{j,Ω_j} generated from a uniform distribution [10, 80] and Ω_j randomly generated from a uniform distribution [1, 7]. The algorithm was coded in C++ and run on a desktop computer with 2.4 GHz CPU and 1 G memory.

Computational results of the experiment are summarized in Table 4.22.

Number of vessel services	Optimal solutions found (%)	Infeasible cases (%)
5 services	100	0
10 services	91	9
15 services	79	21

Table 4.22 Computational results

It is clear from the table that the respective percentages of optimal solutions found by the two-phase algorithm for 5, 10 and 15 services are 100%, 91% and 79%. Although the overall performance of the algorithm for finding optimal solutions in the experiment is satisfactory, it is worthwhile to note the algorithm is unable to find a feasible solution for some of the test problems – 9% of the 10-service problems and 21% of the 15-service problems. As discussed in previous sections, fitting more clusters into a yard template is likely to produce more vacant slots between them, but these oddly-shaped vacant slots make it difficult to be allocated these to other clusters. Thus, more vacant slots lead to a lower utilization of yard storage capacity.

To analyze the impact of storage requirements of vessel services on the performance of the algorithm, another experiment was carried out. For the 10 yard blocks of 40 slots each, the total storage capacity is 400 slots. In the experiment, three levels of average storage requirements, measured as the total number of slots required at the peak period, were considered. The respective average storage requirements at the peak period of the test problems in Levels 1, 2 and 3 are 120, 280 and 360 slots. 500 test problems for each of the three levels of storage requirements were randomly generated with each test problem satisfying the necessary feasibility condition stated in 3.5.2. The computational results are summarized in Table 4.23. It is obvious from the table that the storage requirement level directly has a substantial impact on the ability of the two-phase algorithm to solve the problem optimally.

Level	Optimal solutions found (%)	Infeasible cases (%)
1	100	0
2	100	0
3	65	35

Table 4.23 Computational results for three levels of storage requirements

It is noted that though all the test problems pass the necessary feasibility condition, there is no guarantee that the test problems are feasible. In the computational experiments, the test problems are randomly generated. Hence, some problems which have no feasible solutions may be generated because of the structures of the randomly generated constraints. However, it is difficult to determine whether these infeasible cases are results of the randomly generated constraints as running CPLEX for many hours still cannot produce any feasible solutions.

To evaluate the algorithm more comprehensively, the infeasible cases in aforementioned experiments are examined in detail by testing whether the 'infeasible' test problems satisfy the sufficient feasibility condition in Section 3.5.2. It was found that that 36% of these 'infeasible' test problems satisfied the sufficient feasibility condition and feasible yard templates, in fact, were constructed for the 'infeasible' test problems. For the remaining 'infeasible' test problems, it is difficult to determine whether the 'infeasible' problems are actually infeasible.

To sum up, the results of the extensive computational experiments show that the two-phase algorithm proposed in this research is, indeed, effective and efficient for finding good solutions to the yard planning problem.

4.10 Summary of the Chapter

This chapter has developed an effective and efficient algorithm for finding a good solution to the yard planning problem formulated in Chapter 3. The yard planning problem has been decomposed into two inter-related sub-problems – Cluster Allocation Problem (CAP) and Cluster Configuration Problem (CCP). The first sub-problem CAP has been analyzed in detail. An aggregation-disaggregation solution procedure has been developed to solve CAP. It has been shown that the optimal size of each cluster in each yard block to balance the workload equilibrium can quickly be determined for most cases. With the solutions of CAP, an efficient yard template design heuristic has been developed in the second phase to solve CCP. It has been shown that CCP is similar to the well-known Bin Packing problems which are not solvable in polynomial-bounded time. Computational experiments have shown that although the problem of determining positions and shape of each cluster in a yard block is more complex than the traditional two-dimensional Bin Packing problem, good feasible solutions can be found in a short period of time by the yard template design heuristic. Further experiments have shown that more than 90% test problems of Problem P1 can be solved optimally by the algorithm.

CHAPTER 5

YARD PLANNING FOR IRREGULAR VESSEL SERVICES

5.1 Introduction

Chapter 3 and 4 have analyzed and developed the yard planning problem for vessel services with cyclical calling pattern. Export containers to be loaded onto the same vessel service are stacked in different clusters of slots in different yard blocks to facilitate the loading operation. An effective yard template determined by the two-phase algorithm presented in Chapter 4 can achieve good workload balance among yard blocks for each period.

It is noted that a key assumption for adopting the yard template strategy for yard planning is availability of accurate information about the arrival of vessels and containers. However, in real practice, there are often some vessel services which call at a terminal. The arrival time and container throughput of these vessel services are not known until a few days before their calling dates. The yard planning problem for these vessel services is a more challenging job for a terminal as using a yard template strategy for these vessel services will unlikely be effective as it is difficult for the storage space of clusters planned for each service to nicely match with the actual space demand of the service on a particular day.

To cope with unpredictable vessel arrival time and throughput, scattered stacking strategy, advocated by Steenken et al. (2004), is adopted in this chapter for planning container storage space for vessel services with irregular calling pattern. The basic idea of scattered stacking is to assign a yard area to a berth instead of to a specific ship. Export containers to be loaded at the berth are scattered in assigned yard area. Thus, with this strategy, yard storage space will not be reserved for any particular vessel and no cluster is planned for any such services. To gain a better understanding and insights, the yard planning problem is discussed in detail in the section below.

5.2 Problem Description

This section studies the yard planning problem for vessel services with irregular calling pattern. It is noted that although the vessel calling pattern is not cyclical, terminal planners are still able to estimate the arrival time and the throughput for these vessel services according to historical data and updated requests for berthing from shipping lines. It is a common practice in the terminal industry that shipping lines need to make a request for berthing a few days in advance and export containers normally arrive at the terminal 3-5 days before the arrival time of the vessel service. Terminal planners, thus, usually try to minimize the highly concentrated workload in the yard by finding the appropriate storage locations for the incoming export containers.

Due to the dynamic nature of arrival time and/or container throughput of these services, a rolling horizon planning approach is used to determine the storage locations for incoming export containers to balance the workload for the entire planning horizon. Suppose that there are T planning periods in the entire planning horizon. Typically, terminal planners only implement the planning decisions for the first planning period as shipping lines will keep updating the terminal about the arrival time and the throughput of their services throughout the planning horizon. At the next planning period, the terminal planners will solve the yard planning problem again for the next planning horizon of T periods.

This chapter studies the problem of planning yard storage locations with scattered stacking for vessel services with irregular calling pattern. The following assumptions are made when analyzing the problem:

1) A dedicated group of yard blocks is assigned to store export containers to be transported to the berths allocated to vessel services with irregular calling pattern.

2) The arrival time and the throughput of each vessel service to be considered in a planning horizon are known, whereas the schedule could be updated at the next planning horizon. The storage space requirement of each service is non-decreasing with time until its loading day.

3) Export containers already stacked on the dedicated group of yard blocks at the beginning of a planning horizon will stay at their storage locations until they are loaded onto vessels.

The objective of the yard planning problem is to determine the exact location of the slots in each block assigned to store export containers of vessel services with irregular calling patterns so as to balance workload among blocks. The problem is formulated as an integer program below.

5.3 An Integer Programming Model

As stated in the assumptions, the export container already stored in the yard blocks will remain there until their loading day. The following notation is used to indicate whether a slot is occupied by such containers.

$$X^{i}_{k,j,0} = \begin{cases} 1, & \text{if slot } k \text{ of block } i \text{ is occupied by service } j \text{ at the beginning} \\ & \text{of the planning horizon} \\ 0, & \text{otherwise} \end{cases}.$$

The integer programming formulation for the yard planning problem is as follows:

Problem IP1

Minimize $\sum_{t=1}^{T}(\beta_t - \alpha_t)$

Subject to

$$X^{i}_{k,j,t-1} \le X^{i}_{k,j,t} \qquad k \in \mathbf{K}, i \in \mathbf{I}, j \in \mathbf{J}, t = 1,\ldots,\Omega_j \tag{5.1}$$

$$\sum_{j=1}^{J} X^{i}_{k,j,t} \le 1 \qquad k \in \mathbf{K}, i \in \mathbf{I}, t \in \mathbf{T} \tag{5.2}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{K} X^{i}_{k,j,t} = R_{j,t} \qquad j \in \mathbf{J}, t \in \mathbf{T} \tag{5.3}$$

$$\sum_{j=1}^{J}\sum_{k=1}^{K} X^{i}_{k,j,t} \le K \qquad i \in \mathbf{I}, t = 0,1,..,T \tag{5.4}$$

$$\sum_{j \in \mathbf{J} \& \Omega_j = t}\ \sum_{k=1}^{K} X^{i}_{k,j,t} \ge \alpha_t \qquad i \in \mathbf{I}, t \in \mathbf{T} \tag{5.5}$$

$$\sum_{j \in \mathbf{J} \& \Omega_j = t}\ \sum_{k=1}^{K} X^{i}_{k,j,t} \le \beta_t \qquad i \in \mathbf{I}, t \in \mathbf{T} \tag{5.6}$$

The objective of Problem IP1 is to minimize the sum of workload imbalance among yard blocks for the entire planning horizon. Constraints (5.1) ensure that the size of occupied slots in each cluster grows when time approaches the loading period. Constraints (5.2) state that each slot in a block can only be assigned to at most one service at any period. Constraints (5.3) state that the total number of slots assigned to a service must be equal to the space requirement of the service at each period. Constraints (5.4) give the capacity constraint for each block. Constraints (5.5) and (5.6) state the relationship between $\mathbf{X}$ and α_t and that between $\mathbf{X}$ and β_t, respectively. The above integer program is similar to the one presented in Chapter 3.

The main differences are that the size of a cluster is non-decreasing with t until its loading day and there may be multiple clusters storing containers for the same service. Therefore, constraints corresponding to Constraints (3.2), (3.6) and (3.7) are not required.

5.4 Solution Method

In this section, the two-phase method presented in Chapter 4 is modified to solve the problem. Similar to the analysis in Chapter 4, the aggregate version of Problem IP1 is considered in Phase 1 of determining the number of slots assigned to each vessel service in each yard block to meet the total storage space requirement at each period.

5.4.1 Slot Allocation Problem

Using the approach presented in Chapter 4, the integer program presented in Section 5.3 is simplified into a computationally more manageable problem, Slot Allocation Problem, which determines the number of slots allocated to each service at each period. Since the yard planning problem in this chapter is solved on a rolling horizon basis, there are some slots already allocated to services in last planning cycle at the beginning of the current planning horizon. Denote $U^i_{j,0}$ the number of slots that containers have already been stacked in block i for service j at the beginning of the planning horizon and $U^i_{j,t}$ the number of slots allocated to service j at period t in block i. The Slot Allocation Problem, which is similar Problem CAP in Chapter 4, can be stated as

Problem IP2

Minimize $\sum_{t=1}^{T}(\beta_t - \alpha_t)$

Subject to

$$U^i_{j,t-1} \le U^i_{j,t} \qquad i \in \mathbf{I}, j \in \mathbf{J}, t = 1,\ldots,\Omega_j \qquad (5.7)$$

$$\sum_{i=1}^{I} U^i_{j,t} = R_{j,t} \qquad j \in \mathbf{J}, t \in \mathbf{T} \qquad (5.8)$$

$$\sum_{j=1}^{J} U^i_{j,t} \le K \qquad i \in \mathbf{I}, t \in \mathbf{T} \qquad (5.9)$$

$$\sum_{j \in \mathbf{J} \& \Omega_j = t} U^i_{j,t} \ge \alpha_t \qquad i \in \mathbf{I}, t \in \mathbf{T} \qquad (5.10)$$

$$\sum_{j \in \mathbf{J} \& \Omega_j = t} U^i_{j,t} \le \beta_t \qquad i \in \mathbf{I}, t \in \mathbf{T} \qquad (5.11)$$

where $U^i_{j,t}$ are non-negative integers.

It is noted that the objective of problem IP2 - to minimize the sum of maximum differences of workload among yard blocks at each period is the same as that of Problem IP1. Constraints (5.7) ensure that the number of slots assigned to a service in a yard block is non-decreasing with time until the loading day. Constraints (5.8) state that the total storage space assigned to a vessel equals its storage space requirement at each period. Constraints (5.9) state the capacity constraint of a yard block. Constraints (5.10) and (5.11) establish a linear model for the objective function.

5.4.2 Properties of Slot Allocation Problem

When analyzing Problems IP1 and IP2 in detail, a few interesting properties of the two problems can be derived and summarized as follows:

Lemma 5.1

The optimal solution of Problem IP1 is a feasible solution of Problem IP2.

Proof:

The lemma can easily be proved using the argument for proving Lemma 4.1 - showing that a solution of Problem IP1 satisfies all the constraints of Problem IP2.

Since the objective functions of the two problems are the same, it is clear from Lemma 5.1 that the following corollary is true.

Corollary 5.1

The optimal objective function value of Problem IP1 is greater than or equal to the optimal objective function value of Problem IP2.

It follows from the above corollary that the corollary below is true.

Corollary 5.2

Suppose that $\{\hat{U}, \hat{\alpha}, \hat{\beta}\}$ is the optimal solution of Problem IP2. If $\{\overline{\boldsymbol{X}}, \overline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\beta}}\}$ is a feasible solution to Problem IP1 with $\sum_{k=1}^{K} \overline{X}_{k,j,t}^{i} = \hat{U}_{j,t}^{i}$ for all j and t, $\{\overline{\boldsymbol{X}}, \overline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\beta}}\}$ is the optimal solution can be constructed for Problem IP1.

5.4.3 Algorithm for Solving Slot Allocation Problem

An aggregation-disaggregation approach similar that discussed in Chapter 4 is used to solve Problem IP2. In each planning horizon, vessel services are first grouped by their loading dates. Let M be the number of groups and π_m, $\pi_m \in \mathbf{T}$, be the loading date of the services in group m. Extending the definition of $\gamma_{m,t}$ and $G_{m,t}^{i}$ in Section 4.5 to the index of t, $t = 0,1,\ldots,T$, the relationship of $\gamma_{m,t}$ and $R_{j,t}$, $G_{m,t}^{i}$ and $U_{j,t}^{i}$ can be expressed as

$$\gamma_{m,t} = \sum_{j \in \mathbf{J} \& \Omega_j = \pi_m} R_{j,t} \quad m \in \mathbf{M},\ t = 0,1,\ldots,T \qquad (5.12),$$

$$G_{m,t}^{i} = \sum_{j \in \mathbf{J} \& \Omega_j = \pi_m} U_{j,t}^{i} \quad i \in \mathbf{I}, m \in \mathbf{M}, \; t = 0,1,\ldots,T \qquad (5.13).$$

It follows from Expressions (5.12) that $G_{m,t}^{i}$ is non-decreasing in t until its loading date π_m, i.e.,

$$G_{m,t-1}^{i} \le G_{m,t}^{i} \quad i \in \mathbf{I}, m \in \mathbf{M}, \; t = 1,\ldots,\pi_m \qquad (5.14),$$

It can be shown from Expressions (5.13) and Constraints (5.8) and (5.9) that for $m \in \mathbf{M}, t \in \mathbf{T}$,

$$\sum_{i=1}^{I} G_{m,t}^{i} = \sum_{i=1}^{I} \sum_{j \in \mathbf{J} \& \Omega_j = \pi_m} U_{j,t}^{i} = \sum_{j \in \mathbf{J} \& \Omega_j = \pi_m} R_{j,t} \,,$$

and for $i \in \mathbf{I}, t \in \mathbf{T}$,

$$\sum_{m=1}^{M} G_{m,t}^{i} = \sum_{m=1}^{M} \sum_{j \in \mathbf{J} \& \Omega_j = \pi_m} U_{j,t}^{i} = \sum_{j=1}^{J} U_{j,t}^{i} \le K \,.$$

Hence,

$$\sum_{i=1}^{I} G_{m,t}^{i} = \gamma_{m,t} \quad m \in \mathbf{M}, t \in \mathbf{T} \qquad (5.15),$$

$$\sum_{m=1}^{M} G_{m,t}^{i} \le C_i \quad i \in \mathbf{I}, t \in \mathbf{T} \qquad (5.16).$$

It can be shown from Expressions (5.13), (5.10), (5.11) that

$$G_{m,\pi_m}^{i} \ge \alpha_{\pi_m} \quad i \in \mathbf{I}, m \in \mathbf{M} \qquad (5.17),$$

$$G_{m,\pi_m}^{i} \le \beta_{\pi_m} \quad i \in \mathbf{I}, m \in \mathbf{M} \qquad (5.18).$$

Hence, the problem of finding the optimal number of slots to service groups, IP3, can be stated as follows:

Problem IP3

Minimize $\sum_{m=1}^{M} (\beta_{\pi_m} - \alpha_{\pi_m})$ subject to Expressions (5.14)-(5.18).

It is obvious from the formulation of Problem IP3 that the scale of the problem in each planning horizon is largely reduced with much smaller number of service groups. In view of the manageable size of Problem IP3, it is suggested solving IP3 using CPLEX. The latter the chapter will test the performance of the solution idea by computational experiments. Once a solution of Problem IP3 is found, the remaining work for space allocation is to disaggregate the allocated slots in $G^{i}_{m,t}$ to each vessel service in group *m*. When analyzing Problem IP3 in detail, the following properties are obtained.

Lemma 5.3

The optimal solution of Problem IP2 is a feasible solution of Problem IP3.

The lemma can easily be proved by showing that the optimal solution of Problem IP2 satisfies all constraints of Problem IP3.

Since the objective functions of Problems IP2 and IP3 are identical, the following corollary is true.

Corollary 5.2

The optimal objective function value of Problem IP2 is greater than or equal to the optimal objective function value of Problem IP3.

In this section, a method is developed for determining $U^{i}_{j,t}$ for Problem IP2 based on a good feasible solution of Problem IP3.

Define $A^{i}_{j,t}$ the increment of slots allocated to service *j* in block *i* at period *t*. Hence, for $i \in \mathbf{I}, j \in \mathbf{J}$,

$$A^{i}_{j,t} = U^{i}_{j,t} - U^{i}_{j,t-1}, \quad t = 1,...,\Omega_{j} \ . \tag{5.19}$$

It is obvious from the definition that $A^{i}_{j,t} \geq 0$.

It follows from Expressions (5.8), (5.13) and (5.19) that for $j \in \mathbf{J}$,

$$\sum_{i=1}^{I} A_{j,t}^{i} = R_{j,t} - R_{j,t-1}, \quad t = 1,...,\Omega_j \tag{5.20}$$

and for $i \in \mathbf{I}, m \in \mathbf{M}$

$$\sum_{j \in \mathbf{J} \& \Omega_j = \pi_m} A_{j,t}^{i} = G_{m,t}^{i} - G_{m,t-1}^{i}, \quad t = 1,...,\Omega_j \,. \tag{5.21}$$

On the basis of the above analysis, once the value of **G** for IP3 is determined, the problem of finding **U,** i.e. the set of $U_{j,t}^{i}$, is transformed into determining **A** where $A_{j,t}^{i}$ satisfies Expressions (5.19)-(5.21). The problem of determining **A** can be solved by the following algorithm.

Algorithm IDP

Step 0 Set $m = 1$.

Step 1 Initialize t to1.

Step 2 Find a feasible solution to Expressions (5.20) and (5.21) by Triangularity rule (Dantzig 1997).

Determine $U_{j,t}^{i}$ using the expression that $U_{j,t}^{i} = U_{j,t-1}^{i} + A_{j,t}^{i}$ for $i \in \mathbf{I}, j \in \mathbf{J} \& \Omega_j = \pi_m$.

Step 3 If $m = M$ and $t = \pi_M$, terminate the algorithm;

Otherwise, if $t \neq \pi_m$, then set $t = t + 1$ and go to Step 2; else, set $m = m+1$ and go to Step 1.

Algorithm IDP presents an effective way for finding **U** that satisfies Expressions (5.7), (5.8) and (5.13). As shown in the algorithm, the disaggregation of the slots in **G** to each vessel service of the group is started from t =1 - the first time period of the planning horizon. It determines the value of $A_{j,t}^{i}$ - the increment of required slots for service j by the Triangularity rule proposed by Dantzig and then determines $U^i_{j,t}$ by Expressions (5.19). Repeating the steps on t =2,…, π_m, the solution procedure finds solution of $U^i_{j,t}$ for all periods. Thus, a feasible {**U**, **A**} to Expressions (5.19)- (5.21) can easily be determined by Algorithm IDP. The following lemma states an important property of the solution found by the algorithm.

Lemma 5.4

If **G** is the optimal solution of IP3 and {**U**, **A**} is a feasible solution found by Algorithm IDP, then **U** is a partial optimal solution of IP2.

Proof:

It is shown from the solution procedure of Algorithm IDP that {**U**, **A**} satisfies Expressions (5.19)-(5.21) if it is a feasible solution found by the algorithm.

Suppose that **G** is the optimal solution of Problem IP3. It is obvious that **G** satisfies all constraints of Problem IP3.

It can be shown from Expressions (5.19)-(5.21) that $\sum_{i=1}^{I} U^i_{j,t} - \sum_{i=1}^{I} U^i_{j,t-1} = R_{j,t} - R_{j,t-1}$ and $\sum_{j \in \mathrm{J} \& \Omega_j = \pi_m} U^i_{j,t} - \sum_{j \in \mathrm{J} \& \Omega_j = \pi_m} U^i_{j,t-1} = G^i_{m,t} - G^i_{m,t-1}$ for t =1,2,…, Ω_j.

It is obvious that at the beginning of the planning horizon that $\sum_{i=1}^{I} U^i_{j,0} = R_{j,0}$ and $\sum_{j \in \mathrm{J} \& \Omega_j = \pi_m} U^i_{j,0} = G^i_{m,0}$.

Thus, it follows from $\sum_{i=1}^{I} U^i_{j,t} - \sum_{i=1}^{I} U^i_{j,t-1} = R_{j,t} - R_{j,t-1}$ and $\sum_{i=1}^{I} U^i_{j,0} = R_{j,0}$ that $\sum_{i=1}^{I} U^i_{j,t} =$ $R_{j,t}$ for t =1,2,… Ω_j. Hence, **U** satisfies Expressions (5.8).

Similarly, it can be shown from the above expressions that $\sum_{j \in \mathrm{J} \& \Omega_j = \pi_m} U^i_{j,t} = G^i_{m,t}$.

It can be derived from Expressions (5.16) that for $i \in \mathbf{I}, t \in \mathbf{T}$, $\sum_{m=1}^{M} G^i_{m,t} = \sum_{m=1}^{M} \sum_{j \in \mathrm{J} \& \Omega_j = \pi_m} U^i_{j,t} = \sum_{j=1}^{J} U^i_{j,t} \le C_i$. Hence, Expressions (5.9) are satisfied.

It can be shown from the fact $U^i_{j,t} - U^i_{j,t-1} = A^i_{j,t} \ge 0$ that $U^i_{j,t-1} \le U^i_{j,t}$ for

$i \in \mathbf{I}, j \in \mathbf{J}, t = 1, \ldots, \Omega_j$, i.e., Expressions (5.7) are satisfied.

It is obvious that Expressions (5.10) and (5.11) can always be satisfied by **U**.

Thus, **U** is a feasible solution of Problem IP2.

Since the objective functions of Problem IP2 and IP3 are identical, **U** is a partial optimal solution of Problem IP2 if **G** is the optimal solution of Problem IP3.

The lemma is thus proved.

5.4.4 Algorithm for Solving Yard Planning Problem

The previous section presents an algorithm for solving the Phase 1 problem – Slot Allocation Problem. In this section, an algorithm for solving the problem in the second phase – the problem of finding the exact storage location in a yard block based on a feasible solution of the Phase 1 problem. It is worth noting that the solution of the Phase 2 problem is actually a feasible solution of the yard planning problem stated in Section 5.3.

It is noted that multiple clusters for the same service can exist in a yard block under the scattered stacking strategy. That means, the export containers for the same service need not be stored in a contiguous stretch of slots. Hence, the exact storage location in a yard block can be determined based on a feasible solution of Slot Allocation Problem (Problem IP2). In this section, a simple algorithm is proposed for determining the exact storage locations for export containers. Let $\overline{\mathbf{K}}_t^i$ be the set of available slots in block *i* at time period *t*. Let $\ddot{\mathbf{K}}_t^i$ be the set of slots occupied by service *j* with loading date at period *t* (i.e., $\Omega_j = t$). It is noted that $\overline{\mathbf{K}}_0^i$ and $\ddot{\mathbf{K}}_0^i$ can easily be determined based on the export containers already stored in block *i* at the beginning of a planning horizon. Details of the algorithm for determining the exact storage locations for export containers in block *i* at period *t* are as follows:

Algorithm SAP

Step 0: Initialize t =1.

Step 1: Initialize i =1.

Step 2: Set $\overline{\mathbf{K}}_t^i = \overline{\mathbf{K}}_{t-1}^i + \ddot{\mathbf{K}}_{t-1}^i$.

Step 3: Initialize j =1.

Step 4: If $U_{j,t}^i \neq 0$, allocate a number of $U_{j,t}^i$-$U_{j,t-1}^i$ slots in $\overline{\mathbf{K}}_t^i$ to service j in sequence starting from the slot with the smallest number in the set. Update the set $\overline{\mathbf{K}}_t^i$ by eliminating the slots which have already been allocated to service j.

Step 5: Set $j = j$+1;

If $j \leq J$, go to Step 4;

Otherwise, set $i = i$+1;

If $i \leq I$, go to Step 2;

Otherwise, set $t = t$+1;

If $t \leq T$, go to Step 1;

Otherwise, terminate the algorithm.

It is clear from the procedures of Algorithm SAP that it sequentially allocates available slots in $\overline{\mathbf{K}}_t^i$ to service j when $U_{j,t}^i \neq 0$ to satisfy the number of slots in block i assigned to service j in Phase 1. Therefore, Algorithm SAP can find a feasible solution to Problem IP1 based on a feasible solution of IP2. In real practice, only the yard planning decisions made for the first period of the planning horizon is implemented.

5.5 Performance Evaluation

In this section, the proposed two-phase method for solving Problem IP1- the yard planning problem for services with irregular calling patterns is evaluated by a computational experiment. In the experiment, 500 test problems were randomly generated with 3 different services calling frequencies, namely, 5, 10 and 15 services in a planning horizon of one week. There are 10 export yard blocks, each consisting of 40 slots. The length of a planning period is one day. Hence, T=7.

The largest space requirement R_{j,Ω_j} was randomly generated from a uniform distribution [10, 80] and Ω_j randomly generated from [1, 7]. The two-phase algorithm was coded in C++ and that run on a desktop computer with 2.4 GHz CPU. Problem IP1 was solved by ILOG CPLEX 7.1 and the optimal solution/lower bound obtained was used to benchmark the solutions found by the algorithm. The computational results are summarized below.

	Imbalance workload among blocks			Relative gap to lower bound (%)		
	5 services	10 services	15 services	5 services	10 services	15 services
Average	7.8	12.7	14.9	0	0	0.3
Maximum	16	18	23	0	0	6.1
Minimum	4	7	7	0	0	0

Table 5.1 Computational results

It is noted from the computational results that the optimal solutions were found for all the test problems with 5 and 10 vessel services. For test problems with 15 services, more than 96% of them were solved optimally. The average gap of workload imbalances between generated solutions and the lower bounds provided by CPLEX are 0.3%. The algorithm is effective and efficient for solving the yard planning problem for vessel services with irregular calling patterns.

5.6 Summary of the Chapter

In this chapter, the yard planning problem for vessel services with irregular vessel calling patterns has been studied. The problem has been formulated as an integer programming model. A rolling horizon approach has been suggested to implement the yard planning decisions. Due to the large computational requirement of the problem, the problem has been decomposed two levels of simpler sub-problems using aggregation-disaggregation approach and a two-phase algorithm has been developed for solving the problem. The yard planning problem has first been simplified into an aggregated problem – the problem of finding the number of slots allocated to each service at each period. The aggregated problem has further been simplified into a computationally manageable problem by CPLEX in service groups and the solutions for service groups have been dispatched into services by an algorithm. Then, the chapter has proposed a simple algorithm to disaggregate the solution of the simplified problem into a feasible solution of the original yard planning problem. A computational experiment has been conducted to evaluate the performance of the two-phase algorithm. The computational results have shown that the algorithm is indeed effective and efficient, with the solutions found for more than 96% of test problems being the optimal ones.

CHAPTER 6

Conclusions and Suggestions for Future Work

The objective of this research study is to develop effective algorithms for planning yard blocks to store export containers under two different scenarios of vessel service calling patterns – cyclical calling pattern and irregular calling pattern, for the purpose of balancing workload among all the yard blocks. Unlike most existing studies, the study aims at determining the storage locations for the services over the entire planning horizon to minimize the occurrence of bottlenecks in the yard.

The research has reviewed the role of yard operations in a container terminal. Traditionally, a container yard is a place for temporary storage of containers and is not considered as a prime focus for directly improving the container handling efficiency. However, in recent years, the rapid growth in container traffic has put tremendous pressure on container terminals to handle and store increasing number of containers. Due to the shortage of land supply, many Asian seaports adopt a high-stacking strategy for yard storage which often results in more frequent re-handling, reshuffling and highly uneven workload in different parts of their yards. The highly uneven workload is a major factor causing bottlenecks in yard operations and these bottlenecks slow down a terminal operation and lead to unsatisfactory terminal productivity.

Since handling of export containers for a vessel is performed within a short period of time, a yard block storing export containers for several vessels requesting loading on the same day often has a very high concentration of handling activities in the block and this would likely cause bottlenecks in the block. Therefore, a terminal needs to plan its yard in such a way that the workload arising from handling of export containers stored in various yard blocks are as even as possible.

In this research, the problem of determining the storage locations for export containers of vessel services with cyclical calling pattern has first been studied. For this scenario of vessel service calling plan, terminals can in general produce good forecasts on the service arrival time and the storage requirement of the service and export containers of a service are often stored in different clusters of contiguous stretches of slots in different yard blocks. The yard planning problem has been formulated as an integer program. However, results of computational experiments have shown that the problem is unlikely to be solved optimally by ILOG CPLEX within a reasonable period of time. Indeed, finding a feasible solution of some test problems have taken a long time. The research has decomposed the problem into two sub-problems - Cluster Allocation Problem (CAP), the problem of determining the size of clusters assigned to each yard block storing export containers of different services, and Cluster Configuration Problem (CCP), the problem of determining the location of the shape of each cluster assigned to each yard block in the solution of CAP. A two-phase solution algorithm has been developed to solve the yard planning with phases 1 and 2 solving CAP and CCP, respectively. To solve CAP, an aggregation-disaggregation approach has been used, Vessel services calling at the terminal at the same period have been aggregated as a service group and an integer programming model has been formulated to determine the size of clusters assigned to different service groups in different yard blocks. The integer program has then been transformed into a problem similar to a transportation problem, which can be solved easily. An efficient method has been designed to disaggregate to find the number of slots assigned to each service in each yard block. A condition for a solution to be optimal has been derived and the solution determined by the phase 1 solution method for most test problems of the computational experiment have met the optimality condition.

With the solution of CAP found in Phase 1, an integer programming model has been formulated for CCP which can be interpreted as fitting clusters of different sizes determined in phase 1 into a number of rectangular yard blocks. CCP has been shown to be identical to the well-known two-dimensional bin-packing problem except that the shape of cluster is a decision variable. An effective yard cluster construction heuristic has been developed as the phase 2 solution method for solving CCP. The heuristic uses several simple rules to determine order of fitting clusters into yard blocks and the location of each yard cluster in a block to the minimize number of slots used. Results of computational experiments have shown that the two-phase algorithm is indeed effective and efficient, with over 90% of the solutions found being the optimal ones.

On the basis of the analysis on the yard planning problem for services with cyclical calling pattern, the research has developed an algorithm for solving the yard planning problem for services with irregular calling pattern. To tackle the unpredictable vessel throughput, containers are stacked in scattered areas of the yard instead of in contiguous slots of clusters so as to free up a large number of slots reserved for more flexible adjustments upon updated requests of vessels. The solution approach and the two-phase algorithm developed for cyclical vessel services have been modified. Results of computational experiments have shown the modified algorithm is effective and efficient.

The research on the yard planning problem with two scenarios of vessel service calling patterns has demonstrated the effectiveness of the aggregation-disaggregation approach for improving the utilization of yard space and balancing yard workload under different situations. In view of the various requirements and objectives for designing a yard storage plan, the model and the algorithm proposed in this thesis can be extended to tackle more sophisticated yard planning problems.

The yard planning methodology proposed in this research is very effective in allocating storage space for export containers to achieve workload balances in a container yard. The solution found the proposed method facilitates many other concurrent planning activities in a terminal and enable a terminal to achieve smooth flow of containers in the operations of both the landside and the quayside. The methodology is applicable to various operating situations. When more complicated conditions and constraints in yard operations need to be considered, the parameters and formulations of the yard planning problem can be modified. Indeed, the proposed approach provides an effective way for analyzing the whole container handling process from a systematic point of view. To better plan the storage of containers which interacts with many other terminal activities, the relations between yard operations with remaining container operations need to be analyzed in detail. Yard planning may be conducted simultaneously with other inter-related planning activities to optimize the performance of overall terminal operations. On the basis of this research work, the following interesting research issues can be studied in the future.

- It has been discussed that yard operations are closely linked with many other handling activities in a terminal. Researchers need to develop effective methods for solving the space allocation as well as monitoring the entire container flow in future studies. Specifically, different planning activities such as berth allocation, crane deployment and yard planning can be integrated in a high-level decision making problem which considers the interaction between the yard and berth operations. It is likely that a complex model need to be developed for integrated problem.

- In this research study, two stacking strategies are suggested for the planning of yard blocks assigned to services with two different calling patterns. It is shown that an appropriate stacking strategy for storing containers can improve the storage space utilization and the workload balance. The problem of how to partition an export container yard into two sections, with one for cyclical services and the other for irregular services, is a challenging and interesting direction for future work.

- To incorporate more realistic characteristics of real life operations in container terminals, the assumptions and constraints of the yard planning problem for services with irregular calling pattern need to be modified in further studies. With respect to different terminal layouts and operational strategies, there are a number of potential adjustments for modeling a more realistic planning problem for dynamic calling pattern. Further work can be carried out to extend analysis presented in this thesis to consider different real life scenarios.

REFERENCES

ANDERSON, C. M., PARK, Y.-A., CHANG, Y.-T., YANG, C.-H., LEE, T.-W. & LUO, M. (2008) A game-theoretic analysis of competition among container port hubs: the case of Busan and Shanghai. *Maritime Policy & Management,* 35**,** 5-26.

ARBIB, C., LUCERTINI, M. & NICOLÒ, F. (1991) Workload balance and part-transfer minimization in flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems,* 3**,** 5-25.

AVRIEL, M. & PENN, M. (1993) Exact and approximate solutions of the container ship stowage problem. *Computers & Industrial Engineering,* 25**,** 271-274.

AVRIEL, M., PENN, M. & SHPIRER, N. (2000) Container ship stowage problem: complexity and connection to the coloring of circle graphs. *Discrete Applied Mathematics,* 103**,** 271-279.

AVRIEL, M., PENN, M., SHPIRER, N. & WITTEBOON, S. (1998) Stowage planning for container ships to reduce the number of shifts. *Annals of Operations Research,* 76**,** 55-71.

BAKER & B, S. (1985) A new proof for the first-fit decreasing bin-packing algorithm. *Journal of algorithms,* 6**,** 49-70.

BAO, Q. & LUO, W. (2005) *The Technology of Construction & Operation for the Container Terminals,* Shanghai, Shanghai Scientific & Technical Publishers.

BAZZAZI, M., SAFAEI, N. & JAVADIAN, N. (2009) A genetic algorithm to solve the storage space allocation problem in a container terminal. *Computers & Industrial Engineering,* 56**,** 44-52.

BERKEY, J. O. & WANG, P. Y. (1987) Two-Dimensional Finite Bin-Packing Algorithms. *Journal of the Operational Research Society,* 38**,** 423-429.

BERRADA, M. & STECKE, K. E. (1986) A hierarchical approach to solving machine grouping and loading problems of flexible manufacturing systems. *European Journal of Operational Research,* 24**,** 369-378.

BISH, E. K. (2003) A multiple-crane-constrained scheduling problem in a container terminal. *European Journal of Operational Research,* 144**,** 83-107.

BROWN, G. G., CORMICAN, K. J., LAWPHONGPANICH, S. & WIDDIS, D. B. (1997) Optimizing submarine berthing with a persistence Incentive. *Naval Research Logistics,* 44**,** 301-318.

BROWN, G. G., LAWPHONGPANICH, S. & THURMAN, K. P. (1994) Optimizing ship berthing. *Naval Research Logistics,* 41**,** 1-15.

CHEN, T. (1999) Yard operations in the container terminal - a study in the "unproductive moves". *Maritime Policy & Management,* 26**,** 27-38.

CORDEAU, J. F., LAPORTE, G., LEGATO, P. & MOCCIA, L. (2005) Models and tabu search heuristics for the berth-allocation problem. *Transportation Science,* 39**,** 526-538.

CULLINANE, K., WANG, T. F. & CULLINANE, S. (2004) Container Terminal Development in Mainland China and Its Impact on the Competitiveness of the

Port of Hong Kong *Transport Reviews,* 24**,** 33-56.

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

DAGANZO, C. F. (1990) CRANE PRODUCTIVITY AND SHIP DELAY IN PORTS. *Transportation Research Record* 1-9.

DAI, J., LIN, W., MOORTHY, R. & TEO, C.-P. (2008) Berth Allocation Planning Optimization in Container Terminals. *Supply Chain Analysis.*

DANTZIG, G. B. (1997) *Linear programming,* Edited by M. N. Thapa. New York, Springer.

DE CASTILLO, B. & DAGANZO, C. F. (1993) Handling strategies for import containers at marine terminals. *Transportation Research Part B: Methodological,* 27**,** 151-166.

DEKKER, R., VOOGD, P. & VAN ASPEREN, E. (2006) Advanced methods for container stacking. *OR Spectrum,* 28**,** 563-586.

EGBELU, P. J. (1993) Positioning of automated guided vehicles in a loop layout to improve response time. *European Journal of Operational Research,* 71**,** 32-44.

FRIEDMAN, W. D. (1992) A comprehensive approach to container terminals planning: striking a balance. *ports',* 92**,** 29-42.

GADEMANN, A. J. R. M. & VAN DE VELDE, S. L. (2000) Positioning automated guided vehicles in a loop layout. *European Journal of Operational Research,* 127**,** 565-573.

GAREY, M. R. (1979) *Computers and intractability : a guide to the theory of NP-completeness,* New York :, W. H. Freeman.

GIALLOMBARDO, G., MOCCIA, L., SALANI, M. & VACCA, I. (2008) The Tactical Berth Allocation Problem WITH Quay Crane Assignment AND Transshipment-related Quadratic Yard Costs. *European Transport Conference* Italy.

GOODCHILD, A. V. & DAGANZO, C. F. (2005) Crane Double Cycling in Container Ports: Affect on Ship Dwell Time. Berkeley, Institute of Transportation Studies, UC Berkeley.

GOODCHILD, A. V. & DAGANZO, C. F. (2007) Crane double cycling in container ports: Planning methods and evaluation. *Transportation Research Part B: Methodological,* 41**,** 875-891.

GRUNOW, M., GÜNTHER, H.-O. & LEHMANN, M. (2007) Strategies for dispatching AGVs at automated seaport container terminals. *Container Terminals and Cargo Systems.*

GUAN, Y. & CHEUNG, R. (2004) The berth allocation problem: models and solution methods. *OR Spectrum,* 26**,** 75-92.

GÜNTHER, H.-O. & KIM, K.-H. (2006) Container terminals and terminal operations. *OR Spectrum,* 28**,** 437-445.

HAN, Y., LEE, L., CHEW, E. & TAN, K. (2008) A yard storage strategy for minimizing traffic congestion in a marine container transshipment hub. *OR Spectrum,* 30**,** 697-720.

HANSEN, P., OGUZ, C. & MLADENOVIC, N. (2008) Variable neighborhood search for minimum cost berth allocation. *European Journal of Operational Research,* 191, 636-649.

HEE, K. M. V. & WIJBRANDS, R. J. (1988) Decision support system for container terminal planning. *European Journal of Operational Research,* 34, 262-272.

HOSHINO, S., OTA, J., SHINOZAKI, A. & HASHIMOTO, H. (2006) Design of an AGV transportation system by considering management model of an ACT. IOS PRESS, I. (Ed.) *Intelligent Autonomous Systems.*

HOSHINO, S., OTA, J., SHINOZAKI, A. & HASHIMOTO, H. (2007) Improved design methodology for an existing automated transportation system with automated guided vehicles in a seaport container terminal *Advanced Robotics,* 21, 371-394.

IMAI, A, NAGAIWA, K, I. & CHAN WENG, T. A. T. (1997) Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation,* 31, 75-94.

IMAI, A., CHEN, H. C., NISHIMURA, E. & PAPADIMITRIOU, S. (2008) The simultaneous berth and quay crane allocation problem. *Transportation Research Part E: Logistics and Transportation Review,* 44, 900-920.

IMAI, A., NISHIMURA, E., HATTORI, M. & PAPADIMITRIOU, S. (2007) Berth allocation at indented berths for mega-containerships. *European Journal of Operational Research,* 179, 579-593.

IMAI, A., NISHIMURA, E. & PAPADIMITRIOU, S. (2001) The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological,* 35, 401-417.

IMAI, A., NISHIMURA, E. & PAPADIMITRIOU, S. (2003) Berth allocation with service priority. *Transportation Research Part B: Methodological,* 37, 437-457.

IMAI, A., SASAKI, K., NISHIMURA, E. & PAPADIMITRIOU, S. (2006) Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks. *European Journal of Operational Research,* 171, 373-389.

IMAI, A., SUN, X., NISHIMURA, E. & PAPADIMITRIOU, S. (2005) Berth allocation in a container port: using a continuous location space approach. *Transportation Research Part B: Methodological,* 39, 199-221.

JOHNSON, D. S., DEMERS, A., ULLMAN, J. D., GAREY, M. R. & GRAHAM, R. L. (1974) Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms. *SIAM Journal on Computing,* 3, 299-325.

KIM, K. H. (1995) Positioning of automated guided vehicles in a loop layout to minimize the mean vehicle response time. *International Journal of Production Economics,* 39, 201-214.

KIM, K. H. (1997) Evaluation of the number of reshuffles in storage yards. *Computers & Industrial Engineering,* 32, 701-711.

KIM, K. H. & BAE, J. W. (1998) Re-marshaling export containers in port container terminals. *Computers & Industrial Engineering,* 35, 655-658.

KIM, K. H. & KIM, H. B. (1998) The optimal determination of the space requirement and the number of transfer cranes for import containers. *Computers & Industrial Engineering,* 35**,** 427-430.

KIM, K. H. & KIM, H. B. (1999a) Segregating space allocation models for container inventories in port container terminals. *International Journal of Production Economics,* 59**,** 415-423.

KIM, K. H. & KIM, H. B. (2002) The optimal sizing of the storage space and handling facilities for import containers. *Transportation Research Part B: Methodological,* 36**,** 821-835.

KIM, K. H. & KIM, K. Y. (1999b) Routing straddle carriers for the loading operation of containers using a beam search algorithm. *Computers & Industrial Engineering,* 36**,** 109-136.

KIM, K. H., LEE, K. M. & HWANG, H. (2003) Sequencing delivery and receiving operations for yard cranes in port container terminals. *International Journal of Production Economics,* 84**,** 283-292.

KIM, K. H. & MOON, K. C. (2003) Berth scheduling by simulated annealing. *Transportation Research Part B: Methodological,* 37**,** 541-560.

KIM, K. H. & PARK, K. T. (2003) A note on a dynamic space-allocation method for outbound containers. *European Journal of Operational Research,* 148**,** 92-101.

KIM, K. H. & PARK, Y.-M. (2004) A crane scheduling method for port container terminals. *European Journal of Operational Research,* 156**,** 752-768.

KIM, K. Y. & KIM, K. H. (1999c) A routing algorithm for a single straddle carrier to load export containers onto a containership. *International Journal of Production Economics,* 59, 425-433.

KIMEMIA, J. & GERSHWIN, S. B. (1983) An Algorithm for the Computer Control of a Flexible Manufacturing System. *IIE Transactions,* 15, 353-362.

KU, L., LEE, L., CHEW, E. & TAN, K. (2010) An optimisation framework for yard planning in a container terminal: case with automated rail-mounted gantry cranes. *OR Spectrum* 32, 519-541.

KUSIAK, A. (1986) Application of operational research models and techniques in flexible manufacturing systems. *European Journal of Operational Research,* 24, 336-345.

LAI, K. K. & SHIH, K. (1992) A study of container berth allocation. *Advanced Transportation,* 26, 45-60.

LE-ANH, T. & DE KOSTER, M. B. M. (2006) A review of design and control of automated guided vehicle systems. *European Journal of Operational Research,* 171, 1-23.

LEE, D.-H., CAO, Z. & MENG, Q. (2007) Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. *International Journal of Production Economics,* 107, 115-124.

LEE, L., CHEW, E., TAN, K. & HAN, Y. (2006) An optimization model for storage yard management in transshipment hubs. *OR Spectrum,* 28, 539-561.

LEGATO, P. & MAZZA, R. M. (2001) Berth planning and resources optimisation at a container terminal via discrete event simulation. *European Journal of Operational Research,* 133**,** 537-547.

LI, C.-L., CAI, X. & LEE, C.-Y. (1998) Scheduling with multiple-job-on-one-processor pattern. *IIE Transactions,* 30**,** 433-445.

LIANG, C., HUANG, Y. & YANG, Y. (2009) A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Computers & Industrial Engineering,* 56**,** 1021-1028.

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

LIU, J., WAN, Y.-W. & WANG, L. (2006) Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. *Naval Research Logistics,* 53**,** 60-74.

LODI, A., MARTELLO, S. & VIGO, D. (2002) Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics,* 123**,** 379-396.

MAIMON, O. Z. & GERSHWIN, S. B. (1988) Dynamic Scheduling and Routing for Flexible Manufacturing Systems that Have Unreliable Machines. *Operations Research,* 36**,** 279-292.

MOORTHY, R. & TEO, C.-P. (2007) Berth management in container terminal: The template design problem. *Container Terminals and Cargo Systems.*

MURTY, K. G. (1983) *Linear Programming* New York, Wiley.

NG, W. C. (2005) Crane scheduling in container yards with inter-crane interference.

European Journal of Operational Research, 164, 64-78.

NG, W. C. & MAK, K. L. (2005a) An effective heuristic for scheduling a yard crane to handle jobs with different ready times. *Engineering Optimization,* 37, 867 - 877.

NG, W. C. & MAK, K. L. (2005b) Yard crane scheduling in port container terminals. *Applied Mathematical Modelling,* 29, 263-276.

NG, W. C. & MAK, K. L. (2006) Quay crane scheduling in container terminals *Engineering Optimization,* 38, 723-737.

NG, W. C., MAK, K. L. & LI, M. K. (2010) Yard planning for vessel services with a cyclical calling pattern. *Engineering Optimization,* 42, 1039-1054.

NISHIMURA, E., IMAI, A. & PAPADIMITRIOU, S. (2001) Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research,* 131, 282-292.

OZPEYNIRCI, S. & AZIZOGLU, M. (2009) A Lagrangean relaxation based approach for the capacity allocation problem in flexible manufacturing systems. *Journal of the Operational Research Society,* 61, 872-877.

PAPADIMITRIOU, C. H. & STEIGLITZ, K. (1982) *Combinatorial Optimization: Algorithms and Complexity,* Englewood Cliffs, NJ, Prentice-hall.

PARK, K. T. & KIM, K. H. (2002) Berth Scheduling for Container Terminals by Using a Sub-Gradient Optimization Technique. *The Journal of the Operational Research Society,* 53, 1054-1062.

PARK, Y.-M. & KIM, K. H. (2003) A scheduling method for Berth and Quay cranes. *OR Spectrum,* 25**,** 1-23.

PETERING, M. E. H. & MURTY, K. G. (2009) Effect of block length and yard crane deployment systems on overall performance at a seaport container transshipment terminal. *Computers & Operations Research,* 36**,** 1711-1725.

QIU, L., HSU, W.-J., HUANG, S.-Y. & WANG, H. (2002) Scheduling and routing algorithms for AGVs: a survey. *International Journal of Production Research,* 40**,** 745 - 760.

SABUNCUOGLU, I. & HOMMERTZHEIM, D. L. (1992) Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system. *International Journal of Production Research,* 30**,** 1059-1079.

SHANTHIKUMAR, J. G. & STECKE, K. E. (1986) Reducing work-in-process inventory in certain classes of flexible manufacturing systems. *European Journal of Operational Research,* 26**,** 266-271.

SIMCHI, L., DAVID (1994) New worst-case results for the bin-packing problem. *Naval Research Logistics,* 41**,** 579-585.

STAHLBOCK, R. & VOß, S. (2008) Operations research at container terminals: a literature update. *OR Spectrum,* 30**,** 1-52.

STECKE, K. E. (1983) Formulation and Solution of Nonlinear Integer Production Planning Problems for Flexible Manufacturing Systems. *Management Science,* 29**,** 273-288.

STECKE, K. E. & MORIN, T. L. (1985) The Optimality of Balancing Workloads in Certain Types of Flexible Manufacturing Systems. *European Journal of Operational Research,* 20, 68-82.

STECKE, K. E. & SOLBERG, J. J. (1981) Loading and control policies for a flexible manufacturing system. *International Journal of Production Research,* 19, 481-490.

STEENKEN, D., VOß, S. & STAHLBOCK, R. (2004) Container terminal operation and operations research - a classification and literature review. *OR Spectrum,* 26, 3-49.

STEENKEN, D., WINTER, T. & ZIMMERMANN, U. T. (2001) Stowage and Transport Optimization in Ship Planning. IN GRÖTSCHEL M, K. S. O., RAMBAU J (Ed.) *Online optimization of large scale systems.* Berlin, Springer.

TALEB-IBRAHIMI, M., DE CASTILHO, B. & DAGANZO, C. F. (1993) Storage space vs handling work in container terminals. *Transportation Research Part B: Methodological,* 27, 13-32.

VIS, I. F. A. & DE KOSTER, R. (2003) Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research,* 147, 1-16.

WILSON, I. D. & ROACH, P. A. (1999) Principles of Combinatorial Optimization Applied to Container-Ship Stowage Planning. *Journal of Heuristics,* 5, 403-418.

ZHANG, C. (2000) Resource planning in container storage yards. *Industrial Engineering and Engineering Management.* Hong Kong, Hong Kong

University of Science and Technology.

ZHANG, C., LIU, J., WAN, Y.-W., MURTY, K. G. & LINN, R. J. (2003) Storage space allocation in container terminals. *Transportation Research Part B: Methodological,* 37, 883-903.

ZHANG, C., WAN, Y.-W., LIU, J. & LINN, R. J. (2002) Dynamic crane deployment in container storage yards. *Transportation Research Part B: Methodological,* 36, 537-555.