UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

A HYBRID APPROACH FOR CONTAINER TERMINAL OPERATIONS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

BY

KANG-HUNG YANG
Norman, Oklahoma
2008

A HYBRID APPROACH FOR CONTAINER TERMINAL OPERATIONS

A DISSERTATION APPROVED FOR THE
SCHOOL OF INDUSTRIAL ENGINEERING

BY

Dr. Yongpei Guan, Chair

Dr. Mary Court

Dr. Hillel J. Kumin

Dr. P. Simin Pulat

Dr. Sridhar Radhakrishnan

This is dedicated to my Mother, my Father, my Sisters, my Father in law, my Mother in law, my lovely kids, Shao-yen and Shao-yi, and the most important person in my life, Chen-ya.

Acknowledgments

I would like to express my gratitude to my advisor, Professor Yongpei Guan for his support and guidance to inspire me to finish my PhD degree. I also think Professor Mary Court, Professor Hillel Kumin, Professor Simin Pulat and Professor Sridhar Radhakrishnan for their valuable suggestions to my dissertation work.

Special thanks my “old” friend, Jim Vernon, for always editing my paper, dissertation, and so on, and teaching me about American culture and English.

I am also grateful to all the people and friends in OU IE. To Professor Randa Shehab, Professor Chen Ling, Professor Kash Barker, Ms. Cheryl Carney, Ms. Jean Shingledecker, Ms. Amy Piper, Zhili Zhou, Jiahui Wang, Kitti Setavoraphan, thanks for your support.

To those friends far away but always care about me, Dr. Zsuhin Chuang, Ying Sun, Dr. Hwa Chien, Jennifer Chen, Meng-chu Hsiao.

I can not achieve PhD without all of you.

November 08, 2008

Contents

List of Figures

List of Tables

Abstract

The container trade has increased dramatically in the late decade. Forecast projections indicate that this trend will continue in the future. Thus, effective and efficient operations become more essential for container terminals in the transportation network. This study aims to analyze different operational problems in container terminals to provide effective and efficient policies for on-site operations. The three main topics in this study are the berth allocation problem (BAP), the quay crane scheduling problem (QCSP), and the inspection problem. Instead of analyzing these problems separately, I study the combined problems. The first study addresses BAP and the inspection problem, the second study addresses QCSP, and the third study addresses the combined problem of BAP and QCSP. The first study estimates a reasonable service rate for the inspection operation, so that the inspection will not be the bottleneck of the system. Theoretical lower bound of the inspection service rate is provided through different deterministic berthing heuristics. For the stochastic job processing time cases, the inspection service rate can be acquired by a simulation model with a heuristic approach. In this study, the simulation approach and the heuristic approach are combined by the embedded simulation technique, and this combined approach is successfully applied for BAP with inspection. The second study addresses QCSP. I consider the problem that has one vessel and multiple cranes. Exact and heuristic solution approaches are developed for the problem. A mathematical model with a time-space network flow sub-structure and non-crossing constraints is established for small size problems. Lagrangian relaxation approach is used to obtain tight lower bounds and near-optimal solutions for medium size problems. For large size problems, I adapt the heuristic approach. The theoretical analysis results show that the error bound of the designed heuristics is no more than 100%. Numerical experiments are performed to evaluate efficiency and effectiveness of the solution approaches for the problem. The third study focuses on a general problem that combines BAP and QCSP, named BAQCP. I develop a mathematical formulation and use a heuristic approach to solve BAQCP. Two heuristics, preemption and non-preemption, are developed. Both theoretical and numerical results demonstrate the efficiency and effectiveness of these two heuristics. In order to further improve the heuristic results, I combine the exact solution approach provided by the mathematical formulation and the heuristic approach. The heuristic solutions are set to be the initial inputs for the mathematical model. Numerical experiments show that this combined approach makes the improvement of the solution quality within a reasonable execution time, and that they can be applied for practical cases in the real world.

1 Introduction

1.1 Background

A container is a large, standardized box typically used to transport goods from one destination to another by seagoing vessels or trucks. Shipping goods in containers has increased dramatically in the past twenty years. The United Nations (2005) reports the growth of the container trade will continue to grow at least 3% until 2015. Therefore, a seaport plays an essential role in the global transportation network. Due to the high cost of transportation and cargo handling, designing an efficient scheduling of the loading, unloading, systematic storage processes, and adequate resources is extremely important. When cargo vessels enter the harbor, the operational process begins with the following procedures (shown in Figure 1): being assigned a berth at a dock, unloading the containers with cranes, and storage.

Figure 1: Operations in a container terminal

However, an inspection center becomes a potential bottleneck in the container terminal due to the enhancements on the inspection operations. The goals of this dissertation are to analyze the terminal system, to design approaches to evaluate system performance and to develop new methodologies that will improve the operations. In this study, I will focus on issues related to the berth allocation problem (BAP), the Quay Crane Scheduling Problem (QCSP), and the security inspection operation. Due to the essential role container trade plays in aspects of the U.S. economy, it is imperative to develop efficient and effective container terminal operations. Many container terminal operation studies examined berth allocation problems, crane assignment problems, and storage and stacking problems. However, limited studies solved both BAP and QCSP together. Previous researchers who studied BAP or QCSP normally assumed these two problems were independent, which did not match the real world. For example, in previous BAP studies, researchers assumed vessel processing time is pre-determined. Actually, the vessel processing time is related to the number of cranes working to unload the vessel. In this dissertation, BAP and QCSP are combined to form a new problem, i.e. an integrated problem of berth allocation and quay crane scheduling (BAQCP), to be studied.

Additionally, the previous research, which focused on BAP or QCSP, seldom consider a stochastic job processing time in the real world. For instance, a service rate of a crane is influenced by many factors, such as the varying skills of the operators or the different types of the machines. In order to consider those uncertain factors, BAP or QCSP require both the operations research approach and the simulation approach. In this dissertation, an important study topic focuses on combining BAP with a security inspection in the container terminal. The analysis of inspection operations is discussed to find the proper service rate for an inspection center so that the enhancement of the security inspection operation will not be the bottleneck in the container terminal.

1.2 Structure of the dissertation

This dissertation is divided into three sub-research stages, i.e., three-phase studies. The Phase I study integrates the problem of BAP and inspection operations that is solved by using the simulation and optimization approaches together. The combined approach is studied to find the reasonable inspection service rate by different berthing heuristics, and the service rate is verified by the theoretical lower bound. The Phase

II study combines a special case of the BAP and QCSP, i.e., one vessel multiple cranes problem (QCSP itself), and is solved by exact and heuristic approaches that consider crane traveling time. Theoretical and numerical studies show that the proposed solution approaches were effective and efficient for this problem. The third phase is the extension of the second phase study. The more general case of the combination of BAP and QCSP is solved. Due to the complex nature of the problem, the heuristic approach is the primary research tool. Two heuristics are proposed and verified, both theoretically and numerically. Lastly, the model and heuristic approaches are combined to enhance the solutions quality.

The first study has two main objectives. First, simulation and optimization approaches are combined to find a way to solve BAP with inspection. Second, using combined approaches for deterministic and stochastic processing time cases, a lower bound of the theoretical inspection rate is provided. This analysis of the numerical experiments finds the reasonable inspection rate in order to avoid inefficient inspection operations. The following shows the main structure of the first study:

- Establish an embedded simulation frame for analyzing BAP with security inspection.
- Modify deterministic heuristic algorithms for stochastic job processing time scenarios.
- Derive the theoretical lower bound of an inspection rate from different berthing heuristics.
- Implement the heuristic algorithms to a simulation model by the embedded simulation technique and verify this combined approach for the BAP with inspection operations.
- Perform the numerical experiments to evaluate the validity of the combined

approach and suggest the adequate scheduling policies for the problem.

The second study is the pilot study of the third study, which targets the combined problem of the berth allocation and quay crane scheduling problems (BAQCP). This study focuses on the special case of BAQCP, i.e., one vessel multiple cranes case (QCSP). Distinguished from the most previous studies, the crane traveling factor is considered in the final analysis. The main objective of this study is to establish a solution approach for QCSP, which includes both exact and heuristic approaches. The main tasks in this study are:

- Establish a mathematical model with time-space network flow sub-structure to describe the QCSP and evaluate the capacities of the model.
- Develop a Lagrangian relaxation approach to solve QCSP for medium-size problems.
- Find the lower bound of the problem by Dynamic programming technique.
- Design two heuristics and derive the worst case bounds for the heuristics associated with the lower bound.
- Carry out numerical experiments, which compare the present model with other mathematical formulations, and evaluate the efficiency and the effectiveness of the exact approaches, the Lagrangian relaxation approach, as well as the heuristic approach for different size instances.

The purpose of the third study is to develop a solution approach to solve BAQCP. Due to the complex nature of the problem, it is difficult to solve the problem by the exact solution approach. Therefore, the heuristic approach is adapted as the main approach. The outlines of this study are:

- Extend the results in the Phase II study to build a mathematical model to describe BAQCP.
- Adapt the heuristic approach as the main analysis tool and design two types of heuristics.
- Derive worst case analysis for the two designed heuristics.
- Evaluate the heuristic approach numerically and enhance the heuristic approach by combining the model approach and the heuristic approach.

1.3 Dissertation contributions

The proposed contributions of this dissertation are:

1. With more demand for container security inspections in a container terminal, this study analyzes BAP with inspection operations, which container operations research seldom considers. The relationship between berth allocation and inspection is theoretically established, and the relationship is verified numerically by deterministic and stochastic processing time cases. This allows for the suggestion of proper operation policies for inspections to avoid creating a bottleneck in the container terminal.

2. Analyze QCSP with consideration of crane traveling time. The solution approach includes exact and heuristic approaches. The advantages of these approaches are: the situation or incident is easily described, non-crossing and non-interference constraints make the application of the model possible for real-valued problems, and the model can easily include or exclude crane traveling time effects. Additionally, the solution quality of the proposed heuristics is proved theoretically and numerically to demonstrate the effectiveness and efficiency of the solution approaches.

3. Develop a new mathematical model for BAQCP to describe this problem, analyze the model capabilities, establish solution approaches to solve this problem in deterministic cases, and ensure and evaluate the solution quality theoretically. Finally, I enhance and improve the solution quality by combining the model and the heuristic approaches. The main achievement of this study also includes finding the theoretical upper bound for the problem, which the previous research did not address.

2 Analysis of Berth Allocation and Inspection Operations in a Container Terminal

Abstract

Nowadays, approximately 90% of the world's cargos are moved by ships, and the majority of goods are transported by containers. Each year, around 200 million containers are transported between the world seaports. A container terminal becomes one of the important nodes in the transportation network and plays a significant role in the global supply chain. After the terrorism attack on September 11, 2001, container security issues aroused, especially in the United States. U.S. government proposed and implemented several measures, such as C-TPAT, CSI, and etc., to improve the security systems. However, inefficiency or failure of the container supply chain becomes one of the most concerned issues while implementing container security measures. This chapter studies the container inspection operations in a container terminal. Improper and inefficient inspection operations will make an inspection center to become a bottleneck in the container terminal. In this chapter, I analyze the container inspection system to find out the proper service rate so that the bottleneck can be avoided. I derive the theoretical lower bound of the inspection service rate of the operations and perform the simulation experiments to validate the the bound for both deterministic and stochastic cases.

2.1 Introduction

"The global supply chain is an international system that has evolved to make the transport freight throughout the world amazingly efficient" (i.e., see Simchi-Levi et al. (2002)). Maritime transportation is the most important component of the global supply chain, and highly related to the U.S. economy. According to the statistics of 2005 from U.S. Bureau of Transportation, 75% of goods imported to or exported from U.S. are transported by maritime transportation. Data from World Trade Organization 2004 showed that U.S. is the world's first importer and the second largest exporter. Some other historical statistics (i.e., see Roach (2003)) showed that around 90% of the world's cargo are moved by vessel and each year over 48 million full containers are transported between major ports in the world. For US trade, each year more than six million containers are offloaded at US ports and almost half of them arrives by vessels. Such a huge amount container transportation flow makes a container terminal play an important role in the supply chain transportation network. Therefore, the shipping containers and its transport system are vital components to the global supply chain (RAND, 2004).

Prior to 2001, port security measures mainly focused on reducing cargo theft, stowaways, and smuggling. Only 2% to 4% of several million containers shipped to U.S. were physically examined by U.S. Customs (i.e., see Thibault et al. (2006)). The low examination rate on the containers makes crime have a chance to smuggle weapons into the U.S. by shipping containers. After the terrorism attack of September 11 2001, security issues aroused, especially maritime security. A war game simulation held in 2002 for a major U.S. seaport under several terrorist scenarios (i.e., see Gerencser et al. (2003)) showed that if the attacks would have been successful, it will seriously harm U.S. economy and global trade.

In order to respond future possible terrorist threat, U.S. government adopted and implemented new technologies, regulations, and operating process and protocols. Most of the new measures focused on maritime shipping operations,which included two major categories: policy measures and technology measures. Willis and Ortiz

(2004) listed the measures as follows,

- Policy measures: Customs-Trade Partnership Against Terrorism (C-TPAT), Container Security Initiative (CSI), and Maritime Transportation Security Act of 2002 (MTSA).

- Technology measures: Operation Safe Commerce (OSC), Antitamper Seals, Radio-Frequency Identification, X-Ray and Gamma-Ray Scanning, Radiation Pagers, Portal Sensors, and Remote Monitoring.

C-TPAT is a voluntary program and encourage shippers and carriers to cooperate U.S. Customs and Border Protection (CBP). They will follow the best security practices for shipping containers and goods to U.S.. All those join the C-TPAT program will reward faster processing and avoid the inspection delay. The goal of CSI is to make it harder to ship illegal containers to the U.S. by inspecting the containerized cargo at the ports of the origin. MTSA encouraged national ports and carriers with U.S.-flagged vessels proposing and submitting security plans to U.S. Coast Guard. Technology measures are mainly used with the up-to-date technologies as an assistant tool to detect and inspect unusual shipments or illegal containerized cargo. For example, X-Ray and Gamma-Ray Scanning technologies allow CBP to non-contact inspect for a container's content. Any suspicious containers will be inspected further by CBP physically.

Willis and Ortiz (2004) concluded that security and efficiency are two distinct but interconnected issues in the global supply chain. Improvement on the efficiency of the system may or may not influence the security of the system. However, increasing a security level will reduce efficiency of the system. For example, increasing the number of containers to be inspected could cause delays for transporting container cargo, and further lead to negative economic effects. Sekine et al. (2006) used a simulation-based approach to find the balance of diverse of conflicting objectives, for instance, considering efficiency and security factors into the objective function. In their study, a real case was applied to verify the approach. Conclusions showed that this approach can be applied but needs improvement on the simplified first order model.

Although the demand of inspection operations is increasing recently, it is seldom systematically studied on how inspection operations affect the terminal daily operations. This chapter studies terminal operations with an inspection operation to find out the proper inspection service rate so that an inspection center will not become the bottleneck of the system. The remaining part of this chapter is organized as follows: In Section 2, I explain the overall terminal operations and the relationship between the berth allocation and the inspection operation. In Section 3, three berth allocation policies for the deterministic processing time cases are reviewed and utilized for the problem with stochastic processing time. In Section 4, I study how to embed these three berth allocation policies into a simulation framework to evaluate their performance for practical terminal operations. In Section 5, I perform the worst case analysis for the inspection rate for the deterministic processing time case. I estimate the ratio between the inspection service rate and quay crane service rate such that the inspection operation will not be a bottleneck, once each container is required to go through the inspection operation. In Section 6, I perform simulation study to test the performance of three proposed policies under the stochastic processing time setting. I also show the simulation experiment results to evaluate the inspection rate required under the stochastic processing time setting and to justify the theoretical analysis in Section 5. Finally, in Section 7, I present conclusions and suggest further study.

2.2 Terminal operations

Container terminals play an important role in the container transportation network. They are usually the nodes where transportation modes change in the intermodal freight flow transportation network system. Different container terminal may have different functions. Some terminals mainly serve as the hub to connect incoming ships and outgoing trains and trucks, or vice versa For instance, the port of Long beach. Some terminals mainly serve as the hub to transship containers from one ship to another, such as the port of Singapore. Although different terminals with different functions, most container terminals include inbound and outbound operations.

The inbound operations usually include all operations involved to unload a container from an incoming vessel/train/truck and store it in the storage yard. For instance, let I consider the case that the incoming transportation mode is vessel. First, a vessel needs to be berthed after it arrives at the terminal; second, quay cranes are arranged to unload containers from the vessel; third, an internal truck picks the containers to the inspection center and then to the storage yard; Finally, a container is unloaded from the internal truck and stored in the storage yard. The outbound operations follow a reverse direction except the berth allocation part. Similarly let us consider the case that outgoing transportation mode is also vessel. First, a vessel needs to be berthed after it arrives at the terminal; second, a container is loaded from the storage yard to an internal truck; third, the container is shipped by the internal truck to the berth place; finally, the container is loaded to the vessel for the final departure. The terminal operations are similar for the cases that incoming and outgoing transportation modes are train or truck instead of vessel, except that the berth allocation operations are not needed, which makes the terminal operations less complicated. Without loss of generality, in this chapter, I study the inbound operations with vessel as incoming transportation mode.

There have been extensive research on terminal operations. Recently, Stahlbock and Voβ (2008) did a literature review on terminal operations. They categorized literature in groups that include *ship planing process* (i.e., berth allocation and storage planing), *storage and stacking logistics*, *transport optimization* (i.e., quayside transport, landside transport and crane transport optimization), and *integrative approaches*. Of all the four research categories, integrative approaches are the most challenging problems because those consider various interconnected operations together. Now I need to consider extra inspection operation, which makes the problem even challenge to solve. This chapter will study berth allocation and inspection operation together. The relationship between berth allocation and inspection operation can be simplified as shown in Figure 2.

I define each berth contains one crane and consider one crane as one processor.

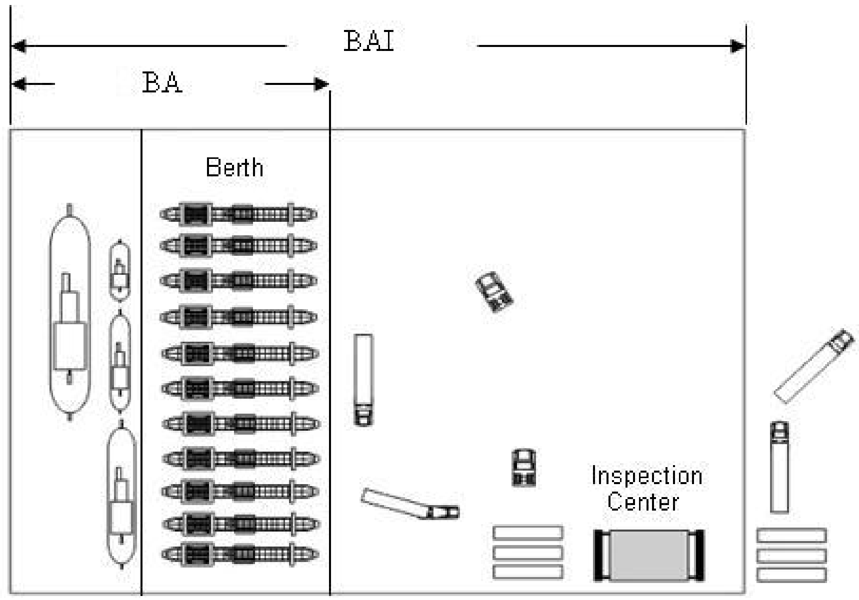

Figure 2: System of the berth allocation with inspection

Within this system, a container is first unloaded from a vessel. Then it is picked by a truck to the inspection center. Once the container completes the inspection, it will leave the system.

2.3 Berth allocation policies

The berth allocation problem can be formulated as a multiprocessor task scheduling (MTS) problem. An MTS problem is defined as one type of task scheduling problems in which each task is processed by multiple processors (machines) simultaneously and preemption is not allowed (i.e., see Drozdowski (1996)).

In the system described in this chapter, I want to let two major entities, vessels and containers, go through the system smoothly with minimum delay. The berth allocation and the inspection operation are two interlinked operations in the terminal area. First, I have to find an efficient berthing policy, which can allocate berths efficiently and vessels can leave the terminal as soon as possible with minimum delay. Also, an efficient service rate of the inspection center needs to be determined such that the inspection center will not be a bottleneck in the terminal area. In this section, I examine three heuristic algorithms for the stochastic processing time cases and in later sections, I apply all three heuristics as the berthing policies to estimate the service rate of the inspection center.

2.3.1 Time-space representation of a vessel

In the approach, I formulate a vessel as a time-space rectangle. The length of the rectangle represents the processing time and the width of the rectangle represents the length of the vessel in terms of number of berths. For example, a vessel with size 3 and processing time 6, assigned at time 2 and sections 2 to 4 can be represented as a time-space rectangle shown in Figure 3.

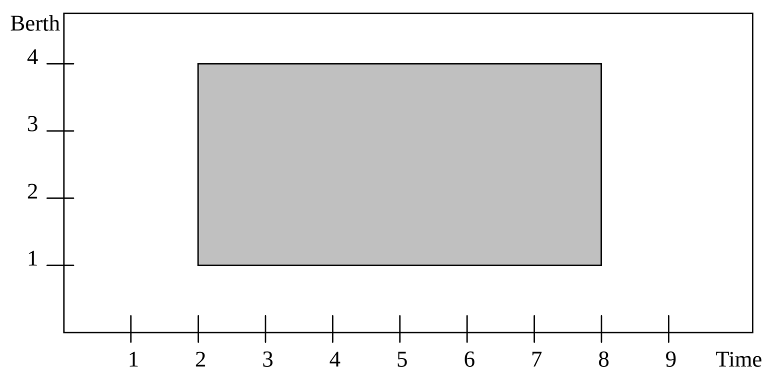

Figure 3: Time-space representation of a vessel

2.3.2 Guan et al.'s heuristic

In recent studies, there are three berthing heuristics, Guan et al.'s heuristic, Small processing time first (SPT) heuristic and Li et al.'s heuristic, developed to assign vessels along the berth. For all these case, the processing time for each vessel is considered as a constant parameter. In practice, due to crane service rate uncertainty and traffic within the terminal, the processing time may not be a constant. In this approach, I consider that the processing time for each vessel is stochastic and I use the expected processing time in the heuristics. Then, I compare the performance among these three heuristics under the stochastic process time setting. In the following part, I describe these three heuristics and later I show an example in Section 2.3.5.

In Guan et al.'s heuristic (i.e., see Guan et al. (2002)), each individual vessel is considered as one job. The heuristic includes three steps. The first step sorts all jobs by vessel's length and renumbers the job index. The second step groups jobs according to the limitation of the total berth length. The final step assigns vessels to the berths group by group. Corresponding to a vessel set $\{V_1, V_2, \cdots, V_N\}$, let P_i and L_i be the expected processing time and vessel length of $V_i, \forall 1 \leq i \leq N$ respectively. I assume $L_i's$ and $P_i's$ are agreeable. That is, if $L_i \leq L_j$, then $P_i \leq P_j$. The detailed step for Guan et al.'s heuristic can be described as follows.

Guan et al.'s heuristic:

Step 0: Sort and renumber the vessels by $P_1 \leq P_2 \leq \cdots \leq P_N$ and $L_1 \leq L_2 \leq \cdots \leq L_N$. Initialize $I = 1$.

Step 1: Let $\{V_\ell, V_{\ell+1}, \ldots, V_N\}$ be the set of the unscheduled vessels. Let

$$u = \max\{q| \sum_{j=\ell}^{q} L_j \leq S \text{ and } q \leq N\}$$

in which S is the number of the berths, and N is the number of the vessels.

Set $G_I \leftarrow \{V_\ell, V_{\ell+1} \cdots, V_u\}$.

Step 2: For $r = \ell, \ell+1, \cdots, u$:

(a) If I is odd, then assign V_r to berths $S - \sum_{j=r}^{u} L_j + 1$, $S - \sum_{j=r}^{u} L_j + 2$,$\cdots$, $S - \sum_{j=r+1}^{u} L_j$

(b) If I is even, then assign V_r to berths $\sum_{j=r+1}^{u} L_j + 1$, $\sum_{j=r+1}^{u} L_j + 2$,$\cdots$, $\sum_{j=r}^{u} L_j$

Schedule V_r behind the existing scheduled jobs on these berths, and make it start as early as possible.

Step 3: Set $I = I + 1$. If there is no more unscheduled job, then stop, else go to Step 1.

2.3.3 SPT heuristic

SPT heuristic contains two main steps: the first step sorts the vessels from the smallest to the largest in terms of vessel's size and reindex the vessels; the second step assigns vessels sequentially by checking if there is enough available space at the earliest time for each vessel. The detailed step can be expressed as follows.

SPT heuristic:

Step 0: Sort and renumber vessels such that $P_1 \leq P_2 \leq \cdots \leq P_N$ and $L_1 \leq L_2 \leq \cdots \leq L_N$, in which $P_i's$ and $L_i's$ are agreeable. Let $\{R_1, R_2, \cdots, R_S\}$ represent berths 1 to S, and put all vessels into a vessel list $\{V_1, V_2, \cdots, V_N\}$. Initialize the earliest available time for each berth $w_1 = w_2 = \cdots = w_S = 0$

Step 1: For each step, take vessel V_k out of the unassigned vessel list where k is the smallest index in the unassigned vessel list and initialize $\prod = \{1, 2, \cdots, S\}$.

Step 2: Select a berth $\tilde{r} = \arg\min\{w_j : j \in \prod\}$. If $\tilde{r} + L_k - 1 \leq S$ and $w_j \leq w_{\tilde{r}}$ for $\tilde{r} \leq j \leq \tilde{r} + L_k - 1$, allocate the k^{th} vessel for the berths from $R_{\tilde{r}}$ to $R_{\tilde{r}+L_k-1}$. Update $w_j = w_{\tilde{r}} + P_j$ for $\tilde{r} \leq j \leq \tilde{r} + L_k - 1$. Else, let $\prod = \prod \setminus \{\tilde{r}\}$ and repeat Step 2 until V_k is assigned.

Step 3: Go to Step 1 and stop until all vessels are assigned.

2.3.4 Li et al.'s heuristic

In Li et al.'s heuristic, the vessels are sorted from the largest to the smallest and are assigned sequentially. Comparing to the above two algorithms, Li et al.'s heuristic tries to put a job as close to one of the two ends of the terminal as possible so that the middle part of the terminal is reserved for other unassigned vessels. From the largest to the smallest in the unassigned vessel list, to assign each vessel k, I check starting from the earliest available time slot to see if there is enough space to assign the current vessel k.

Let $C_{[1]}, C_{[2]}, \ldots, C_{[k-1]}$ represent the finish times of the first $k-1$ assigned vessels and $\Pi_{[i]k} = \{\gamma|$position is unoccupied throughout the entire time period$(C_{[i]}, C_{[i]} + P_k]$. By the definition, $\Pi_{[i]k}$ can be expressed as $[a_{[i]k1}, b_{[i]k1}] \cup [a_{[i]k2}, b_{[i]k2}] \cup \ldots \cup [a_{[i]kn_{[i]k}}, b_{[i]kn_{[i]k}}]$, where $0 \leq a_{[i]k1} < b_{[i]k1} < a_{[i]k2} < b_{[i]k2} < \ldots < a_{[i]kn_{[i]k}} < b_{[i]kn_{[i]k}} \leq S$, where S is the number of berths, and $n_{[i]k}$ is the number of spaces which are unoccupied throughout the entire time period $(C_{[i]}, C_{[i]} + P_k]$. To assign the current vessel k, the algorithm checks all possible positions in $\Pi_{[i]k}$ at each completion time $C_{[i]}$. Figure 4 shows an example that explains the variables used in Li et al.'s heuristic. In the example, $C_{[1]} = 4$ (the completion time of V_3), $C_{[2]} = 6$ (the completion time of V_2), $C_{[3]} = 6$ (the completion time of V_1), $C_{[4]} = 8$ (completion time of V_4), $C_{[5]} = 9$ (completion time of V_5), and $C_{[6]} = 10$ (completion time of V_6).

Now assume I want to add the seventh vessel (V_7) with the processing time $P_7 = 3$ and the length $L_7 = 2$. If I consider the finishing time $C_{[3]}$, the time period is $(C_{[3]}, C_{[3]} + P_3] = (6, 9]$ and there are two unoccupied berthing space ($n_{[3]7} = 2$). The corresponding available time slots are ($a_{[3]71} = 2$, $b_{[3]71} = 3$) and ($a_{[3]72} = 6$, $b_{[3]72} = 8$). Since $b_{[3]71} - a_{[3]71} < 2$ and $b_{[3]72} - a_{[3]72} = 8 - 6 = 2$, I can allocate the seventh vessel from berths 7 to 8 at time 6. The details of Li et al.'s heuristic are shown as follows (ref. Li et al. (1998)).

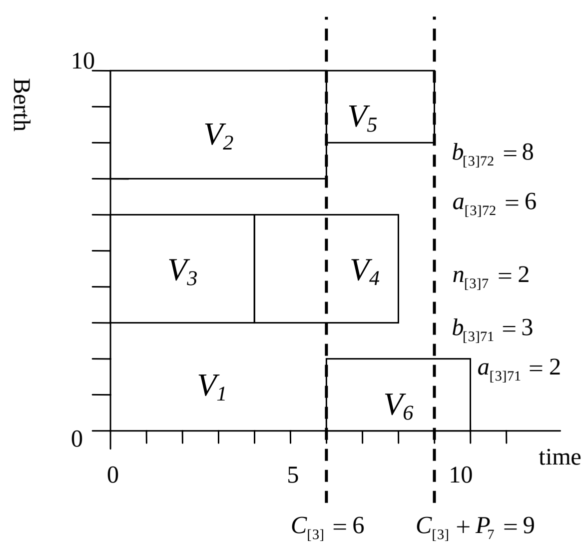

Figure 4: Example of $a_{[i]kn_{[i]k}}$, $b_{[i]kn_{[i]k}}$, and $n_{[i]k}$

Li et al.'s heuristic:

Step 0: Sort and renumber the vessels such that $P_1 \geq P_2 \geq \cdots \geq P_N$ and $L_1 \geq L_2 \geq \cdots \geq L_N$.

Step 1: Set $C_{[0]} \leftarrow 0$.

Step 2: For $k = 1, \cdots, N$ do:

Step 2-1: Set $i \leftarrow 0$

Step 2-2: Determine $n_{[i]k}$, and $a_{[i]kj}$, $b_{[i]kj}$ for $j = 1, \cdots, n_{[i]k}$;

$\triangle \leftarrow \max_{j=1,\cdots,n_{[i]k}}\{b_{[i]kj} - a_{[i]kj}\}$;

if $L_k \leq \triangle$ then

$l \leftarrow \min\{j | b_{[i]kj} - a_{[i]kj} \geq L_k$ and $1 \leq j \leq n_{[i]k}\}$;

$u \leftarrow \max\{j | b_{[i]kj} - a_{[i]kj} \geq L_k$ and $1 \leq j \leq n_{[i]k}\}$;

if $a_{[i]kl} \leq S - b_{[i]ku}$ then

schedule V_k to start at time $C_{[i]}$ between berths $a_{[i]kl}$ and $a_{[i]kl} + L_k - 1$;

and update $C_{[h]}$, $h = 1, \cdots, k$;

else

schedule V_k to start at time $C_{[i]}$ between position $b_{[i]ku} - L_k + 1$ and $b_{[i]ku}$;

and update $C_{[h]}$, $h = 1, \cdots, k$;

else

set $i \leftarrow i + 1$ and go to Step 2-2.

2.3.5 An example of three heuristics

Table 1 shows a group of 5 sorted vessels with fix processing time by their size. Assume there are 12 berths. The time-space graph for 5 vessels allocation by three algorithms are shown in Figure 5.

According to Guan et al.'s heuristic, vessels will be grouped first. Then, according to vessels' sizes and processing times, vessels will be allocated to different berths. SPT heuristic works similar to Guan et al.'s heuristic, but for each vessel group, the smaller index berths are always allocated for the smaller index vessels, which implies that the

Table 1: Example data information

j	1	2	3	4	5
Vessel size L_j	2	3	4	4	5
Processing time P_j	3	4	5	5	8

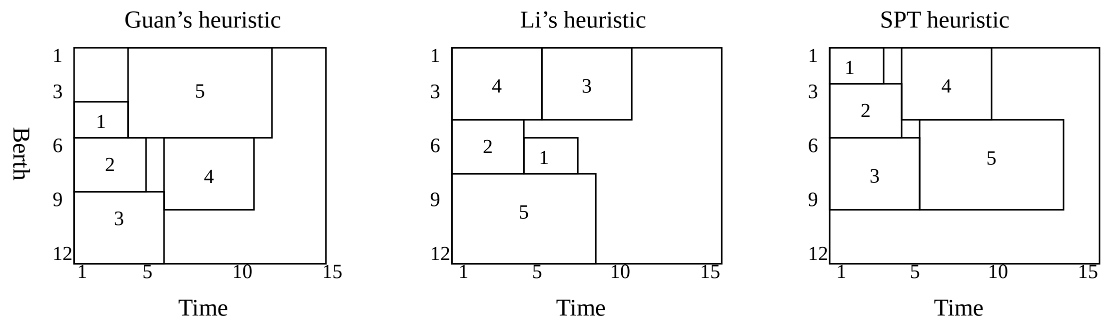

Figure 5: Examples of the three heuristics

makespan by SPT heuristic is larger than that by Guan et al.'s heuristic. The sorting rule of Li et al.'s heuristic is different from the other two heuristics, the smaller the vessel index, the larger processing time of the vessel. Therefore, in this example, the fifth vessel is assigned first, then the fourth vessel and so on. In terms of the makespan, Li et al.'s heuristic has the best performance of all these three heuristic. However, for stochastic processing time cases, it is not easy to judge which heuristic is the best because different parameter settings will influence the performance of these three heuristics.

2.4 Analysis of inspection service rate

In the MTSI system, I treat a berth as a processor. In current system, there are S berths and one inspection center in the terminal. I expect that if the service rate of the inspection center is S times of the service rate of the processors, the inspection center will not be the bottleneck of the system, that is, all containers can go through the system without any delay. However, in the system, I find one processor is often waiting for others, which indicates the service rate of the inspection is not necessary S times of the service rate of a processor. To explain this phenomenon, I use time-space graph representation and find out the vessel rectangles can not fully occupy the time-space domain. One example of a six vessels case is shown in Figure 6.

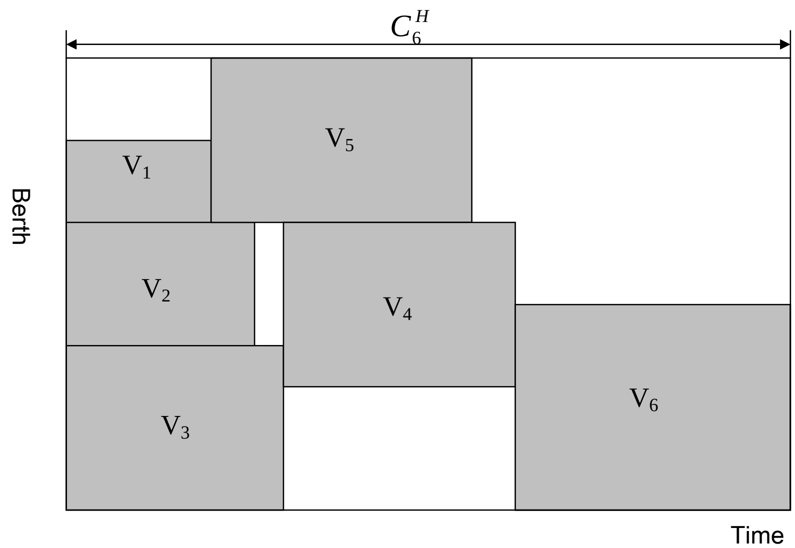

Figure 6: Space-time representation of a 6-vessels case by Guan et al.'s heuristic

In order to estimate the proper service rate of the inspection center, I establish the relationship between the service rate of a processor and the service rate of the inspection center. If I know the service rate of a processor and number of the processors, I can estimate the service rate of the inspection center by the established relationship. The following parameters are defined for the formula,

***Total Area*:** A_1, the product of makespan and total number of processors.

***Solid Area*:** A_2, the summation of the product of job size times processing time of each vessel.

S**:** number of processors

ρ**:** The ratio of the occupied area and the total area.

γ_p**:** Service rate of a processor.

γ_i: Service rate of an inspection center.

If I know S, makespan, service rate of a processor, and length and processing time of each vessel, I can estimate the service rate of the inspection center,

$$\gamma_i = S\rho\gamma_p = S\frac{A_2}{A_1}\gamma_p \tag{1}$$

In reality, a vessel-incoming pattern is not easily predictable. I can estimate service rate of the inspection center while there is a sufficient large number of vessels coming to the system. In the following section, the lower bound of ρ is derived by Guan et al.'s heuristic, Li et al.'s heuristic, and SPT heuristic.

2.4.1 Service rate for inspection by Guan et al.'s heuristic

In Guan et al. (2002), a relaxed problem is provided for the berth allocation problem. The relaxed problem indicates that I use multiple one unit size vessel rectangles replace the original vessel rectangle. Figure 7 demonstrates how I can replace a vessel rectangle by multiple one unit length rectangles. Assume a vessel j is with size L_j and processing time P_j. The relaxation means for every V_j (a vessel is treated as one job), I replace it by L_j identical jobs $\{V_{j1}, V_{j2}, \cdots, V_{jL_j}\}$ with each of unit length. Figure 8 shows an relaxed example of the case in Figure 6.

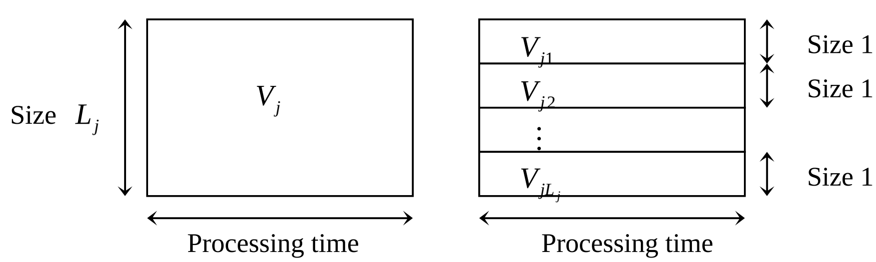

Figure 7: Relaxation demonstration

In Guan et al. (2002), it is shown that the relaxed problem provides a lower bound in terms of the total completion time for the proposed heuristics in Section 2.3.2. Especially in Guan et al. (2002), the following conclusion holds.

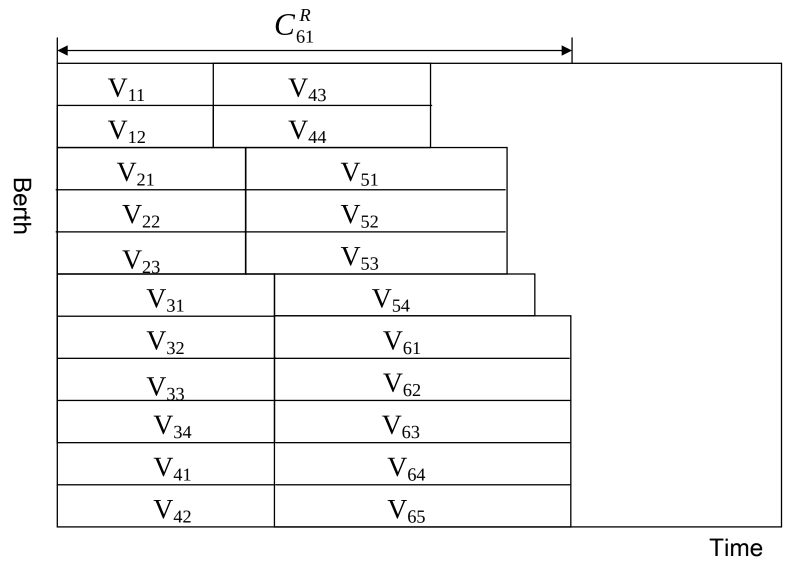

Figure 8: Relaxation of a 6-vessels case in Figure 6

Lemma 1 *The completion time of V_j in the heuristic solution (i.e., C_j^H) is at most twice the completion time of the first relaxed vessel rectangle of V_j in the relaxed problem (i.e., C_{j1}^R) for each $j = 1, 2, \cdots, N$.*

Proposition 1 *If the processing time for each vessel P_i is bounded as a constant number, then the ratio between the solid area and the total area for Guan et al.'s heuristic converges to one half asymptotically as the number of vessels goes to infinity.*

Proof: It can be first observed that the solid area for the original problem is the same as the total occupied area for the relaxed problem. In general, based on the relaxed problem, as shown an example in Figure 8, I have

$$\sum_{i=1}^{N} P_i L_i \geq S(C_{N1}^R - \max\{P_i\})$$

$$\text{and} \sum_{i=1}^{N} P_i L_i/(C_N^H S) \geq S(C_{N1}^R - \max\{P_i\})/(C_N^H S) = C_{N1}^R/C_N^H - \max\{P_i\}/C_N^H.$$

Also, according to Lemma 1, I have $C_{N1}^R/C_N^H \geq 1/2$. Then, $\lim_{N\to\infty} \sum_{i=1}^N P_i L_i/(C_N^H S) \geq \lim_{N\to\infty}(C_{N1}^R/C_N^H - \max\{P_i\}/C_N^H) = 1/2$ since C_N^H tends to be infinity and P_i is a finite number. Therefore, the conclusion holds. □

2.4.2 Service rate for inspection by Li et al.'s heuristic

In Li et al. (1998), the worst case bound is derived for their heuristic. Here I can use the similar concept and obtain the following proposition.

Proposition 2 *The ratio between the solid area and the total area is at least one half for Li et al.'s heuristics.*

Proof: In Li et al. (1998), it is shown that the ratio between the makespan Z_f^H and the lower bound Z_f^* is greater or equal to 1/2. It is shown that $Z_f^* \geq R_1+R_2+R_3 > 0.5Z_f^H$, where

$$\begin{aligned} R_1 &= \int_0^{t_n} (\text{total size of the vessels being processed at time } t)\ \mathrm{d}t, \\ R_2 &= \int_{t_n}^{Z_f^H - t_n} (\text{total size of the vessels being processed at time } t)\ \mathrm{d}t, \\ R_3 &= \int_{Z_f^H - t_n}^{Z_f^H} (\text{total size of the vessels being processed at time } t)\ \mathrm{d}t. \end{aligned}$$

In Li et al. (1998), the total length of the space is only one unit. In the case, the total berth length is S. But the conclusion still holds. I can modify the formula derived in Li et al. (1998) to $Z_f^* \geq R_1+R_2+R_3 > 0.5Z_f^H S$. Also, I can see that the *Solid Area* $A_2 = R_1 + R_2 + R_3$ and *Total Area* $A_1 = Z_f^H S$. Therefore,

$$\begin{aligned} \rho &= A_2/A_1 \\ &= (R_1 + R_2 + R_3)/(Z_f^H S) \\ &> 1/2. \end{aligned}$$

The conclusion holds. □

2.4.3 Service rate for inspection by SPT heuristic

The vessel assignments by SPT is similar to Guan et al.'s heuristic. The difference is that vessels in the same group for SPT will always be assigned from the smallest to the largest index berths. For instance, the smallest vessel in one group by SPT heuristic is always assigned from the first berth. Vessels are grouped as shown the third time-space diagram in Figure 5. In the following, I also show the ratio between the solid area and the total area is bounded below.

Proposition 3 *If the processing time for each vessel P_i is bounded as a constant number, then the lower bound of the ratio between the solid area and the total area for SPT heuristic converges to 0.4 asymptotically as the number of vessels goes to infinity.*

Proof: I consider two cases to analyze the worst case bound of the inspection service rate by SPT heuristic while there is an infinite number of vessels: 1) $\max_{i \in N} L_i \leq 0.5S$ and 2) $\max_{i \in N} L_i > 0.5S$. First of all, due to the reason that vessel processing times and vessel lengthes are agreeable, I have that the start time for the $i+1^{th}$ group is larger than the finish time for the $i-1^{th}$ group.

Case 1: $\max_{i \in N} L_i \leq 0.5S$. In this case, except the last group, I have at least two vessels in each group. Figure 9 shows an example for this case. Each vessel group is expressed as the area between two dashed lines. An area of the i^{th} vessel group between two dashed lines can be divided into three parts. The first is the solid area occupied by vessel rectangles (denoted as a_i), the second is the blank area under the largest index of the vessel in that vessel group (denoted as b_{i1}), and the third is other blank area that not belongs to the b_{i1} (denoted as b_{i2}). I can observe that $b_{i2} \leq a_{i+1}$. The area of b_{i1} is smaller than the area of the smallest indexed vessel rectangle in the $(i+1)^{th}$ group. Therefore, $b_{i1} \leq 0.5a_i$ since there are at lease 2 vessels within each vessel group due to the assumption that $\max_{i \in N} L_i \leq 0.5S$.

By induction, assuming there are n groups, I have

$$\begin{aligned} b_{11} + b_{12} &\leq a_2 + \frac{1}{2}a_2 \\ b_{21} + b_{22} &\leq a_3 + \frac{1}{2}a_3 \\ &\cdots \\ b_{n1} + b_{n2} &\leq a_{n+1} + \frac{1}{2}a_{n+1}. \end{aligned}$$

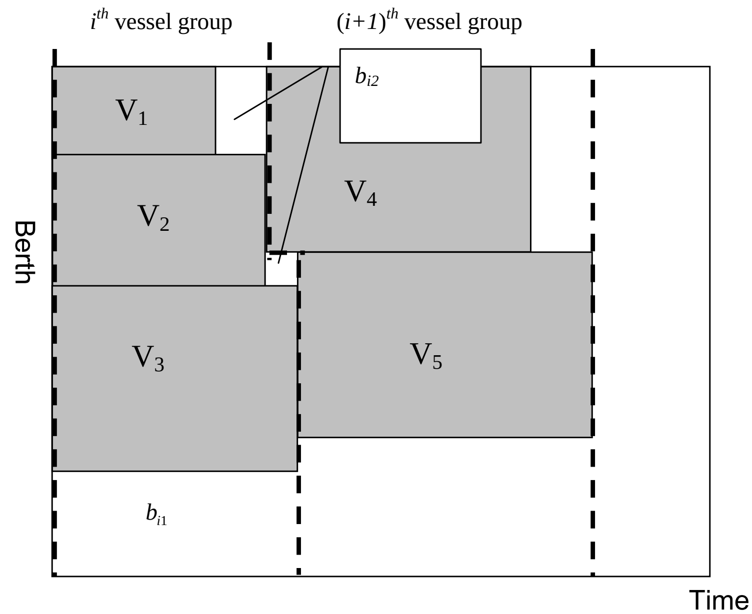

Figure 9: Case 1 for SPT

Accordingly, I have

$$\begin{aligned} \text{Total Blank Area} &\leq a_2 + a_3 + \cdots + a_{n+1} + 0.5(a_2 + a_3 + \cdots + a_{n+1}) \\ &\leq 3/2(a_1 + a_2 + \cdots + a_n) + 3/2(a_{n+1} - a_1). \end{aligned}$$

Then, I have

$$\begin{aligned} \frac{\text{Total Solid Area}}{\text{Total Area}} &= \frac{\text{Total Solid Area}}{\text{Total Solid Area} + \text{Total Blank Area}} \\ &= \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + \text{Total Blank Area}} \\ &\geq \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + 3/2\sum_{i=1}^{n} a_i + 3/2(a_{n+1} - a_1)} \\ &= \frac{2}{5} - \frac{3/5(a_{n+1} - a_1)}{\sum_{i=1}^{n} a_i + 3/2\sum_{i=1}^{n} a_i + 3/2(a_{n+1} - a_1)}. \end{aligned}$$

Therefore, I have

$$\begin{aligned} \lim_{n\to\infty} \frac{A_2}{A_1} &= \lim_{n\to\infty} \frac{\text{Total Solid Area}}{\text{Total Area}} \\ &= \lim_{n\to\infty} \frac{2}{5} - \frac{3/5(a_{n+1} - a_1)}{(\sum_{i=1}^{n} a_i + 3/2\sum_{i=1}^{n} a_i + 3/2(a_{n+1} - a_1)} \\ &= \frac{2}{5}. \end{aligned}$$

The final equation follows from the fact that $a_{n+1} - a_1$ is bounded by a constant number and $(\sum_{i=1}^{n} a_i + 3/2 \sum_{i=1}^{n} a_i + 3/2(a_{n+1} - a_1)) \to \infty$ as $n \to \infty$.

Case 2: $\max_{i \in N} L_i > 0.5S$. Under this case, there exists a number k^* such that the vessel groups after group k^* contain only one vessel with its size larger than $0.5S$ as shown in Figure 10. Therefore, I only have b_{i1} appears for $i > k^*$ and $b_{i2} = 0$ for $i > k^*$. Thus, $b_{i1} \leq a_{i+1}$ for $i > k^*$. For group k^*, I have that the blank area is no larger than $2a_{k^*+1}$. Thus, for groups after group k^* (including group k^*), I have

$$\text{Total Blank Area}_2 \leq a_{k^*+1} + a_{k^*+2} + \cdots + a_{n+1} + a_{k^*+1}.$$

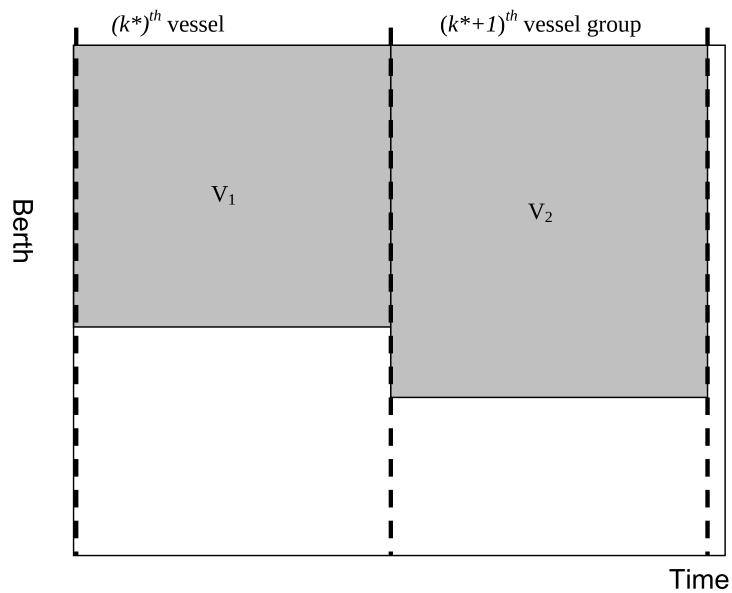

Figure 10: Case 2 for SPT

Before group k^* (including group k^*), I have at least two vessels assigned for each group. Therefore, I have the blank areas before group k^* described as follows:

$$\begin{aligned}\text{Total Blank Area}_1 &\leq a_2 + a_3 + \cdots + a_{k^*} + 0.5(a_2 + a_3 + \cdots + a_{k^*}) \\ &\leq 3/2(a_1 + a_2 + \cdots + a_{k^*-1}) + 3/2(a_{k^*} - a_1).\end{aligned}$$

Therefore, in total, I have

$$\begin{aligned}\text{Total Blank Area} &\leq a_2 + a_3 + \cdots + a_{n+1} + 0.5(a_2 + a_3 + \cdots + a_{k^*+1}) + 0.5a_{k^*+1} \\ &\leq 3/2(a_1 + a_2 + \cdots + a_n) + 1/2a_{k^*+1} + 3/2(a_{n+1} - a_1).\end{aligned}$$

Then, similarly as the proof in Case 1, I have

$$
\begin{aligned}
\lim_{n\to\infty} A_2/A_1 &= \lim_{n\to\infty} \text{Total Solid Area/Total Area} \\
&= \lim_{n\to\infty} 2/5 - \frac{1/5a_{k^*+1} + 3/5(a_{n+1} - a_1)}{\sum_{i=1}^{n} a_i + 3/2\sum_{i=1}^{n} a_i + 1/2a_{k^*+1} + 3/2(a_{n+1} - a_1)} \\
&= 2/5.
\end{aligned}
$$

Therefore, the conclusion holds. □

2.5 Embed heuristic policies into a simulation framework

For the stochastic processing time case, I run the experiments with the aid of embedded simulation modeling technique. In the approach, I embed berthing allocation policies into the simulation model. Figure 11 shows the concept of a two-level simulation model for implementing the approach for terminal operations.

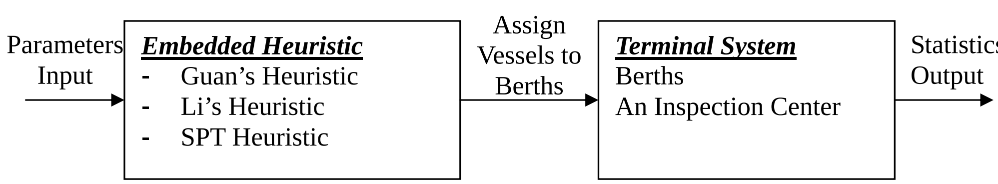

Figure 11: Conceptual simulation structure

I choose Visual SLAM and AweSim simulation software with C++ programming embedded to simulate the current terminal system. External C++ programming codes and an internal simulation model can be combined in the "EVENT" function node in the AweSim model. Once there is an entity going through the "EVENT" node, the simulation time will temporarily suspend. At this moment, the simulation process will go to the embedded codes, and then the simulation time will continue. "Event" node provides a way to embed the heuristics into a terminal simulation model. Figure 12 shows the detailed simulation model flow.

In the simulation flow, there are two systems, including the berthing system and the inspection system. The vessel entities will go through the berthing system, and

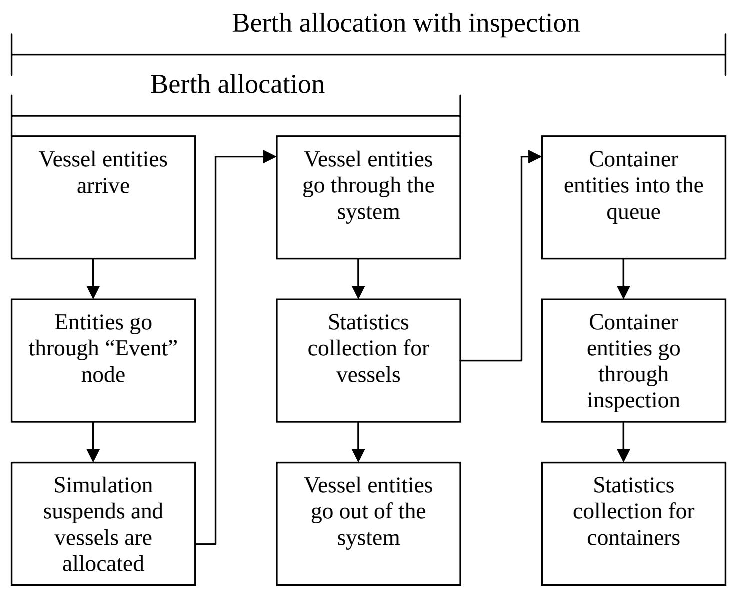

Figure 12: Simulation flow

the container entities will go through the inspection system. When the simulation starts, the vessel entities will be generated periodically in a batch. At this moment, an event (a batch of vessels arrive to the terminal) happens which triggers the embedded heuristic to allocate arriving vessels to different berths.

If the vessels are assigned and the corresponding berths are available, vessel entities will go though the berthing system, which also indicates cranes start to unload the containers. Otherwise, the vessels will wait in the queue for next assignments where some assigned berths are available. Once a vessel entity goes through the berthing system, it will trigger the other event to generate the container entities to go through the inspection system. The termination of generating the container entities is at the time the corresponding vessel entity leaves the system. After the vessel entities or container entities go through the systems, related statistics (average queue length, average time in the queue, and flow time) will be collected for comparisons of the three berthing heuristic. At the same time, I can evaluate the performance of the inspection center and find a proper service rate for the inspection center.

For a berth allocation simulation, the most challenging job in the simulation is to handle the case that each entity occupies several consecutive processors. In this study, I create a “copy entity” strategy to overcome the simulation difficulty. The main idea of the copy entity strategy is that several entities instead of the original vessel entity go through the simulation system. The number of the entities is the same number of the size of the vessel. The following are brief descriptions of the copy entity strategy.

1. Record a vessel's attributes that includes vessel generation time and size of the vessel once a vessel entity is generated.
2. Generate a number of copy entities in which the number is equal to the vessel's size. For example, if the length of a vessel is three, then 3 copy entities are generated each with length 1. All attributes of the copy entities are the same as those of the original vessel.
3. Replace the original entity by copy entities, and terminate the original entity to avoid double counting performance statistics.
4. Put copy entities in temporary stacks and let them wait to go through a berthing system.
5. Let copy entities start together once all designated berths are ready for those copy entities
6. The statistics of a vessel entity comes from the average of all corresponding copy entities.

2.6 Simulation experiments

In the simulation experiments, I first compare the performance among the three different heuristics for the berth allocation under stochastic processing time setting. I carry out numerical experiments for two cases. For the first case, all processors have the same service rate and the vessels with the same workload have the same processing time. In the second case, I assume that different processors have a slightly different service rate such that two vessels with the same workload may or may not have the same processing time.

2.6.1 Experiment designs for comparing three heuristic for berth allocation

In order to verify the efficiency of the berthing policies and the effectiveness of the service rate of the inspection center later on, I create a scenario with 12 processors, i.e. a terminal with 12 berths. For convenience, I assume the service rate of all berths is 1. After containers are unloaded from a vessel, a truck will pick up the unloaded container immediately and move them to the inspection center. Land limitations and truck capacities are not considered in this study.

For the deterministic cases, I generate 5 different runs. Each run contain 5000 vessels. The vessel sizes are from 1 unit length to 6 units length. The vessel size and the vessel processing time are agreeable, ranging from 2 time units to 10 time units. Every 6 minutes, the inspection center can check one container.

For stochastic cases, I create 3 different scenarios and name them *low loading*, *medium loading* and *high loading*. For each scenario, there are 10 runs. The simulation time horizon is 2 months. Vessels are arriving in the terminal for a batch every 3 days. For low loading scenario, every time, 20 to 60 vessels arrive. For medium loading scenario, 60 to 100 vessels arrive. For high loading scenario, 100 to 140 vessels arrive. The same vessel size has the same expected value of the processing time; however, a maximum of 10% variance of the processing time difference is allowed. The processing time of the vessel depends on the containers on the vessel. Unloading a container takes 10 minutes in this scenario.

2.6.2 Results comparison among three heuristics for berth allocation

Table 2 shows the simulation results for different scenarios. The simulation results show that Guan et al.'s heuristic approach outperforms the SPT heuristic and Li et al.'s heuristic. For Guan et al.'s heuristic and the SPT heuristic, the smaller size vessels are processed first rather than the lager size vessels, which leads to smaller queue length when comparing to Li et al.'s heuristic.

I also compare the standard deviation (the number within parentheses in Table 2). Overall, the standard deviations from Li et al.'s heuristic are smaller than those from Guan et al.'s heuristic and the SPT heuristic, which implies that the Li et al.'s algorithm is more robust and more stable than other two policies. If I consider both the mean and standard deviation results, Guan et al.'s algorithm are better than SPT heuristic and Li et al.'s heuristic.

Table 2: Simulation of 10-run results

Scenario	Average time in the queue (hours)			Flow time (hours)			Average queue length		
	Guan	Li	SPT	Guan	Li	SPT	Guan	Li	SPT
Low loading	84.67 (13.65)	106.93 (13.74)	88.94 (14.61)	202.36 (26.49)	259.01 (25.43)	211.87 (28.66)	198.4 (26.47)	255 (25.42)	207.89 (28.65)
Medium loading	487.7 (19.28)	526.16 (18.18)	492.03 (17.92)	486.85 (18.73)	549.13 (16.25)	491.45 (20.03)	482.93 (18.69)	545.09 (16.2)	487.54 (20)
High loading	908.38 (21.8)	960.98 (22.79)	912.88 (20.93)	580.26 (18.69)	654.93 (20.66)	579.62 (15.36)	576.36 (18.68)	650.84 (20.7)	575.71 (15.38)

2.6.3 Deterministic case verification

Figures 13, 14, and 15 show the simulation results for the ratio of *Solid Area* and *Total area* (ρ) v.s. number of vessels. I generate 5000 vessels for each simulation run. Initially, because of small number of vessels, ρ diverges at the very beginning and converges as the number of the vessels increases. When the number of the vessels reaches around 2000, the ratio tends to converge. For Guan et al.'s heuristic, the ρ value converges to 0.81; for Li et al.'s heuristic, the ρ value converges to 0.78; for the SPT heuristic, ρ doesn't converge. The reason comes from the behavior of the heuristic. Unlike Guan et al.'s heuristic and Li et al.'s heuristic, which try to make the vessel rectangles compact as possible. The SPT heuristic always assign the smallest index vessel in a vessel group from the smallest index berth first. Therefore, it is difficult to converge to a fixed values for different instances by SPT heuristic. Instead, instances reach certain fix values. For the current system, if the service rate of a processor is 1, then the service rate of the inspection center is around $12 \times 1 \times 0.81 = 9.72$ by Guan et al.'s heuristic; $12 \times 1 \times 0.78 = 9.36$ by Li et al.'s heuristic. With the service rate, all containers can go through the inspection center without any delay.

The simulation results also verify the theoretical lower bound of inspection service rate by different heuristics when there are large enough number of vessels going through the current system.

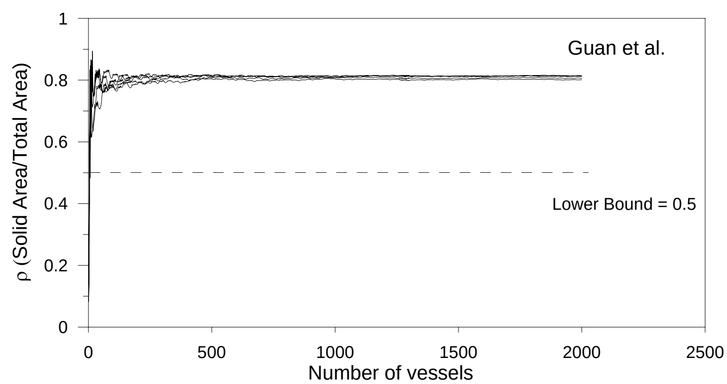

Figure 13: Deterministic experimental runs by Guan et al.'s heuristic

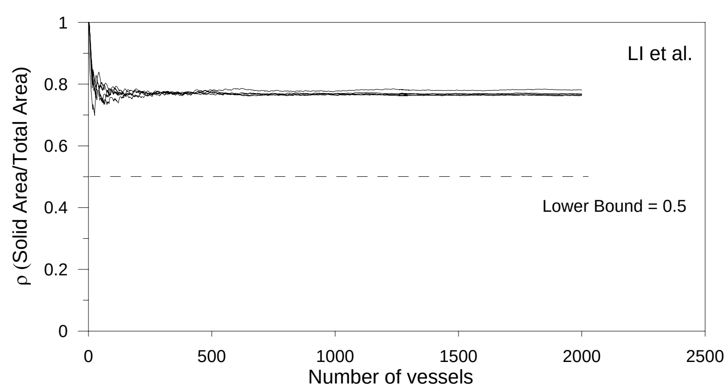

Figure 14: Deterministic experimental runs by Li et al.'s heuristic

2.6.4 Stochastic case verification

The deterministic approach provides a way to estimate the service rate of the inspection center. However, in the real world, the deterministic cases are rare. Therefore, in

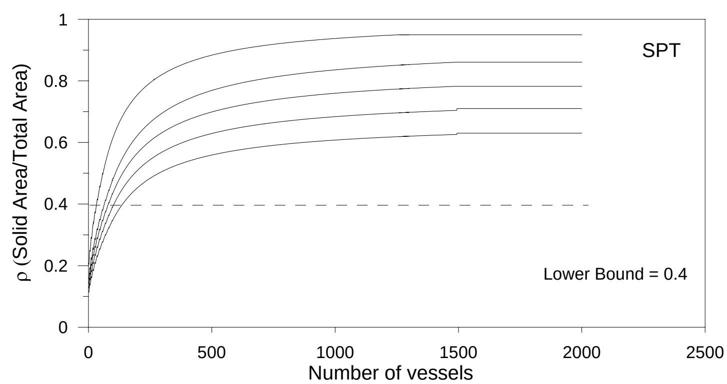

Figure 15: Deterministic experimental runs by SPT heuristic

this section, I carry out the numerical experiments with the aid of simulation tools. I build a terminal system with 12 berths and one inspection center with three heuristics, then I run the simulation model over a 2 months simulation period. I run different scenarios. The scenarios use different the service rate of the inspection center.

In this study, the stochastic vessel processing time follows a normal distribution with the mean value as the expected processing time. When applying a deterministic heuristic to berth vessels, a vessel size is agreeable to a vessel processing time. However, under the assumption of stochastic processing time, the agreeable assumption is no longer valid. In order to apply a deterministic heuristic, I use expected processing time as input and let the vessel size be agreeable with the expected processing time. While the vessel or container entities go through the system, actual processing time is used instead of expected processing time.

Table 3 shows the average throughput percentages under different service rates of the inspection center for the stochastic case. The throughput percentage is defined as ratio between the container entities going through the inspection center and total containers needed to inspect. The results show are slightly different from the analytical results. From the figure, resource capacity or service rate needs to reach 9 times (instead of 9.72 by Guan et al.'s heuristic and 9.36 by Li et al.'s heuristic) as the original value. The major differences come from two sources. First, analytical results based on the assumption of deterministic processing time of each processor. Second, the assumption of vessel size being agreeable to vessel processing is violated.

Table 3: Average throughput % under different service rate setting

Service rate multiple	Guan et al.'s heuristic	Li et al.'s heuristic	SPT heuristic
5	56.10	58.43	56.54
6	67.44	70.25	67.98
7	78.64	81.91	79.26
8	89.64	93.45	93.35
9	99.35	99.61	99.69
10	99.95	99.94	99.97
11	99.98	99.98	99.99
12	99.99	99.99	99.99

In practice, it is hard to increase service rate of the original setting by 9 times. For example, in the harbor area, the land is limited, and most land will be planed for other commercial activities, such as transportation, storage, etc. Expanding an inspection center by increasing the number of inspection machines will influence commercial activities and will be very expensive. In case that there is no enough inspection capacity, I can utilize sampling strategy to find out a proper sampling rate such that the system can run smoothly. Sampling rate can be decided by the inspection service rate and the processor rate. For instance, the sampling rate can be $C\gamma_i/(S\gamma_p)\times 100\%$. C is a parameter, and need to be further determined by policy makers. Table 4 shows average throughput % under different sampling rate policies. At the present system, only about 11% can go through the inspection operation. The results also show if sampling rate is 10%, all sampled containers will go though the inspection without any delay. Table 4 can be a throughput results reference for a policy maker to determine the proper sampling rate for the current system.

2.7 Summaries

In recent years, the container security inspection becomes more important for terminal operations. In this chapter, I develop an approach to evaluate the service rate of the inspection center so that the whole system can run smoothly.

I consider a system within a container terminal, which contains two operations, the berth allocation and the security inspection. Theoretically, the service rate of the

Table 4: Average throughput % under different sampling rate setting

Sampling rate (%)	Guan et al.'s heuristic	Li et al.'s heuristic	SPT heuristic
10	99.8	99.9	99.9
12	86.2	89.8	87
14	74.8	77.8	75.4
16	66	68.7	66.5
18	59	59.5	59.5
20	56.1	58.4	56.6
30	37.4	38.9	37.7
40	28	29.2	28.3
50	22.4	23.4	22.6
60	18.7	19.5	18.9
70	16	16.7	16.2
80	14	14.6	14.1
90	12.5	13	12.6
100	11.2	11.7	11.3

inspection operation needs at least 0.4 of the service rate of the processors when the SPT berthing policy is applied. By using different policies (Guan et al.'s or Li et al.'s algorithms), the theoretical lower bound may increase to half of the service rate of the processors. Deterministic and stochastic experiments show this argument is correct for the designed scenarios. However, the deterministic and stochastic experiments have similar results with minor difference. These differences mainly come from two sources. First, analytical results based on the assumption of deterministic processing time of each processor. Second, the assumption of vessel size being agreeable to vessel processing for the deterministic cases may not be valid any more under the stochastic processing time setting.

In the stochastic experiments, the simulation results also show that the service rate of the security inspection center should be around 9 times the processor service rate for the berth allocation. In practice, it is usually hard to achieve such a high service rate. A high service rate examination machine may not worth for investment, or maybe there is limited land to expand the inspection center. Therefore, to increase the inspection rates or provide more inspection machines may not be feasible. I can combine "increase service rate" and "reasonable sampling" to find the proper alternatives for the container security inspection operation.

Teo and Dai (2003) carried out berthing planing project to deal with home berth allocation problem and dynamic berth planing. In their research, they used the rectangle packing problem concept to deal with BAP. The advantage of this study approach is BAP can be treated as one strip packing problem. For this type of problem, there exists developed efficient algorithms such as First-Fit Decreasing Height (FFDH) algorithm, Split-Fit algorithm (SF), Next-Fit Decreasing Height (NFDH) algorithm (Coffman et al. (1980)), Baker's Up-Down (UD) algorithm (Baker et al. (1981)), Reverse-fit (RF) algorithm and Steinberg's algorithm (Steinberg (1997)). These algorithms can be provided as the heuristics I use in this study for investigating more effective berthing heuristics for BAP with inspection operations.

3 The crane scheduling problem: models and solution approaches

Abstract

In this chapter, I study the crane scheduling problem for a vessel after the vessel is berthed on a terminal and develop both exact and heuristic solution approaches for the problem. For the exact solution approach, I develop a network flow formulation with non-crossing constraints to obtain an optimal solution for small size problems. For medium-size problems, I develop a Lagrangian relaxation approach that allows us to obtain tight lower bounds and near-optimal solutions. For the heuristic solution approach, I develop two heuristics and show that the error bound of the heuristics is no more than 100%. Finally, I perform computational experiments to compare different solution approaches as well as evaluate efficiency and effectiveness of the solution approaches.

3.1 Introduction

A container is a large and standardized box which can be used to transport goods from one destination to another through different transportation modes such as vessels, trucks, and etc. From the yearly container throughput historical data report shown in UN (2005), I can find that container vessels and container trade had increased dramatically in the past twenty years, and the container trade projection will keep at least 3% growth till year 2015. Therefore, operation efficiency of container terminals is essential to accommodate the increment of container flows through the global transportation network.

Container terminal operations usually contain four main sections: assigning berths for each particular vessel, loading or unloading containers between vessels and internal trucks, transportation of containers by internal trucks between berths and storage yard, and container loading or unloading service in the storage yard. Container loading or unloading activities by cranes is one of the most important operations in the container terminal. Effective and efficient crane scheduling policy will facilitate container flows so as to shorten container flow time and improve terminal operation performance.

Be aware of the importance of the crane operations, this study focuses on the quay crane scheduling problem (QCSP). Quay cranes are equipments that load/unload containers to or from a vessel in container terminals. A quay crane is made by heavy steal frame and all quay cranes are mounted on the same rail. Thus, cranes need safety distance between each other and they are not allowed to cross each other. Based on real data obtained from a container terminal, I observe quay crane characteristics. The safety distance between two adjacent cranes is 60 to 80 meters. The service rate of each quay cane varies based on different types of cranes with the average service rate to be 30 to 35 containers per hour. The velocity of a quay crane moving along the rail is 30 to 45 meters per minute. When a crane is on duty, it is assigned on a specific section. The crane will not move to other areas until it finishes all the workload in this section. Besides this, no two cranes work in the same section at the same time. Finally, for a quay crane moving from a section to another, there is travel time involved.

Several approaches have been developed to study QCSP. The first study was conducted by Daganzo (1989) who developed a model in which non-crossing constraints due to crane interference with each other are not considered. Peterkofsky and Daganzo (1990) followed the previous study, and applied branch and bound technique to solve the problem. Kim and Park (2004) took precedence relationship of jobs into account in their model and developed a branch and bound algorithm to solve small size problems. They also adopted the greedy randomized adaptive search procedure to improve the performance of their branch and bound algorithm. Following Kim and Park's study, Moccia et al. (2006) developed a branch and cut algorithm to solve some instances which can not be solved by Kim and Park's approach.

Recently, there is also significant research done for the case that non-crossing constraints are considered. Lim et al. (2002) designed a dynamic programming algorithm and compared its performance with tabu search approach. **?** formulated the problem as an integer programming problem and proved the problem is NP-hard. This indicates that it is difficult to find an optimum solution within a reasonable amount of time. They developed a branch and bound algorithm with a special a branching strategy to solve small and medium size problems. Their algorithms outperform default CPLEX solver. They also developed a simulated annealing algorithm with a neighborhood search embedded to find near-optimal solutions for large size problems. Most recently, Lim et al. (2007) compared the performance between the dynamic programming and the simulated annealing technique. Lee et al. (2008b) applied a similar approach as described in **?** to show that QCSP with non-crossing constraints is an NP-hard problem and elaborated a genetic algorithm.

In the above previous QCSP studies, most researchers ignored crane travel time in their model. In this study, I will take crane travel time into consideration and develop the corresponding model and algorithm. Since QCSP with non-crossing constraints is a NP-hard problem, which is a special of the problem with zero travel time between sections, the problem is clearly NP-hard and it is hard to find an optimal solution in short time. In the approach, I will develop a time-space network flow model to formulate the problem with the objective of minimizing the makespan to serve the particular vessel. The model will capture the non-crossing constraints and it is also capable to cover crane travel time constraints. For large size problems, I develop three heuristics: a Lagrangian relaxation based heuristic, a simple threshold heuristic and a dynamic programming based heuristic. I perform worst case analysis and compare the computational performance of different approaches.

In this study, I develop different approaches to solve QCSP, including the analytical approach, the Lagrangian relaxation approach, and the heuristic approach. The remaining part of this chapter is organized as follows: Section 2 describes the assumptions and the problem definition. Section 3 presents a mathematical model for QCSP. A developed model demonstrates the advantages by comparing with other two different QCSP models. The performance of this study model performance is also evaluated. I solve the model by the Lagrangian relaxation approach to find out the tight lower bound and near-optimum solution in Section 4. In Section 5, two heuristics for large-size instances are developed. One simple heuristic is based on threshold value and its worst case performance is studied. The other one is a dynamic programming algorithm that requires each crane to work on consecutive sections. Its performance is compared with the simple heuristic. In Section 6, I perform experiments to evaluate the performance and effectiveness of the model and the proposed heuristic approaches. Finally, Section 7 concludes this study.

3.2 Problem descriptions

In this study, I consider one vessel multiple crane instances. As shown in Figure 16, when there is a vessel moored in the berth, several cranes are assigned to that vessel. Then the assigned cranes move to their work areas along the rail and load or unload the containers from that vessel to the land side truck or from the land side truck to that vessel until the assigned cranes finish their jobs. The cranes then wait for the next assignments. If no further assignment, the cranes stay at the present positions. The vessel leaves the assigned berth after all jobs are finished.

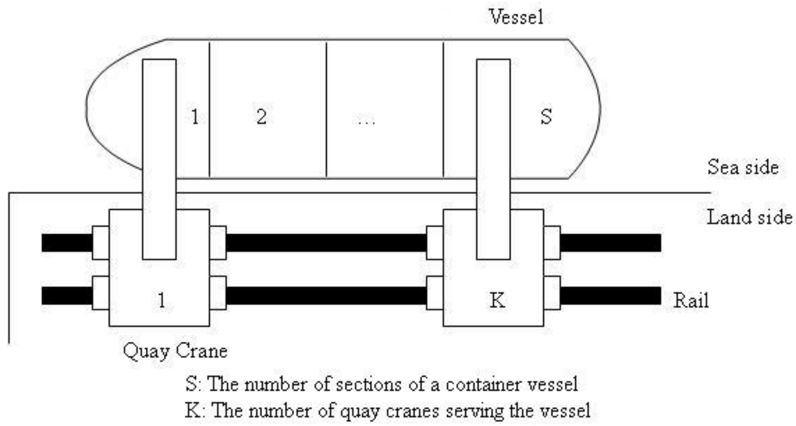

Figure 16: The illustration of the QCSP (ref. Lee et al. (2008b))

According to the descriptions, without loss of generality, I can make the following reasonable assumptions for QCSP.

1. There are K cranes in total available to serve the moored vessel;

2. The total workload, assumed to be N units of workload in total, on the vessel can be divided into S sections. There are W_j units of workload for each section $j = 1, 2, \ldots, S$. The partition is usually based on the safety distance required for each crane operation. Note here each unit of workload represent the number of containers. It is not necessary to be one container per workload unit.

3. Each section can only holds one crane at one time, i.e. no two cranes can work for the same section at the same time.

4. Once a crane starts to work on a section, it will not stop until it finishes loading or unloading all the containers in this section. For instance, no-preemption is allowed.

5. A crane can move from one section to another along the rail and the travel time between two consecutive sections is constant (i.e., ν_1). This also indicates that no two cranes can cross each other.

6. I assume all cranes have the same service rate Y. For instance, there are Y units of workloads served per time unit.

In this chpater, I study how to schedule quay cranes such that the total workload of the vessel can be finished as soon as possible. Besides the parameters described above in the assumption, I let binary parameter b_j to represent if there is a crane originally located in section j for $j = 1, 2, ..., K$ at time period $t = 1$. For instance, $b_j = 1$ if originally there is a crane available in section j and 0 otherwise. I will develop a time-space network follow model with side constraints to solve the problem with the objective to minimize the makespace τ to finish all workloads in the vessel. I let a binary decision variable x_{jt} to indicate if there is workload left in section j at time period t. I have $x_{jt} = 1$ if there is still workload left in section j at time t, and 0 otherwise, where $1 \leq j \leq S$ and $1 \leq t \leq T$. Here T is the planning horizon for the problem. It can be any upper bound of the makespan τ. For instance, $T = S\nu_1 + N/Y$ is an upper bound for τ and therefore it is a candidate for the planning horizon. I also let decision variable w_{jt} to represent the remaining workload in section j at time period t with $w_{j0} = W_j$ for all $j : 1 \leq j \leq S$ and $t : 1 \leq t \leq T$. Finally, I let an integer decision variable $z_{j,j'}^{t,t+1}$ to represent the number of cranes moving from section j to section j', from time t to time $t + 1$, where $1 \leq j, j' \leq S$ and $1 \leq t \leq T$.

3.2.1 Mathematical formulation

According to the problem description in the previous section, assuming each time unit to be ν_1, a mixed integer programming formulation with the objective to minimize makespan for QCSP can be formulated as follows:

$$\min \quad \tau \tag{2}$$

$$tx_{jt} + 1 \leq \tau, \tag{3}$$

$$w_{jt} - Mx_{jt} \leq 0, \tag{4}$$

$$w_{jt-1} - Yz_{j,j}^{t-1,t} \leq w_{jt}, \tag{5}$$

$$z_{j,j+1}^{t,t+1} + z_{j,j-1}^{t,t+1} + z_{j,j}^{t,t+1} = b_j, \quad t = 1 \tag{6}$$

$$z_{j-1,j}^{t-1,t} + z_{j,j}^{t-1,t} + z_{j+1,j}^{t-1,t} - z_{j,j+1}^{t,t+1} - z_{j,j-1}^{t,t+1} - z_{j,j}^{t,t+1} = 0, \quad t = 1, \ldots, T-1 \tag{7}$$

$$\sum_{j=1}^{S} z_{j,0}^{t-1,t} = \sum_{j=1}^{S} b_j, \quad t = T \tag{8}$$

$$z_{j-1,j}^{t-1,t} + z_{j,j}^{t-1,t} + z_{j+1,j}^{t-1,t} \leq 1, \tag{9}$$

$$z_{j-1,j}^{t-1,t} + z_{j,j-1}^{t-1,t} \leq 1, \tag{10}$$

$$z_{j,j}^{t-1,t} + x_{jt} - z_{j,j}^{t,t+1} \leq 1, \quad t = 1, \ldots, T-1 \tag{11}$$

$$w_{jt} \geq 0, x_{jt} \in \{0,1\}, \text{ and } z_{j,j'}^{t,t+1} \in \{0,1\}. \quad t = 1, \ldots, T-1 \tag{12}$$

where M is a sufficient large number.

The objective function (2) is to minimize the makespan of all assigned cranes to handle all containers, which is the completion time of the latest job. Constraint (3) determines the finish time of the last job, i.e. makespan. Constraint (4) states that if there is workload left in the section, the section is under a crane loading or unloading process. Constraint (5) indicates the workload flow balance constraints. For instance, for each section j. I have initial workload $w_{j0} = W_j$ and then for each following time period, the remaining workload will reduce by Y units before it reaches zero if there is a crane working on the section during the period. Constraints (6) to (8) are network flow balance constraints. Constraints (9) indicate that at each time unit, each section can only hold at most one crane to work on the section. (10) indicates that cranes cannot move cross each other. These two restrictions are shown in Figures 17. Finally, constrains (11) represent no-preemption restrictions. For instance, once a crane starts to work on a section, it has to finish all workload in this section before it moves to other sections. Note here due to the restrictions for crane moving as described in

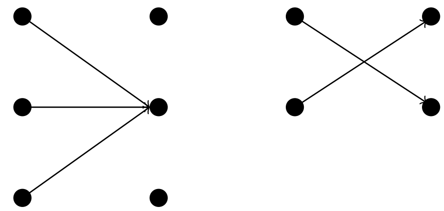

Figure 17: Crane bump and non-crossing constraints

(9) and (10), I can limit the decision variables $z_{j,j'}^{t,t+1}$ to be binary instead of general integer as described in (12).

3.2.2 Model analysis

In the approach, I introduce a mathematical formulation with network flow constraints as a subset of the formulation. In the model described in the previous section. I assume each time unit to be ν_1. For more general cases, if each time unit is much larger than ν_1, then I do approximation that the travel time between two sections is zero. In the model, I use decision variables $z_{j,j+1}^{t,t}$ instead of $z_{j,j+1}^{t,t+1}$ and $z_{j,j-1}^{t,t}$ instead of $z_{j,j-1}^{t,t+1}$. If each time unit is smaller than ν_1, then I can use decision variables $z_{j,j+\nu}^{t,t}$ instead of $z_{j,j+\nu}^{t,t+1}$ and $z_{j,j-\nu}^{t,t}$ instead of $z_{j,j-\nu}^{t,t+1}$, where ν is the closest integer for ν_1/ν_0 with ν_0 representing the time for each time period.

From this approach, I can describe a network flow based model that solve the problem to obtain an exact optimal solution for the problem. I first test the model under a default CPLEX setting and compare its performance with other existing formulations. By taking advantage of network flow substructure in the formulation, this study model has better performance than other formulations, such as the models described in Lee et al. (2008b) and **?**. The detailed results are shown in Section 3.5.1.

3.3 Lagrangian relaxation approach

In order to evaluate the capacities of the developed model, I run the experiments in different settings. The experimental results are shown in section 3.5.2. They show that the model works for small size instances. For the medium and large size instances, I solve the problem by the Lagrangian relaxation approach and the heuristic approach.

In this section, I purpose a Lagrangian relaxation heuristic based on the mathematical model to improve the analytic model approach.

3.3.1 A relaxed problem

Lagrangian relaxation approach relaxes some constraints and puts the constraints into the objective function. The optimal objective value of the sub-problem provides a lower bound for the original problem. The first step for this approach is to determine which constraints are relaxed. Those related to the network flow constraints are not relaxed because the CPLEX solver performs well for efficiently solving network flow structure models. Except the network flow constraints, there are four candidate constraints, (3), (4), (5), and (11), to be relaxed. I test 135 instances of the relaxed problem of the 15 combinations to evaluate average objective values as shown in Table 5. I also compare the optimum by the original model to the lower bound with the Lagrangian relaxation method. I relax the constraints with the objective value close to the optimum to form a relaxed problem. The results of Table 5 indicate that constraint (11) should be constraints to be relaxed due to the tightest lower bound it provides.

Table 5: Candidate constraints tests

	Run								
Relaxed constraints	1	2	3	4	5	6	7	8	9
(3) (4) (11) (5)	-111.37	-95.35	-51.55	-46.46	-34.73	-72.01	-68.36	-63.09	-71.9
(3) (4) (11)	1	-45.85	-30.3	1.01	1.01	1	1	1	1
(3) (4) (5)	-111.53	-97.13	-93.34	-46.44	-34.7	-71.81	-68.15	-62.97	-71.89
(3) (11) (5)	3.3	2.17	1.8	1.61	1.5	1.44	1.44	1.44	1.44
(4) (11) (5)	-224.8	-182.09	-82.41	-56.48	-88.96	-35.41	-34.79	-36.61	-46.21
(3) (4)	-29.88	-45.52	-30.09	-10.86	-5.46	-14.43	-14.43	-14.43	-14.44
(3) (11)	26.8	13.78	9.33	7.67	9	7.97	7.97	7.97	7.97
(3) (5)	3.3	2.18	1.8	1.61	1.5	1.44	1.44	1.44	1.44
(4) (11)	8.38	5.11	8.08	9.75	8.3	4.52	9.37	4.55	10.23
(4) (5)	-464.99	-182.1	-191.07	-162	-149.64	-1036.85	-1067.13	-1192.78	-1096.49
(11) (5)	3.3	2.18	1.8	1.61	1.5	1.45	1.45	1.45	1.45
(3)	0	0	11.8	7.67	9	7.93	7.97	7.97	7.97
(4)	8.38	6.85	6.08	7.72	7.26	6.85	6.68	6.72	8.84
(11)	91	46	31	24	19	16	14	13	13
(5)	3.3	2.18	1.8	1.61	1.5	1.45	1.45	1.45	1.45
Objective value	-	-	31	-	21	20	19	18	13

Let μ_{jt} be the Lagrangian multipliers for a given j and t and μ be the vector of

μ_{jt}. The relaxed problem can be written as

$$L(\mu) \quad = \quad \min \tau + \sum_{j=1}^{S} \sum_{t=1}^{T} \mu_{jt}(z_{j,j}^{t-1,t} + x_{jt} - z_{j,j}^{t,t+1} - 1) \qquad (13)$$

subject to (3),(4),(5),(6),(7),(9),(10).

3.3.2 Subgradient algorithm

In this study, I adopt Subgradient method to obtain the optimal dual for the Lagrangian relaxation problem. Kalvelagen (2002) summarized Subgradient algorithm. In the algorithm, μ^j represents the Lagrangian multiplier μ_{jt} in the j^{th} iteration. The details of the Subgradient algorithm in this study are listed in the following:

Step 1: Input the parameters, such as the number of cranes and the number of containers per section

Step 2: Calculate the initial upper bound L^*, which can be provided by any feasible solution of the QCSP. For convenience, in this approach, I set the value to be time horizon T

Step 3: Calculate initial values for $\mu^0 \geq 0$. In this approach, I set the initial value of Lagrangian multiplier to be the dual values of the linear programming relaxation of the mixed integer program

Step 4: Set the initial value $\theta_0 = 2$

Step 5: Subgradient iterations

for $j = 0, 1, \ldots, N$, do (where N is the maximum number of iterations)

1. let $\gamma^j = \partial L(\mu^j)/\partial \mu^j$. For this case, γ^j corresponds to constraint (11)
2. set the step size $t_j = \theta_j(L^* - L(\mu^j))/||\gamma^j||^2$
3. update the dual multiplier $\mu^{j+1} = \max\{0, \mu^j + t_j\gamma^j\}$
4. **if** $||\mu^{j+1} - \mu^j|| < \epsilon$, **then** stop

end if

5. **if** no progress in more than M iterations (M is a pre-defined limited execution iterations)

   then update $\theta_{j+1} = \theta_j/2$

   else $\theta_{j+1} = \theta_j$

   end if

6. update $j = j + 1$

end for

3.3.3 Lagrangian relaxation heuristic

I find in Table 5 that there are three cases where the original model can not return the optimum within one day. However, lower bounds can be returned by Lagrangian relaxation in all test cases, which indicates that the relaxed problem is easier to solve than the original problem.

The experiment results also show that the duality gap in some cases are very small, but they might still return an infeasible solution for the original problem. The infeasible solutions come from the violation of the non-preemption constraint (11) which I relax. Table 6 demonstrates a case with 6 cranes and 9 section jobs by Lagrangian relaxation approach. The third column of Table 6 shows multiple cranes serve the same section. In order to let an infeasible solution become feasible, I propose the following method.

Algorithm for making an infeasible solution feasible (LR Heuristic)

Step 0: Sort the cranes in non-decreasing order from one end to the other end of the terminal. Renumber them from the smallest to the largest accordingly.

Step 1: Record the cranes serving a corresponding section obtained by Lagrangian relaxation approach.

Step 2: For each section, find the minimum index crane serving the section, in other words, allocate section jobs to the minimal index crane for that section.

Step 3: Calculate the makespan for each indexed crane. The maximum value among makespans of all those cranes provides the first makespan for the problem.

Step 4: Repeat Steps 2 and 3 but find the maximum index crane instead of the minimum index crane.

Step 5: The smaller makespan of the two feasible solutions provides the makespan for the problem.

An example that shows how to apply LR heuristic to make an infeasible solution to be feasible is described in Table 6. After I apply the proposed heuristic to make an infeasible solution to a feasible solution, each crane will serve consecutive section jobs, as shown in Table 6. In the example, columns 4 and 5 represent the feasible solutions obtained by the minimum crane index and the maximum crane index cases respectively. The maximum crane index solution provides a better solution, in which the first crane works for section 1, the second crane works for section 2, the third crane works for section 3, the fourth crane works from sections 4 to 6, the fifth crane works for section 7, and the last crane works for sections 8 and 9.

Table 6: An example by LR heuristic to make an infeasible solution to be feasible

Section	Workload	Crane	Min. Crane (1)	Max. Crane (2)	
1	7	1	1	1	Makespan (1)
2	12	1,2	1	2	19.1
3	10	2,3	2	3	
4	5	2,3,4	2	4	Makespan (2)
5	7	3,4	3	4	18.1
6	5	3,4	3	4	
7	12	4,5	4	5	Makespan
8	11	5,6	5	6	18.1
9	7	6	6	6	

3.4 Heuristic approach

In this section, I study two different heuristics to solve large size instances. The first heuristic is based on a threshold policy obtained by a lower bound generated by a dynamic programming algorithm. By this threshold, I can allocate jobs in several consecutive section to different cranes. The worst case analysis shows that any feasible solution obtained by this heuristic is bounded by 2 times of the lower bound. The second heuristic utilizes a dynamic programming algorithm to obtain a feasible solution directly. I first introduce the following notation before I describe the heuristics.

s_j^0 : The initial location of crane j

$t_j^0(s)$: The time to move crane j from its initial location to section s, i.e., $t_j^0(s) = \nu_1|s_j^0 - s|$

ν_2 : The time to load or unload a container

$s(i)$: The section index for the i^{th} container where $i = 1, 2, ..., N$

m_j : The first working section of crane j

n_j : The last working section of crane j

τ_j : The total time that the j^{th} crane spend on loading or unloading containers between sections m_j and n_j

Z^{H_1} : The makespan by H_1 heuristic

Z^{H_2} : The makespan by H_2 heuristic

$\mathcal{H}(i, j)$: The optimal value function (i.e., makespan) for the first i containers served by the first j cranes.

3.4.1 Lower bound by dynamic programming

Non-preemption is a common assumption for QCSP. It requires that once a crane starts loading or unloading containers in a section, the crane will not stop until all jobs in this section are finished. If I assume preemption is allowed for QCSP, i.e., jobs in one section can be shared by multiple cranes. It can be observe that, under this scenario, a crane will serve consecutive sections in an optimal solution as shown in Figure 18 in which there are 4 cranes and 14 sections. This case demonstrates that the workload of section 4 and section 9 are shared by two cranes.

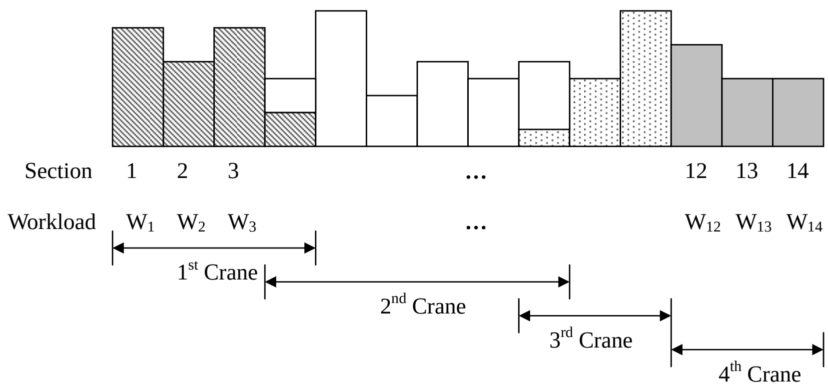

Figure 18: Consecutive relaxed problem

The optimal solution of this preemption problem is the lower bound of QCSP in which preemption is not allowed. By applying dynamic programming I can derive an algorithm to find the lower bound of QCSP. The detailed step can be described as follows.

Algorithm for the threshold by dynamic programming (TDP)

Step 0: Index the cranes and containers from the smallest to the largest along the same direction as the index for sections along the vessel. For instance, I have $s(1) = 1$, $s(i) \leq s(j)$ and $s_i^0 \leq s_j^0$ if $i \leq j$

Step 1: Initialize the dynamic programming value function

$\mathcal{H}(i, 1) = \min\{t_1^0(s(1)), t_1^0(s(i))\} + (s(i) - s(1))\nu_1 + \nu_2 i$ for all $i = 1, 2, ..., N$

$\mathcal{H}(1, j) = t_j^0(s(1)) + \nu_2$ for all $j = 1, 2, ..., K$

Step 2: Obtain the dynamic programming recursive function For $i = 2, 3, \cdots, N$ and $j = 2, 3, \cdots, K$

$$\mathcal{H}(i, j) = \min_{k=1}^{i} \max\left\{\mathcal{H}(k, j-1), \min\left\{t_j^0(s(k+1)), t_j^0(s(i))\right\} + (s(i) - s(k+1))\nu_1 + (i-k)\nu_2\right\}$$

In above Step 1, I calculate the start value for dynamic programming. $\mathcal{H}(i, 1)$ represents the time required that one crane works for the first i containers. It includes the time the crane moving from its initial position to the first loading or unloading position (note here, the crane can start from the either end of the consecutive sections), the traveling time the crane moving among the working sections, and the time for a crane loading/unloading i containers. $\mathcal{H}(1, j)$ represents the time that the j^{th} crane will work for the first container and the first $j-1$ cranes are idle.

In Step 2, I obtain the dynamic programming recursive value function corresponding to each particular container i and crane j. The optimal makespan corresponding to the state that the first i containers served by the first j cranes is determined by choosing the maximum of two parts: first, the optimal makespan for the first $j-1$ cranes serving k containers ($k \leq i$), and second, the time that the j^{th} crane serves the $k+1^{th}$ container to the i^{th} container. The optimal makespan for the state (i, j) is the minimum value among the choices $k = 1$ to i.

According to the algorithm, $\mathcal{H}(N, K)$ is equal to the minimum makespan of K cranes serving N containers, which is the lower bound of QCSP.

For the dynamic programming procedure, I have to calculate all states, i.e. $\mathcal{H}(i, j)$ for all $i = 1, 2, \cdots, N, j = 1, 2, \cdots, K$. To obtain $\mathcal{H}(i, j)$, I need to do calculations corresponding to all states $(i', j-1 | i' \leq i)$ and the present state. Therefore, the computational complexity to find the optimal objective value for the relaxed problem of OCSP is $\mathcal{O}(N^2K)$. Accordingly, the computational time of this algorithm depends on the number of containers and the number of cranes of an instance.

Nowadays, the capacity of the largest container vessel is between 4000 containers to 5000 containers. Normally, there are at most 6 cranes serving one vessel. Based on these conditions, I test a practical instance to estimate the computational time of the algorithm to evaluate the efficiency of this algorithm. I test the case that one vessel is with 5000 containers, and there are 6 cranes serving that vessel. The computational time to find the lower bound is around 72 seconds. In some studies, the projection of a container vessel capacity will reach 12000 containers in 2010. Therefore, I test a scenario that 10 cranes serve one vessel with 12000 containers. This scenario takes 1102 seconds.

3.4.2 Heuristic by threshold method

I name this heuristic as H_1 heuristic. In the previous section I can get a lower bound $\mathcal{H}(N,K)$ by DPT algorithm. I define $\mathcal{H}(N,K) - \nu_1$ as time threshold for H_1 to determine a consecutive section jobs for a crane. The algorithm starts assigning from the first crane to the last crane, and from the lowest section to the highest section to follow the non-crossing constraint. A section I assume that its horizon distance is wide enough when two neighboring cranes will not interfere with each other.

Algorithm for the threshold heuristic (H_1)

Step 1: Find $\mathcal{H}(N,K)$ by DPT. The threshold is $\mathcal{H}(N,K) - \nu_1$

Step 2: Initialize $\tau_j = 0$ for each $j = 1, 2, \ldots, K$. Then, find partitions and makespans for each crane $j = 1, 2, \ldots, K$

Step 2-1: Set the start section for the first crane $m_j = 1$ for $j = 1$

Step 2-2: Corresponding to the j^{th} crane, I find an integer number $1 \le \alpha \le S$ such that

$$\nu_2 \sum_{i=m_j}^{\alpha-1} W_i + |\alpha - 1 - m_j|\nu_1 + \min\{t_j^0(m_j), t_j^0(\alpha - 1)\} \le \mathcal{H}(N,K) - \nu_1$$

and

$$\nu_2 \sum_{i=m_j}^{\alpha} W_i + |\alpha - m_j|\nu_1 + \min\{t_j^0(m_j), t_j^0(\alpha)\} > \mathcal{H}(N,K) - \nu_1.$$

Let $n_j = \alpha$, $\tau_j = \nu_2 \sum_{i=m_j}^{\alpha} W_i + |\alpha - m_j|\nu_1 + \min\{t_j^0(m_j), t_j^0(\alpha)\}$.

Step 2-3: Update the crane index $j = j + 1$ and the start section of j^{th} to be $m_j = \alpha + 1$. If $\alpha < S$, then repeat Step 2-2, else go to Step 2-4.

Step 2-4: The makespan $Z^{H_1} = \max\{\tau_j | 1 \le j \le K\}$.

The Step 1 of H_1 is to find a threshold. In Step 2 of H_1, the total sections are divided into K' partitions ($K' \leq K$), that is, each assigned crane only serves one work zone. A specific working zone for a crane is determined by observing if the accumulation time for the crane working on this zone is greater than or equal to the threshold. The accumulation time includes the crane traveling time between sections, the time serving containers, and the time for a crane traveling from its original location to the one end of the working zone. The section providing the marginal pass is the last section for the working zone.

H_1 contains two steps. The computational complexity of the first step is $\mathcal{O}(N^2K)$ and that of the second step is $\mathcal{O}(S)$. In practice, $N \gg S$. It is clearly that $\mathcal{O}(N^2K)$ is much larger than S. The first step of H_1 dominates whole H_1 calculation time, and therefore the total computational complexity for H_1 is $\mathcal{O}(N^2K)$.

3.4.3 Worst-case analysis of the threshold heuristic

In this section I examine the solution quality of H_1 for QCSP by the worst case analysis. The worst case analysis shows that relative error of H_1 is not more than 100%, that is, the ratio between any feasible solution and the lower bound is bounded by 2.

Theorem 1 *The worst case ratio for H_1 is at most 2. That is, $Z^{H_1}/Z^* \leq 2$, where Z^{H_1} and Z^* are the objective values for the feasible solution and the optimal solution, respectively.*

Proof: I first derive the lower bounds for the problem. At first, I can observe that, due to non-preemption constraints, I have the first lower bound

$$\underline{Z}_1 = \max_{i=1}^{S}\{W_i\}\nu_2. \tag{14}$$

From the conclusion for the relaxed problem, I can also observe that

$$\underline{Z}_2 = \mathcal{H}(N, K). \tag{15}$$

Therefore, based on (14) and (15), I have

$$Z^* \geq \max\{\underline{Z}_1, \underline{Z}_2\}. \tag{16}$$

Now I prove the claim for two cases:

Case 1. Consider a special case in which there is only one section. If the initial position of a crane is exactly on that section, then $Z^{H_1} = W_1\nu_2 \leq \underline{Z}_1 \leq Z^*$. If the initial position of a crane is not on that section, then according to the heuristic, the section is assigned to the first crane, which is the closest one. Therefore, I have $Z^{H_1} = t_1^0(s(1)) + W_1\nu_2 \leq \underline{Z}_2 + \underline{Z}_1 \leq 2Z^*$ since $t_1^0(s(1)) \leq \mathcal{H}(N,K) = \underline{Z}_2$.

Case 2. Consider the case that there are multiple sections. For each general step to assign a crane j, suppose that crane j serves multiple section jobs from sections m_j to n_j as shown in Figure 19. The makespan of the crane j can be divided into two parts. The total time for the first part (i.e., T_1) is the service time of unloading containers and traveling time from crane j's original location to section m_j, and to section $(n_j - 1)$. The total time for the second part (i.e., T_2) is the crane traveling time from sections $(n_j - 1)$ to n_j and the service time of unloading containers in the section n_j. According to the heuristic policy, I have

$$T_1 \leq \mathcal{H}(N,K) - \nu_1.$$

I also have

$$T_2 = \nu_1 + W_{n_j}\nu_2 \leq \nu_1 + \nu_2 \max_{i=1}^{S}\{W_i\}.$$

Therefore,

$$\begin{aligned} Z^{H_1} = T_1 + T_2 &\leq \mathcal{H}(N,K) - \nu_1 + \nu_2 \max_{i=1}^{S}\{W_i\} + \nu_1 \\ &= \underline{Z}_1 + \underline{Z}_2 \\ &\leq 2Z^*. \end{aligned}$$

It can also be verified that, starting from the smallest index section, corresponding to each crane j, I have that the index of the finishing section by crane j in the heuristic is no smaller than the index of the finishing section by crane j in the relaxed problem. Therefore, if I use K cranes for the relaxed problem and achieve the makespan $\mathcal{H}(N,K)$. Then I need no more than K cranes to finish the workload for all sections in the heuristic approach.

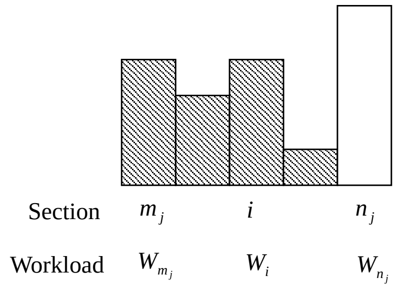

Figure 19: one partition by H_1

According to above two cases, I can conclude that $Z^{H_1}/Z^* \leq 2$. To prove the bound is tight, I use an example with 2 sections and 2 cranes shown in Figure 20. In this example, I assume ϵ is a very small positive number and $\nu_1 < \epsilon$.

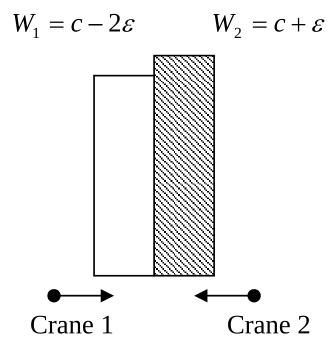

Figure 20: two sections jobs

In this case, the locations of cranes 1 and 2 are one section away from the section with workload. Considering the relaxed problem, I know that the threshold is $(c+\varepsilon)\nu_2+\nu_1$. According to the heuristic algorithm, I have $Z^{H_1} = (2c-\epsilon)\nu_2+2\nu_1$. The optimum solution $Z^* = \max\{(c+\epsilon)\nu_2+\nu_1, (c-\epsilon)\nu_2+\nu_1\} = (c+\epsilon)\nu_2+\nu_1$. Therefore, $Z^{H_1}/Z^* = ((2c-\epsilon)\nu_2+2\nu_1)/((c+\epsilon)\nu_2+\nu_1) \to 2$ as $\epsilon \to 0$ and $\nu_1 \to 0$. This result indicates that the worst case bound is asymptotically tight and is not possible to be improved. $\square$

3.4.4 Heuristic by dynamic programming

For H_2, I consider applying dynamic programming for the crane scheduling problem with the consideration that all containers belonging to the same section is considered as a job, comparing to the case that each individual container is treated as one job for the relaxed problem. The detailed steps of the algorithm are shown as follows.

Algorithm for dynamic programming heuristic (H_2)

Step 0: Index the cranes from the smallest to the largest along the same direction as the index for sections along the vessel.

Step 1: Initialize the dynamic programming value function $\mathcal{H}(1, j) = \nu_2 W_1 + t_j^0(s1)$ for all $j = 1, 2, ..., K$
and $\mathcal{H}(i, 1) = \nu_2 \sum_{k=1}^{i} W_k + (i-1)\nu_1 + \min\{t_1^0(1), t_1^0(i)\}$ for all $i = 1, 2, ..., S$

Step 2: For $i = 2, \ldots, S$ and $j = 2, ..., K$, I calculate

$$\mathcal{H}(i, j) = \min_{k=1}^{i} \max\{\mathcal{H}(k, j-1), \nu_2 \sum_{r=k+1}^{i} W_r + (i-k-1)\nu_1 + \min\{t_j^0(k+1), t_j^0(i)\}\}.$$

QCSP by H_2 is similar to a partition problem. For this problem, I can apply H_2 to find the optimal partition pattern by assuming all containers in one section as one job. The computational complexity of H_2 is $O(S^2K)$. In practice, $N \gg S$. Therefore, the computation time of H_2 is much smaller than that of H_1. I can save a lot of computational time by applying H_2, especially in cases where there are a large number of containers. In this study approach, the dynamic programming approach is the way systemically check all of the possible partition patterns to find the optimum, which indicates that the makespan of the same case by making H_2 equal to or smaller than that of H_1. This ensures that the worst case bound of H_2 is also bounded by 2.

3.5 Computational results

3.5.1 Model comparisons

In the computational experiments, I first compare the performance of this study mathematical formulation with two other existing models (i.e., Lee et al. (2008b) and

?). In the experiment setting, I create 16 instances: 8 small size instances and 8 medium size instances. The results are shown in Figures 21 and 22. The instances are expressed in 3 parameters, $a \times b - c$, in which a represents the number of sections, b represents the number of cranes, and c represents the number of jobs. For these problem, I set one time unit to be the process time for 50 containers.

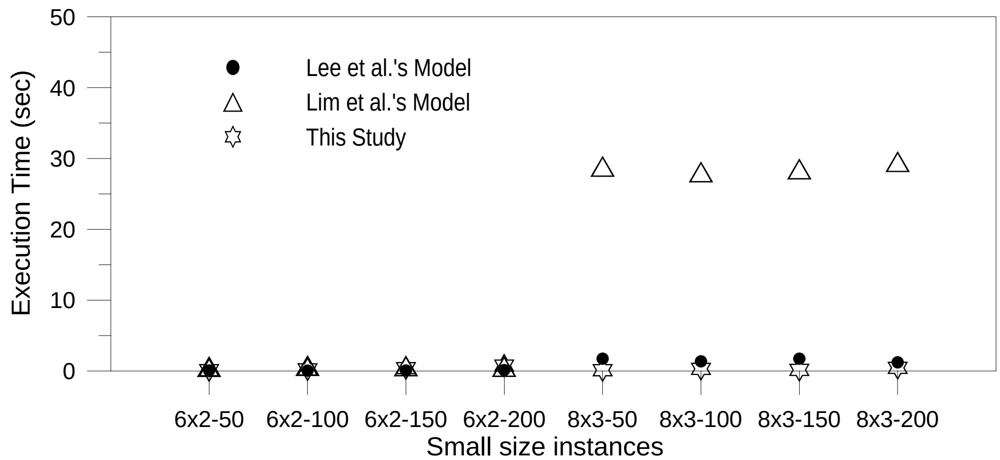

Figure 21: comparisons for small size instances

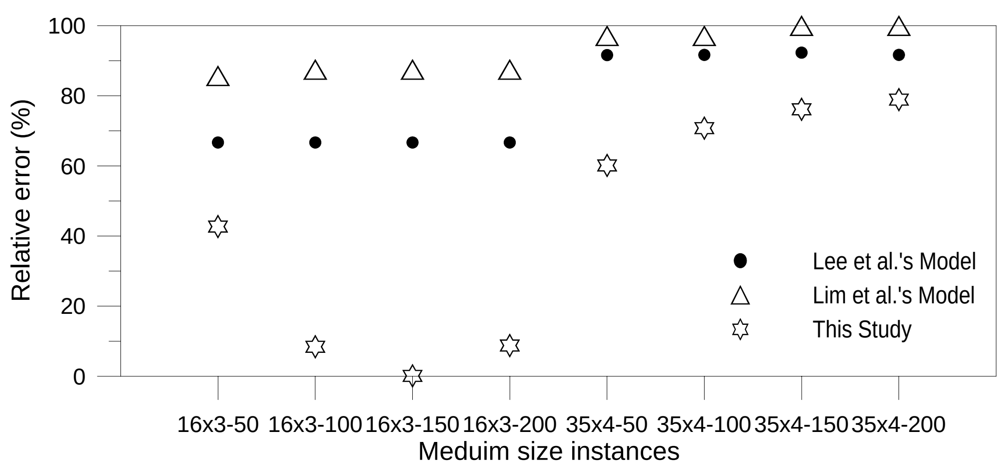

Figure 22: comparisons for medium size instances

For small size instances, all these three models can find the optimal solutions. This study model and Lee's model have very similar performance. Lim's model spends more time on some scenarios, but still can finish within 30 seconds. For medium size instances, I set 30 minutes as the time limit. All three models fail to find optimal solutions. I examine the feasible solutions and relative optimality gap provided by CPLEX solver. Overall the gap by running this study model is around 20% smaller than the other two models.

3.5.2 Model performance

In order to further examine the limitation of the model I propose, I create different scenarios to test the model. The experimental settings are listed in Table 7. I run 5 instances with the same parameter setting except the workload per section, and take the average value as the representing value for the setting. The tolerance of the optimality gap is set to be 1%. The final results are shown in Table 8.

Table 7: Experimental settings for testing the model

Number of cranes	1 crane to 3 cranes
Ship size	3 sections to 9 sections
Execution time limit	1800 seconds
Containers per section	workload is randomly generated between 100 and 300 containers

From Table 8, I can observe that the model can find the optimal solutions within the time limit for 22.78% of the cases. For around 54.43% of instances, default CPLEX can not return any feasible solution within the time limit. I can observe that as the number of nodes in the time space network is larger than 500. Default CPLEX to run the model can not provide an optimal solution within the time limit. Therefore, in the following, I describe Lagrangian relaxation and heuristic approaches to solve medium and large size instances.

Table 8: Experimental results for the model approach

Case	Results	Number of runs	Relative error (%)
1	Optimum	72	0
2	Feasible Solution	72	27.03
3	No solution return	172	100

3.5.3 Performance of the Lagrangian relaxation heuristic

I implemented the Lagrangian relaxation heuristic for three cases in which the number of cranes are 2, 3 and 4 respectively. For each case, I consider the size of the problem to be 9 sections and the planning horizon is 100 time units. Under this setting, the number of nodes is 900. Each time unit represents the time to process 25 containers. Initial workload for each section are randomly generated from 4 unit containers (i.e., 100 containers in total) to 12 unit containers (i.e., 300 containers in total). For each case, I perform and record 5 runs. The results are shown in Table 9. I report the ratio between the objective value for the best feasible solutio obtained by Lagrangian relaxation heuristic (Z^{LRH}) and the lower bound provided by Lagrangian relaxation (Z^L), and the computational time for each run.

In order to test the effectiveness of the heuristic, I run the model with the same experiment setting for comparison purpose. For each of the 15 runs, the default CPLEX can not return any feasible solution within one day. However, the proposed Lagrangian relaxation approach can return feasible solutions for all 15 experimental cases with the computational time range from 5.9 hours to 9 hours, and the average ratio ($\frac{Z^{LRH}}{Z^L}$) of the 15 cases is under 1.5, which indicates that the proposed Lagrangian approach improves the exact mathematical model approach in terms of solving medium size problems.

Table 9: Performance of the Lagrangian relaxation heuristic

No. of crane	$K = 2$		$K = 3$		$K = 4$	
	$\frac{Z^{LRH}}{Z^L}$	Time (hrs)	$\frac{Z^{LRH}}{Z^L}$	Time (hrs)	$\frac{Z^{LRH}}{Z^L}$	Time (hrs)
run 1	1.25	7.4	1.4	8.5	1.37	6.8
run 2	1.44	7.5	1.37	8.5	1.17	7.1
run 3	1.28	7.1	1.23	9	1.6	7.1
run 4	1.21	7.5	1.29	7.6	1.18	5.9
run 5	1.12	8	1.76	8.5	1.48	7.6
average	1.26	7.5	1.41	8.42	1.36	6.9

3.5.4 The Heuristic performance

In previous discussions, I report the computational experiment results of the default CPLEX for the mathematical formulation and Lagrangian relaxation approach to solve small size and medium size instances. In this section, I evaluate the performance of the two heuristics H_1 and H_2 by comparing the two heuristics with other analytic approaches, including the analytic model and the Lagrangian relaxation heuristic. Then, I compare H_1 with H_2 for large size instances.

I generate small size instances with known optimum (shown as "*Obj*" column in Table 10) as benchmarks to determine if these two heuristics can run efficiently and effectively comparing to the exact solution approach. For all instances, the default CPLEX for the mathematical formulation can reach optimal values within ten minutes ("*M*" column in Table 10). It is much larger than the computational time of H_1 and H_2, which are less than one second. H_2 yields optimal solutions for four instances. Since H_2 examines all possible consecutive sections and form working zones for cranes, when the working zones in the optimal solution are not consecutive, H_2 can not return the optimal value. H_1 does not yield optimal solutions for any case.

Table 10: Model and Heuristics Comparisons

	$K=2$				$K=3$				$K=4$				$K=5$			
	Obj	M	H_1	H_2	*Obj*	M	H_1	H_2	*Obj*	M	H_1	H_2	*Obj*	M	H_1	H_2
$S=6$	33	33	53	34	17	17	22	21	50	50	92	50*	33	33	37	33*
$S=7$	23	23	36	25	18	18	22	21	42	42	76	44	20	20	36	22
$S=8$	19	19	28	19*	28	28	52	30	22	22	42	23	38	38	70	42
$S=9$	20	20	33	22	30	30	54	30*	39	39	76	40	38	38	72	42

An asterisk indicates that the heuristic provides optimal value

Lagrangian relaxation heuristic, H_1, and H_2 are compared among each other for medium size problems. The results are shown in Table 11. The experimental settings are the same as described in Section 3.5.3, and the only difference is that the number of cranes is from 2 to 9. The average values of the ratios between the upper bounds generated by feasible solutions and lower bounds generated by Lagrangian relaxation are list in the table. The results indicate that H_2 has the best solution quality of the three heuristics. LR heuristic has the same solution quality comparing to H_1. However, Lagrangian relaxation heuristic requires large computational time in all instances.

In order to further evaluate two heuristics, I create large size instances. There are

Table 11: Comparisons among LR heuristic, H_1, and H_2

		LR heuristic	H_1	H_2
Case	No. of Cranes	$E(\frac{Z^{LRH}}{Z^L})$	$E(\frac{Z^{H_1}}{Z^L})$	$E(\frac{Z^{H_2}}{Z^L})$
1	2	1.26	1.08	1.04
2	3	1.41	1.24	1.08
3	4	1.36	1.32	1.12
4	5	1.46	1.31	1.15
5	6	1.37	1.56	1.32
6	7	1.36	1.39	1.18
7	8	1.15	1.5	1.1
8	9	1	1.26	1
Overall average		1.3	1.33	1.12

4 types of vessel sizes, which contain 20, 30, 40, and 50 sections respectively. The number of containers per section is generated uniformly distributed and categorized into three different categories: Case I: 10 to 50 containers per section, Case II: 50 to 100 containers per section, and Case III: 100 to 300 containers per section. The numbers of cranes are 5, 10, and 20 respectively. For each combination setting, I run 5 instances and take the average of the results for each combination setting. Table 12 shows the computational results for H_1 and H_2. In the table, "T" represents the average computational time for a specific setting in terms of seconds, and "G" represents the ratio between the upper bound generated by the feasible solution and the Lagrangian relaxation lower bound.

According to the analysis in Section 3.4.2, the computational complexity of H_1 is $O(N^2K)$, that is, the computational time of the experiment increases while the number of containers increases or the number of cranes increases. Results also show the same tendency; the number of containers dominates the computational time of H_1. In practice, one vessel normally needs 3 to 5 cranes to serve it. The close examples in the experiments are the number of cranes K equals to 5. I can see that average computational time by H_1 in these cases is no more than 200 seconds, and the ratio bound is under 1.3. H_2 outperforms H_1 in terms of both computational time and optimality gaps, because the computational complexity of H_2 is related to the number of sections (i.e., S) instead of the number of containers (i.e., N) and H_2 systemically examine all combinations of consecutive sections served by all cranes, which finds the minimum makespan if several consecutive working zones are served by the same crane.

Table 12: Large instance comparisons by H_1 and H_2

		$K=5$				$K=10$				$K=20$			
Sections	Containers	H_1		H_2		H_1		H_2		H_1		H_2	
		T	G	T	G	T	G	T	G	T	G	T	G
20	Case I	0.26	1.19	0.06	1.1	0.56	1.04	0.06	1.18	1.08	1.66	0.07	1.08
	Case II	2.72	1.19	0.05	1.08	6.01	1.42	0.1	1.16	11.98	1.76	0.07	1.02
	Case III	40.9	1.25	0.06	1.11	79.18	1.5	0.06	1.22	155.58	1.73	0.08	1.02
30	Case I	0.62	1.19	0.06	1.07	1.46	1.32	0.06	1.14	2.67	1.67	0.06	1.17
	Case II	5.67	1.14	0.06	1.06	12.29	1.31	0.06	1.14	24.95	1.56	0.08	1.29
	Case III	80.88	1.13	0.06	1.04	190.55	1.3	0.06	1.15	329.93	1.68	0.0.8	1.32
40	Case I	0.96	1.12	0.06	1.05	2.32	1.31	0.06	1.11	4.14	1.51	0.07	1.26
	Case II	10.96	1.11	0.06	1.03	21.54	1.26	0.06	1.08	45.08	1.63	0.07	1.17
	Case III	131.51	1.1	0.06	1.05	322.13	1.29	0.05	1.12	657.74	1.6	0.07	1.27
50	Case I	1.61	1.09	0.06	1.04	3.47	1.27	0.07	1.09	6.83	1.53	0.07	1.14
	Case II	15.28	1.08	0.09	1.03	35.81	1.19	0.07	1.08	69.01	1.4	0.07	1.19
	Case III	185.09	1.09	0.06	1.03	517.99	1.21	0.06	1.09	988.57	1.45	0.07	1.16

3.6 Summaries

In this chapter, I study the quay crane scheduling problem for one vessel case. In the study, I first developed a time-space network flow formulation for the problem. This model is very flexible and it can easily include non-crossing and non-interference constraints and easily include or exclude crane travel times. By aggregating containers into batches, under some scenarios, it will have better performance than other models.

I then extended the exact solution approach to the Lagrangian relaxation setting to solve medium size problems. Lagrangian relaxation heuristic is proposed to make an infeasible solution to a feasible solution. In this approach, I improve the analytic approach by getting the feasible solutions within reasonable computational time. However, the Lagrangian relaxation approach is only applied for small and medium size instances.

In order to solve the real world large size problems, I develop two heuristics. The first heuristic is based on the threshold to find the consecutive sections as a working zone for a crane. The threshold is determined by the lower bound from a dynamic programming technique. This approach ensures the solution quality in which worst case bound is 2. However, in extreme cases, this heuristic takes time to find a lower bound to provide the threshold. In order to conquer these issues in the first heuristic, I develop the second heuristic. In this heuristic I treat all containers in one section as one job. I do this to find the best partitions to form a working zone for cranes by dynamic programming. This leads to the problem size decreased substantially. Accordingly, the computational time decreases dramatically. The second heuristic is more efficient and effective than the first heuristic from the computational point of view as well.

In this study I focus on the crane scheduling problem. However, berth allocation and crane scheduling problems are two inter-connected activities in the container terminal operations. I will study combine berth allocation and quay crane scheduling in the next chapter.

4 Integrated Study of Berth Allocation and Quay Crane Scheduling Problems

Abstract

In this chapter, I study the integrated problem of combining berth allocation (BAP) and quay crane scheduling (QCSP) together. That is, I solve BAP and QCSP simultaneously. We first develop a Mixed Integer Programming formulation for the problem and obtain optimal solutions for small-size problems. For large-size problems, I develop two heuristic solution approaches and perform worst case analysis for each solution approach. I also use the solution obtained from the heuristic as an initial solution to be inserted into the exact solution framework to obtain a better feasible solution. Finally, computational experiment results are provided to compare different solution approaches and evaluate the effectiveness of our proposed solution approaches.

4.1 Introduction

Most cargos in the world nowadays are transported by containers. Each year, around 200 million containers are transported among world seaports. The containerized maritime transportation is important to the world trade network. Therefore, a seaport becomes a crucial node in the transportation network. Efficient container terminal operations not only affect the competitiveness of a port, but also release traffic congestion in the transportation network. The berths and cranes are two important resources in the container terminal related to the capacity of a port. The berths are high cost facilities comparing to other facilities in a container terminal, therefore, an efficient and effective usage of berths is critical to a container terminal. When a vessel is allocated to berths, the cranes are scheduled to serve the vessel. The processing time of a vessel is related to the number of cranes serving on the vessel. Therefore, berth allocation and cranes scheduling need to be considered simultaneously for terminal operations.

In the previous studies, there are fruitful researches on berth allocation problem (BAP) (For instance, see Nishimura et al. (2001), Guan et al. (2002), Kim and Moon (2003), Guan and Cheung (2004), Cordeau et al. (2005), Monaco and Sammarra (2007), Imai et al. (2007), Lee and Chen (2008)), and quay crane scheduling problem (QCSP) (For instance,see Daganzo (1989), Peterkofsky and Daganzo (1990), Lim et al. (2002), Li and Vairaktarakis (2004), Lim et al. (2004), Kim and Park (2004), Goodchild and Daganzo (2006), Mak (2006), Lim et al. (2007), Sammarra et al. (2007), Lee et al. (2008b), Lee et al. (2008a)). For BAP, researchers assumed the processing time of each vessel is independent of the number of cranes assigned for that vessel. That is, the processing time is a constant parameter and predefined. For QCSP, researchers mainly dealt with the arrangement of the cranes so that all assigned jobs can be finished as soon as possible . A lot of researchers focused on solving QCSP with non-interference and non-crossing constraints. The vessel berthing allocation is not considered in their studies. Up to now, only a few researchers tried to study BAP and QCSP simultaneously.

In the study of Daganzo (1989), quay cranes are scheduled for incoming vessels for both static and dynamic problems. In his approach, only the scheduling sequence of vessels was considered and no policies were studied on how to allocate vessels to berths. Park and Kim (2003) suggested a two-phase procedure for scheduling berth and quay cranes at the same time. In their approach, they first established a mathematical model to describe the problem. By using Lagrangian relaxation approach, the berth positions for a vessel and number of cranes assigned to that vessel are determined. In the model, he made an assumption that the berthing processing time of a vessel is inversely proportional to the number of the cranes assigned to the vessel. Another important assumption was that every vessel has a most favorable location of berthing. If a solution from Lagrangian relaxation is infeasible, he tuned the Lagrangian multiplier to make an infeasible solution to a feasible solution. Once the berthing position and the number of cranes are determined, in the second phase, the detail schedule of each crane is found by a dynamic programming algorithm. Liu et al. (2005) studied QCSP with the objective to minimize the maximum relative tardiness of vessel departures. Therefore, in his approach, he had to involve the vessel berthing factor into his model. Although he adapted two phases approach to solve the problem, unlike the study of Park and Kim (2003), a vessel berthing pattern is pre-defined already, that is, the berthing positions of each vessel are parameters instead of decision variables. In the study, he focused on how to solve the quay crane scheduling problem. Ak and Erera (2007) established a mixed integer programming model to solve small size instances of simultaneous assignments of berths and quay cranes for vessels. For large size problems, he develop Tabu Search algorithm to solve the near-optimal solutions. Imai et al. (2008b) considered the vessel berthing with limited quay crane capacity. He made an assumption that the handling time of a vessel depends on its allocated berth length, which indicates the processing time of a vessel is not determined by quay crane scheduling. Instead of solving BAP and QCSP together, he solved the variation of BAP to approach BAP and QCSP. Imai et al. (2008a) studied a simultaneous berth and quay crane allocation problem. He introduced a mathematical formulation and employed a genetic algorithm to solve the problem. He improved the genetic algorithm to determine cran transfer scheduling across the berth maximum flow problem-based algorithm.

In this chapter, I first develop a mixed integer programming formulation to describe the problem which combines BAP and QCSP together. I denote this combined problem by BAQCP as an abbreviation. Then, I evaluate the capability of the model and verify that it can solve small size problems. For large size problems, it is hard to solve the problem into optimality in a reasonable amount of time by the exact solution approach. Thus, I develop a heuristic approach to solve the problems to find near-optimal solutions. I analyze the worst case ratio of the heuristics to measure the solution quality of our proposed heuristics. In reality, First-in-First-out (FIFO) is a common policy used to schedule vessel and cranes. Therefore, I also compare the heuristics with FIFO approaches to evaluate the effectiveness of our designed heuristics. Finally, I combine non-preemption heuristic together with exact solution approach to further improve the solution quality.

The remaining part of this chapter is organized as follows. Section **??** describes the problem, lists the notation used in this chapter, and provides a mathematical formulation for the combined problem of berth allocation and quay crane scheduling. In Section 4.3, the preemption heuristic and the non-preemption heuristic are developed. Worst case analysis for both approaches is provided to ensure the solution quality of each heuristics. I also describe the common used FIFO heuristic, which is used as the base for comparison for the two designed heuristic. Section 4.4 provides the computational results, which include the evaluations of the model performance and evaluations of the heuristic performance. I also combine the model and the non-preemption heuristic together to further improve the solution from the non-preemption heuristic. Finally, Section 4.5 concludes this chapter.

4.2 Mathematical formulation

In this section, I discuss the mathematical formulation for the problem. In this formulation, I decide the location of each vessel and the detailed scheduling for each crane. I first make some assumptions which are listed in the following.

- Vessels arrive a terminal at the same time.
- Once a vessel is assigned to moor for berths, it will not leave until all jobs of the vessel are finished.
- A vessel contains several holds, where store containers. The capacity of a hold is agreeable to a vessel size. That is, the larger of a vessel, the larger capacity of the hold in that vessel.
- Once a crane starts to its assigned jobs, it will not stop or work for another assignment. After a crane finish its assigned jobs, if it is not assigned to a new job, it will stay in the same position.
- the crane traveling time is ignored in this study.
- The width of a berth section is wide enough so that two neighboring working cranes will not interfere with each other.
- Each quay crane has the same loading or unloading capacity.
- Workload on a vessel is uniformly distributed.

I can use a time-space diagram to describe BAQCP. The horizontal axis represents time units and the vertical axis represents the berth sections. In this study, one berth section distance represents as one hold width. In the time-space diagram, a vessel is viewed as a rectangle. The height of the rectangle is the vessel size, and the length of the rectangle is the processing time of the vessel, which depends on cranes work on the vessel. A directional arrow linking with the two nodes of a time-space represents

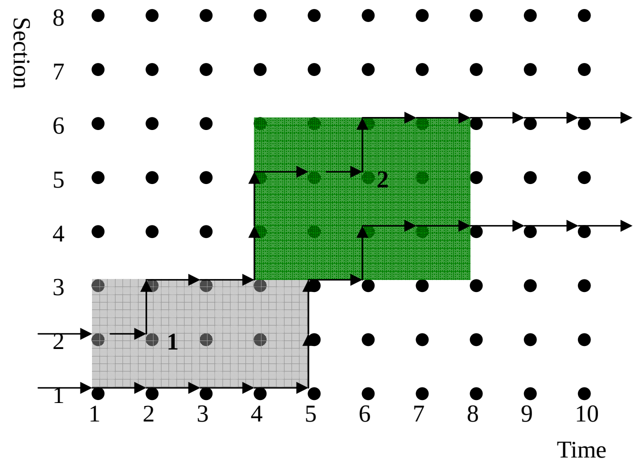

Figure 23: An example of a time-space diagram for two vessels

a crane moving from one position to another position and from one time to next time unit. Figure 23 shows an example of a time-space diagram with two vessels.

In Figure 23, there are two cranes initially located in the section one and section 2. I use the left-bottom of a vessel rectangle to show the vessel is assigned. In Figure 23, one vessel is assigned at time one, on the first section, and the other vessel is assigned at time four, on the section 3. The completion time of each vessel depends on how cranes working for the vessel. In this example, the completion of the first vessel is five, and the completion time of the second vessel is eight.

4.2.1 Notation for the model

Before I describe the mathematical formulation, I at first list all notation used in the mathematical model.

Problem parameters

N: The set of vessels and the total number of vessels is N

K: The set of cranes and the total number of cranes is K

S: The set of berths and the total number of berths is S

T: Time horizon

M: A sufficient large number

Y: The service rate of a crane with the unit "containers/unit time"

L_i: The length of vessel i

U_i: Number of containers in vessel i

B_j: The number of cranes initially located in section j

ν: Processing time of each unit of containers

Decision variables

τ: The completion time of the last vessel

c_i: The completion time of vessel i

w_{ijt}: Number of containers left in vessel i in section j at time t

y_{ijt}: Binary decision variable to indicate if vessel i is assigned in berths j to $j+L_i-1$ starting at time period t. If yes, then $y_{ijt}=1$; otherwise $y_{ijt}=0$

o_{ijt}: Binary decision variable to indicate if there is workload left from vessel i at section j at time t. If yes, then $o_{ijt}=1$; otherwise $o_{ijt}=0$

x_{ijt}: Binary decision variable to indicate if vessel i occupies section j at time t. If yes, then $x_{ijt}=1$; otherwise $x_{ijt}=0$

δ_{it}: Binary decision variable to indicate if the completion time of vessel i is earlier than or equal to time t. If yes, then $\delta_{it}=1$; otherwise $\delta_{it}=0$

p_{jt}: Binary decision variable to indicate if a crane starts to load or unload containers in section j at time t. If yes, then $p_{jt}=1$; otherwise $p_{jt}=0$

$z_{j,j'}^{t,t+1}$: Binary decision variable to indicate if there is a crane moving from section j to section j', from time t to time $t+1$. If yes, then $z_{j,j'}^{t,t+1}=1$; otherwise $z_{j,j'}^{t,t+1}=0$

Due to the fact that a vessel i can not leave until there is no workload left for all sections of the vessel, it can happen that the vessel i occupies section j at time t even though there is no workload left on section j at time t. Thus, I introduce two types of binary decision variables o_{ijt} and x_{ijt} respectively corresponding to vessel i in section j at time t.

4.2.2 Model formulation

I establish a mixed integer programming model to describe and solve BAQCP. According to the descriptions of BAQCP and the notation defined, the model formulation is listed in the following.

$$
\begin{aligned}
& \min \quad \tau \\
c_i &\leq \tau, && (17)\\
to_{ijt} + 1 &\leq c_i, && (18)\\
\sum_{j=1}^{S-L_i+1} \sum_{t=1}^{T} y_{ijt} &= 1, && (19)\\
U_i/L_i \sum_{k=j-L_i+1}^{j} y_{ikt} &\leq w_{ijt}, && (20)\\
w_{ijt-1} - Y z_{j,j}^{t-1,t} &\leq w_{ijt}, && (21)\\
w_{ijt} - M o_{ijt} &\leq 0, && (22)\\
c_i - t &\leq M(1-\delta_{it}), && (23)\\
1 + t - c_i &\leq M\delta_{it}, && (24)\\
1 - M(2 - \sum_{t'=1}^{t} \sum_{k=\max(0,j-L_i+1)}^{j} y_{ikt'} - (1-\delta_{it})) &\leq x_{ijt}, && (25)\\
\sum_{i=1}^{N} x_{ijt} &\leq 1, && (26)\\
z_{j,j+1}^{t,t} + z_{j,j-1}^{t,t} + z_{j,j}^{t,t+1} &= B_j, && (27)\\
z_{j-1,j}^{t,t} + z_{j,j}^{t-1,t} + z_{j+1,j}^{t,t} - z_{j,j+1}^{t,t} - z_{j,j-1}^{t,t} - z_{j,j}^{t,t+1} &= 0, && (28)\\
\sum_{j=1}^{S} z_{j,0}^{t-1,t} &= \sum_{j=1}^{S} B_j, && (29)\\
z_{j-1,j}^{t,t} + z_{j,j}^{t-1,t} + z_{j+1,j}^{t,t} &\leq 1, && (30)\\
z_{j-1,j}^{t,t} + z_{j,j-1}^{t,t} &\leq 1, && (31)\\
w_{ijt} - w_{ijt+1} &\leq M p_{jt}, && (32)\\
z_{j,j}^{t-1,t} + o_{ijt} + p_{jt} - 2 &\leq z_{j,j}^{t,t+1}, && (33)\\
y_{ijt}, o_{ijt}, x_{ijt}, p_{jt}, z_{j,j'}^{t,t+1} \in \{0,1\}, & \qquad \tau, c_i, w_{ijt} \geq 0,
\end{aligned}
$$

The objective of the problem is to minimize the makespan to serve all vessels by assigning cranes to handle all containers.

Constraints (17) and (18) determine the makespan by the completion time of each vessel. For instance, the makespan infers the completion time of the latest finished job, and the completion time for each vessel should be the earliest time that no containers left in each section on this vessel.

Constraints (19) to (22) represent vessel assignment and quay crane loading/unloading containers. For instance, constraint (19) indicates that one vessel is assigned exactly once. Constraint (20) describes that once a vessel is assigned, the corresponding workload will be generated accordingly. Constraint (21) indicates that the remaining workload will be reduced by Y if there is a crane working in the section. Constraint (22) indicates if there is workload left in section j for vessel i at time t.

Constraints (23) to (26) decide the sections and time slots that will be occupied by each vessel i and separate vessels from overlapping with each other. For instance, constraints (23) and (24) indicate if vessel i is finished before or at time t. Constraint (25) shows that section j at time t will be occupied by vessel i if the assignment time of vessel i is before time t, the first assignment section is from $j - L_i + 1$ to j, and vessel i is finished after time t. Constraint (26) indicates each section at each particular time unit can not be occupied by more than one vessel.

Constraints (27), (28), and (29) are network flow balance constraints for quay cranes, which determine the cranes movement.

Constraints (30) and (31) represent the non-interference constraints, in which constraint (30) describes that no more than one crane will stay work for the same section at the same time and constraint (31) prevents cranes from crossing with each other.

Constraints (32) and (33) represent non-preemption constraints, in which constraint (32) represent if crane starts to work on vessel i in section j at time t and constraint (33) indicates that a crane will work for the same vessel once it starts to work for this vessel.

4.3 Methodology

In this section, I develop four heuristics (for the preemption, non-preemption, modified non-preemption heuristic, and FIFO cases) to assign the berths and the cranes for a vessel simultaneously. All assumptions used for the model are applicable for the heuristics. The preemption and non-preemption heuristics contain two steps. The first step is to group the vessels into batches by the limitation of the berth length. The modified non-preemption heuristic is a little different from non-preemption heuristic. For those vessel lengths less then or equal to K berths, I use the limitation of the length K instead of the length S. The second step is to assign certain number of cranes to the vessels. The rule to group vessels is modified from the heuristic developed by Guan et al. (2002). The preemption heuristic allows multiple cranes share the same section jobs. The non-preemption heuristic refers that once a crane starts to load or unload the containers, it will not leave the current section for another job until it finishes the assigned job. FIFO heuristic is a commonly used scheduling policy in the real operation. FIFO is used as a comparison base to examine the performance of the preemption and non-preemption heuristics.

4.3.1 Preemption heuristic H_1

In this study, the preemption is only allowed for the cranes but not for the vessels, because the cost of any re-berthing of a vessel is relatively high and re-berthing is a rare event in the real world. The crane preemption sometimes happens while a crane temporarily is out of order, or a high efficient crane finishes its assigned job and takes over the job of the neighboring low efficient crane. In this section, We design a heuristic that allows a crane preemption happening. Figure 24 shows a three-vessels-and-two-cranes example to explain the preemption heuristic.

The first vessel has two holds, and there are one container within each hold. The second vessel has three holds, and there are two containers within each hold. The third vessel has four holds, and there are three containers within each hold. According to the preemption heuristic, vessels' assignments are divided into two different

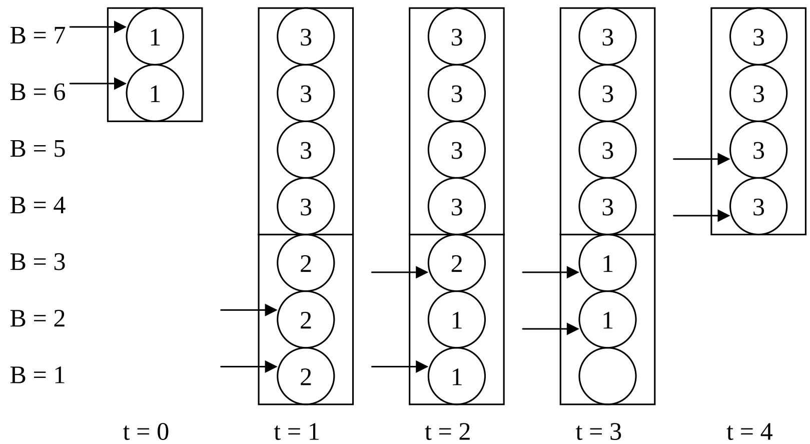

Figure 24: An example to explain H_1 preemption heuristic

groups. The berth allocation of those vessels with size less then or equal to K berths is assigned by the limitation length of K berths (the vessels belong to category 1 in the Step 1 of the heuristic). Cranes are fixed from berth 1 to berth K to serve those vessels of category 1. Other vessels are assigned by the limitation length of S (the vessels belong to category 2 in the Step 1 of the heuristic). At first, these three vessels are grouped into two vessel categories. Because vessel 1 belongs to the category 1, berth 6 and berth 7 are allocated for vessel 1, and two cranes are fixed at section 1 and section 2 to serve vessel 1. At $t = 1$ vessel 1 leave the terminal. vessel 2 and vessel 3 are allocated based on Step 3-1 and Step 3-2. The cranes assignments follow two rules. The first rule is that the vessels belong the smaller vessel group index have higher priority to be served. The second rule is that when the left containers on the berth sections are tight, the sections of the smaller vessel index has higher priority to be served. While a condition that tight sections belongs two different vessel groups happens, the first rule needs to be considered first. By observation by this example, the vessels belong category 2 have a property that a smaller index vessel will leave its assigned berths before a lager index vessel. The details of the H_1 are listed in the following.

The preemption heuristic H_1

Step 0 Sort and renumber the vessels according to the vessel length from the smallest to the largest; that is $L_1 \leq L_2 \cdots \leq L_N$.

Step 1 Separate the vessels to two categories. If $L_i \leq K$, V_i goes to category 1, named V_i^1. Otherwise, V_i is in category 2, named V_i^2. Set $i_0 = \arg\max\{L_i^1\}$.

Step 2 Assign vessels to berth sections and assign cranes to vessels in category 1.

Step 2-1 Assign crane k to berth section k, where $k = 1, \cdots, K$. Crane k keeps working in the section k.

Step 2-2 Let $\{V_l^1, V_{l+1}^1, \cdots, V_{i_0}^1\}$ be the set of ungrouped vessels.

$u^1 = \max\{q | \sum_{i=l}^{q} L_i^1 \leq K \text{ and } q \leq i_0\}$

$G_x^1 = \{V_l^1, \cdots, V_{u^1}^1\}$

Step 2-3 For $r = l, l+1, \cdots, u^1$

(a) if x is odd, then assign V_r^1 to berth sections

$K - \sum_{j=r}^{u^1} L_j + 1$,$K - \sum_{j=r}^{u^1} L_j + 2$,$\cdots$, $K - \sum_{j=r+1}^{u^1} L_j$

(b) if x is even, then assign V_r^1 to berth sections

$\sum_{j=r}^{u^1} L_j + 1$, $\sum_{j=r}^{u^1} L_j + 2$,$\cdots$, $\sum_{j=r}^{u^1} L_j$.

Step 2-4 Set $x = x+1$, repeat Step 2-1 and Step 2-2 until no more vessel in category 1 need to be assigned for berth sections.

Step 2-5 Let $t = 0$. Assign crane k to berth section k, where $k = 1, \cdots, K$. Crane k keeps working in the section k. Set $t = t + 1$. Repeat Step 2-5, until no workload occupied any berth sections. Record the first category finished time t_0.

Step 3 Set $t = t_0$ and assign vessels to berth sections and assign cranes to vessels in category 2.

Step 3-1 Let $\{V_l^2, V_{l+1}^2, \cdots, V_N^2\}$ be the set of ungrouped vessels.

$u^2 = \max\{q| \sum_{i=l}^{q} L_i^2 \leq S \text{ and } q \leq i_0\}$

$G_x^2 = \{V_l^2, \cdots, V_{u^2}^2\}$

Step 3-2 For $r = l, l+1, \cdots, u^2$

(a) if x is odd, then assign V_r^1 to berth sections

$S - \sum_{j=r}^{u^2} L_j + 1$,$S - \sum_{j=r}^{u^2} L_j + 2$,$\cdots$, $S - \sum_{j=r+1}^{u^2} L_j$

(b) if x is even, then assign V_r^2 to berth sections

$\sum_{j=r}^{u^2} L_j + 1$, $\sum_{j=r}^{u^2} L_j + 2$,$\cdots$, $\sum_{j=r}^{u^2} L_j$. Let $t = t_0$.

Step 3-3 Set $p(s,t)$ is priority of section s at time t for $s = 1 \cdots, S$, $t = 1, 2, \cdots$.

Initially set $p(s, t_0) = 0$.

Let $r(s,t)$ record the index of group occupied berth section s at time t.

Let $g(i)$ record the index of group in which the vessel i belongs to.

Let $j(s,t)$ record the index of vessel occupied berth section s at time t.

Let $w(i,s,t)$ record the workload left in vessel i which occupy the berth section s at time t.

Step 3-3-1 At time t,

if $G_{r(s,t)} < G_{r(d,t)}$, $p(s,t) = p(s,t) + 1$

if $g(j(s,t)) = g(j(d,t))$ and $w((j(s,t),s,t) > w(j(d,t),d,t)$, $p(s,t) = p(s,t)+1$.

if $g(j(s,t)) = g(j(d,t))$, $w(j(s,t),s,t) = w(j(d,t),d,t)$, and $j(s,t) < j(d,t)$, $p(s,t) = p(s,t) + 1$.

Step 3-3-2 Sorting $p(s,t)$ at time t, $s = 1, \cdots, S$.

Assign the cranes to the k largest $p(s,t)$ sections.

$w(j(s,t),s,t) = w(j(s,t),s,t) - 1$.

Step 3-4 if there are still workload left $t = t + 1$, go to Step 3-3, else $\tau = t$.

4.3.2 Worst case analysis of H_1

In this section, if H_1 is applied, a smaller size vessel has a smaller processing time whiles a larger size vessel has a longer processing time. This fact indicates the vessel size is agreeable to vessel processing time by applying H_1. By this characteristic, I can apply the study results of Guan et al. (2002) on the relaxed problem to find the worst case bound of the preemption heuristic H_1.

The relaxed problem means for every V_j (V_j is treated as one job), I replace it by L_j identical jobs $\{V_{j1}, V_{j2}, \cdots, V_{jL_j}\}$ with each of unit size, which is shown in Figure 25.

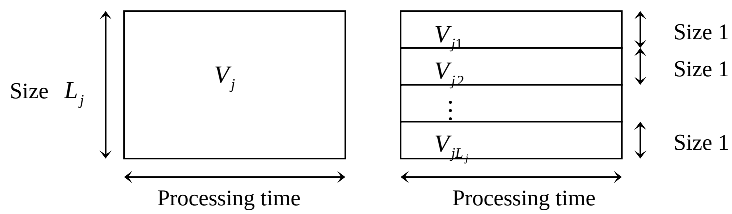

Figure 25: An original job v.s. a relaxed job

Figure 26 shows an example of 5 vessels berth allocation by the heuristic of Guan et al. (2002), and Figure 27 is the corresponding relaxed instance. In Guan et al. (2002), the relationship between the completion time of a job and the completion time of its relaxed job is shown as the following lemma.

Lemma 2 *The completion time of V_j in the heuristic solution (i.e., C_j^H) is at most twice the completion time of the first relaxed vessel rectangle of V_j in the relaxed problem (i.e., C_{j1}^R) for each $j = 1, 2, \cdots, N$. That is, $C_j^H \leq C_{j1}^R$.*

Proposition 4 *Vessel group G_{j+2}^2 will be not served by cranes before all jobs of vessel group G_j^2 is finished.*

Proof: In H_1, there are three priorities settings; first, within each vessel group, sections with larger containers left have higher priority to be served by cranes than sections with smaller containers do. second, within each vessel group, if the sections have the same number of containers left, a smaller index vessel has higher priority to be served by cranes than a larger vessel does; third, if the sections belonging to

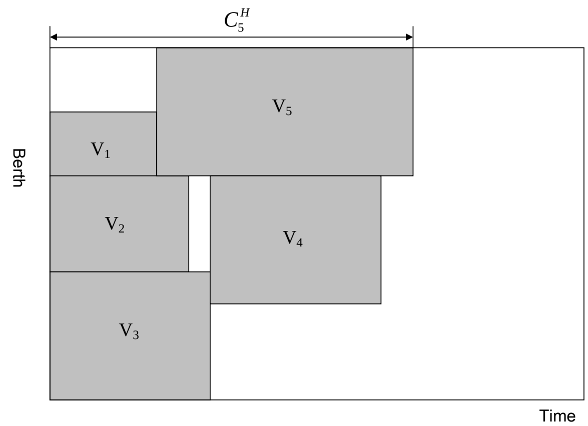

Figure 26: 5 vessels berth allocation by the heuristic developed by Guan et al. (2002)

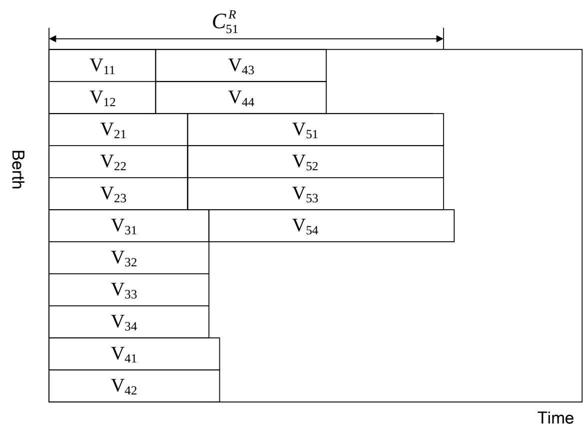

Figure 27: Relaxation of the example of 5 vessels of Figure 26

different vessel groups have the same number of containers left, a smaller index vessel group has higher priority to be served by cranes than a larger vessel does;

According to these three priorities settings, a smaller index vessel has a smaller completion time than a larger vessel does. For example in the Figure 29, the completion time of V_a is less than the completion time of V_b, less than the completion time of V_c. Therefore, $\forall\ V_i \in G_j$ and $\forall\ V_{i'} \in G_j$, the completion time of V_i is smaller than the completion time of $V_{i'}$.

Consider a case, $\exists V_{i''} \in G_{j+2}$, $V_{i''}$ starts to be served by cranes before all vessels belonging to G_j leave berths. There exists at least one $V_{i'} \in G_{j+1}$ leave its assigned berths and its completion time is less than the completion time of the vessel belonging

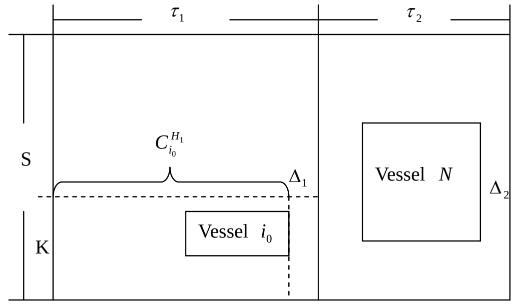

Figure 28: An illustration of H_1

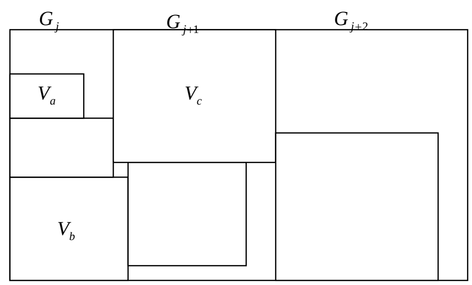

Figure 29: Three vessel groups in category 2

to G_j , which is a contradiction. Therefore, Proposition 4 is valid. □

Proposition 5 *At each time t, at most two vessel group in category 2 will be served by cranes.*

Proof: By Proposition 4, at each time t, there is no chance that cranes will serve G_j, G_{j+1}, and G_{j+2}, therefore, Proposition 5 is valid. □

Proposition 6 *If the number of vessels tends to infinity, the ratio between makespan by H_1 and the optimal value of BAQCP is at most two.*

Proof: By observing Figure 28, the time before τ_1, that is, for the vessels of category 1, I can apply Lemma 2, i.e., $C_{i_0}^{H_1} \leq 2C_{i_0 1}^{R}$, and $\tau_1 = C_{i_0}^{H_1} + \Delta_1 \leq 2C_{i_0 1}^{R} + \Delta_1$
For the vessels of category 2, according to Proposition 4, Proposition 5, and priority settings of H_1, each vessel size is greater than K berths length, which indicates at

each time, all K cranes are serving vessels. Therefore, $\tau_2 = \sum_{i=i_0+1}^{N} U_i/K + \Delta_2 \leq 2\sum_{i=i_0+1}^{N} U_i/K + \Delta_2$

$$\tau = \tau_1 + \tau_2 \leq 2C_{i_0 1}^{R} + \Delta_1 + 2\sum_{i=i_0+1}^{N} U_i/K + \Delta_2$$

Let lower bound $L = C_{i_0 1}^{R} + \sum_{i=i_0+1}^{N} U_i/K \leq Z^*$, therefore,

$$\tau \leq 2Z^* + \Delta_1 + \Delta_2$$

$$\tau/Z^* \leq 2 + (\Delta_1 + \Delta_2)/Z^*$$

As $N \to \infty$, $(\Delta_1 + \Delta_2)/Z^* \to 0$

$$\tau/Z^* \leq 2$$

□

4.3.3 Preemption heuristic H_{1a}

In the Section 4.3.1, H_1 separates vessels into different categories according to the vessel size by the length limitation of K berths. Worst case analysis of H_1 shows that the worst case bound is two. In this section, I relax the constraint of the K berth length limitation, more berth space can be used by vessels with the length less than K berths length. I name this heuristic as H_{1a}, which provides a comparison base for H_1 to examine whether more space used by vessels will be more efficient or not. The results will show in the Section 4.4 and the detail steps of H_{1a} are listed in the following.

The preemption heuristic H_{1a}

Step 0 Sort and renumber the vessels according to the vessel length from the smallest to the largest; that is $L_1 \leq L_2 \cdots \leq L_N$.

Step 1 Let $\{V_l, V_{l+1}, \cdots, V_N\}$ be the set of ungrouped vessels.

$u = \max\{q | \sum_{i=l}^{q} L_i \leq S \text{ and } q \leq i_0\}$

$G_x = \{V_l, \cdots, V_u\}$

Step 2 For $r = l, l+1, \cdots, u^2$

(a) if x is odd, then assign V_r^1 to berth sections

$S - \sum_{j=r}^{u^2} L_j + 1, \cdots, S - \sum_{j=r+1}^{u^2} L_j$

(b) if x is even, then assign V_r^2 to berth sections

$\sum_{j=r}^{u^2} L_j + 1, \sum_{j=r}^{u^2} L_j + 2, \cdots, \sum_{j=r}^{u^2} L_j$. Let $t = 0$.

Step 3 Set $p(s,t)$ is priority of section s at time t for $s = 1\cdots, S,\ t = 1, 2, \cdots$.

Initially set $p(s, t_0) = 0$.

Let $r(s,t)$ record the index of group occupied berth section s at time t.

Let $g(i)$ record the index of group in which the vessel i belongs to.

Let $j(s,t)$ record the index of vessel occupied berth section s at time t.

Let $w(i,s,t)$ record the workload left in vessel i which occupy the berth section s at time t.

Step 3-1 At time t,

if $G_{r(s,t)} < G_{r(d,t)}$, $p(s,t) = p(s,t) + 1$

if $g(j(s,t)) = g(j(d,t))$ and $w((j(s,t), s, t) > w(j(d,t), d, t)$, $p(s,t) = p(s,t) + 1$.

if $g(j(s,t)) = g(j(d,t))$, $w(j(s,t), s, t) = w(j(d,t), d, t)$, and $j(s,t) < j(d,t)$, $p(s,t) = p(s,t) + 1$.

Step 3-2 Sorting $p(s,t)$ at time t, $s = 1, \cdots, S$.

Assign the cranes to the k largest $p(s,t)$ sections.

$w(j(s,t), s, t) = w(j(s,t), s, t) - 1$.

Step 4 if there are still workload left $t = t + 1$, go to Step 3, else $\tau = t$.

4.3.4 Non-preemption heuristic (H_2)

H_2 has the only different assumption from H_1, preemption is not allowed not only for vessels, but also for cranes. That is, a section job can not be shared by multiple cranes. Vessels $V_1, V_2, \cdots, V_N$ are sorted and renumbered according to the vessel lengths from the smallest to the largest. If there are i_0 vessels with the lengths less than K berths length, vessels are assigned by Step 2 of H_2 by using the length limitation of K berths length. K cranes are fixed from berth 1 to berth K and do not move. For the vessels in which lengths are larger than K $(V_{i_0+1}, V_{i_0+2}, \cdots, V_N)$, vessel assignments are based on the berth length S instead of length K. The crane assignments are based on the threshold method in the Step 4-6 of H_2.

Figure 30 shows an example with three vessels and four cranes by H_2. In the example, the length of the first vessel is two and each hold has one container; the length of the second vessel is two and each hold has two containers; the length of the third vessel is six and each hold has two containers. According to the H_2, the first two vessels belong to category 1, and the third vessel belongs to category 2. For vessels in the category 1, 4 cranes are fixed from berth 4 to berth 7. At t = 3, the first and the second vessel are finished serving, and at this time, vessel assignments are based on the total berth length limitation. Therefore, the third vessel are assigned from berth 1 to berth 6. Crane assignments is based on the threshold method and therefore, the first crane serves section 1 and section 2, the second crane serves section 3 and section 4, and the third crane serves section 5 and second 6. The fourth crane is idle after t = 3. The makespan of this example is seven. The details of the H_2 are listed in the following.

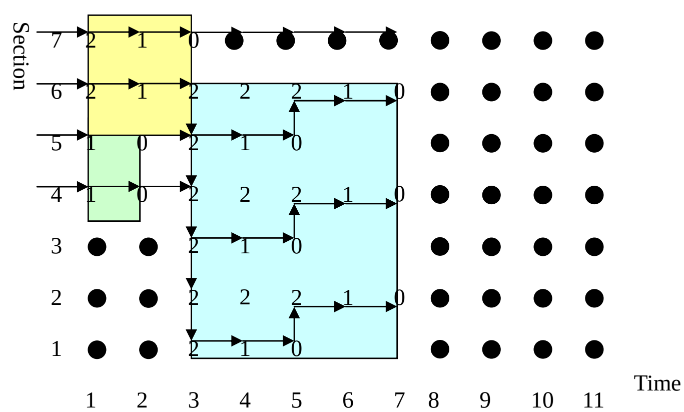

Figure 30: An example of 3 vessels and 4 cranes by H_2

The non-preemption heuristic H_2

Step 0 Sort and renumber the vessels according to the vessel length from the smallest to the largest; that is $L_1 \leq L_2 \cdots \leq L_N$. Set $j \leftarrow 1$

Step 1 Separate the vessels to two categories. If $L_i \leq K$, V_i goes to category 1, named V_i^1. Otherwise, V_i is in category 2, named V_i^2. Set $i_0 = \arg\max\{L_i^1\}$.

Step 2 Assign vessels to berth sections and assign cranes to vessels in category 1.

Step 2-1 Assign crane k to berth section k, where $k = 1, \cdots, K$. Crane k keeps working in the section k.

Step 2-2 Let $\{V_l^1, V_{l+1}^1, \cdots, V_{i_0}^1\}$ be the set of ungrouped vessels.

$u^1 = \max\{q| \sum_{i=l}^{q} L_i^1 \leq K \text{ and } q \leq i_0\}$

$G_j^1 = \{V_l^1, \cdots, V_{u^1}^1\}$

Step 2-3 For $r = l, l+1, \cdots, u^1$

(a) if j is odd, then assign V_r^1 to berth sections

$K - \sum_{i=r}^{u^1} L_i + 1$, $K - \sum_{i=r}^{u^1} L_i + 2 \cdots$, $K - \sum_{i=r+1}^{u^1} L_i$

(b) if j is even, then assign V_r^1 to berth sections

$\sum_{i=r}^{u^1} L_i + 1$, $\sum_{i=r}^{u^1} L_i + 2$,$\cdots$, $\sum_{i=r}^{u^1} L_i$.

Step 2-4 Set $j = j+1$, repeat Step 2-1 and Step 2-2 until no more vessel in category 1 need to be assigned for berth sections. Record the completion time of Vessel i_0, denoted t_{i_0}, and start to assign vessels in category 2 from time t_{i_0}.

Step 3 $j^0 = \arg\max\{G_j^1 | \exists V_i^1 \in G_j^1\}$ Let $j \leftarrow j^0 + 1$.

Step 4 Assign vessels to berth sections and assign cranes to vessels in category 2.

Step 4-1 Let $\{V_\ell^2, V_{\ell+1}^2, \cdots, V_N^2\}$ be the set of ungrouped vessels.

$u^2 = \max\{q| \sum_{i=\ell}^{q} L_i \leq S \text{ and } q \leq N\}$

$G_j^2 \leftarrow \{V_\ell^2, V_{\ell+1}^2, \cdots, V_u^2\}$

Step 4-2 For $r = \ell, \ell+1, \cdots, u$

(a) if j is odd, then assign V_r^2 to berth sections

$S - \sum_{i=r}^{u^2} L_i + 1$, $S - \sum_{i=r}^{u^2} L_i + 2$, $\cdots$, $S - \sum_{i=r+1}^{u^2} L_i$

(b) if j is even, then assign V_r^2 to berth $\sum_{i=r+1}^{u^2} L_i + 1$, $\sum_{i=r+1}^{u^2} L_i + 2$, $\cdots$, $\sum_{i=r}^{u^2} L_i$

Step 4-3 Set $j \leftarrow j+1$, repeat Step 4-1 and Step 4-2 until no more vessel needs to be determined for berths. Then let $t = t_{i_0}$ and $j \leftarrow j^0 + 1$.

Step 4-4 Calculate h_j the threshold value for assignments of cranes in G_j^2,i.e., $h_j = \nu \sum_i U_{ji}/K$, in which U_{ji} is the total number of containers in V_i^2 where $V_i^2 \in G_j^2$. K is the number of cranes.

Step 4-5 Let c_x be the initial location of Group G_x^2. If x is odd, $c_x = S - \sum_i L_i$; if x is even, $c_x = 1$. L_i is the size of the vessel V_i^2, where $V_i^2 \in G_x^2$.

Step 4-6 $R = \{R_{x1}, R_{x2}, \cdots, R_{x_K}\}$ records partitions of group G_x^2. R_{xq} is the q^{th} partition of group G_x^2 worked by the crane q, and $1 \leq q \leq K$. c_{jq} is the first section of R_{jq} and d_{jq} is the last section of R_{jq}. R_{jq} is determined by $\nu \sum_{i=c_{jq}}^{d_{jq}-1} W_{ji} \leq h_j$ and $\nu \sum_{i=c_{jq}}^{d_{jq}} W_{ji} > h_j$

if j is odd, then assign partitions from R_{j_K} to R_{j_1}

if j is even, then assign partitions from R_{j_1} to R_{j_K}

Step 4-7 Let F_{jq} represent processing time of a crane q serving the partition R_{xq} for vessels in group G_j^2, where $q = 1, 2, \cdots, K$. Then, $F_{jq} = \sum_{i=c_{jq}}^{d_{jq}} W_{ji}$ and $T_j = \max_{q=1}^{K} \sec\{F_{jq}\}$.

Step 4-8 Update $t = t + T_j$.

Step 4-9 Set $j \leftarrow j+1$, repeat Step 4-4 to Step 4-8 until no vessel group left.

Step 5 Makespan $\tau = t$

4.3.5 Worst case analysis of H_2

Before I precede the worst case analysis of the H_2, I consider one vessel group which are served by K cranes and within in this vessel group, the size of each vessel is larger then K berths. For this special case, the following Lemma is obtained.

Lemma 3 *The ratio between makespan and lower bound is at most two for the one vessel group and multiple cranes problem.*

Proof: Assume a working zone partition of one crane for an one-vessel-group with-multiple-cranes instance start from section a to section b. This working zone is determined by time threshold $h_j = \nu \sum_{i=1}^{S} W_{ji}/K$, where W_{ji} represents the initial number of containers for a section i of an arbitrary vessel group j. Let T_1 is the processing time for a crane to work from section a to section $(b-1)$, which is less than h_j and T_2 is the processing time for a crane to work for the section b, which is less than $\nu \max_i\{W_{ji}\}$. In this special case, lower bound $L = \max\{\nu \sum_{i=1}^{S} W_{ji}/K, \nu \max_i\{W_{ji}\}\}$. A feasible solution $\tau = T_1 + T_2$,

$$\tau = T_1 + T_2 \leq \frac{\nu \sum_{i=1}^{S} W_{ji}}{K} + \nu \max_i\{W_{ji}\} \leq 2L$$

Therefore, I can conclude that the Lemma 3 is valid. □

In the following, I apply the Lemma 2 and Lemma 3 to find the worst case bound of the a multiple vessels with multiple cranes case by H_2.

Proposition 7 *The ratio between makespan by H_2 and the optimum (Z^*)of BAQCP is at most two.*

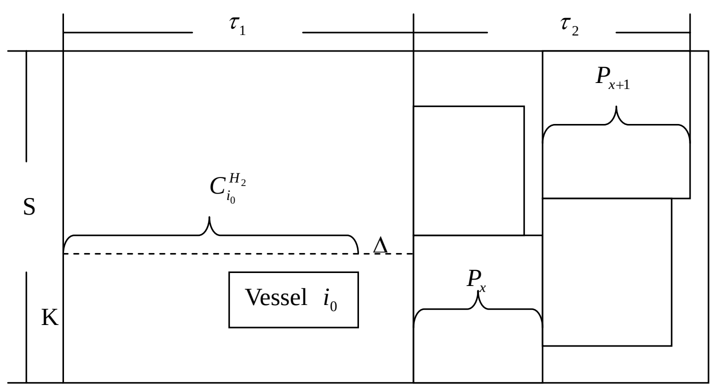

Figure 31: An illustration of H_2

Proof: Figure 31 shows a general case by H_2. Similar to the proof for H_1, by Lemma 2, $\tau_1 = C_{i_0}^{H_2} + \Delta \leq 2C_{i_01}^{R} + \Delta$. And $\tau_2 = P_{j'+1} + P_{j'+2} + \cdots + P_{\mathcal{N}}$, where j' is a vessel group which contains the vessel i_0.

Let L_{j1} and L_{j2} be two lower bounds of the vessel group j, and $\Gamma_j = \max\{L_{j1}, L_{j2}\}$, in which $L_{j1} = \nu \max_i\{W_{ji}\}$ and $L_{x2} = \nu \sum_i W_{ji}/K$. P_j is defined as a processing time for a vessel group j. According to Lemma 3, for each vessel group, $P_j \leq 2\Gamma_j$. Therefore,

$$\begin{aligned}
\tau_2 &= P_{j'+1} + P_{j'+2} + \cdots + P_{\mathcal{N}} \\
&\leq 2(\Gamma_{j'+1} + \Gamma_{j'+2} + \cdots + \Gamma_{\mathcal{N}}) \\
&\leq 2(\Gamma_{j'+1} + L_{(j'+1)2} - L_{(j'+1)2} + \\
&\quad \Gamma_{j'+2} + L_{(j'+2)2} - L_{(j'+2)2} + \cdots + \\
&\quad \Gamma_{\mathcal{N}} + L_{\mathcal{N}2} - L_{\mathcal{N}2})
\end{aligned}$$

It is easy to prove that $\Gamma_{j-1} \leq L_{j2}$ is valid while vessel group length is lager then K berth length. Hence,

$$\begin{aligned}
\tau_2 &= P_{j'+1} + P_{j'+2} + \cdots + P_{\mathcal{N}} \\
&\leq 2\nu \sum_i U_i/K + 2(\Gamma_{\mathcal{N}} - L_{(j'+1)2} - \delta) \\
\delta &= L_{(j'+2)2} - \Gamma_{j'+1} + L_{(j'+3)2} - \Gamma_{j'+2} \cdots + L_{\mathcal{N}2} - \Gamma_{\mathcal{N}-1}
\end{aligned}$$

$$\tau = \tau_1 + \tau_2 \leq 2C^R_{i_0 1} + \Delta + 2(\nu \sum_i U_i/K) + 2\Gamma_{\mathcal{N}} \leq 2Z^* + \Delta + 2\Gamma_{\mathcal{N}}$$

$$\tau/Z^* \leq 2 + (\Delta + 2\Gamma_{\mathcal{N}})/Z^*$$

As $N \to \infty$, $(\Delta + 2\Gamma_{\mathcal{N}})/Z^* \to 0$

$$\tau/Z^* \leq 2$$

□

4.3.6 Non-preemption heuristic (H_{2a})

H_{2a} is an alternative non-preemption heuristic that does not separates vessels into different categories according to the vessel size by the length limitation of K berths. The detail steps of H_{2a} are listed in the following.

The non-preemption heuristic H_{2a}

Step 0 Sort and renumber the vessels according to the vessel length from the smallest to the largest; that is $L_1 \leq L_2 \cdots \leq L_N$. Set $j \leftarrow 1$

Step 1 Let $\{V_\ell^2, V_{\ell+1}^2, \cdots, V_N^2\}$ be the set of ungrouped vessels.
$u^2 = \max\{q | \sum_{i=\ell}^{q} L_i \leq S \text{ and } q \leq N\}$
$G_j^2 \leftarrow \{V_\ell^2, V_{\ell+1}^2, \cdots, V_u^2\}$

Step 2 For $r = \ell, \ell+1, \cdots, u$

(a) if j is odd, then assign V_r^2 to berth sections
$S - \sum_{i=r}^{u^2} L_i + 1,\ S - \sum_{i=r}^{u^2} L_i + 2,\ \cdots,\ S - \sum_{i=r+1}^{u^2} L_i$

(b) if j is even, then assign V_r^2 to berth $\sum_{i=r+1}^{u^2} L_i + 1,\ \sum_{i=r+1}^{u^2} L_i + 2,\ \cdots,$
$\sum_{i=r}^{u^2} L_i$

Step 3 Set $j \leftarrow j+1$, repeat Step 1 and Step 2 until no more vessel needs to be determined for berths. Then let $t = t_{i_0}$ and $j \leftarrow j^0 + 1$.

Step 4 Calculate h_j the threshold value for assignments of cranes in G_j^2,i.e., $h_j = \nu \sum_i U_{ji}/K$, in which U_{ji} is the total number of containers in V_i^2 where $V_i^2 \in G_j^2$. K is the number of cranes.

Step 5 Let c_x be the initial location of Group G_x^2. If x is odd, $c_x = S - \sum_i L_i$; if x is even, $c_x = 1$. L_i is the size of the vessel V_i^2, where $V_i^2 \in G_x^2$.

Step 6 $R = \{R_{x1}, R_{x2}, \cdots, R_{x_K}\}$ records partitions of group G_x^2. R_{xq} is the q^{th} partition of group G_x^2 worked by the crane q, and $1 \leq q \leq K$. c_{jq} is the first section of R_{jq} and d_{jq} is the last section of R_{jq}. R_{jq} is determined by $\nu \sum_{i=c_{jq}}^{d_{jq}-1} W_{ji} \leq h_j$ and $\nu \sum_{i=c_{jq}}^{d_{jq}} W_{ji} > h_j$
if j is odd, then assign partitions from R_{j_K} to R_{j_1}
if j is even, then assign partitions from R_{j_1} to R_{j_K}

Step 7 Let F_{jq} represent processing time of a crane q serving the partition R_{xq} for vessels in group G_j^2, where $q = 1, 2, \cdots, K$. Then, $F_{jq} = \sum_{i=c_{jq}}^{d_{jq}} W_{ji}$ and $T_j = \max_{q=1}^{K} \sec\{F_{jq}\}$.

Step 8 Update $t = t + T_j$.

Step 9 Set $j \leftarrow j + 1$, repeat Step 4 to Step 8 until no vessel group left.

Step 10 Makespan $\tau = t$

4.3.7 FIFO heuristic

In the real world, the FIFO heuristic is commonly used for allocating the berths and the quay cranes to the vessels. In FIFO algorithm, each time-space node has two main attributes: if the node is occupied by a vessel or not and if a crane is available or not. The algorithm continuously checks time by time and vessel by vessel if there are enough consecutive berths and available cranes for a vessel until no vessel needs to be allocated for berths and cranes. Figure 32 shows an example with two cranes and three vessels. At time 1, the first two berths and two cranes are available and therefore, the first vessel can be assigned for the berths and the cranes. Although there are still space for the second vessel, there is no crane available. Therefore, the second vessel is not assigned until time 2. With the same reason, the third vessel has to wait until there are both berths and cranes available, then it can be assigned at time 4. In this section, the FIFO heuristic is proposed as a comparison base for the H_1, H_{1a}, H_2, and H_{2a} which are listed in the following.

The FIFO heuristic

Step 1 Let b_v^i record the finish time of vessel occupying the berth i. Let a_v^k record the finish time of vessel worked by the crane k at time v. Initially set $b_0^i = 0$ and $a_t^k = 0$ for $s = 1, ..., S$, $k = 1, \cdots, K$, and $t = 0$.

Step 2 Check whether there exist enough berth sections and cranes for vessel V_i at time t.

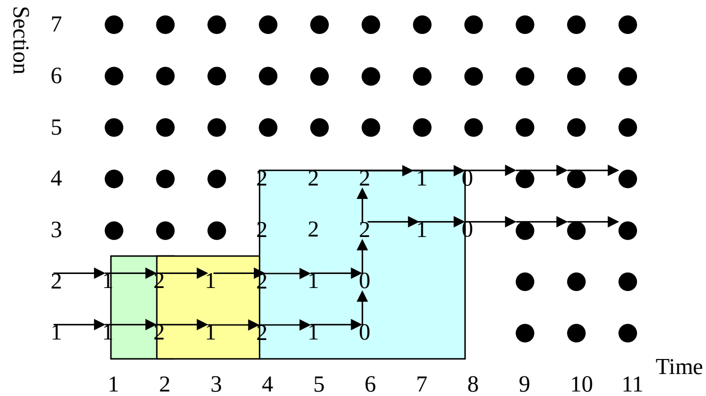

Figure 32: An example of 3 vessels and 2 cranes by $FIFO$

Step 2-1 Check whether there exist enough berth sections at time t for vessel V_i.

Step 2-1-1 Let $I_t = \{i|b_t^i < t\}$ be the assignable section set at time t. Record as $I_t = \{i_t^1, \cdots, i_t^n\}$

If $I_t = \emptyset$, $t = t + 1$, go to Step 2.

Let l_t be the length of assignable sections, f_t^B be the first assignable section for V_i.

And initially set $l_t = 1$, $f_t^B = 0$, index $j = 0$.

Step 2-1-2 From jth element in I_t,

if $i_t^{j+1} = i_t^j + 1$, $l_t = l_t + 1$, $j = j + 1$, go to Step 2-1-2; otherwise,

if $j = n$ and $l_t < L_i$, set $t = t + 1$, Go to Step 2,;

if $j < n$ and $l_t < L_i$, $l_t = 1$, $f_t^B = i_t^j$, go to Step 2-1-2 until $j = n$;

if $j \leq n$ and $l_t > L_i$ then go to Step 2-2.

Step 2-2 Check whether there exist enough cranes at time t for vessel V_i.

Step 2-2-1 Let $A_t = \{k|a_t^k < t\}$ be the available crane set at time t. Record as $A_t = \{k_t^1, k_t^2, \ldots, k_t^m\}$.

If $A_t = \emptyset$, $t = t + 1$, go to Step 2.

Let h_t record the number of available cranes for V_i at time t. Let f_t^C be the the first available crane. Initially set $h_t = 1$, $f_t^C = 0$, index $j = 0$.

Step 2-2-2 From j^{th} element of A_t,

if $k_t^{j+1} = k_t^j + 1$, $h_t = h_t + 1$, $j = j + 1$, go to Step 2-2-2; otherwise,

if $j = m$, $t = t + 1$, go to Step 2;

if $j < m$, $h_t = 1$, $f_t^C = k_t^j$, go to Step 2-2-2.

Step 3 Assign vessel V_i to berth sections and assign cranes for V_i at time t. Set $i = i + 1$.

Step 3-1 Assign vessel V_i to the berth sections $f_t^B, \cdots, f_t^B + L_i - 1$.

Step 3-2 Assign cranes to vessel V_i and update b_t^i and a_t^k

If $h_t \geq L_i$,

assign crane $f_t^C + j$ to section$f_t^B + j$, $j = 0, \cdots, L_i - 1$.

Set available time for berths and cranes which assign to vessel i:

$b_v^i = a_v^k = t + t_i$,$i = f_t^B, \cdots, f_t^B + L_i - 1$, $k =$$f_t^C, \cdots, f_t^C + L_i - 1$,

$v = t, \cdots, t + t_i$.

If $h_t < L_i$,

assign crane $f_t^C + j$ to sections, $f_t^B + j \times \lfloor L_i/h_t \rfloor, \cdots, f_t^B + (j+1) \times \lfloor L_i/h_t \rfloor$,

$j = 1$,...,$h_t - 2$;

assign crane $f_t^C + h_t - 1$ to sections $f_t^B + (L_i - 1) \times \lfloor L_i/h_t \rfloor, \cdots, f_t^B + L_i - 1$.

Set available time for berth sections and cranes which assign to

V_i: $f_t^B + (h_t - 1) \times \lfloor L_i/h_t \rfloor, \cdots, f_t^B + L_i - 1$

$b_v^i = a_v^k = t + t_i \times (L_i - h_t + 1) \times \lfloor L_i/h_t \rfloor$, $k =$$f_t^C, \cdots, f_t^C + h_i - 1$,

$i = f_t^B$,...,$f_t^B + L_i - 1$, $v = t, \cdots, t + t_i \times (L_i - h_t + 1) \times \lfloor L_i/h_t \rfloor$.

Step 4 If $i = N$, the maximum makespan $\tau_0 = t$.

4.4 Computational results

In this section, I first examine the capabilities of the mathematical model. Then, I compare H_1, H_2, modified H_2, and $FIFO$ for different experimental settings. Finally, I combine the exact solution approach and the heuristic approach together to take advantages from both exact and heuristic approaches to further improve the solution quality. This approach has two steps, in which the first step is to find an initial feasible solution from non-preemption heuristic H_2. Then in the second step, I will use this initial solution as a starting point for the mathematical model and run the mathematical model for a period of time to improve the solution quality. The experiments are executed by an INTEL dual core computer with T3200 CPU and 1 GB ram memory.

4.4.1 Performance of the model

Both BAP and QCSP are proved to be NP-hard problems. Since BAP and QCSP are two sub-problems of BAQCP, it implies that BAQCP is also an NP-hard problem. For some instances, especially for large size problems, it is hard to get the optimum solution within a reasonable amount of time. In order to examine the capacities of the model, I create different scenarios to test the model by using default CPLEX. In the experiments, I set the execution time limitation to be 30 minutes and optimality tolerance is 1%. For instance, the default CPLEX will stop and return the best objective value if the execution time is more than 30 minutes. If the error gap is under 1%, the default CPLEX will return the best solution as an optimal solution. In the time space model, I apply an aggregation strategy to let 1 time unit to be the time processing 50 containers. I test different combinations such that the number of vessels to be 2 and 4, the number of cranes to be 4, 7 and 10, and the number of berth sections to be 10, and 15 respectively. Table 13 lists the experimental settings for the model evaluation. For each combination setting, I run 5 instances and report the average value to represent the outcome for the corresponding combination setting.

The performance of the model is shown in Table 14. In the table, I use the

Table 13: Experimental setting for the model evaluations

Execution Time	30 minutes
Time unit	50 containers processing time
Number of vessels (V)	2 and 4
Number of cranes (K)	4,7, and 10
Vessel size (S)	5 and 10
Berth sections (B)	10 and 15
Number of containers per section	Case 1: 50 to 200 containers (1 to 4 time units) Case 2: 250 to 450 containers (5 to 9 time units) Case 3: 500 to 700 containers (10 to 14 time units)

optimality gap, namely, the percentage difference between feasible solution and lower bound, to show the performance of default CPLEX. For instance, I let $gap = (Z - Z_L)/Z \times 100\%$, where Z and Z_L represent the best feasible solution and lower bound that default CPLEX finds within the time limit. If the result shows 100%, it indicates no solution return within time limitation; if the result is 0%, it represents the default CPLEX can return the optimum; and if the result shows the optimality gap between 1% and 100%, the default CPLEX returns a feasible solution. Table 14 shows that when there are 2 vessels, most of the instance can return feasible solutions. However, if I increase the number of vessels from 2 to 4, most of cases have no feasible solution return.

Table 14: Results of relative gap % to the optimum for two vessel cases

V=2		B=10			B=15		
		Case 1	Case2	Case3	Case 1	Case 2	Case 3
K=4	S=5	87.9%	88.9%	92.3%	88.9%	94.6%	98.3%
	S=10	100%	100%	100%	100%	100%	100%
K=7	S=5	0%	3.8%	30.6%	22.2%	31%	54.2%
	S=10	100%	100%	100%	100%	100%	100%
K=10	S=5	0%	0%	0%	0%	0%	0%
	S=10	0%	0%	0%	0%	0%	0%

Table 16 summarizes the experiments including the number of optimum returns, the number of instances that there are feasible solution returning, and the number of

Table 15: Results of relative gap % to the optimum for four vessel cases

V=4		B=10			B=15		
		Case 1	Case2	Case3	Case 1	Case 2	Case 3
K=4	S=5	100%	100%	100%	100%	100%	100%
	S=10	100%	100%	100%	100%	100%	100%
K=7	S=5	51.9%	90.9%	97.2%	78.2%	99.5%	97.9%
	S=10	100%	100%	100%	100%	100%	100%
K=10	S=5	0%	30%	73.4%	23.9%	82.1%	79.9%
	S=10	0%	17.2%	87.9%	49.4%	100%	100%

instances that there are no feasible solution returning. It can be observed that the modeling approach can only solve small size problems.

Table 16: Results of two and four vessel cases

	# of optimum	# of feasible	# of infeasible	Total # of instances
2 vessels	69	83	28	180
4 vessels	13	50	117	180

4.4.2 Performance of the heuristics

I use the heuristic approaches for large size problems. In this section, I evaluate the performance of four heuristics, H_1, H_{1a}, H_2 and H_{2a}, and compare them with the commonly used FIFO heuristic. I generate different combinations in terms of the number of berth sections and the number of vessels. For each combination, I test 10 instances and report the average value of the 10 instances. Table 17 shows the results of performance of the heuristics. In the table, S, K, and V represent the number of berth sections, the number of cranes, and the number of vessels respectively. The vessel size is randomly generated in the interval $[2, 7]$. The number of containers per section is uniformly distributed in the interval $[100, 300]$. The number in the table represents the ratio between the makespan of the heuristics and the lower bound for BAQCP. The lower bound of BAQCP is the time that total containers are served by total cranes without any delay. "AVE" represents the average "ratio" of the "10" instances, and "MAX" represents the maximum "ratio" of the "10" instances. After testing all combinations, I report the overall average ratio for each approach and the corresponding standard derivation.

From the experiment results, it is easy to observe that among all the heuristics, the preemption heuristic H_{1a} has the smallest ratio. For instance, the maximum average ratio is 1.49 for H_1a, which is much smaller than that of H_2a and FIFO heuristic. Besides this, Non-preemption heuristic (H_2, H_{2a}) performs better than FIFO heuristic for most cases. The computational results also show that the average ratios obtained from H_2 heuristic are smaller than those from H_{2a}, which indicates that if I separate vessels into two different groups, one vessel group contains vessels' length less then or equal to K berths, the other contains vessels' length larger than K berths, solution quality of this approach is better than if I do not separate vessels into two different groups for non-preemption heuristics. However, preemption heuristics show an opposite tendency.

I can also observe that numerical results indicate that the ratios for both preemption and non-preemption heuristics are smaller than the theoretical results I prove in the previous sections. From the reported standard deviation results, I can observe that the preemption heuristic H_{1a} and non-preemption heuristic H_2 are more stable than those from FIFO heuristic. In practice, the non-preemptive operation is more popular than preemptive operation. From both the analytical point and the practical point of views, the non-preemption heuristics are proper for BAQCP.

4.4.3 Post optimization by applying the exact solution approach

Although the heuristic approach is reasonable for large size problems, the solution quality could be possibly improved by applying the exact solution approach, for the given initial solution provided by the heuristics. In this section, I combine the exact modeling and the heuristic approaches to take advantages from both approaches to improve the solution quality.

I propose a two phase method to combine the model and the heuristic approaches. In the first stage, I run the non-preemption heuristic to acquire an solution that includes all initial values of the decision variables in the model. I use this feasible

Table 17: Heuristics performance evaluations

	AVE					MAX				
(S,K,V)	H_1	H_{1a}	H_2	H_{2a}	FIFO	H_1	H_{1a}	H_2	H_{2a}	FIFO
$(8,2,20)$	1.02	1.02	1.14	1.15	1.13	1.04	1.04	1.17	1.21	1.17
$(8,2,30)$	1.03	1.02	1.16	1.16	1.13	1.05	1.02	1.20	1.19	1.17
$(8,6,20)$	1.23	1.13	1.21	1.69	1.87	1.28	1.23	1.35	1.85	2.07
$(8,6,30)$	1.16	1.12	1.29	1.63	1.85	1.24	1.13	1.42	1.7	1.91
$(12,4,20)$	1.09	1.06	1.2	1.26	1.31	1.17	1.11	1.25	1.3	1.46
$(12,4,30)$	1.1	1.04	1.22	1.22	1.31	1.13	1.06	1.24	1.24	1.39
$(12,9,20)$	1.38	1.24	1.38	1.54	1.83	1.46	1.49	1.46	1.64	2.12
$(12,9,30)$	1.36	1.2	1.36	1.57	1.84	1.43	1.38	1.43	1.63	1.97
$(16,4,20)$	1.09	1.05	1.16	1.19	1.33	1.17	1.09	1.19	1.25	1.48
$(16,4,30)$	1.1	1.06	1.16	1.19	1.32	1.14	1.09	1.2	1.21	1.51
$(16,12,20)$	1.34	1.14	1.34	1.72	1.93	1.51	1.26	1.51	1.78	2.22
$(16,12,30)$	1.23	1.09	1.28	1.72	1.85	1.39	1.19	1.39	1.8	2.07
Average	1.18	1.1	1.24	1.42	1.56					
S.D.	0.13	0.07	0.08	0.24	0.32					

solution as an initial point for the Branch and Bound method of default CPLEX solver for further improvements. I create 18 instances for the experiments. I generate instances that contain 8, 10 and 12 vessels and 4 and 6 cranes respectively. For each combination of the number of the vessels and the number of the cranes, I let the number of berth sections to be 10 and generate 3 different parameter settings in terms of vessel sizes and the number of containers per section. The vessel sizes are uniformly generated in the interval $[1, 8]$, and the number of containers per section is uniformly generated in the interval $[4, 11]$, where 1 unit represents the working time for 50 containers.

I set time limit to be 24 hours for the post optimization procedure and Table 18 demonstrates the results. In the table, for each combination, I list the average objective value corresponding to initial feasible solutions obtained by the heuristic (i.e., the first row), the final objective value obtained by the mathematical formulation (i.e., the second row), and the relative gap between these two values to demonstrate the improvements (i.e., the third row). If the gap is larger than 0%, which means the model finds better feasible solutions. Otherwise, there is no improvement (i.e., 0%). From the Table, I can observe that when the numbers of the vessels are 8 and 10, the solution quality has been improved for most instances by applying the mathematical formulation approaches. However, when the number of the vessels is increased to 12, only the solution quality for 2 cases has been improved. From the experiments, I can conclude that the two stage method is suitable for the case that the number of vessels is under 10 and average improvement is around 14.5%.

Table 18: Improvement on combination of the model and the heuristic approaches

No. of Cranes	No. of Vessel	V=8	V=10	V=12
K=4	Model	38	105	71
	Heuristic	54	115	71
	Improve %	29.6	11.8	0
K=4	Model	79	44	154
	Heuristic	81	57	154
	Improve %	2.5	22.8	0
K=4	Model	41	93	69
	Heuristic	51	121	69
	Improve %	19.6	23.1	0
K=6	Model	82	64	105
	Heuristic	86	65	105
	Improve %	4.7	1.5	0
K=6	Model	26	72	41
	Heuristic	35	73	67
	Improve %	25.7	1.4	24.4
K=6	Model	40	54	91
	Heuristic	53	58	99
	Improve %	24.5	6.9	8.0

4.5 Summaries

In this paper, I study algorithms to solve the integrated problem of combining berth allocation and quay crane scheduling problems together (BAQCP). I first establish a mathematical formulation to describe BAQCP. In the mathematical model, I provide decisions on when and where to allocate each vessel and movements of quay cranes. The approach provides one of the first mathematical formulations for the problem. The model performs well for small size problems.

For large size problems, I develop two types of fast efficient heuristic algorithms. The first type of heuristic is developed for the case that the preemption is allowed. For instance, it is allowed that cranes can move among sections from time to time and one section job can be shared by multiple cranes. The second type of heuristics are developed for the case that non-preemption is not allowed. For this case, the cranes are not allowed to move to other sections until they finish all the workload in their allocated sections. I study the worst case ratios for both types of heuristics. I prove that the worst case bounds of the preemption heuristic and the non-preemption are both tending to two when the number of vessels is increased to infinity. I also compare the two designed heuristics to the commonly used FIFO heuristic. The results show that in practice the designed heuristics have better performance than the commonly used FIFO policy.

In order to further improve the solution quality, I combine the modeling approach with the non-preemption heuristic. The results showed that the mathematical model can be considered as a post-optimization approach to provide better solutions based on the initial solution provided by the non-preemption heuristic.

In general, the research provided one of the first studies on developing exact solution models for the integrated berth allocation and quay crane scheduling problem, as well as the worst case ratio study for this type of problems. Although the study shows that the worst case bound of the preemption heuristic and non-preemption heuristic are both tending to two times of the lower bound, the numerical experiments indicate that all ratios between feasible solutions and the lower bounds are under two. In the future study, I will consider studying other solution approaches such as the Lagrangian Decomposition, Tabu Search and Genetic Algorithm to investigate BAQCP.

5 Conclusions and suggestions

5.1 Conclusions

The Phase I study focuses on the berth allocation and the security inspection for a container terminal system. Three berthing heuristics (SPT, Guan et al.'s and Li et al. 's algorithms) are applied to estimate the service rate of the inspection operation. The theoretical lower bounds are derived from these three heuristics. The experiments for the stochastic setting verify the analytic results and show that the combined optimization and simulation approaches work for the problem. Conclusions for the Phase I Study are summarized as follows:

1. Combine optimization and simulation approaches to establish a model for the berth allocation and the inspection systems. The processors of the berth system must serve a vessel consecutively. The consecutive characteristic of the processors becomes a bottleneck to implement a berth system into a simulation model. By the embedded simulation technique, a copy entity scheme is designed to conquer the modeling difficulty in the simulation model.

2. Modify the deterministic berthing algorithm used in this study for dealing with stochastic processing time scenarios with the aids of the simulation model.

3. Derive a theoretical lower bound of the inspection service rate related to the service rate of the processors by different berthing heuristics.

4. To allow more containers to be inspected, the inspection service rate has to increase. However, due to some limitations, such as budget and land usages, increasing the service rate of the inspection operation becomes a difficult scenario. Therefore, the sampling policy can be chosen as another alternative.

In the Phase II study, QCSP is analyzed by the exact approach and the heuristic approach. A mathematical formulation is established by using a network flow modeling technique, which makes the application of the model easier. This model can easily include or exclude the crane traveling factor. In order to solve larger size problems, Lagrangian relaxation approach is developed, which may result in an infeasible solution. The Lagrangian relaxation heuristic is proposed to make an invisible solution to a feasible one. Finally, two heuristics are developed for the large size problems. The first heuristic is based on the time threshold to find the consecutive sections as a working zone for a crane. The time threshold is determined by the lower bound from a dynamic programming algorithm. The second heuristic directly applies the dynamic programming algorithm to determine consecutive sections as a working zone for a crane. Conclusions for the Phase II Study are summarized as follows:

1. Establish a mathematical model by network flow modeling technique to describe QCSP with the consideration of the crane traveling factor.

2. Show the advantages of the proposed model as compared to other mathematical formulations.

3. Derive the worst case bound theoretically for the designed heuristics to ensure the solutions quality. The analysis shows that worst case bound is two.

4. Design different solution approaches for QCSP. The model is capable for the small size instances. The Lagrangian relaxation approach and the heuristic approach can deal with medium and large size instances. The numerical results conclude that the solution approaches that I propose are efficient and effective for QCSP.

The Phase III Study focuses on the study of the combined problem of berth allocation and quay crane scheduling. The heuristic approach is the main solution approach. First, a mathematical model is established. The numerical experiments show that the capabilities of the model are only for small size instances. Two heuristics for large size instances are developed. The first one is the preemption heuristic, and the second one is the non-preemption heuristic. The worst case analysis of these two heuristics is performed to show that the worst case bound of the preemption heuristic is less than three times the lower bound, and that of the non-preemption heuristic is less than four times the lower bound. However, as the number of vessels tends to infinity, the worst case bound of the two heuristics is tending to two. The numerical experiments verify the theoretical proofs. Further, the model and the non-preemption heuristic are combined to improve the solution quality successfully. Conclusions for the Phase III Study are summarized as follows:

1. Establish a mathematical model from the extension of the mathematical model of the Phase II study to describe BAQCP without considering the crane traveling factor.

2. Adapt the heuristic approach to develop two heuristics and derive the worst case bounds for those two heuristics.

3. Numerical experiments show the effectiveness and efficiency of the solution approaches, which are proper for the real world cases.

4. Combine the model and the non-preemption heuristic to improve the solution quality of the heuristic approach.

5.2 Future study

In the Phase I Study, a simulation frame is established to allow deterministic berthing heuristics applying for the stochastic processing time cases, which indicates that this approach can be used for the problem with uncertain data. Validation of this approach is suggested by the real data to make this approach more realistic. In addition, the designed deterministic heuristic for the combined problem of the berth allocation problem and quay crane scheduling problem needs to be verified and validated through the hybrid approach established by real data. All approaches developed in this dissertation may form a basis for a decision support system for the container terminal operations.

BAQCP is still a new research field. In this dissertation, the worst case bound analysis indicates that more room is needed to improve the solution quality of the proposed heuristics. Currently, no evidence shows that the worst case bound is tight.

Lastly, other solution approaches such as the Lagrangian Relaxation approach or other meta heuristics (Tabu Search, Genetic Algorithm, and so on) are worthy investigating for BAQCP in the future studies.

References

Ak, A., A. L. Erera. 2007. Simultaneous berth and quay crane scheduling for container ports. Working paper, H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.

Baker, B. S., D. J. Brown, H. P. Katseff. 1981. A 5/4 algorithm for two-dimensional packing. *Journal of Algorithms* **2** 348–368.

Coffman, E. G., M. R. Garey, D. S. Johnson, R. E. Tarjan. 1980. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing* **9** 808–826.

Cordeau, J., G. Laporte, P. Legato, L. Moccia. 2005. Models and tabu search heuristics for the berth-allocation problem. *Transportation Science* **39-4** 526–538.

Daganzo, C. F. 1989. The crane scheduling problem. *Transportation Research B* **23-3** 159–175.

Drozdowski, M. 1996. Scheduling multiprocessor tasks - an overview. *European Journal of Operational Research* **94** 215–230.

Gerencser, M., J. Weinberg, J. Vincent. 2003. Port security wargame implications for U.S. supply chain. Tech. rep., Booze Allen Hamilton.

Goodchild, A. V., C. F. Daganzo. 2006. Double-cycling strategies for container ships and their effect on ship loading and unloading operations. *Transportation Science* **40-4** 473–483.

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

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

Imai, A., H. C. Chen, E. Nishimura, S. Papadimitriou. 2008a. The simultaneous berth and quay cran allocation problem. *Transportation Research E* **44** 900–920.

Imai, A., E. Nishmura, S. Papadimitriou. 2008b. Berthing ships at a multi-user container terminal with a limited quay capacity. *Transportation Research E* **44** 136–151.

Imai, A., J. T. Zhang, E. Nishmura, S. Papadimitriou. 2007. The berth allocation problem with service time and delay time objectives. *Maritime Economics & Logistics* **9** 269–290.

Kalvelagen, E. 2002. Lagrangian relaxation with GAMS. Tech. rep., GAMS Development Corp.

Kim, K. H., K. C. Moon. 2003. Berth scheduling by simulated annealing. *Transportation Research B* **37** 541–560.

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

Lee, D. H., H. Q. Wang, L. Miao. 2008a. Quay crane scheduling with handling priority in port container terminals. *Engineering Optimization* **40-2** 179–189.

Lee, D. H., H. Q. Wang, L. Miao. 2008b. Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research E* **44** 124–135.

Lee, Y., C. Y. Chen. 2008. An optimization heuristic for the berth scheduling problem. In press, European Journal of Operational Research.

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

Li, C. L., G. L. Vairaktarakis. 2004. Loading and unloading operations in container terminals. *IIE Transactions* **36** 287–297.

Lim, A., B. Rodrigues, F. Xiao, Y. Zhu. 2002. Crane scheduling using tabu search. *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*.

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

Lim, A., B. Rodrigues, Z. Xu. 2007. A m-parallel crane scheduling problem with a non-crossing constraint. *Naval Research Logistics* **54** 115–127.

Liu, J., Y. Wan, L. Wang. 2005. Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. *Naval Research Logistics* **53** 60–74.

Mak, K. L. 2006. Quay crane scheduling in container terminals. *Engineering Optimization* **38-6** 723–737.

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

Monaco, M. F., M. Sammarra. 2007. The berth allocation problem: A strong formulation solved by a lagrangean approach. *Transportation Science* **41-2** 265–280.

Nishimura, E., A. Imai, S. Papadimitriou. 2001. Berth allocation planning in the public berth system by genetic algorithm. *European Journal of Operational Research* **131** 282–292.

Park, Y. M., K. H. Kim. 2003. A scheduling method for berth and quay cranes. *OR Spectrum* **25** 1–23.

Peterkofsky, R. I., C. F. Daganzo. 1990. A branch and bound solution method for the crane scheduling problem. *Transportation Research B* **24-3** 159–172.

Roach, A. 2003. Container and port security: a bilateral perspective. *The International Journal of Marine and Coastal Law* **18-3** 341–361.

Sammarra, M., J. F. Cordeau, G. Laporte, M. F. Monaco. 2007. A tabu search heuristic for the quay crane scheduling problem. *Journal of Scheduling* **10** 327–336.

Sekine, S., E. Campos-Nanez, J. R. Harrald, H. Abeledo. 2006. A simulation-based approach to trade-off analysis of port security. *Proceedings of the 2006 Winter Simulation Conference* 521–528.

Simchi-Levi, D., P. Laminsky, E. Simchi-Levi. 2002. *Designing and managing the supply chain*. 2nd ed. McGraw-Hill/Irwin.

Stahlbock, R., S. Voβ. 2008. Operations research at container terminals: a literature update. *OR Spectrum* **30** 1–52.

Steinberg, A. 1997. A strip-packing algorithm with absolute performance bound 2. *SIAM Journal on Computing* **9** 401–409.

Teo, C. P., J. Dai. 2003. Berth planing project. `http://www.tliap.nus.edu.sg/tliap/Project_BerthManagement/default.aspx`.

Thibault, M., M. R. Brooks, K. J. Button. 2006. The response of the `U.S.` maritime industry to the new container security initiatives. *Transportation Journal* **45** 5–15.

UN. 2005. *Regional shipping and port development strategies (Container Traffic Forecast)*. United Nations.

Willis, H. H., D. S. Ortiz. 2004. Evaluating the security of the global containerized supply chain. Tech. rep., RAND.