Berth Allocation Problem in Dry Bulk Terminals

by

Gita Taherkhani

A Thesis Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

Koç University

July, 2015

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a M.S. thesis by

Gita Taherkhani

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

____

Prof. Dr. Ceyda Oğuz (Advisor)

____

Assoc. Prof. Dr. Emre Alper Yildirim

____

Prof. Dr. Necati Aras

Date: ______________________

Dedicated to my parents

ABSTRACT

One aspect of better managing the port terminals is to efficiently allocate the vessels to the berth locations which is referred to as Berth Allocation Problem (BAP). In BAPs, one of the basic criteria is to decide what sequence of arriving vessels minimizes the time spent in the terminal.

This study focuses on the berth allocation problem in dry bulk terminal, and proposes a mixed integer linear programming model focusing on the partitioned BAP, i.e. the total length of the quay is partitioned into several sections and to each section only one vessel can be allocated at a specific time. We consider the allocation of vessels to a location on a berth as well as the sequence, in which the vessels should be handled in order to minimize the sum of arriving vessels' completion times. In addition, the effects of the tidal condition that happens periodically in the time horizon are considered. At low tide, available depth of water is not adequate for the movement of vessels. The vessel assigned to the berth location is therefore able to depart the terminal only in the high tide periods. We also introduce three additional sets of constraints and add them to the model to make it computationally more tractable. Moreover, we develop two simple heuristic algorithms that enable us to obtain near optimal solutions to the problem within a short computational time.

To better understand the performance of the proposed model and the heuristic algorithms, we test them on instances generated based on the real data of a dry bulk terminal. The computational results show that adding the additional constraints to the model improves its performance both in terms of the running time and the optimality gap. The results also show that our second heuristic algorithm outperforms the first one in terms of the quality of final solutions. Furthermore, we compare the solutions obtained by the proposed model with the solutions obtained by a continuous model

proposed in the literature. According to the results, we conclude that both models are strong and effective, and their applicability depends on the decision plans of the port management. To reflect the importance of some parameters in the proposed model and the heuristic algorithms, we apply sensitivity analysis to determine how different values of some parameters affect the performance of the proposed model as well as the heuristic algorithms.

ÖZETÇE

Yük terminallerinin daha iyi yönetilmesini sağlamanın bir yolu gemileri rıhtımlara verimli bir şekilde atamaktır ve bu problem rıhtım atama problemi (RAP) olarak adlandırılır. RAP'ın temel prensiplerinden biri gemilerin limanda geçirdiği zamanı enazlayacak şekilde rıhtıma yanaşma sırasını belirlemektir.

Bu çalışmada kuru yük dökme terminallerinde rıhtım atama problemini ele almaktayız ve bölümlere ayrılmış (partitioned) RAP'a; şöyle ki, rıhtımın toplam uzunluğu birkaç kısma ayrılmıştır, odaklanan bir karma tam sayılı doğrusal programlama modeli sunulmaktadır. Limana yanaşan gemilerin elleçleme sürelerini enazlamak üzere gemilerin rıhtımlara atanma konumlarının yanı sıra gemilerin rıhtımlarda elleçlenme sıralamasını da ele almaktayız. Her bir t anında bir kısma yalnızca bir gemi atanabilmektedir. Buna ek olarak, periyodik olarak gerçekleşen gel-git koşullarının atama problemine etkisi de göz önünde bulundurulmuştur. Suların alçak olduğu periyotta, gemilerin hareketi için yeterli su derinliği bulunmamaktadır. Buna bağlı olarak, rıhtıma atanan gemiler ancak suların yükseldiği periyotlarda limandan ayrılabilirler. Ek olarak üç kısıt seti sunduk ve bu kısıtları ekleyerek modelin izlenebilir bir sürede çözüme ulaşmasını sağladık. Bunun yanı sıra, eniyi çözüme kısa bir süre içinde yakınsayan sonuçlar elde eden iki basit sezgisel algoritma geliştirdik.

Performanslarını daha iyi ölçebilmek adına önerilen modelleri bir kuru dökme yük terminalinden elde edilen veriler ile oluşturulan örneklerle test ettik. Sayısal sonuçların ışığında ek kısıtların model performansını çözüm süresi ve çözüm kalitesi açısından iyileştirdiği görülmüştür. Ayrıca sonuçlar ikinci sezgisel algoritmamızın birinciyi çözüm kalitesi bakımından geçtiğini göstermektedir. Bunlara ek olarak, sunduğumuz model çözümlerini literatürde var olan sürekli RAP modeli ile karşılaştırdık. Sonuçlara göre, her iki modelin de güçlü ve etkili olduğunu, uygulanabilirliğinin liman yönetiminin

kararlarına bağlı olduğunu söyleyebiliriz. Sunulan modeldeki ve sezgisel algoritmalardaki değişkenlerin önemini yansıtması açısından ve farklı değerler alan değişkenlerin modelin ve sezgisel algoritmanın performansları üzerindeki etkisini gözlemlemek için duyarlılık analizi gerçekleştirdik.

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to Prof. Ceyda Oğuz. I am thankful to her for providing me the chance to be a member of the Optimization Lab at Koç University. She has not only provided me her professional expertise, valuable guidance and the financial support I needed to complete my M.S. studies but also been a real friend and a true mentor. I feel extremely fortunate to work with her. I learned a lot from her, and I am very grateful to have her as my advisor.

I am also very grateful to Prof. Emre Alper Yildirim. He was always enthusiastic to share our problems and to find a better attitude. I certainly believe that his ideas upgraded my thesis. I am indebted to other members of my dissertation committee: Prof. Necati Aras for willingly accepting to be a member of my committee and to read and review this thesis.

I recognize that the whole research through my master studies would not have been possible without the financial support of TÜBİTAK.

I am grateful to my parents and my whole family for all the love and support I received over the years.

Finally, I would like to express my gratitude to my whole friends who have always been there for me, and to all the current and previous members of the Lab ENG216.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

DBT	:	Dry Bulk Terminal
BAP	:	Berth Allocation Problems
BAPDBT	:	Berth Allocation Problem in Dry Bulk Terminal
B & B	:	Branch-and-Bound
GA 1	:	Greedy Algorithm 1
GA 2	:	Greedy Algorithm 2
MIP	:	Mixed Integer Programming
MILP	:	Mixed Integer Linear Programming
PM	:	Partitioned Model
VI	:	Valid Inequality
AC	:	Additional Constraints
QCSP	:	Quay Crane Scheduling Problem
SA	:	Simulated Annealing-based Algorithm
TTW	:	Tidal Time Windows
LB	:	Lower Bound
OPT	:	Optimal (Final) Solution
UB	:	Upper Bound
CPU	:	Central Processing Unit

Chapter 1

INTRODUCTION

The last decade of the 20th century witnessed significant growth in worldwide maritime transportation. This steady increase in international maritime traffic has led to changes in the business of ocean transports and terminal operators. International shipping lines connect countries, markets, businesses and people by allowing them to buy and sell goods in large scale. The maritime transportation industry has developed in recent decades, and now is one of the efficient modes of transporting goods. Because of that, carriers face with higher and higher shipping demands. In order to satisfy high shipping demands, larger vessels have been built. A single large transport vessel may carry a large number of goods. It may require hundreds of freight aircraft, miles of rail cars, and fleets of trucks to carry the same amount of goods that can fit with the capacity of one large vessel. So, it is one of the cheapest ways of transporting large amounts of goods compared to other transportation methods. Thus, by utilizing maritime transportation, consumer costs are kept down and industrial efficiency is improved. However, beside costs, available services to traders and vessels as well as service quality, concerning speed, reliability, safety and security are of increasing significance in the context of globalized transportation processes.

In maritime transportation, ports are zones between two geographical interfaces - the quayside and landside. The quayside is introduced as a place for loading and unloading of vessels. The landside is related to the port's region and locality. It is available to support maritime access, e.g. the place where cargoes are accommodated. The design and the facilities used in the quayside and the landside depending on the type of port terminals are different from each other. The terminals can be classified into two groups: container terminals, in which all cargo is packed into containers, and bulk port terminals, which contains bulk cargoes, mostly in loose forms.

The remainder of the chapter is organized as follows: Container terminal, dry bulk terminal and their differences are discussed in Sections 1.1, 1.2 and 1.3, respectively. In Section 1.4, berth allocation problems are discussed in detail. Finally, the outline of the thesis is given in Section 1.5.

1.1 Container Terminal

A container terminal is the zone of the port where vessels dock on a berth and containers are loaded, unloaded and stored in a buffer area called yard. The terminal can be divided into three areas: the quayside, the yard and the gate. Figure 1.1, taken from Buhrkal et al. [1], illustrates a container terminal, where we can identify the quayside in the upper part of the picture, the yard in the middle part and the gate in the bottom part. The quay is an interface of landing place into the water to facilitate the loading and unloading of cargo. The locations where mooring can take place are called berths. In the berth, there are cranes to pick up the containers carried by vessels. In the land, there are some trucks and trains to move the container from stock yards to quay part. In the container terminal, a yard connects the landside to the quayside, and provides space for shipment and storage. Mainly, container terminal operations can be grouped into four major classes that are associated with specific processes and stages in the material flow:

- **Berth Allocation Problems**: The decisions on assigning vessels to the berth locations are associated with the vessel arrival. When a vessel arrives in a seaport, first, it has to be moored in the quayside for loading and unloading over a time period. For this purpose, a number of berths are available at port terminals. A berth is a quay location, equipped with one or more quay cranes. It is usual to say that a berth may accommodate one vessel at a time.

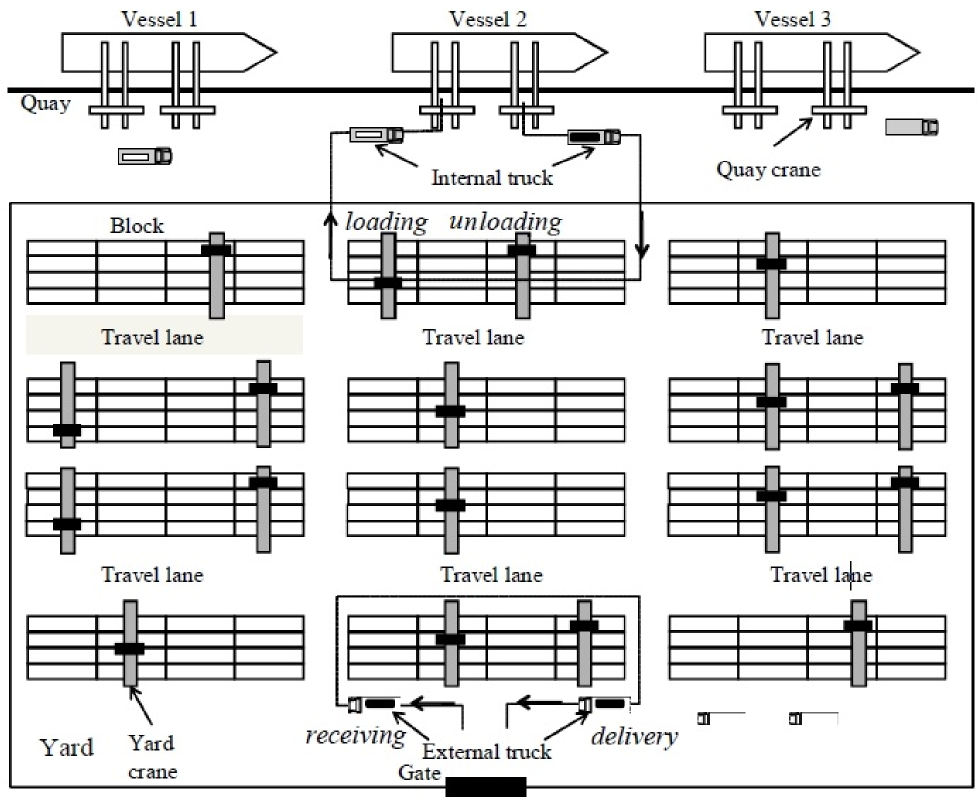

Figure 1.1: A typical container terminal layout (Buhrkal et al. [1])

The vessels arriving to the berth locations have different length and width, so the number and the length of berths at a port terminal are the most important strategic decisions that must be made at the strategic level. A schematic representation of a vessel berthing is provided in Figure 1.2. Vessel is scheduled over time (x-axis) according to its expected handling time and assigned to the berthing position on a quay (y-axis) according to its length.

- **Quay crane allocation and scheduling**: These decisions are associated with the loading and the unloading operations to and from the vessels. The objective of the quay crane allocation problem is to assign quay cranes to the vessels that must be operated over the time horizon efficiently.

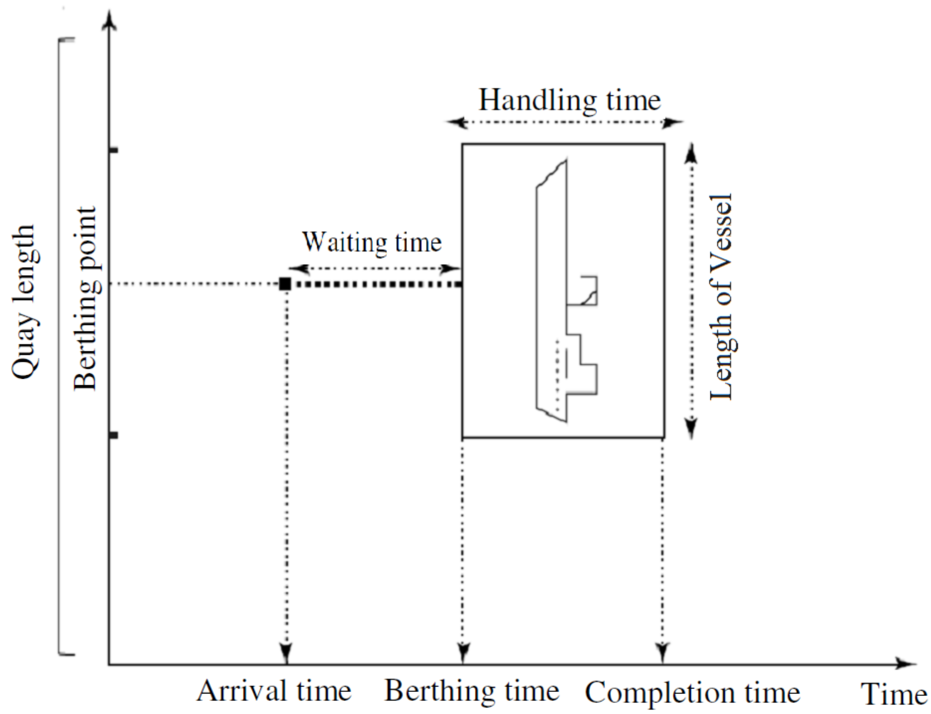

Figure 1.2: A schematic representation of the allocation of vessel in the berth-time space

- **Transfer operations**: Decisions related to transfer operations are usually made inside the container terminal. Transferring is carried out by internal trucks or automated vehicles. Objective functions are to minimize the total distance traveled to complete the tasks or to minimize the total operations delay.

- **Yard operations decisions**: These decisions are associated with the stock and storage part. The management of yard operations involves several decision problems. The yard allocation problem refers to the design of storage policies according to the containers.

1.2 *Dry Bulk Terminal*

Dry Bulk Terminals (DBT) are used all around the world to handle large quantities of bulk materials. Due to the high demand for energy and mineral resources, many DBTs are expanding and increasing their capacity. DBTs are used worldwide as a buffer between either international or intercontinental transportation and inland or domestic transportation or other ways. DBTs essentially operate with dry bulk materials that are shipped in large and unpackaged amounts. This kind of materials are usually classified into two categories; major bulks and minor bulks. Some examples of major dry bulk materials include coal, ore and iron. Minor bulks include grain, steel, plastic and cements.

There are two types of dry bulk terminals: export or import terminals. Unlike other types of terminals (e.g. container terminals, general cargo terminals), for DBTs it is important to distinguish if they are export or import terminals. In exporting terminal the main decision is to load the incoming vessels, while in importing terminals, the objective is to unload the loaded vessels. Because of the differences in objectives, the design of an export bulk terminal is different from an import bulk terminal. Export terminals are often located closer to the sources of bulk materials to make the shipment of bulk material easier. Import terminals need to fit services in both the waterside and landside. Unlike export terminals, import terminals usually handle multiple types of bulk materials. So the complexity of waterside and landside services will be higher in import terminals. Figure 1.3 schematically shows the layout of a typical dry bulk terminal. This figure shows dry bulk materials that are being exported from the terminal of the port. Vessels are loaded using either conveyor or pipelines. Silos or stockpiles for the embedding of bulk cargo are situated alongside the quay.

The terminal should have a balanced system to allow continuous flow with some spare capacity. Inflow and outflow rates define the need for storage. To construct a balanced system for dry bulk terminals, it is of great significance to know their characteristics. In the next section, we discuss the characteristics of dry bulk terminals in more details.

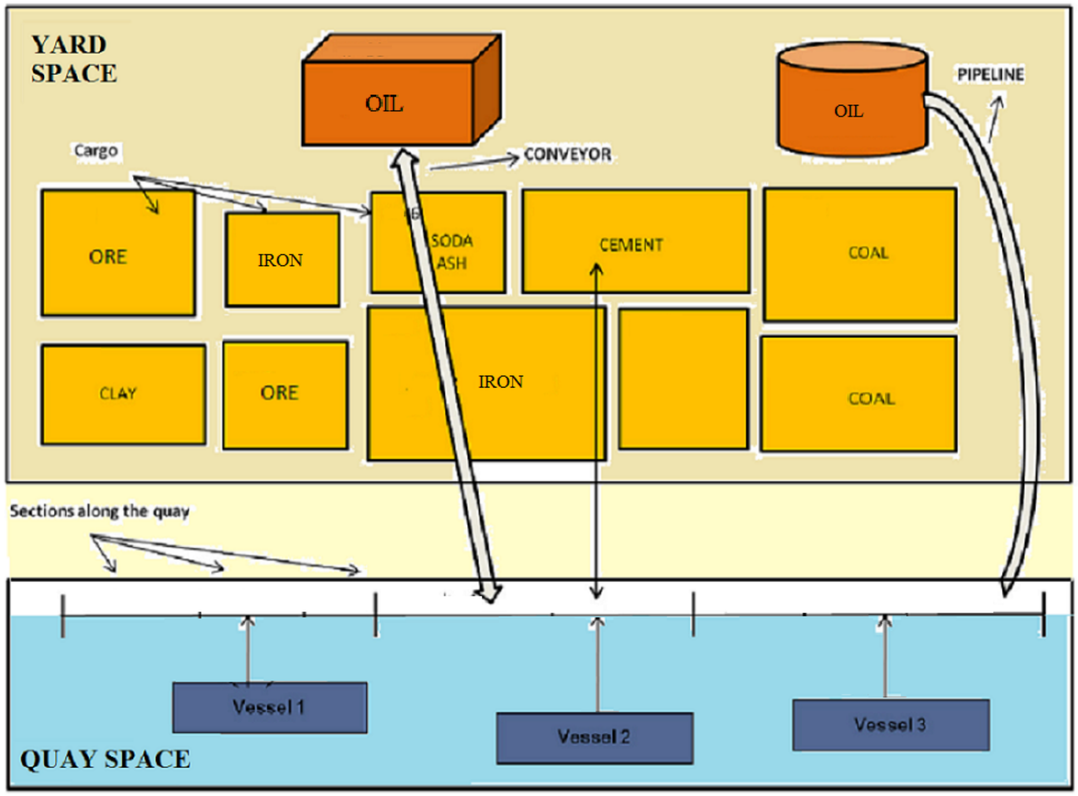

Figure 1.3: A typical configuration for a dry bulk terminal

1.2.1 Characteristics of Dry Bulk Terminals

The dry bulk terminals can be analyzed in three parts to define their characteristics: seaside, stockyard and land side. The main focus of the dry bulk terminals is on the decision problems that usually originate between the seaside and the landside.

- **Seaside**: In dry bulk terminals, seaside is referred to the location that connects the sea part to the land part. Quay has a significant role in this connection. The quayside is introduced as a place for loading and unloading of vessels. When vessels arrive to the port, they enter in the harbor and wait for mooring at the quay. The locations, where mooring can take place, are called berths. It is usual to say that a berth may accommodate one vessel at a time. Since, vessels have direct connection with berth locations, the vessels' loader or unloader, pipelines and conveyors are located in berths. So, the main decisions at seaside focus on the allocation of the vessels to the berth locations. Consequently, design of a seaside in dry bulk terminal incorporates the quay length, the selection of the number and capacity of vessels' loading or unloading machines and connecting quay conveyors. Figure 1.4 shows bulk vessels that are loaded in the berth location.

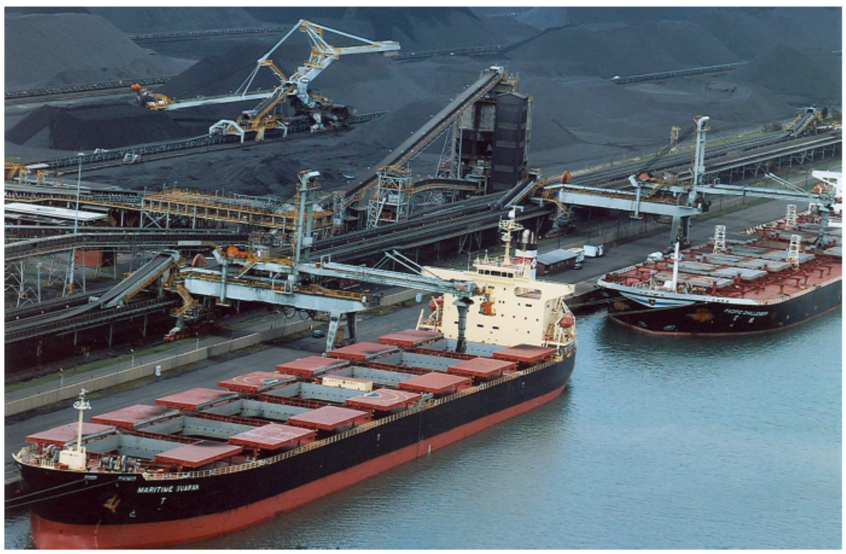

Figure 1.4: Bulk vessels are loaded

Berthing of vessels with significant drafts can be limited by the water depth alongside the quay. One specific characteristic of dry bulk terminals is the existence of tides. Tide means the change of water levels in seaside. The tides in the port area generally happen daily, that is the occurrence of one high and one low water depth every day. The tidal effect, which is quite typical, causes transportation companies to face with big challenges in berth planning. For instance, in export terminals the decision of when the loaded vessels with heavy weight depart the terminal, considering water depth has significant importance. At low tide, available depth is not adequate for the movement of vessels. On the other hand, in import terminals under tidal condition, even when a berth position is available, vessels may need to wait for mooring. Hence, the water depth should be considered in the design of seaside as well.

- **Stockyard**: The dry bulk supply chains typically include a number of transportation processes which are separated by buffer storage facilities located at dry bulk terminals in ports. These buffers are essential for absorbing unavoidable differences between incoming and outgoing flows of bulk materials (Lodewijks et al. [2]). Due to the large volumes of coal, iron and ore and the possibility to store these dry bulk materials in open air, stockyards are generally used. In other words, bulk materials are stacked onto and reclaimed from piles at the stockyard. Piles are sprayed with mixtures of water and wax-containing substances to accelerate crust formation on stockpile surfaces and to avoid wind erosion. Common machines installed at stockyards are dual-purpose stacker-reclaimers or single-purpose stackers and reclaimers. Stacker-reclaimers combine the two functions of stacking and reclaiming into a single unit. Stockyard sizing is crucial during the design of dry bulk terminals. An undersized stockyard results in excessive vessel waiting times and forces terminal operators to pay penalty costs. An oversized stockyard prevents the recovery of the huge investment costs. Figure 1.5 shows an example of a stockyard where dry bulk materials are stored in separated piles on several stockyard lanes.

- **Landside**: In dry bulk terminals, landside is the place where bulk materials are delivered or exported. Figure 1.6 shows a schematic representation of landside connections. Based on the type of bulk port terminals - exporting or importing - the connections are different. For example, for the transport of bulk materials from mines to export terminals, there are a number of empty locomotives and rail-cars, which are in line at rail yards in ports. Rail-cars pass and arrive at specified mines, load the material, return to the port and unload at the export terminal, where the material is stacked in stockpiles. The shipment can start after all required materials are stacked. Whereas for

Figure 1.5: A stockyard with lanes and separated piles

import terminals, orders are received from industrial clients to deliver material at a predefined time to their facilities. In fact, the train's journey time is defined and determined in consultation with terminal operators on the time when rail-cars must be loaded. Just before the loading time, the empty rail-cars are railed from the yard to the terminal. After loading, the train is railed to the industrial client, unloaded and returned to the yard.

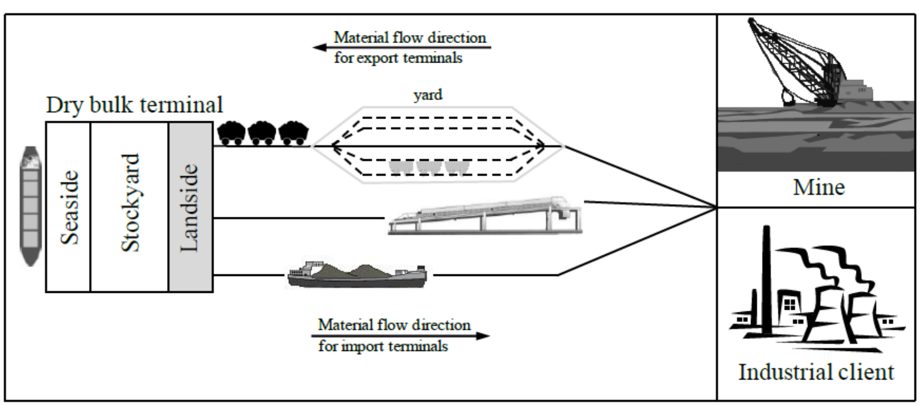

Figure 1.6: Transport between the terminal's landside, mines and industrial clients

In general, The landside design is limited by the determination of the number and the capacity of the loading and the unloading machines as well as their locations at the terminal.

1.3 Dry Bulk Terminals vs Container Terminals

To compare the container and the dry bulk terminals briefly, packed cargoes along with shipping containers are handled in container terminals and subsequently specialized equipments, e.g. cranes and trucks, are used to load, to unload and to store cargoes in the terminal. On the other hand, dry bulk terminals handle products like coal, iron, ore, grain and cement in the unpacked and bulk form. Dry bulk terminals have specialized equipments, e.g. stackers and reclaimers, for loading, unloading and shipping of bulk materials. Most bulk terminals have a singular direction of the cargo movement - they are either import or export. This is almost true for the "big four" bulk cargoes - coal, iron, ore and grain. Besides, one of the common features of the dry bulk terminals is the significance of tide in the seaside of the terminal that is uncommon in container terminals and considerably changes the structures of terminals. Despite the similarities between dry bulk terminal and container terminals, some fundamental differences implicate them to be designed and considered separately. So, it is rarely seen one terminal serving both types of cargo, for container terminal and dry bulk terminal.

To meet the growing global demand for energy and steel, the seaborne trade flows for coal, iron and ore will have to increase. Despite the expected increase of the maritime transportation for bulk commodities, the main focus in the field of port logistics is on container terminals and bulk port terminals receive less attention than they deserve. In this research, we address an exporting bulk terminal, which is covered by major dry bulk materials. As mentioned, to realize the realistic design of the dry bulk terminals, specifically the seaside, assignment of vessels to the berth location is of strategic importance. Therefore, in this study, we focus on the berth allocation problem in dry bulk terminal which is one of the main decisions at the seaside of the dry bulk terminals. In the next part, we explain berth allocation problem as well as its characteristics.

1.4 Berth Allocation Problems

The scope of this study is restricted to the Berth Allocation Problem (BAP) at the dry bulk terminals. The berth allocation problem is the problem of assigning vessels to positions on the quayside in a terminal such that the performance of a port terminal system is optimized. Managing the allocation of arriving vessels to the berth location is important in terminals, since bad decisions related to allocating vessels to the berth location can cause unnecessary waiting times in vessels' processing. The berth is a bottleneck resource for determining the overall capacity of the port terminal because the cost of constructing a berth is very high compared to other terminal facilities. As the operations of berth directly affect the overall operations of the terminal, the operational level decision of allocating vessels to the berth space is critical. BAPs are typically divided into three main versions: in continuous locations, in discrete locations and in partitioned locations which are described in the following subsections.

1.4.1 Continuous BAPs

In continuous BAPs, the quay is considered as a continuous line and each vessel can be moored at any place along the quay with respect to the non-overlapping constraint. Non-overlapping constraint ensures that there is no intersection between the location space of any two vessels allocated simultaneously.

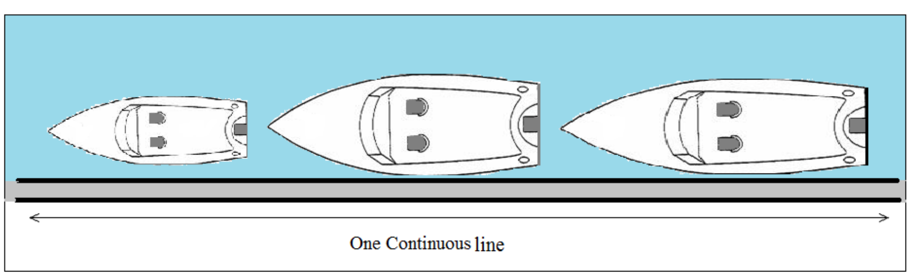

Figure 1.7: A schematic representation of continuous BAP

Figure 1.7 demonstrates a simple example of the continuous BAP where vessels are allocated along the quay. In this case the only thing that may restrict the assignment of vessels is the non-overlapping constraint.

1.4.2 Discrete BAPs

The berth allocation problem in discrete location considers the quay as a finite set of berths. In discrete case, it is assumed that a quay, which is represented as a continuous line, is discretized into several berths with same and fixed lengths, and only one vessel can be allocated to a berth at a specific time. Figure 1.8 provides a simple representation of the BAP with discrete structure. Vessels arrive to the terminal for mooring. There are three discrete berths located at the terminal. Only one vessel can be moored to each berth at a time even if the berth appears to have enough space for more than one vessel. For example, even if there exist more than one vessels that are short enough to moor simultaneously at berth, due to the discrete BAP structure, they must be assigned to another berth location.

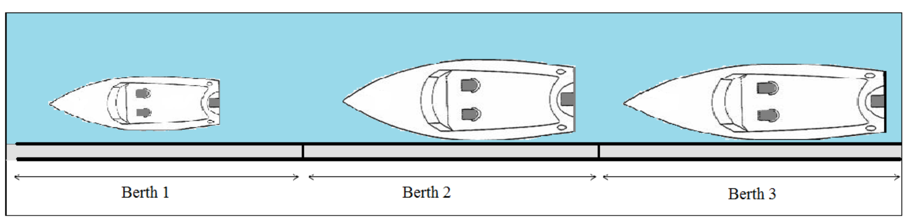

Figure 1.8: A schematic representation of discrete BAP

1.4.3 Partitioned BAPs

In partitioned BAP, the quay, which is represented as a continuous line, is partitioned into several sections and at each section only one vessel can be allocated at a specific time. Figure 1.9 provides a simple representation of the BAP with partitioned structure. Vessels arrive to the terminal for mooring, simultaneously as usual. There are separated sections such as Section 1, Section 2 and Section 3 located at the terminal as depicted in Figure 1.9. Only one vessel can be moored at one section at any time.

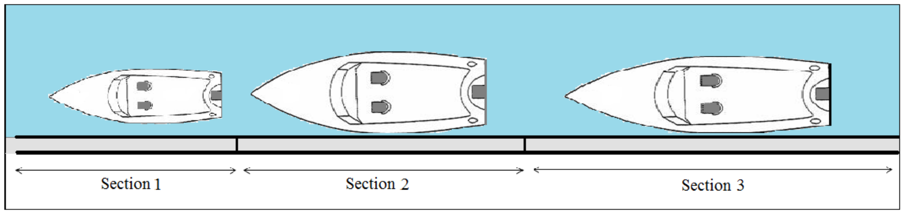

Figure 1.9: A schematic representation of partitioned BAP

One of the particular layout of partitioned BAP is called "hybrid" BAP. In hybrid layout, the quay is partitioned into several sections like partitioned BAP, but a large vessel can be moored in more than one section, and small vessels can share one section at a time. It is also worth mentioning that discrete BAP is a special case of partitioned BAP in which the quay is segmented into several unit-length sections. Since the partitioned BAP discretizes the quay into several sections with different lengths instead of several berths of unit length, it brings more flexibility to the model in comparison with discrete BAP. In other words, the partitioned BAP is a generalization of the discrete BAP and subsequently provides better quay utilization.

As mentioned above, in partitioned layout, there is a finite set of sections where each section can accommodate only one vessel at a time. With these layouts, the BAP is treated as a parallel machine scheduling problem, where a vessel is treated as a job and a section as a machine. On the other hand, continuous BAP allows the vessels to place anywhere along the quay. To compare these two layouts, continuous BAP provides more freedom in the decision of assigning vessels to berthing positions. However, solving continuous BAP is typically harder than partitioned one.

1.5 Outline of the Thesis

This study concentrates on the partitioned version of exporting BAPs in dry bulk terminals. As mentioned before, in partitioned version, we separate the total length of the quay into several sections and only one vessel can be allocated to each section at a specific time. The manner that sections are defined along the quay is critical. So, the partitioning should be applied in a rational way. For example, the interval of the lower bound and the upper bound of sections' length should be respectively between the smallest length of vessel and the largest one.

In this study, we also consider the tidal condition as commonly occur in dry bulk terminals. Figure 1.10 represents a schematic example of partitioned BAP in an exporting dry bulk terminal. Vessels are scheduled over time (horizontal axis) according to their handling times and assigned to the sections (vertical axis) according to their lengths. The blue vertical bars show high tide periods that happen periodically in time horizon. Boxes represent the exact location as well as handling time of each vessel in the allocation. The right hand side of boxes should place in high tide bars which means that the vessels departure time coincide with the high tide periods.

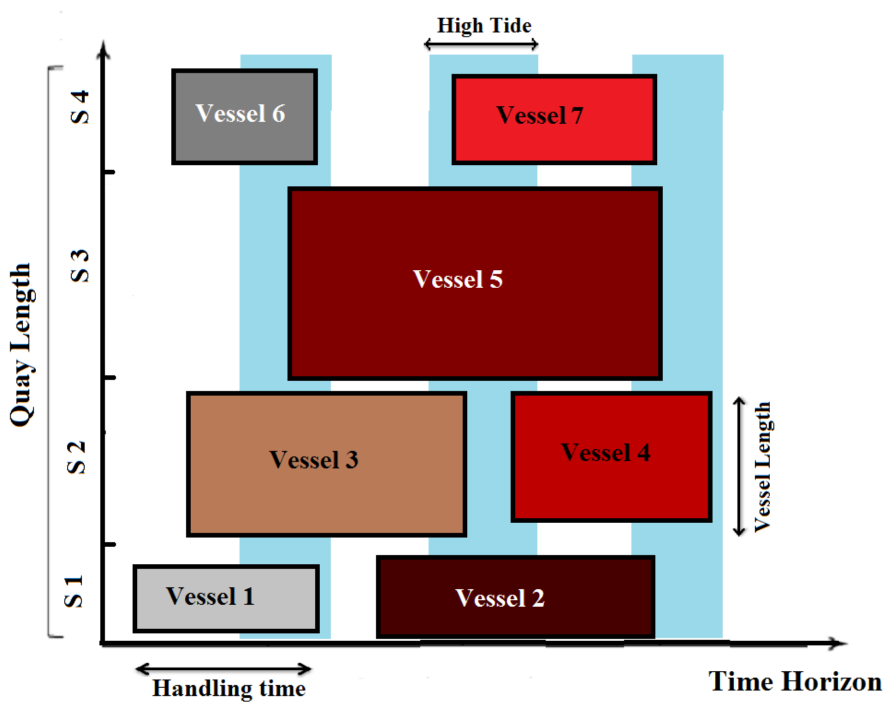

Figure 1.10: Assignment of vessels in a partitioned berth location with tidal conditions

We model our berth allocation problem mathematically with respect to all assumptions made related to the practical berth allocation situations in dry bulk terminals. The aim of our problem is to minimize the total completion time of vessels. A brief outline of this study is as follows:

- In Chapter 2, we present a survey of relevant literature that review various BAPs. In this review, all previous studies are classified into four main sections: 1. Discrete BAP, 2. Continuous BAP, 3. Hybrid BAP, and 4. Integrated BAP.

- In Chapter 3, first we define the problem formally. Next, we model the berth allocation problem mathematically under the partitioned berth structures. The model consists of various constraints with respect to all assumptions made related to the berth allocation situations in dry bulk terminals (i.e. existence of tidal periods in dry bulk terminals). The objective is to minimize the total completion time of vessels. Afterwards, we add three additional constraints to the mathematical model to obtain tighter upper bounds. In the rest of the chapter, two simple heuristic algorithms are used (greedy algorithm 1 and greedy algorithm 2) to obtain a feasible solution as an upper bound for the problem.

- In Chapter 4, comprehensive computational results are presented to determine the performance of the proposed models. The instances used in the computational experiments are generated based on the real data of a dry bulk terminal in Newcastle, Australia. We present the computational results of proposed model without and with additional constraints. Then, the comparison results of proposed model with the continuous model are given. Two heuristic algorithms are compared with each other, as well. Finally, the effects of some parameters on the performance of proposed model and heuristic algorithm are investigated.

- In Chapter 5, a conclusion for the entire thesis is provided to present the contributions, limitations and the future directions of this research.

Chapter 2

LITERATURE REVIEW

In this chapter we review the literature on berth allocation problem (BAP). Several berth allocation problems are proposed in the literature to cover different statements appear in ports. Most of the papers about berth allocation problems focus on the problem of allocating vessels to the quay location, where the objective is to obtain a non-overlapping berth plan for the vessels arriving to berth location.

Despite significant contributions on container terminals, relatively little effort has been devoted to dry bulk terminals. In spite of some similarities between dry bulk terminals and container terminals, which may enable us to apply possible analogies to dry bulk terminals by considering the container terminals, in general, there are some fundamental differences, e.g. existence of tidal constraint in dry bulk terminals which necessitate them to be considered separately.

To make a better comparison, we classify the related BAP studies according to the berth layout and investigate them one by one. On this basis, berth allocation studies can be classified as discrete BAP, continuous BAP, hybrid BAP and integrated BAP. In discrete BAP, the quayside is divided into sections of equal size. Each section is called a berth and at most one vessel can be assigned to a berth location at a time. In continuous BAP, the quay itself is considered as a single berth and hence it is possible to allocate more than one vessel simultaneously. In the hybrid case, the quay is partitioned into several sections, but a large vessel can be moored in more than one section at a time. The integrated problem consists of two tactical decision problems of berth allocation and yard assignment which are strongly relevant to each other in real world. The yard assignment problem refers to decisions that concern the storage location of materials. The integration differs for various types of decisions related to yard design: BAP integrated with Quay Crane Scheduling Problem (QCSP) and BAP integrated with yard assignment problem.

2.1 Discrete BAP

One of the early works that appeared in the literature is by Lai and Shih [3]. The authors propose a heuristic algorithm considering first-come-first-served rule for berth allocation problems. Their problem and algorithm are considered in discrete space. They use simulation models to evaluate different berthing policies. Through a simulation experiment three berth-allocation rules for container vessels are compared. The numerical results indicate that different policies may be used for different vessel arrival patterns.

Imai et al. [4] develop the discrete location version of the berth allocation problem in container terminals. They conclude that in order to achieve high port productivity, an optimal set of vessel assignments should be found without considering the first-come-first-served rule. However, this may result in some vessels being dissatisfied with the order of service. In order to deal with the two criteria to evaluate, i.e. berth performance and dissatisfaction on order of service, they present an heuristic algorithm to find a set of non-inferior solutions while maximizing the former and minimizing the latter. The algorithm is demonstrated with some sample problems and the results indicate the importance of the problem in efficient terminal utilization.

Imai et al. ([5], [6]) extend the static version of the BAP to a dynamic version. In their study dynamic treatment is similar to the static treatment, but with the difference that some vessels arrive while work is in progress. They present a Mixed Integer Programming (MIP) model for their problem. Due to the difficulty in finding an exact solution, they also develop a heuristic by using a subgradient method with Lagrangian relaxation. Their computational experiments indicate that the proposed algorithm is applicable to the container terminals.

Imai et al. [7] extend the BAP of Imai et al. [5] to treat the vessels with different handling time. They assume that a vessels' handling time depends on its quay location, that means different locations have a specific speed rate for loading and unloading of vessels. They modify the formulation with the objective of minimizing the total handling time of vessels.

Monaco and Sammarra [8] give a more compact formulation in comparison with Imai et al. [5]. They study the discrete dynamic berth allocation problem. In their paper the term of dynamic means vessels can arrive to the berth location after handling the assignment in planning time. They analyze the model described in Imai et al. [5] to derive a new mathematical formulation which uses fewer variables as well as constraints and is stronger than the previous one. They also present a heuristic algorithm for solving the new model based on a Lagrangian relaxation. The results of the computational experiments show the efficiency of their algorithm.

Hansen et al. [9] present a minimum cost berth allocation problem, which is an extension of Imai et al.'s [7] model. Extensions are as follows: Instead of considering just handling times, they consider for both each vessel and berth a handling time and a handling cost, respectively. In addition to an arrival time or release time, a due time or due date is specified for each vessel.

Cordeau et al. [10] present a tabu search algorithm for solving the berth allocation problem. The study is based on data from a container terminal in Gioia Tauro. They introduce a new model for the discrete version of the berth allocation problems. Their tabu search algorithm is based on their proposed BAP model and can only solve small- and medium-size instances, optimally. They extend the algorithm to the continuous case to solve the large-size problems. It should be noted that, their mathematical model is in discrete location; however, they develop tabu search heuristics for both discrete and continuous cases.

One of the main assumptions made in the berth allocation problems relates to tidal constraints. The tidal constraint as the restriction that certifies the water depth of the berth must be deep enough to situate the vessel, is added to the model assumption. Xu et al. [11] exploit the assignment of vessels to the berths limited by the physical condition of water depth in discrete environment. They model their problem as a parallel-machine scheduling problem, where the time horizon is divided into two periods: a low-water period and a high-water period. They consider both the static and dynamic cases of the problem, and present different solution methods for them.

Lorenzoni et al. [12] discuss a problem of arriving vessel within agreed time limits at a port under the condition of the first-come-first-served order. The authors include the tidal conditions in the port, which can restrict the port entrance to vessels at certain time intervals. They illustrate the implementation by computational tests with data generated base on the characteristics of a real port environment.

More recently, Barros et al. [13] consider the berth allocation problem for the positions of vessels in tidal bulk port terminals. In their study, available depth at low tide is not adequate for the movement of vessels, so draft conditions depend on high tide periods. They present an integer linear programming model based on the transportation problem as well as Simulated Annealing-based algorithm (SA) to represent the berth allocation problem in tidal bulk ports with stock level conditions. Stock level constraints are important for some dry bulk materials such as minerals because the stock level sometimes depends on a continuous process of consumption or production of minerals. Hence, the decision to load or to unload the vessels must consider the amount of the bulk cargo stored in the port yards. Therefore, a basic criterion for decision making is to give priority to the vessels related to the most critical mineral stock level. Problem instances are solved by a commercial solver and by a simulated annealing-based algorithm. The SA becomes a valid alternative for finding out good solutions for difficult instances.

2.2 Continuous BAP

Kim and Moon [14] develop an MIP model for the continuous BAP to determine the berthing times and positions of containerships in container terminals. A simulated annealing algorithm is also applied to the berth-scheduling problem to find near-optimal solutions. Experimental results show that the simulated annealing algorithm obtains solutions that are similar to the optimal solutions found by the MIP model.

The static continuous BAP is considered by Li et al. [15]. This paper addresses the problem of determining the berthing position and time of each vessel at a container terminal. The objective of the problem is to minimize the sum of the handling time, the waiting time and the delay time for every vessel. They introduce a formulation for the berth allocation problem. Next, they combine genetic algorithm with a heuristic to find an approximate solution to the problem. Computational experiments show that the proposed approaches are applicable to solve the problem.

Guan and Cheung [16] consider continuous dynamic BAP with fixed handling times to minimize the total weighted port service time of vessels. They differentiate the vessels in the objective function based on their importance and subsequently proposed two MILP models. One of the MILP models is similar to the model proposed by Kim and Moon [14] and the other MILP is used to obtain a lower bound.

The continuous BAP with handling times depending on berthing positions is studied by Imai et al. [17]. They present the continuous berth allocation problem to minimize the total service time of vessels. They also present a heuristic for the berth allocation problem which solves the problem in two stages, by improving the solution for the discrete case. A wide variety of experiments are conducted and the results show that the heuristic works well in practice.

Chang et al. [18] consider a BAP which requires the determination of exact berthing times and positions of incoming vessels in a container port. The problem is solved by optimizing the berth schedule to minimize concurrently three objectives of makespan, waiting time, and degree of deviation from a predetermined priority schedule. They propose a multi-objective evolutionary algorithm for solving the multi-objective BAP.

2.3 Hybrid BAP

Moorthy and Teo [19] investigate the hybrid berth allocation problem with fixed handling time. In their study, a container terminal is divided into a number of berthing space in a line, which are further subdivided into sections that vessels need more than one section to be moored. The goal is to identify the impact of vessel delays.

The processing of vessels are considered as activities and represented in a precedence graph which is analyzed using the project evaluation review technique. The dynamic hybrid BAP with position-dependent handling times is studied by Imai et al. [20] for indented berths. They model their problem in hybrid location space with integer linear programming formulation to minimize the delay in handling time of vessels.

Draft restrictions in dynamic hybrid BAP are considered by Nishimura et al. [21]. In their study, up to two vessels can be served at the same berth simultaneously if their total length is less than the overall berth length. To obtain a good solution with considerably small computational effort, they develop a heuristic procedure based on the Lagrangian relaxation of the original problem. They conduct a large amount of computational experiments which show that the proposed algorithm is adaptable to real world applications.

Umang et al. [22] study the hybrid berth allocation problem in bulk ports with the objective to minimize the total service times of the vessels. They discretize the quay into a set of sections. In their berthing assignment, the vessel may occupy more than one section, however a section cannot be occupied by more than one vessel or part of a vessel at any time. They propose two exact methods based on mixed integer programming and generalized set partitioning problem. Due to the difficulty of the problem, they propose a heuristic method based on squeaky wheel optimization for solving the BAP. The formulations are compared through extensive numerical experiments based on instances inspired from real bulk port data. The results indicate that the set partitioning method and the heuristic method can be used to obtain near-optimal solutions for even large size instances.

2.4 Integrated BAP

Since BAP is closely related to other operational decisions in a terminal, some studies in the literature attempted to integrate BAP with some of these decisions.

Kim and Park [23] introduce a nonlinear integer programming model for BAP that considers quay crane assignments as well. The quay is represented as a continuous line and the objective function is to minimize the sum of penalty terms over all vessels.

Cordeau et al. [24] discuss the service allocation problem at a container transshipment terminal based on Gioia Tauro port. They solve the version of Quay Crane Scheduling Problem (QCSP) that is defined by Kim and Park [23]. Cordeau et al. [24] integrate the continuous berth allocation problem with QCSP. The objective is to minimize the container handling time inside the yard. There are two mathematical formulations in the paper.

Liang et al. ([25], [26]) introduce a formulation for the simultaneous berth and quay crane scheduling problem. The objective of the problem is minimizing the sum of the handling time, waiting time and the delay time for every vessel. They propose a genetic algorithm approach with a priority-based encoding method to find an approximate solution to the problem. Computational experiments show that the proposed approaches are applicable to solve the problem.

Meisel and Bierwirth [27] describe the integrated problem of BAP and QCSP in detail against the background of different terminal properties and objectives. A quay crane allocation problem formulation is derived from a new classification scheme for the berth allocation problems. Particular focus is put on integrated solution approaches which have importance for the terminal management.

Blazewicz et al. [28] study the problem of allocating berths to incoming vessels and assigning the necessary quay cranes to the vessels at a container terminal port. They formulate the problem as the moldable task scheduling problem by considering the tasks as vessels and processors as quay cranes assigned to the vessels. This observation is based on the number of quay cranes allocated to a vessel. In other words, the duration of a vessel in the berth location depends on the number of quay cranes allocated to the vessel. In the model, the processing speed of a vessel is considered to be a non-linear function of the number of quay crane allocated to it. Blazewicz et al. [28] present a suboptimal algorithm that obtains a feasible solution for the discrete version of the problem and employ computational experiments to evaluate the performance of the algorithm. The computational results show that the behavior of the algorithm is very good.

Robenek et al. [29] study the integrated problem of berth allocation and yard assignment in the context of bulk ports. The authors assume that a cargo type (in their case liquid and dry bulk) is stored at its specific location. In their research, two crucial optimization problems are studied. They discuss how these problems are interrelated and can be combined and solved as a single large scale optimization problem. More importantly they underline the differences in operations between bulk ports and container terminals which highlights the need to devise specific solutions for bulk ports. The objective is to minimize the total service time of vessels berthing at the port. They propose an exact solution algorithm based on a branch and price framework to solve the integrated problem.

Chapter 3

BERTH ALLOCATION PROBLEM

In this chapter, the general version of partitioned BAP in an exporting dry bulk terminal is considered in which the tidal condition is also investigated. We present a mixed integer linear programming model for the problem which is based on the sequence-variables and the objective is to minimize the vessels' completion time.

The organization of this chapter is as follows: In Section 3.1 we give a definition for the BAP with respect to all assumptions made related to the problem. In Section 3.2, we introduce our Mixed Integer Linear Programming (MILP) model and formulations. In Section 3.3, we introduce three sets of additional constraints to the proposed model to tighten the formulation. Finally, in Section 3.4, we develop two simple heuristic algorithms to obtain a feasible solution, which can be used as an upper bound for the mathematical formulation.

3.1 Problem Definition

In this section, we discuss BAP problem to consider the question of how to allocate vessels to a location on a berth and the sequence in which the vessels should be processed in order to minimize the total completion time of vessels. In this problem, the partitioned layout in which the quay is discretized into a set of sections, and tidal constraints which separate the time horizon into high and low tide periods, are highly potent.

Similar to Umang et al. [22] and Robenek et al. [29], we also consider the partitioned version of BAP in which the quay is discretized into a set of sections of variable lengths. Umang et al. [22] consider the BAP with hybrid berth layout, which is a particular layout of partitioned BAP. In their berthing assignment, a given vessel may occupy more than one section; however, a section cannot be occupied by more than one vessel at any time. Robenek et al. [29] extend this berth allocation problem to integrate it with the yard assignment problem, which is much more complex and extensive than the berth allocation problem studied in Umang et al. [22]. On the other hand, in our berthing assignment, the arrival time of vessels is static and at each section only one vessel can be allocated at a specific time.

In this problem, we consider the berthing area into two dimensions: vertical and horizontal axes. The vertical axis is devoted to the length of the quay, which is separated into several sections and the horizontal axis is devoted to time periods. The incoming vessels to the berth location are considered as rectangles, where the vertical length defines the length of the vessel, and the horizontal length defines the handling time of the vessel. The decisions on placing a rectangle to a berth location are associated with the vessel length and the section length. The vertical length of the rectangle should meet the length of the section. By definition of partitioned BAP, only one rectangle can be allocated to each section at a time. On the other hand, the decisions on assigning a rectangle to the time horizon depend on the vessel arrival time and its handling time. The left-hand side of the rectangle should meet the arrival time of vessel. Hence, the right-hand side of the rectangle, which is the left-hand side plus handling time, indicates the completion time of the vessel.

Regarding the tidal constraint, although Xu et al. [11] consider the BAP problem with tidal effects, there are some limitations in their consideration. They assume that there is only one tidal period in time horizon. So, the schedule should be updated periodically to apply the model. Unlike their assumption, Barros et al. [11] consider the effect of the tidal condition periodically in time horizon. They define their berth setting as a transportation problem in which vessels are seen as suppliers and Tidal Time Windows (TTW) as consumers. Each vessel must be allocated to a subset of TTW whose length corresponds to the handling time which is necessary for operation completion. For this reason, the continuous time scale is changed into a discrete tidal scale, in a way that it is easy to compute the number of TTWs for each allocated vessel. In this study, we also take the effect of the tidal condition periodically in time horizon. In the time horizon there are several high tide and low tide periods that happen alternatively. At low tide, available depth is not adequate for vessels to depart the berth location. The assigned vessel to the berth location is just able to leave the terminal in the high tide period of the time horizon. So, in this study, the high tide which makes the departure time of moored vessel possible is of significant importance.

In an earlier study (Ernest et al. [30]) a continuous berth allocation problem under tidal effects is studied. Since, it is one of the most difficult versions of BAP problems, the computational time is the limiting issue for that study. In order to overcome this limitation we consider a partitioned berth allocation problem. We assume that the quay can be separated into several sections and each section can serve only one vessel at any time. To compare two models, continuous one provides more freedom for assigning vessels to the berth location, but solving the model is harder than the partitioned one.

To define our problem mathematically, we make a number of assumptions, most of which are valid for any berth allocation problem:

1. Once a vessel is moored, it will remain in its location until all the required processing is done. It means the loading and unloading of each vessel occur without interruption.

2. The total length of quay is divided into several sections and each section can handle at most one vessel at a time. The partitioning should be applied in a rational way. For example, the interval of the lower bound and the upper bound of section's length should be respectively between the smallest length of vessel and the largest one. If arriving vessels have approximately the same length, the quay part can be equally partitioned into sections as large as the common vessel size and the problem will become a discrete BAP.

3. The departure time of a vessel from quay part must occur in a high tide period. We examine the effects of tidal condition that happens periodically in the time

horizon. At low tide, it is impossible for vessels to depart the berth location. Since the loading and unloading time of each vessel (handling time) is continuous, the decision to start allocating each vessel depends on its handling time whether it will be finished before the end of high tide condition or not. So, the assigned vessel to the berth location is just able to depart the berth location in the high tide period of the time horizon.

In the continuous BAP version of this problem [30], the problem is similar to a strip packing problem with fixed orientation, where the length of the quay is the width of a strip and the time dimension is the length of the strip. However, considering the tidal constraints prevents directly applying the mathematical models developed for strip packing problems to the problem in [30]. In their model, they define two sequence binary variables, one associated with the decision of sequencing the vessels over the time horizon and the other one associated with the decision of sequencing the vessels over the quay side. Furthermore, a continuous variable is defined which defines the position of an assigned vessel along the quay. (the full description of the model is in [30]). On the other hand, in the partitioned BAP, instead of quay side sequence variables and position variables, a new binary variable, which verifies if a certain vessel is allocated to a certain section, is defined. Besides, a section index is added to the binary variables associated with the decision of sequencing the vessels over the time horizon in each section. So, we define our decision variables based on section, vessel and time.

3.2 Mathematical Model

We consider that the quay is divided into a number of sections and there is a set of vessels that need to be moored at one of sections. The set of sections is denoted by S where $S = \{1, 2, ..., M\}$ and M is the number of sections, and V represents the set of vessels where $V = \{1, 2, ..., J\}$ and J is the number of vessels. For the set of vessels with different lengths, we discretize the length of sections into L_m with uniform distribution between the smallest length of vessel and the largest one. For the set of vessels with the same size, we discretize the length of sections into equal (unit) size sections. Hence, the length of the quay can be represented by the summation of the length of sections. We also discretize the time assuming a finite planning horizon, i.e. one week or two weeks, into high tide and low tide periods. The set of high tide periods is denoted by H where $H = \{1, 2, ..., I\}$ and I is the number of high tide periods.

We model the problem using mixed integer linear programming, which is referred to as Partitioned Model (PM) throughout the rest of the study, as follows:

Indices:

m : sections,

j, k : vessels,

i : high tide periods,

Sets:

S : set of sections,

V : set of vessels,

H : set of high tide periods,

Parameters:

$L_{\max}$: the maximum length of quay location

L_m : length of section m, $m \in S$

a_j : arrival time of vessel j, $j \in V$

v_j : length of vessel j, $j \in V$

P_j : handling time of vessel j, $j \in V$

B_i : the start of high tide period i, $i \in H$

E_i : the end of high tide period i, $i \in H$

Three sets of binary variables are considered:

(i) Variables y_{mj} are associated with the decision of vessels position based on sections, $m \in S, j \in V$:

$$y_{mj} = \begin{cases} 1 & \text{if vessel } j \text{ is allocated in section } m, \\ 0 & \text{otherwise.} \end{cases}$$

(ii) Variables x_{mjk} are associated with the decision of sequencing the vessels over the time horizon at the defined sections, $m \in S, j, k \in V : j \neq k$:

$$x_{mjk} = \begin{cases} 1 & \text{if vessel } k \text{ is allocated after vessel } j \text{ in section } m, \\ 0 & \text{otherwise.} \end{cases}$$

(iii) Variables z_{ji} are associated with the decision of using the high tide period i, $j \in V, i \in H$:

$$z_{ji} = \begin{cases} 1 & \text{if the handling time of vessel } j \text{ completes in high tide period } i, \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, two sets of continuous variables are considered:

(iv) S_j : when the handling time of vessel j begins, $j \in V$.

(v) C_j : when the handling time of vessel j is complete, $j \in V$.

The mathematical formulation of the partitioned berth allocation (PM) can be written as follows:

$$\min \sum_{j \in V} C_j \tag{3.1}$$

subject to

$$\sum_{m \in S} y_{mj} = 1 \qquad \forall j \in V \tag{3.2}$$

$$v_j y_{mj} \leq L_m \qquad \forall m \in S, j \in V \tag{3.3}$$

$$S_j \geq a_j \qquad \forall j \in V \tag{3.4}$$

$$C_j = S_j + P_j \qquad \forall j \in V \tag{3.5}$$

$$S_k \geq C_j - M(1 - x_{mjk}) \qquad \forall m \in S, j, k \in V : j \neq k \tag{3.6}$$

$$x_{mjk} \leq y_{mj} \qquad \forall m \in S, j, k \in V : j \neq k \tag{3.7}$$

$$x_{mjk} \leq y_{mk} \qquad \forall m \in S, j, k \in V : j \neq k \tag{3.8}$$

$$x_{mjk} + x_{mkj} \leq 1 \qquad \forall m \in S, j, k \in V : j \neq k \tag{3.9}$$

$$x_{mjk} + x_{mkj} \geq y_{mj} + y_{mk} - 1 \qquad \forall m \in S, j, k \in V : j \neq k \tag{3.10}$$

$$C_j \geq \sum_{i \in H} B_i z_{ji} \qquad \forall j \in V \tag{3.11}$$

$$C_j \leq \sum_{i \in H} E_i z_{ji} \qquad \forall j \in V \tag{3.12}$$

$$\sum_{i \in H} z_{ji} = 1 \qquad \forall j \in V \tag{3.13}$$

$$y_{mj} \in \{0, 1\} \qquad \forall m \in S, j \in V \tag{3.14}$$

$$x_{mjk} \in \{0, 1\} \qquad \forall m \in S, j, k \in V \tag{3.15}$$

$$z_{ji} \in \{0, 1\} \qquad \forall j \in V, i \in H \tag{3.16}$$

$$S_j \geq 0 \qquad \forall j \in V \tag{3.17}$$

$$C_j \geq 0 \qquad \forall j \in V \tag{3.18}$$

The objective function (3.1) minimizes the sum of completion time for all vessels. Constraint set (3.2) expresses that each vessel should be assigned to one section. Constraint set (3.3) ensures that the length of vessel allocated in section m must be less than the length of that section. Constraint set (3.4) requires that each vessel starts its processing only after it has arrived at the terminal. Constraint set (3.5) shows that the completion time of each vessel is equal with the handling time and the start time of it. Constraint set (3.6) implies that if x_{mjk} is equal to one, the start time of vessel k cannot be earlier than C_j. In other words, if x_{mjk} is equal to one (vessel j is handled before vessel k), the start time of vessel k cannot be earlier than the completion time of vessel j. Otherwise, where M is a large constant, the constraint will be relaxed. Constraint sets (3.7) and (3.8) ensure that the amount of binary variables x_{mjk} and x_{mkj} should be less than y_{mj} and y_{mk}, respectively. Constraint sets (3.9) and (3.10) enforce that one of x_{mjk} and x_{mkj} equals 1 if vessels j and k are both assigned to section m and vessel j (k) is processed before vessel k (j). They also ensure that $x_{mjk} = x_{mkj} = 0$, if one of vessels j and k is not assigned to section m. Constraint set (3.11) indicates that for every vessel the departure time (completion time) should be equal to or greater than one of the possible beginning of high tide period. Constraint set (3.12) ensures that every vessel at the berth location should start loading such that the completion time (start time of loading plus handling time) of it will finish before the last possible high tide period. Note that the binary variable z_{ji} defines whether the handling time of vessel j completes in high tide period i or not. So constraint sets (3.11) and (3.12) construct a lower bound and an upper bound, respectively, for the departure time of vessels. Note that both the lower bound and the upper bound should be in the same high tide period (i.e. $B_i \leq C_j \leq E_i$). Constraint set (3.13) shows that the departure time of each vessel from berth location should happen just in one of the possible high tide periods.

This formulation is not very tight and does not perform well computationally, i.e. for large-size problems it even takes more than one hour to solve the problem. In the next section we discuss some ways to improve the formulation.

3.3 Valid Inequalities and Symmetry Elimination Constraints

In this study, we use PM model for the mathematical formulations in order to find optimal solutions for the BAP problem. Analyzing the properties of optimal solutions allows us to add additional constraints to the PM model. These constraints help us to generate tighter upper bounds with less computational time. We develop three sets of additional constraints, two of which are valid inequalities. The constraints are as following:

1. The first set of constraints, which are valid inequalities, is called “tightening constraint and variable” and introduce some cuts based on tidal conditions. Since the assumptions about tidal conditions in this study are similar to that of [30], the properties of the optimal solution related to binary variables z_{ji} are the same as well. These cuts fix some z_{ji} variables to zero. For instance, for the vessel for which the arrival time plus handling time is greater than some E_i's, the summation of z_{ji} should be zero. So, we will have the following equation:

$$\sum_{i \in H: E_i < a_j + P_j} z_{ji} = 0 \qquad \forall j \in V \tag{3.19}$$

2. The second set of constraints is called "symmetry elimination". There are a lot of cases that swapping the precedence of two vessels does not affect the objective function value in the optimal solution or feasibility of the solution. One such case is that two vessels with identical handling time can be assigned to the same section. In Ernest et al. [30], they show that in continuous version of BAP, in the case where vessels have identical length and handling time, the symmetry can be simply eliminated. In this study, the assumption that two vessels can be allocated to an identical section, is equivalent to the assumption in continuous study that these vessels have the same length. Therefore, their symmetry elimination constraint can be applied in this study as well.

Consider two vessels j and k, that both can be allocated to section m. Besides, the handling time of vessel j and k are equal. Without loss of generality, let $a_j \leq a_k$, we can consider the following cases:

Case 1: j can be assigned to section m as soon as it arrives. In this case, according to objective function, j automatically precedes k.

Case 2: Neither j nor k can be assigned to section m upon their arrival. In this case, objective function is indifferent to precedence order of j and k, i.e. there is a symmetry in precedence order of j and k. Therefore, vessel j can be allocated before vessel k without affecting the objective function.

Hence, in general case, we can write:

$$\sum_{m \in S: V_j \leq L_m, V_k \leq L_m} x_{mkj} = 0 \qquad \forall j, k \in V : j \neq k, a_j \leq a_k, P_j = P_k \tag{3.20}$$

Note that in case 1, the symmetry elimination constraint will be redundant.

3. The third set of constraints, which are valid inequalities, refers to the binary variables y_{mj}. In mathematical model, the binary variable y_{mj} defines the position of vessel j in section m with respect to the constraint (3.3). So, when the length of an arbitrary vessel is greater than the length of an arbitrary section, according to constraint (3.3), it is impossible to assign the vessel to the section. $j \in V, k \in V$ and $m \in S$, if $V_k \leq L_m, V_j > L_m$, we have:

$$y_{mj} = 0 \qquad \forall j \in V, m \in S : V_j > L_m \tag{3.21}$$

and this cut dominates constraint (3.3). Consequently, the precedence variable associated to m and j must be zero for each vessel k that can be allocated to section m:

$$x_{mjk} + x_{mkj} = 0 \qquad \forall j, k \in V, m \in S : j \neq k, V_k \leq L_m, V_j > L_m. \tag{3.22}$$

3.4 Heuristic Algorithms

Since the computational time of the PM is prohibitive for large-size instances, we developed two simple heuristic algorithms and present their details in this section.

3.4.1 Greedy Algorithm 1

We present a simple greedy heuristic algorithm that we call it "Greedy Algorithm 1". The algorithm assigns vessels to the sections one by one. At first (Algorithm 1, step 1), array of vessels, high tide periods, vessels waiting for empty section, vessels waiting for departure and list of future events are created, then vessels are sorted in a non-decreasing order of vessels' arrival time. For each vessel, an arrival event is created and is added to the list of future events. Finally, for each high tide period, two events respectively associated to its start time and end time are created and added to the list of future events. In step 2, as long as there exists any vessel in the system, the event with least time is chosen according to the type of event, i.e. arrival, service start, service finish, tide begin and tide end, and routines 1, 2, 3, 5 and 6 are called, respectively.

In routine 1, upon arrival of a new vessel, if the waiting list is empty, the list of sections is scanned for a section that is large enough to accommodate the vessel. If such section exists, the vessel is allocated to the section and a new service start event is added to the list of future events. Otherwise, if the waiting list is not empty or the length of chosen vessel is larger than the length of empty sections, the vessel enters the waiting list.

In routine 2, handling time of vessel starts and the finish time is calculated according to its handling time and a respective event is added to the list of future events.

In routine 3, high tide status is checked, since the departure time of vessel should coincide with a high tide period. If the end of the handling of the vessel is within a high tide period, then the vessel departs the section and subsequently the section is set to empty state. Besides, if the waiting list is not empty, routine 4 is called. If the end of the handling time of the vessel is not within a high tide period, the vessel enters the list of waiting vessels for departure.

In routine 4, we enumerate all vessels in the waiting list, from first vessel through the end. For each vessel, if there is an appropriate empty section, the vessel is allocated to the section and a new service start event is added to the list of future events. Otherwise, the algorithm terminates.

In routine 5, upon beginning of high tide period, for each vessel in the list of vessels waiting for departure, the completion time of the vessel is set and the state of associated section is set to empty.

```
Input: Required parameters for solving the problem;    // Parameters related
       to section, vessel and time
Output: final solution for the BAP problem;    // It obtains near optimal
        solution for PM model
Step 1: Initialize;
Let V denote the array of vessels.
Let T denote the array of high-tide periods.
Let Q^S denote the array of vessels waiting for empty section.
Let Q^D denote the array of vessels waiting for departure.
Let FEL denote the list of future events.
Sort V in a non-decreasing order of arrival time.
for each vessel in V do
|   Add new arrival event to FEL.
end
for each t in T do
|   Add new high-tide begin event to FEL.
|   Add new high-tide end event to FEL.
end
Step 2: Check overlapping;
repeat
|   while there exists any vessel in system; do
|   |   Let e be the event with least time.;
|   |   if e is arrival then
|   |   |   Run the routine 1 for e.
|   |   end
|   |   if e is service start then
|   |   |   Run the routine 2 for e.
|   |   end
|   |   if e is service finish then
|   |   |   Run the routine 3 for e.
|   |   end
|   |   if e is high tide begin then
|   |   |   Run the routine 5 for e.
|   |   end
|   |   if e is high tide end then
|   |   |   High-tide:=false
|   |   end
|   |   Remove e from FEL.
|   end
until terminate;
```

Algorithm 1: Greedy Algorithm 1

Input: event e; // event e is vessel arrival with least time

Routine 1: Arrival for event e;

Let v, t and m be the vessel, time and section of e, respectively.

if $Q^S = 0$ *and an appropriate empty section exists* **then**

  Assign v to the section.

  Add new service start event to FEL for v, t and m.

else

  Add the vessel to the list of Q^S.

end

Algorithm 2: Routine 1

Input: event e;

Routine 2: Service start for event e;

Let v, t and m be the vessel, time and assigned section of e, respectively.

Let $C := t + P$ (P is the handling time of vessel v).

Add new service finish event to FEL for v, C and section.

Algorithm 3: Routine 2

Input: event e;

Routine 3: Service finish for event e;

Let v, t and m be the vessel, time and assigned section of e, respectively.

if *high-tide period is true;* **then**

  Set service completion time of $v := t$.

  Set section status to empty.

  if *length of* $Q^S > 0$ **then**

    Run routine 4.

  end

else

  Add the vessel to the list of Q^D.

end

Algorithm 4: Routine 3

Input: event e;

1 **Routine 4: Sending waiting vessels in Q^S to empty sections**

2 Let t be the time of e, and let $n :=$ length of Q^S.

3 **if** $n > 0$ **then**

4   Let v denote the n^{th} vessel in Q^S.

5   **if** *there is no appropriate empty section for* v **then**

6   Terminate.

7   **end**

8   Let m be the appropriate empty section for v.

9   Assign v to the section.

10   Add new service start event to FEL for v, t and m.

11   n:=$n - 1$.

12   Go to 3.

13 **end**

Algorithm 5: Routine 4

Input: event e;

1 **Routine 5: High tide begin for event** e

2 High-tide:=true.

3 **if** *length of* $Q^D > 0$ **then**

4   **for** *each vessel in* Q^D **do**

5   Send out the vessel.

6   Set the status of the associated section to empty.

7   **end**

8 **end**

9 Run routine 4.

Algorithm 6: Routine 5

3.4.2 Greedy Algorithm 2

Now we present the second greedy algorithm which is called " greedy algorithm 2". Basically, it is similar to greedy algorithm 1. In greedy algorithm 1, a vessel is allocated sooner if and only if it arrive sooner. On the other hand, in greedy algorithm 2, based on the fitness of the vessel, one may be allocated even if it arrives later. There are two differences between greedy algorithms 1 and 2: First, upon arrival of a new vessel (routine 6), the list of vessels waiting for empty section is not checked anymore and the sections are directly scanned for an appropriate section, if no such section exist, then the vessel enters the list of vessels waiting for empty section. Second, upon trying to allocate the vessels in the list of vessels waiting for empty section (routine 8), as long as any empty section exists, all of the vessels in the list are scanned for allocation. In spite of routine 4 that stops scanning the list as soon as there is no appropriate section for the first vessel in the list, in routine 8, other vessels are scanned as well. So, the algorithms are as the following:

Input: Required parameters for solving the problem; // Parameters related to section, vessel and time

Output: final solution for the BAP problem; // It obtains near optimal solution for PM model

```
Step 1: Same as step 1 in greedy algorithm 1
Step 2: Check overlapping;
repeat
    while there exists any vessel in system; do
        Let e be the event with least time.;
        if e is arrival then
            Run the routine 6 for e.
        end
        if e is service start then
            Run the routine 2 for e.
        end
        if e is service finish then
            Run the routine 7 for e.
        end
        if e is tide begin then
            Run the routine 9 for e.
        end
        if e is tide end then
            High-tide:=false
        end
        Remove e from FEL.
    end
until terminate;
```

Algorithm 7: Greedy Algorithm 2

Input: event e;

Routine 6: Arrival for event e

Let v, t and m be the vessel, time and section of e, respectively.

if *an appropriate empty section exists* **then**

  Assign v to the section.

  Add new service start event to FEL for v, t and m.

else

  Add the vessel to the list of Q^S.

end

Algorithm 8: Routine 6

Input: event e;

Routine 7: Service finish for event e;

Let v, t and m be the vessel, time and assigned section of e, respectively.

if *high-tide period is true;* **then**

  Set service completion time of $v := t$.

  Set section status to empty.

  if *length of* $Q^S > 0$ **then**

    Run routine 8.

  end

else

  Add the vessel to the list of Q^D.

end

Algorithm 9: Routine 7

Input: event e;

1 **Routine 8: Sending waiting vessels in Q^S to empty sections**

2 Let t be the time of e and let n:= length of Q^S.

3 **if** $n > 0$ **then**

4   Let v denote the n^{th} vessel in Q^S.

5   **if** *there is any appropriate empty section for* v **then**

6   Let m be the appropriate empty section for v.

7   Assign v to the section.

8   Add new service start event to FEL for v, t and m.

9   **end**

10     $n := n - 1$.

11   Go to 3.

12 **end**

Algorithm 10: Routine 8

Input: event e;

Routine 9: High tide begin for event e

High-tide:=true.

if *length of* $Q^D > 0$ **then**

    for *each vessel in* Q^D **do**

        Send out the vessel.

        Set the status of the associated section to empty.

    end

end

Run routine 8.

Algorithm 11: Routine 9

Chapter 4

COMPUTATIONAL ANALYSIS

In this chapter, we perform computational experiments to analyze the performance of the proposed mathematical model and the additional constraints as well as two simple heuristic algorithms.

The organization of this chapter is as follows: In Section 4.1, we describe the procedure of random data generation. In Section 4.2, we present the computational results of PM model with and without additional constraints. In Section 4.3, first, we compare the results of PM model with that of the continuous model, then, we compare the results of two greedy algorithms with each other. Finally, in Section 4.4, the effects of some parameters on the performance of PM model and greedy algorithm 2 are investigated.

4.1 Data Generation

In this section, test instances, which are randomly generated to evaluate the effectiveness of the proposed models are explained. The instances are generated based on the real data of a dry bulk terminal in Newcastle, Australia. The details of the data generation methods are introduced as following.

1. The length of vessels is considered in three cases.

    (a) In one set of the instances, each vessel has the same length of 1.

    (b) In the second set of instances, we have a random value for the length of the vessels that are generated from a uniform distribution of $U(0, 2)$ (where the uniform distribution is denoted by U).

    (c) In the third set of instances, we have the following structure:

   i. 10% of the vessels will have a v_j generated of $U(0, 0.85)$,
   ii. 30% of the vessels will have a v_j generated of $U(0.85, 1.36)$,
   iii. 60% of the vessels will have a v_j generated of $U(1.36, 2)$.

2. The corresponding handling time (in hour) of vessels (P_j) are generated in three different ways:

   (a) For the first and second sets, handling time is generated from either of the following uniform distribution:

      i. $U(16, 20)$
      ii. $U(5, 20)$

   (b) For the third set, handling times are as follows:

      i. $P_j = 7$ for the 'case1. (c)i. above',
      ii. $P_j = 12$ for the 'case1. (c)ii. above',
      iii. $P_j = 15$ for the 'case1. (c)iii. above'.

3. We consider the arrival time (in hour) of vessels (a_j) in two structures:

   (a) Vessels arrive randomly according to a uniform distribution of $U(0, 150)$.

   (b) Vessels arrive at 12:00 noon every day.

4. We also consider three cases for the number of arriving vessels per weeks as J=16, 18, and 20.

5. The tidal times (in hour), i.e. B_i and E_i, are obtained from the public website of Newcastle port. A sample of such periods for one week can be seen in Table 4.1.

Table 4.1: A sample of high tide periods for one week

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
B_i	1	13	26	38	51	63	76	89	101	114	126	139	151	164
E_i	7	20	32	44	57	69	83	94	108	120	133	145	158	169

6. The length of sections (L_m) are generated according to a uniform distribution $U(a, b)$, where a is less than the smallest length of vessels and b is greater than the largest one. A pseudo-code of the generation scheme is provided in Algorithm 12.

Input: a as minimum length of sections, b as maximum length of sections, M as number of sections, VL as set of vessel lengths, and L_{max} as the total length of quay side;

Output: Length of sections;

1 Let $SL = \Phi$ denote the set of section lengths.
2 **for** m *from 1 to* $M - 1$ **do**
3   Generate random number r from $U(a, b)$.
4   Set L_m equal with r rounded to one decimal place.
5   Add L_m to SL.
6 **end**
7 Set $L_M = L_{max} - \sum_{m=1}^{M-1} L_m$.
8 **if** $L_M \notin [a, b]$ **then**
9   Go to 1.
10 **end**
11 Add L_M to SL.
12 **if** $Max(SL) < Max(VL)$ **then**
13   Go to 1.
14 **end**
15 Return SL as the set of length of sections.

Algorithm 12: The procedure of generating length of sections

To sum, the combination of the mentioned structure results in 30 different scenarios, i.e. ($3 \times 2 \times 2 \times 2 + 3 \times 2 \times 1 \times 1 = 30$) and for each scenario we generate 10 random instances. We use these scenarios in two different sets of instances (set 1 and set 2). In set 1, the maximum length of the quay is equal to 3 ($L_{\max} = 3$), the quay is partitioned into 3 sections with different lengths and the set consists of 200 instances in 20 scenarios. While in set 2, $L_{\max} = 5$, the total number of sections is $M = 5$, and it consists of 30 scenarios with 300 instances.

4.2 Computational Results

This section presents computational experiments to test the performance of the PM model without and with additional constraints. Computational experiments are carried out on a HP-Z800 Work Station that contains: Intel(R) Xeon(R) CPU, 2.4GHz (2 processors) and 12 GB of RAM. The mathematical models are solved by IBM ILOG CPLEX 12.5.1, and an elapsed time limited to one hour is enforced.

To reflect the exact performance of each set of additional constraints, first we add them to the PM model with different combinations, i.e. one by one or with the combination of set 1 with set 2, set 1 with set 3 and set 2 with set 3. For each combination, then we consider the running time and optimality gap of instances separately. To evaluate the exact performance of each set of additional constraints, we use the set of instances (set 1) in which, $L_{\max} = 3$, it is partitioned into 3 sections with different lengths and it consists of 200 instances in 20 scenarios. Tables 4.2 and 4.3 indicate the running time and optimality gap (Gap_{opt}) of instances, respectively. The optimality gap is calculated based on the following formula:

$$Gap_{opt}\% = (\frac{UB - LB}{LB}) \times 100,$$

where UB and LB are the Upper Bound (UB) and the Lower Bound (LB) obtained from PM, respectively.

According to Table 4.2, the first column shows the scenarios that are used to indicate the characteristics of each instance. Each scenario can be read as follows: $J\_T\_v_j\_a_j\_p_j$, for example scenario $16\_1\_Unit\_Uniform\_16$ indicates that, there exist 16 vessels with unit size that arrive to quay part with uniform distribution during one week, and their handling time is generated from $U(16, 20)$. The column "avg. time" indicates the running time of instances obtained by solving the PM model without and with different combinations of additional constraints, respectively. For example, the columns 2 to 9 in order indicate the running time of PM model, PM model with the first set of constraints (PM+AC1), PM model with the second set of constraints (PM+AC2), PM model with the third set of constraints (PM+AC3), PM model with the first and the second sets of constraints (PM+AC1+AC2), PM model with the first and the third sets of constraints (PM+AC1+AC3), PM model with the second and the third sets of constraints (PM+AC2+AC3) and PM model with all three sets of constraints (PM+AC). For each scenario, each entry of columns 2 to 9 is the average value over 10 instances. To evaluate the effects of additional sets of constraints one by one, after comparing the results of PM model with each set, i.e. columns 3 to 5, we realize that the second set of constraints called "symmetry elimination" has the most contribution to PM. Besides, to find the best combination of additional constraints, we add column 10 (min. time) to Table 4.2, which shows the minimum running times for each scenario. By referring to the "min. time" column, we find out that the combination of all three sets of additional constraints with the PM model almost outperforms other combinations.

In Table 4.3, the results of instances in which optimality gap is non-zero for different combinations of sets of additional constraints with PM model are shown. The columns indicate the average optimality gap (avg. Gap_{opt}) over 10 instances for each scenario for PM model without and with different combinations of additional constraints, respectively. We also add column "min. Gap_{opt}" to Table 4.3, which shows the minimum amount of optimality gaps for each scenario.

Results presented in Tables 4.2 and 4.3 support the claim that the combination of all three sets of additional constraints with the PM model outperforms other combinations, in both running time and optimality gap. Hence, for further computational experiments, we incorporate all these three additional constraints with PM.

In Tables 4.4 and 4.5, we present the fundamental attributes of the PM related to its performance in the columns: The "time" column indicates the running time and the root time of the model in seconds, respectively, and the column "Gap_{opt}" presents the optimality gap in percentage between the upper bound and the lower bound found at termination. The column “Number” indicates the number of instances solved to optimality, for each scenario.

In each table, the first column shows the scenarios we consider. For each scenario, we present the result of the mathematical model based on the attributes explained. As we explained before, for each scenario, we generated 10 random instances. So, each entry for all attributes is the average values over 10 instances. The “Overall” indicates the average of the total amount of each column, except the “Number” column, which shows the total number of optimal instances.

In Tables 4.4 and 4.5, all parameters are the same except the total length of the quay and the number of partitioned sections (M). Table 4.4 corresponds to set of instances (set 1) in which, $L_{\max} = 3$, and it is partitioned into 3 sections with different lengths, while Table 4.5 corresponds to set of instances (set 2) in which, $L_{\max} = 5$, and the total number of sections is $M = 5$. Set 1 consists of 200 instances in 20 scenarios, while set 2 consists of 30 scenarios with 300 instances. For all instances, PM is first solved, and then, three additional constraints are added to the PM (PM+AC) to solve the problem. To test the performance of the mathematical model, without and with additional constraints in each table, we split the tables into two sections to illustrate the effect of additional constraints on each scenario one by one.

Tables 4.4 and Table 4.5 demonstrate that both set of instances are solved in a shorter time and with a smaller gap using PM+AC, compared to PM. So, we conclude that the model with additional constraints always outperforms the PM model. Besides, it is informative to mention that the effect of additional constraints in Table 4.4 is more significant than Table 4.5. For instance in Table 4.4, the number of instances with optimal solution without and with additional constraints are 188 and 193 out of 200 instances, respectively, while in Table 4.5, they are in order 298 and 299 out of 300 instances.

4.3 Comparative Analysis

In this section, at first, we compare the results of PM model with the computational results of continuous BAP proposed in Ernst et al. [30]. Then, we compare two heuristic algorithms to show the superiority of greedy algorithm 2 over greedy algorithm 1. In this section, the computational results are performed on the set of instances introduced in Section 4.1 (set 1 and set 2).

4.3.1 Comparison of the Results of PM Model with the Continuous Model

In this part, we compare the results of partitioned model (PM) with the continuous model proposed in Ernest et al. [30]. To compare these two models, the output of PM model gives us an approximation of the optimal solution instead of the exact one if the problem is considered with a continuous quay. To analyze and compare the results, we define three different values, A, B and C, where:

- A denotes the number of instances in each scenario where the PM model obtains optimal solution.
- B denotes the number of instances in each scenario where the continuous model obtains optimal solution.
- C denotes the number of instances in each scenario where the objective function values (either optimal values or upper bounds) are equal to each other in both models.

To evaluate and compare the results of two models with each other, all parameters used in the models should be same and fixed. Note that in PM model in addition to $L_{\max}$, we have L_m (the length of section) and M (the number of section), whereas in continuous model we only consider $L_{\max}$. In Table 4.6, all parameters of instances are the same as the instances in Table 4.7, except the parameter $L_{\max}$ and the number of sections (in Table 4.6 $L_{\max} = 3$ and $M = 3$, in Table 4.7 $L_{\max} = 5$ and $M = 5$).

Tables 4.6 And 4.7 compare the optimal solutions for both PM model and continuous model in terms of running times (in second) and values A, B and C. We consider each scenario which consists of 10 random instances for both models, separately. In "$Gap_{(PM,C)}$" column, we calculate the gap between the upper bounds obtained by two different models:

$$Gap_{(PM,C)}\% = (\frac{UB_{PM} - UB_C}{UB_C}) \times 100,$$

where: UB_{PM} is upper bound of PM model, and UB_C is the upper bound of continuous model.

As expected, the overall results indicate that the continuous BAP provides better solutions in terms of objective function value. We remark that the objective function for both models refers to a minimization problem, so with same parameters, the model that obtains lower values for objective functions, gives better results. However, in all instances and particularly in larger ones, PM model performs better, and gives better result in terms of running time. Also note that, although PM model is an approximation to the continuous model, the cases in which PM obtains objective function values as good as the objective function values obtained by continuous model, are very frequent, e.g. in Tables 4.6 and 4.7 in order 115 out of 200 and 233 out of 300 such instances are observed. Moreover, since the gap values are not so large, we can conclude that both models have their strengths and can be used for different planning horizon related to the priority in time or the value of objective function.

4.3.2 Comparison of Greedy Algorithm 2 with the Greedy Algorithm 1

To evaluate the performance of two heuristic algorithms, we compare the results of them with each other. In the comparison procedure, all the instances used in the PM model and two heuristic algorithms are same and fixed.

To analyze the results, for each instance, we calculate the gaps between the upper bound of PM model and the solutions obtained from heuristic algorithms, greedy algorithm 1 and greedy algorithm 2, respectively:

$$Gap_{(GA1,PM)}\% = (\frac{UB_{GA1} - UB_{PM}}{UB_{PM}}) \times 100,$$

where UB_{GA1} is the solution obtained from the greedy algorithm 1, and UB_{PM} is the upper bound of the PM model.

$$Gap_{(GA2,PM)}\% = (\frac{UB_{GA2} - UB_{PM}}{UB_{PM}}) \times 100,$$

where UB_{GA2} is the solution obtained from the greedy algorithm 2, and UB_{PM} is the upper bound of the PM model.

To evaluate two algorithms, we compare the gap values for each scenario one by one. Figures 4.1 and 4.2 illustrate the comparison between the gap values of two greedy algorithms as calculated using above-mentioned formulas, for set 1 and set 2, respectively. In Figure 4.1 we used 200 instances in 20 scenarios with $L_{\max} = 3$ and the number of sections is $M = 3$. While in Figure 4.2, there are 30 scenarios with 300 instances with $L_{\max} = 5$ and the total number of sections is $M = 5$.

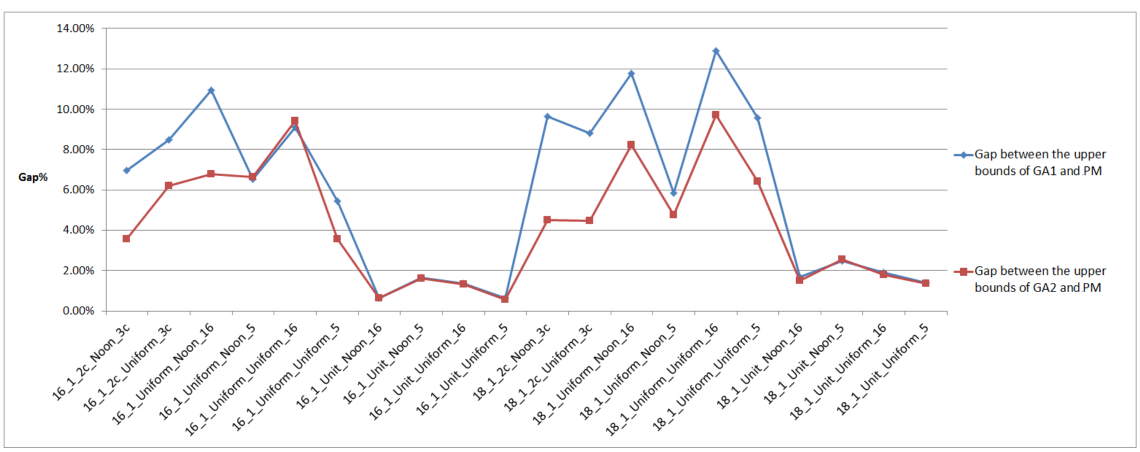

Figure 4.1: Comparison between the greedy algorithm 1 and the greedy algorithm 2 over instances in set 1

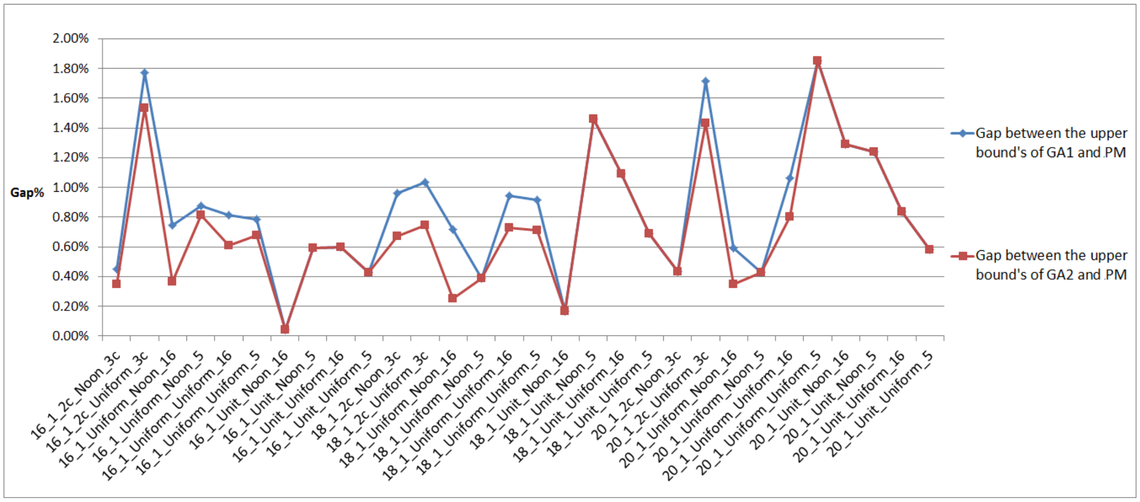

Figure 4.2: Comparison between the greedy algorithm 1 and the greedy algorithm 2 over instances in set 2

In both figures (Figures 4.1 and 4.2), the gaps between the upper bound of the PM model and the greedy algorithm 1 are always equal to or greater than the gaps between the upper bound of the PM model and the greedy algorithm 2. As, the smaller the gap, the better the result, it is concluded that the greedy algorithm 2 outperforms the greedy algorithm 1.

4.4 Effect of the Parameter Settings on the Performance of PM Model and Greedy Algorithm 2

To reflect the importance of some parameters in PM model and heuristic algorithm, we apply sensitivity analysis. As shown in Section 4.2, the greedy algorithm 2 always outperforms the greedy algorithm 1. Therefore, in this section, we only perform sensitivity analysis on greedy algorithm 2. In this study sensitivity analysis is a technique used to determine how different values of some key parameters impact on the performance of PM model as well as the greedy algorithm 2. The parameters include: (1) the length of each section (L_m) with identical($L_{\max}$), (2) the length of each section (L_m) with different L_{max}, and (3) the length of vessels (v_j). Note that, the sets of instances used in this section may differ with the sets of instances generated in Sections 4.1 (set 1 and set 2 in Sections 4.1).

4.4.1 Sensitivity Analysis of Parameter L_m with Identical L_{max}

To interpret the importance of the length of partitioned sections, we use two sets of test instances (with 300 instances and 30 scenarios for each set) with identical $L_{\max}$ and number of sections (M), but with different partition lengths. In both sets parameters $L_{\max}$ and M are set to 7 and 5, respectively. Partitions' lengths in the first set (set 1) are generated from a uniform distribution of $U(0.5, 2)$ as explained in 12 and the values of second set (set 2) are generated after considering the distribution of length of vessels. The value of length of sections for set 1 and set 2 are in order, (2, 1, 1.2, 0.8, 2) and (1.9, 1.9, 1.9, 0.9, 0.4).

According to Table 4.8, the PM model can solve to optimality over all instance in both sets. Therefore, to analyze the results, we consider the running time of PM model for both sets. To compare the results, in the first sight, it seems that there are no perceptible changes in the running time. To generalize our claim, we construct a statistical hypothesis for running time like:

$$H_0 : \mu_D = 0$$

where H_0 is null hypothesis and μ_D is the average of difference of running times of two sets. We use t-test to decide whether two sets have the same running times or not. In this case, we should use dependent two-sample t-test with equal sample sizes. In general, the formula for dependent two-sample t-test, is like below:

$$\mathrm{t} = \frac{\bar{X_D}}{\frac{S_D}{\sqrt{n}}},$$

where t is the static test, $\bar{X_D}$ is the sample average of difference of running times of two sets, S_D is the statistic variance of difference of running times of two sets, and n is the number of instances for both sets 1 and 2. For t-test, we used $\alpha = 0.05$ as the significance level, which is the most common value used in the literature. According to calculation results presented in Table 4.9, the null hypothesis is accepted.

Having the optimal solutions, to analyze the sensitivity of greedy algorithm 2 to the parameter L_m, we calculate the gap between the final solution obtained from PM model and the solution obtained from greedy algorithm 2 for each instance, in each set:

$$Gap_{(GA2,PM)}\% = (\frac{UB_{GA2} - OPT_{PM}}{OPT_{PM}}) \times 100,$$

where UB_{GA2} is the solution obtained from greedy algorithm 2, and OPT_{PM} is the optimal value of PM model.

Table 4.10 indicates the results of two sets, which are different in terms of parameter L_m. To evaluate the effect of parameter L_m on the solutions, we compare the gap values for each instance in two different sets one by one. The computational results show that set 2 corresponds to smaller average gap. In order to generalize our claim, we construct a statistical hypothesis like:

$$H_0 : \mu_D \geq 0$$

where H_0 is null hypothesis, μ_D is the average of gap of set 1 minus gap of set 2. We use t-test to decide whether set 1 has larger gap values compare to set 2 or not. In this case, we also use dependent two-sample t-test with equal sample sizes (n, here n is 30) with the formula that mentioned above. For t-test, we again used $\alpha = 0.05$ as the significance level.

According to calculation results presented in Table 4.11, the null hypothesis is accepted. Hence, set 2 indeed corresponds to smaller gap values. Figure 4.3 also illustrates the effect of parameter L_m on the solutions by displaying the gaps between the optimal value obtained from PM model and the solution obtained from the greedy algorithm 2 for two sets with different L_m and identical L_{max}. The horizontal and vertical axes indicate the scenarios and the gap values (average over 10 instances for each scenario), respectively. The entries of column "avg" in Table 4.10 are the input data as the gap values for Figure 4.3.

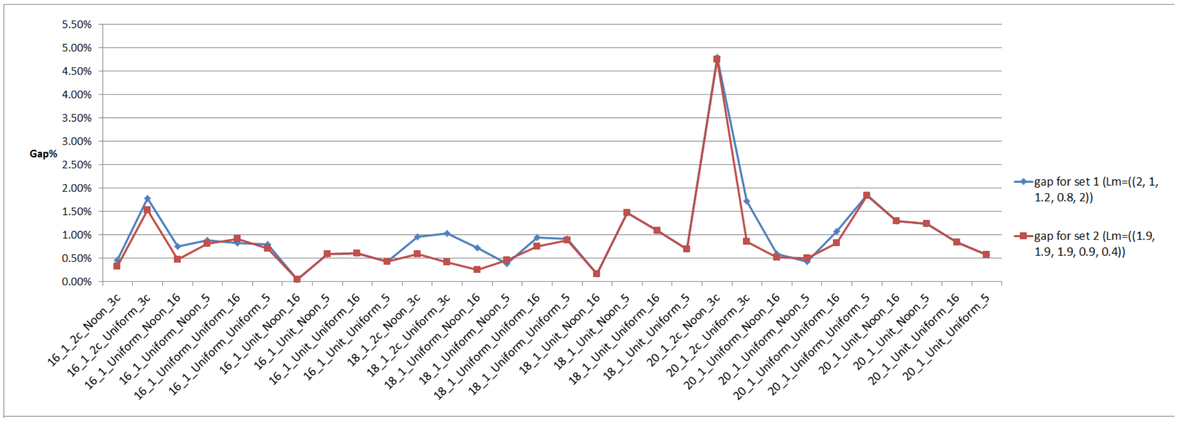

Figure 4.3: The gap between the upper bounds of PM and the solution of greedy algorithm 2 for two sets with different L_m and identical L_{max}

4.4.2 Sensitivity Analysis of Parameter L_m with Different $L_{\max}$

To evaluate the influence of the parameter settings on the PM model, two sets of test instances (with 200 instances and 20 scenarios for each set) are used in this section. All parameters used in the instances are set similarly unless mentioned otherwise. Both of these sets are based on the set of instances introduced in the first part of this chapter (Section 4.1). The number of sections are identical for both sets and equal to 3. While the value of quay lengths and length of sections are different, i.e. for the first set (set 1), $L_{\max} = 3$, and for the second set (set 2), $L_{\max} = 6$. Furthermore, in set 1, as mentioned in the Section 4.1, we discretize the length of sections into L_m with uniform distribution between the smallest length of vessels and the largest one. While in set 2, we discretize the length of sections based on the largest length of vessel, so all vessels are able to be allocated in all sections.

Table 4.12 is split into two parts called "set 1" and "set 2", which correspond to running times and optimality gaps of set 1 and set 2, respectively. To compare the results of these two sets, we consider the running times and the optimality gaps:

$$Gap_{opt}\% = (\frac{UB - LB}{LB}) \times 100,$$

where UB and LB denote the upper bound and the lower bound obtained from PM model, respectively.

According to Table 4.12, for the instances solved to optimality, the running time of instances in set 1 is better than set 2. Moreover, the number of instances with the optimal solution in set 1 is more than set 2. For the remaining instances, where optimality is not met, and are terminated due to time limit, when we compare the results of two sets, the overall results indicate that the optimality gap in set 1 is smaller than set 2. Generally, the most important reason behind the increase in running time and optimality gap is the increase of number of nonzero decision variables (y_{mj} and x_{mjk}). In fact in this case (when the value of $L_{\max}$ increases and the number of sections is fixed), the number of vessels and sections meeting constraint 3.3 increases (in our case when $L_{\max}$=6 and the number of section is $M = 3$, constraint 3.3 will be redundant), so the number of options for solving the problem will increase. Therefore, the growth in running time is reasonable. Thus, it is concluded that, with number of sections fixed and length of sections identical, larger $L_{\max}$, hence larger L_m, corresponds to higher computational complexity. Furthermore, larger $L_{\max}$, hence larger L_m, corresponds to higher number of instances with nonzero optimality gap. For instance in Table 4.12, the number of instances with optimal solution are 193 and 181 out of 200 instances, when $L_{\max} = 3$ and $L_{\max} = 6$, respectively.

4.4.3 Sensitivity Analysis of Parameter v_j

To analyze the importance of the length of vessels, the procedure is almost similar to the previous analysis (analysis of parameter L_m). We use two sets of test instances (with 20 scenarios for each set) in which all parameters except the length of vessels are same and fixed for both sets. The values of length of vessels in the first set (set 1) are generated randomly from a uniform distribution of $U(0, 2)$ and the values of second set (set 2) are generated randomly from a uniform distribution of $U(0, 1.5)$. Figure 4.4 and Figure 4.5 show in order the average and the standard deviation of vessel length for each set and each scenario.

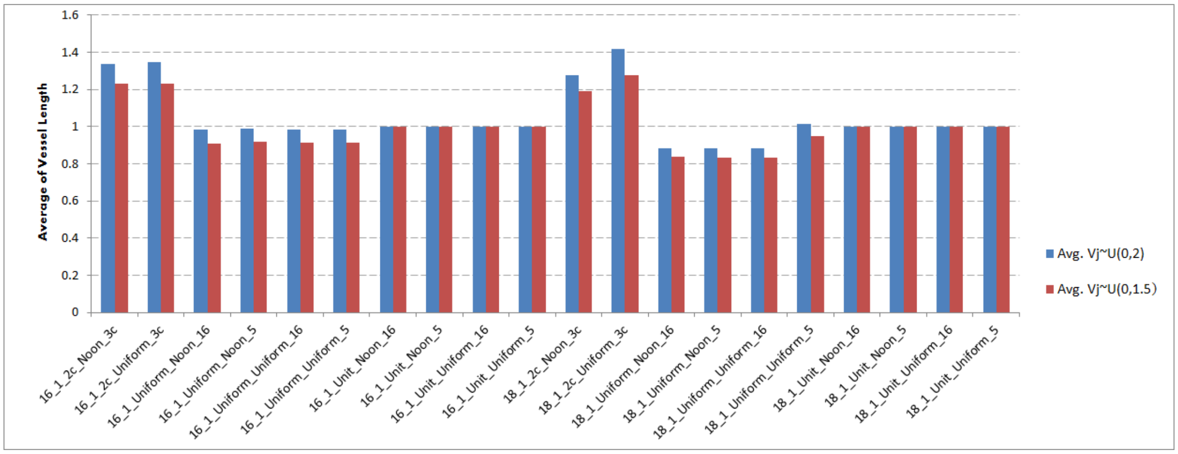

Figure 4.4: The average of vessel length for each set

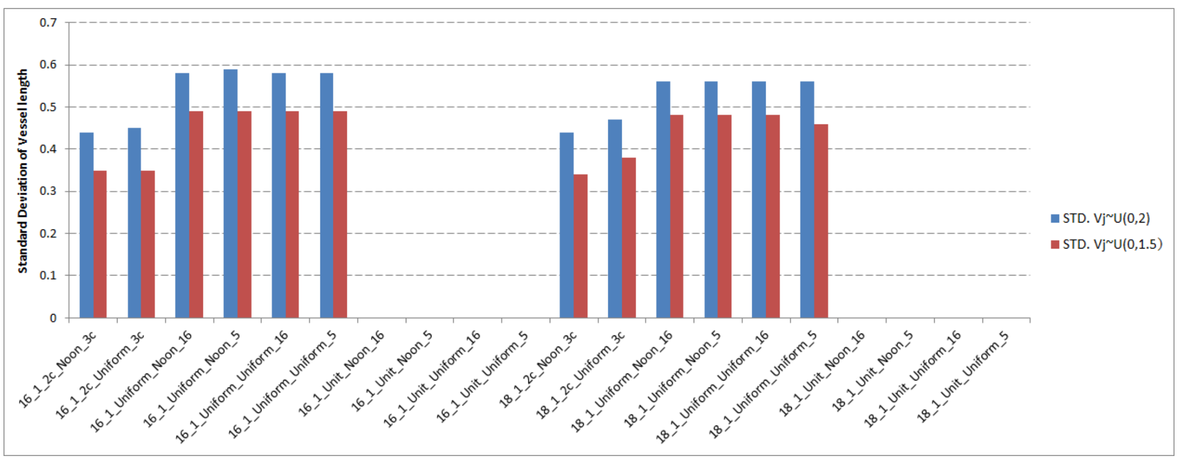

Figure 4.5: The standard deviation of vessel length for each set

According to Table 4.13, we consider the running time of PM model for both sets. To compare the results, in the first sight, it seems that there are no perceptible changes in the running time. To generalize our claim, we construct a statistical hypothesis for running time like:

$$H_0 : \mu_D = 0$$

where H_0 is null hypothesis, μ_D is the average of difference of running times of two sets. We use t-test to decide whether two sets have different running times or not.

In this case, we should use dependent two-sample t-test with equal sample sizes. In general, the formula for dependent two-sample t-test, is like below:

$$\mathrm{t} = \frac{\bar{X_D}}{\frac{S_D}{\sqrt{n}}},$$

where t is the static test, $\bar{X_D}$ is the sample average of difference of running times of two sets, S_D is the statistic variance of difference of running times of two sets, and n is the number of instances for both sets 1 and 2. For t-test, we used $\alpha = 0.05$ as the significance level, which is the most common value used in the literature. According to calculation results presented in Table 4.14, the null hypothesis is accepted.

Having the optimal solutions, to analyze the sensitivity of greedy algorithm 2 to the parameter v_j, we calculate the gap between the final solution obtained from PM model and the solution obtained from greedy algorithm 2 for each instance, in each set:

$$Gap_{(GA2,PM)}\% = (\frac{UB_{GA2} - OPT_{PM}}{OPT_{PM}}) \times 100,$$

where UB_{GA2} is the solution obtained from heuristic algorithm, and OPT_{PM} is the optimal value of PM model. Table 4.15 indicates the results of gap values for two sets, which are different in parameter v_j. Each entry is the average value over 10 instances for each scenario. To evaluate the effect of parameter v_j on the solutions, we compare the gap values with each instance in two different sets one by one. The compared results suggest that the length of vessels, generated by uniform distributions with smaller ranges, results in smaller gaps. To generalize our claim, we construct the below statistical hypothesis:

$$H_0 : \mu_D \geq 0$$

where H_0 is null hypothesis, μ_D is the average of gap of set 1 minus gap of set 2. We use t-test to decide whether set 1 has larger gap values compare to set 2 or not. In this case, we also use dependent two-sample t-test with equal sample sizes (n, here n is 20) with the formula that mentioned above. For t-test, we again used $\alpha = 0.05$

as the significance level. The calculation results (Table 4.16) interpret that the null hypothesis is correct. Hence, set 2 indeed corresponds to smaller gap values. It means vessels with large length, are more critical in allocation, because the number of sections that are able to place them is limited. Figure 4.6 illustrates the comparison between the gap values of two sets. The entries of column "avg" in Table 4.15 are the input data as the gap value for Figure 4.6. The horizontal and vertical axes in Figure 4.6 indicate the scenarios and the gap values (average over 10 instances for each scenario), respectively.

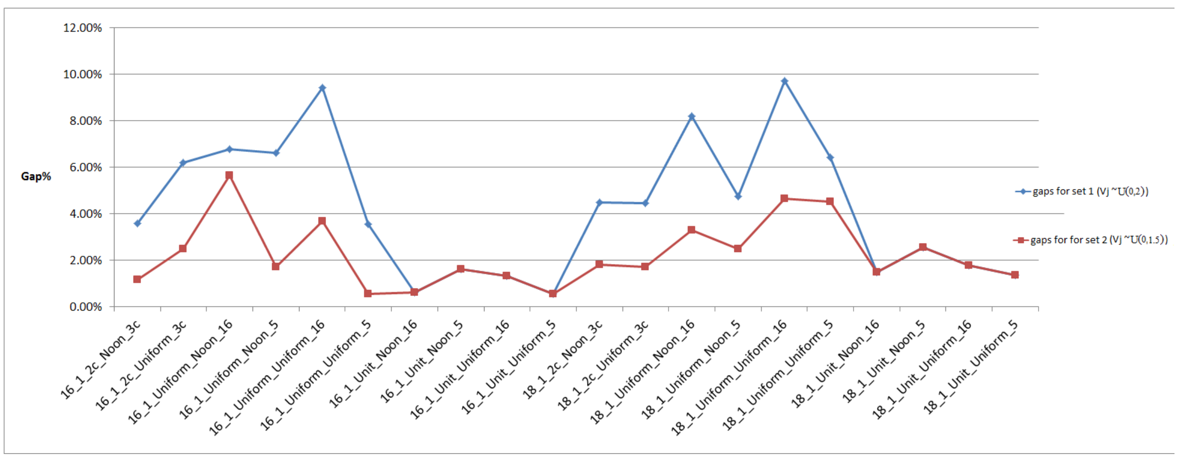

Figure 4.6: The gap between the upper bounds of PM and the solution of greedy algorithm 2 for two sets with different v_j

Table 4.2: CPU time comparison for different combinations of sets of additional constraints over instances in set 1

	avg. time (sec)								
Scenario	PM	PM+ AC1	PM+ AC2	PM+ AC3	PM+ AC1+AC2	PM+ AC1+AC3	PM+ AC2+AC3	PM+ AC	min. time
16_1_2c_Noon_3c	2.85	1.89	2.91	2.38	2.89	**1.76**	2.63	3.28	**1.76**
16_1_2c_Uniform_3c	15.17	12.14	11.35	14.76	12.50	11.93	11.24	**10.55**	**10.55**
16_1_Uniform_Noon_16	85.03	71.22	**66.23**	90.23	69.60	70.00	67.64	67.16	**66.23**
16_1_Uniform_Noon_5	2.26	2.12	1.90	2.01	**1.83**	2.12	1.99	1.95	**1.83**
16_1_Uniform_Uniform_16	37.55	26.37	15.17	35.09	14.77	25.00	15.93	**13.03**	**13.03**
16_1_Uniform_Uniform_5	1.07	1.01	0.99	1.12	**0.91**	1.07	0.99	0.95	**0.91**
16_1_Unit_Noon_16	14.41	13.07	5.04	12.03	5.36	11.68	5.01	**4.60**	**4.60**
16_1_Unit_Noon_5	2.80	3.08	1.45	2.03	1.70	2.08	**1.45**	1.59	**1.45**
16_1_Unit_Uniform_16	4.59	4.07	2.38	4.78	2.40	3.91	2.19	**2.10**	**2.10**
16_1_Unit_Uniform_5	0.85	0.61	**0.44**	0.91	0.95	0.76	0.56	0.75	**0.44**
18_1_2c_Noon_3c	35.88	30.38	27.68	31.46	28.85	30.79	26.98	**26.91**	**26.91**
18_1_2c_Uniform_3c	1159.00	1068.12	1015.68	1100.27	1014.79	1072.21	1012.72	**1008.58**	**1008.58**
18_1_Uniform_Noon_16	1299.87	1201.68	999.00	1301.67	989.20	1195.73	991.00	**963.58**	**963.58**
18_1_Uniform_Noon_5	83.56	85.15	65.99	79.92	66.06	83.48	**65.84**	66.11	**65.84**
18_1_Uniform_Uniform_16	793.13	767.32	772.15	789.23	771.53	762.98	778.42	**761.89**	**761.89**
18_1_Uniform_Uniform_5	115.67	107.45	78.32	113.61	75.88	101.00	75.12	**73.70**	**73.70**
18_1_Unit_Noon_16	2018.45	1827.21	803.43	2004.22	794.90	1830.76	799.23	**778.54**	**778.54**
18_1_Unit_Noon_5	77.21	24.08	**11.90**	60.47	14.90	23.80	12.95	13.52	**11.90**
18_1_Unit_Uniform_16	748.42	727.11	732.18	740.21	726.05	731.08	728.77	**725.15**	**725.15**
18_1_Unit_Uniform_5	1.61	1.57	**1.20**	1.72	1.41	1.83	1.59	1.32	**1.20**
Overall	324.97	298.78	230.72	319.41	229.92	298.20	230.11	**226.26**	**226.26**

Table 4.3: Comparison of optimality gap for different combinations of sets of additional constraints over instances in set 1

	avg. Gap_{opt}								
Scenario	PM	PM+ AC1	PM+ AC2	PM+ AC3	PM+ AC1+AC2	PM+ AC1+AC3	PM+ AC2+AC3	PM+ AC	min. Gap_{opt}
18_1_2c_Uniform_3c	2.07%	1.83%	1.19%	2.04%	0.71%	1.33%	1.04%	**0.64%**	**0.64%**
18_1_Uniform_Noon_16	0.97%	0.96%	**0.95%**	0.97%	**0.95%**	0.96%	**0.95%**	**0.95%**	**0.95%**
18_1_Uniform_Uniform_16	1.95%	1.81%	1.59%	1.90%	1.46%	1.76%	1.54%	**1.34%**	**1.34%**
18_1_Unit_Noon_16	0.16%	0.08%	0.07%	0.13%	0.02%	0.06%	0.03%	**0.00%**	**0.00%**
18_1_Unit_Uniform_16	0.80%	0.78%	0.69%	0.76%	0.63%	0.71%	0.66%	**0.60%**	**0.60%**
Overall	1.19%	1.09%	0.90%	1.16%	0.75%	0.96%	0.84%	**0.71%**	**0.71%**

Table 4.4: Summary of computational results of PM without and with additional constraints over instances in set 1

	PM				PM+ AC			
	time (sec)		Gap_{opt}	number	time (sec)		Gap_{opt}	number
Scenario	avg.	avg. root	avg.	opt. instances	avg	avg. root	avg.	opt. instances
16_1_2c_Noon_3c	2.85	0.44	0.00%	10	3.28	0.46	0.00%	10
16_1_2c_Uniform_3c	15.17	0.45	0.00%	10	10.55	0.44	0.00%	10
16_1_Uniform_Noon_16	85.03	0.54	0.00%	10	67.16	0.51	0.00%	10
16_1_Uniform_Noon_5	2.26	0.53	0.00%	10	1.95	0.51	0.00%	10
16_1_Uniform_Uniform_16	37.55	0.57	0.00%	10	13.03	0.54	0.00%	10
16_1_Uniform_Uniform_5	1.07	0.44	0.00%	10	0.95	0.50	0.00%	10
16_1_Unit_Noon_16	14.41	0.46	0.00%	10	4.60	0.38	0.00%	10
16_1_Unit_Noon_5	2.80	0.42	0.00%	10	1.59	0.39	0.00%	10
16_1_Unit_Uniform_16	4.59	0.47	0.00%	10	2.10	0.37	0.00%	10
16_1_Unit_Uniform_5	0.85	0.38	0.00%	10	0.75	0.30	0.00%	10
18_1_2c_Noon_3c	35.88	0.67	0.00%	10	26.91	0.46	0.00%	10
18_1_2c_Uniform_3c	1159.00	0.52	2.07%	8	1008.58	0.51	0.64%	8
18_1_Uniform_Noon_16	1299.87	0.61	0.97%	8	963.58	0.55	0.95%	9
18_1_Uniform_Noon_5	83.56	0.58	0.00%	10	66.11	0.51	0.00%	10
18_1_Uniform_Uniform_16	793.13	0.57	1.95%	8	761.89	0.50	1.34%	8
18_1_Uniform_Uniform_5	115.67	0.59	0.00%	10	73.70	0.52	0.00%	10
18_1_Unit_Noon_16	2018.45	0.90	0.16%	6	778.54	0.70	0.00%	10
18_1_Unit_Noon_5	77.21	0.87	0.00%	10	13.52	0.50	0.00%	10
18_1_Unit_Uniform_16	748.42	0.93	0.80%	8	725.15	0.53	0.60%	8
18_1_Unit_Uniform_5	1.61	0.75	0.00%	10	1.32	0.49	0.00%	10
Overall	324.97	0.58	0.30%	188	226.26	0.48	0.18%	193

Table 4.5: Summary of computational results of PM without and with additional constraints over instances in set 2

	PM				PM+ AC			
	time (sec)		Gap_{opt}	number	time (sec)		Gap_{opt}	number
Scenario	avg.	avg. root	avg.	opt. instances	avg	avg. root	avg.	opt. instances
16_1_2c_Noon_3c	1.05	0.42	0.00%	10	0.90	0.40	0.00%	10
16_1_2c_Uniform_3c	0.86	0.39	0.00%	10	0.81	0.39	0.00%	10
16_1_Uniform_Noon_16	0.50	0.38	0.00%	10	0.70	0.19	0.00%	10
16_1_Uniform_Noon_5	0.55	0.18	0.00%	10	0.73	0.39	0.00%	10
16_1_Uniform_Uniform_16	0.62	0.40	0.00%	10	0.41	0.19	0.00%	10
16_1_Uniform_Uniform_5	0.49	0.34	0.00%	10	0.30	0.13	0.00%	10
16_1_Unit_Noon_16	0.26	0.22	0.00%	10	0.10	0.07	0.00%	10
16_1_Unit_Noon_5	0.22	0.16	0.00%	10	0.20	0.12	0.00%	10
16_1_Unit_Uniform_16	0.48	0.23	0.00%	10	0.45	0.28	0.00%	10
16_1_Unit_Uniform_5	0.15	0.11	0.00%	10	0.19	0.14	0.00%	10
18_1_2c_Noon_3c	1.50	0.43	0.00%	10	1.34	0.37	0.00%	10
18_1_2c_Uniform_3c	8.60	0.46	0.00%	10	2.20	0.42	0.00%	10
18_1_Uniform_Noon_16	0.29	0.15	0.00%	10	0.53	0.37	0.00%	10
18_1_Uniform_Noon_5	0.54	0.40	0.00%	10	0.19	0.16	0.00%	10
18_1_Uniform_Uniform_16	0.78	0.46	0.00%	10	0.64	0.31	0.00%	10
18_1_Uniform_Uniform_5	0.46	0.31	0.00%	10	0.30	0.16	0.00%	10
18_1_Unit_Noon_16	0.16	0.12	0.00%	10	0.20	0.14	0.00%	10
18_1_Unit_Noon_5	0.15	0.11	0.00%	10	0.13	0.11	0.00%	10
18_1_Unit_Uniform_16	0.26	0.21	0.00%	10	0.17	0.11	0.00%	10
18_1_Unit_Uniform_5	0.22	0.18	0.00%	10	0.18	0.12	0.00%	10
20_1_2c_Noon_3c	365.04	0.87	3.02%	9	327.54	0.49	2.95%	9
20_1_2c_Uniform_3c	381.65	0.73	0.43%	9	92.23	1.94	0.00%	10
20_1_Uniform_Noon_16	2.58	2.00	0.00%	10	1.03	0.42	0.00%	10
20_1_Uniform_Noon_5	1.97	1.77	0.00%	10	0.51	0.35	0.00%	10
20_1_Uniform_Uniform_16	2.16	1.85	0.00%	10	0.76	0.40	0.00%	10
20_1_Uniform_Uniform_5	1.68	1.47	0.00%	10	0.36	0.25	0.00%	10
20_1_Unit_Noon_16	0.77	0.43	0.00%	10	0.34	0.27	0.00%	10
20_1_Unit_Noon_5	1.51	0.80	0.00%	10	0.81	0.37	0.00%	10
20_1_Unit_Uniform_16	0.38	0.35	0.00%	10	0.16	0.15	0.00%	10
20_1_Unit_Uniform_5	0.26	0.24	0.00%	10	0.19	0.16	0.00%	10
overall	25.87	0.54	0.11%	298	14.49	0.31	0.10%	299

Table 4.6: Performance of PM model and continuous models over instances in set 1

Scenario	(A)	(B)	(C)	$Gap_{(PM,C)}$	PM time(sec)	Continuous time(sec)
16_1_2c_Noon_3c	10	10	8	0.15%	3.28	9.079
16_1_2c_Uniform_3c	10	10	7	0.27%	10.55	18.299
16_1_Uniform_Noon_16	10	10	3	3.40%	67.16	79.68
16_1_Uniform_Noon_5	10	10	0	2.81%	1.95	3.590
16_1_Uniform_Uniform_16	10	10	1	4.46%	13.03	23.011
16_1_Uniform_Uniform_5	10	10	2	2.34%	0.95	1.685
16_1_Unit_Noon_16	10	10	10	0.00%	4.60	5.258
16_1_Unit_Noon_5	10	10	10	0.00%	1.59	2.636
16_1_Unit_Uniform_16	10	10	10	0.00%	2.10	4.920
16_1_Unit_Uniform_5	10	10	10	0.00%	0.75	1.419
18_1_2c_Noon_3c	10	10	5	0.38%	26.91	50.670
18_1_2c_Uniform_3c	8	9	7	0.13%	1008.58	1127.940
18_1_Uniform_Noon_16	9	10	0	5.43%	963.58	1090.270
18_1_Uniform_Noon_5	10	10	5	1.32%	66.11	113.340
18_1_Uniform_Uniform_16	8	10	0	3.42%	761.89	816.290
18_1_Uniform_Uniform_5	10	10	1	1.83%	73.70	167.340
18_1_Unit_Noon_16	10	4	7	0.00%	778.54	2765
18_1_Unit_Noon_5	10	10	10	0.00%	13.52	18.544
18_1_Unit_Uniform_16	8	9	9	0.00%	725.15	931.740
18_1_Unit_Uniform_5	10	10	10	0.00%	1.32	1.978
Overall	193	192	115	1.30%	226.26	361.63

Table 4.7: Performance of PM model and continuous models over instances in set 2

Scenario	(A)	(B)	(C)	$Gap_{(PM,C)}$	PM time(sec)	Continuous time(sec)
16_1_2c_Noon_3c	10	10	3	0.66%	0.90	1.45
16_1_2c_Uniform_3c	10	10	5	0.30%	0.81	4.36
16_1_Uniform_Noon_16	10	10	7	0.32%	0.70	1.29
16_1_Uniform_Noon_5	10	10	9	0.15%	0.73	3.04
16_1_Uniform_Uniform_16	10	10	8	0.14%	0.41	1.29
16_1_Uniform_Uniform_5	10	10	9	0.03%	0.30	4.78
16_1_Unit_Noon_16	10	10	10	0.00%	0.10	6.16
16_1_Unit_Noon_5	10	10	10	0.00%	0.20	9.84
16_1_Unit_Uniform_16	10	10	10	0.00%	0.45	3.84
16_1_Unit_Uniform_5	10	10	10	0.00%	0.19	7.44
18_1_2c_Noon_3c	10	10	4	1.05%	1.34	6.62
18_1_2c_Uniform_3c	10	10	1	0.37%	2.20	70.33
18_1_Uniform_Noon_16	10	10	9	0.08%	0.53	1.48
18_1_Uniform_Noon_5	10	10	8	0.11%	0.19	9.48
18_1_Uniform_Uniform_16	10	10	5	0.44%	0.64	1.23
18_1_Uniform_Uniform_5	10	10	9	0.08%	0.30	12.39
18_1_Unit_Noon_16	10	10	10	0.00%	0.20	1.25
18_1_Unit_Noon_5	10	10	10	0.00%	0.13	12.20
18_1_Unit_Uniform_16	10	10	10	0.00%	0.17	9.25
18_1_Unit_Uniform_5	10	10	10	0.00%	0.18	5.23
20_1_2c_Noon_3c	9	9	1	1.36%	327.54	570.84
20_1_2c_Uniform_3c	10	10	3	0.84%	92.23	186.66
20_1_Uniform_Noon_16	10	10	7	0.37%	1.03	3.54
20_1_Uniform_Noon_5	10	10	8	0.09%	0.51	1.14
20_1_Uniform_Uniform_16	10	10	7	0.32%	0.76	9.55
20_1_Uniform_Uniform_5	10	10	10	0.00%	0.36	7.89
20_1_Unit_Noon_16	10	10	10	0.00%	0.34	29.01
20_1_Unit_Noon_5	10	10	10	0.00%	0.81	3.57
20_1_Unit_Uniform_16	10	10	10	0.00%	0.16	9.17
20_1_Unit_Uniform_5	10	10	10	0.00%	0.19	1.14
Overall	299	299	233	0.22%	14.49	33.18

Table 4.8: Time comparison for two sets which are different in parameter L_m with identical L_{max}

	time(sec)			time(sec)	
Scenario	set 1	set 2	Scenario	set 1	set 2
16_1_2c_Noon_3c	0.9048	1.0451	18_1_Uniform_Uniform_5	0.518	0.1559
16_1_2c_Uniform_3c	0.744	0.8597	18_1_Unit_Noon_16	0.5053	0.1653
16_1_Uniform_Noon_16	0.7254	0.499	18_1_Unit_Noon_5	0.3683	0.1685
16_1_Uniform_Noon_5	0.7129	0.5507	18_1_Unit_Uniform_16	0.847	1.9516
16_1_Uniform_Uniform_16	0.6568	0.407	18_1_Unit_Uniform_5	0.7098	0.138
16_1_Uniform_Uniform_5	2.8441	0.2965	20_1_2c_Noon_3c	0.6692	2.1059
16_1_Unit_Noon_16	0.3604	0.0969	20_1_2c_Uniform_3c	0.6676	1.7908
16_1_Unit_Noon_5	0.518	0.2011	20_1_Uniform_Noon_16	0.7301	1.0295
16_1_Unit_Uniform_16	0.4383	0.4836	20_1_Uniform_Noon_5	0.7754	0.691
16_1_Unit_Uniform_5	0.4399	0.145	20_1_Uniform_Uniform_16	0.8658	0.7159
18_1_2c_Noon_3c	1.8391	1.4992	20_1_Uniform_Uniform_5	0.9048	0.7159
18_1_2c_Uniform_3c	0.1824	0.5955	20_1_Unit_Noon_16	0.7316	0.4929
18_1_Uniform_Noon_16	0.4044	0.2946	20_1_Unit_Noon_5	0.6036	0.5256
18_1_Uniform_Noon_5	0.6068	0.1934	20_1_Unit_Uniform_16	0.6083	0.5802
18_1_Uniform_Uniform_16	0.2528	0.1561	20_1_Unit_Uniform_5	0.6489	0.5834
set 1: $M = 5, L_m = (2, 1, 1.2, 0.8, 2)$			set 2: $M = 5, L_m(1.9, 1.9, 1.9, 0.9, 0.4)$		

Table 4.9: Summary results of t-test for the difference of running times of two sets, which are different in parameter L_m with identical L_{max}

Item	Value
H_0	$\mu_D = 0$
α	0.05
n	30
t	0.67
A	$[-t_{\alpha/2,n-1}, t_{\alpha/2,n-1}]$
$[-t_{0.025,29}, t_{0.025,29}]$	$[-2.045, 2.045]$
acceptance region	$t \in A$
rejection region	$t \notin A$

Table 4.10: Results of gap value for two sets, which are different in parameter L_m with identical L_{max}

	set 1 ($M = 5, L_m =$ (2,1,1.2,0.8,2))			set 2 ($M = 5, L_m =$ (1.9,1.9,1.9,0.9,0.4))		
	$Gap_{(GA2,PM)}$			$Gap_{(GA2,PM)}$		
Scenario	avg	min	max	avg	min	max
16_1_2c_Noon_3c	0.45%	0.00%	1.53%	0.32%	0.00%	0.93%
16_1_2c_Uniform_3c	1.77%	0.00%	9.06%	1.53%	0.00%	8.86%
16_1_Uniform_Noon_16	0.74%	0.00%	3.99%	0.46%	0.00%	1.62%
16_1_Uniform_Noon_5	0.87%	0.00%	1.53%	0.81%	0.00%	2.19%
16_1_Uniform_Uniform_16	0.82%	0.00%	2.70%	0.91%	0.00%	5.59%
16_1_Uniform_Uniform_5	0.79%	0.00%	2.54%	0.70%	0.00%	2.85%
16_1_Unit_Noon_16	0.04%	0.00%	0.40%	0.04%	0.00%	0.40%
16_1_Unit_Noon_5	0.59%	0.00%	1.70%	0.59%	0.00%	1.70%
16_1_Unit_Uniform_16	0.60%	0.35%	1.04%	0.60%	0.35%	1.04%
16_1_Unit_Uniform_5	0.43%	0.00%	1.40%	0.43%	0.00%	1.40%
18_1_2c_Noon_3c	0.96%	0.34%	2.59%	0.59%	0.00%	1.30%
18_1_2c_Uniform_3c	1.03%	0.00%	3.06%	0.40%	0.00%	0.92%
18_1_Uniform_Noon_16	0.72%	0.00%	3.01%	0.25%	0.00%	0.76%
18_1_Uniform_Noon_5	0.39%	0.00%	1.45%	0.45%	0.00%	2.58%
18_1_Uniform_Uniform_16	0.94%	0.31%	1.70%	0.75%	0.06%	1.37%
18_1_Uniform_Uniform_5	0.92%	0.00%	2.68%	0.87%	0.06%	2.52%
18_1_Unit_Noon_16	0.17%	0.00%	0.90%	0.17%	0.00%	0.90%
18_1_Unit_Noon_5	1.46%	0.00%	4.82%	1.46%	0.00%	4.82%
18_1_Unit_Uniform_16	1.09%	0.34%	2.44%	1.09%	0.34%	2.44%
18_1_Unit_Uniform_5	0.69%	0.00%	1.55%	0.69%	0.00%	1.55%
20_1_2c_Noon_3c	4.78%	0.00%	37.91%	4.74%	0.00%	44.84%
20_1_2c_Uniform_3c	1.72%	0.00%	5.73%	0.85%	0.00%	2.78%
20_1_Uniform_Noon_16	0.59%	0.00%	1.69%	0.51%	0.00%	1.95%
20_1_Uniform_Noon_5	0.43%	0.00%	1.42%	0.50%	0.00%	1.77%
20_1_Uniform_Uniform_16	1.06%	0.00%	4.64%	0.82%	0.00%	2.88%
20_1_Uniform_Uniform_5	1.85%	0.00%	13.52%	1.83%	0.00%	13.52%
20_1_Unit_Noon_16	1.29%	0.00%	3.75%	1.29%	0.00%	3.75%
20_1_Unit_Noon_5	1.24%	0.39%	2.99%	1.24%	0.39%	2.99%
20_1_Unit_Uniform_16	0.84%	0.00%	2.58%	0.84%	0.00%	2.58%
20_1_Unit_Uniform_5	0.58%	0.00%	1.19%	0.58%	0.00%	1.19%

Table 4.11: Summary results of t-test for the difference of gap values of two sets, which are different in parameter L_m with identical L_{max}

Item	Value
H_0	$\mu_D \geq 0$
α	0.05
n	30
t	0.30
A	$[-t_{\alpha,n-1}, +\infty)$
$[-t_{0.05,29}, +\infty)$	$[-1.699, +\infty)$
acceptance region	$t \in A$
rejection region	$t \notin A$

Table 4.12: Summary of computational results of PM for two sets, which are different in parameter L_m with different$L_{\max}$

	set 1 ($M = 3, L_{max} = 3$)		set 2 ($M = 3, L_{max} = 6$)	
Scenario	time (sec)	Gap_{opt}	time(sec)	Gap_{opt}
16_1_2c_Noon_3c	3.28	0.00%	5.07	0.00%
16_1_2c_Uniform_3c	10.55	0.00%	2.92	0.00%
16_1_Uniform_Noon_16	67.16	0.00%	1138.80	0.40%
16_1_Uniform_Noon_5	1.95	0.00%	19.10	0.00%
16_1_Uniform_Uniform_16	13.03	0.00%	451.79	0.24%
16_1_Uniform_Uniform_5	0.95	0.00%	6.60	0.00%
16_1_Unit_Noon_16	4.60	0.00%	16.64	0.00%
16_1_Unit_Noon_5	1.59	0.00%	3.08	0.00%
16_1_Unit_Uniform_16	2.10	0.00%	4.06	0.00%
16_1_Unit_Uniform_5	0.75	0.00%	1.07	0.00%
18_1_2c_Noon_3c	26.91	0.00%	586.54	0.38%
18_1_2c_Uniform_3c	1008.58	0.64%	739.29	0.45%
18_1_Uniform_Noon_16	963.58	0.95%	2315.74	3.72%
18_1_Uniform_Noon_5	66.11	0.00%	98.98	0.00%
18_1_Uniform_Uniform_16	761.89	1.34%	1935.04	3.50%
18_1_Uniform_Uniform_5	73.70	0.00%	510.03	0.53%
18_1_Unit_Noon_16	778.54	0.00%	796.87	0.00%
18_1_Unit_Noon_5	13.52	0.00%	17.42	0.00%
18_1_Unit_Uniform_16	725.15	0.60%	730.38	0.60%
18_1_Unit_Uniform_5	1.32	0.00%	1.34	0.00%
Overall	226.26	0.18%	469.04	0.49%

Table 4.13: CPU time comparison for two sets which are different in parameter v_j

	time(sec)			time(sec)	
Scenario	set 1	set 2	Scenario	set 1	set 2
16_1_2c_Noon_3c	0.7971	0.808	18_1_2c_Noon_3c	0.3746	0.1559
16_1_2c_Uniform_3c	0.7627	0.808	18_1_2c_Uniform_3c	0.3808	0.1559
16_1_Uniform_Noon_16	0.7222	0.8064	18_1_Uniform_Noon_16	0.4198	0.1575
16_1_Uniform_Noon_5	0.6677	0.8064	18_1_Uniform_Noon_5	0.4884	0.1575
16_1_Uniform_Uniform_16	0.6599	0.8048	18_1_Uniform_Uniform_16	0.4915	0.1575
16_1_Uniform_Uniform_5	0.7176	0.8063	18_1_Uniform_Uniform_5	0.4307	0.1576
16_1_Unit_Noon_16	0.5928	0.4631	18_1_Unit_Noon_16	0.4339	0.2575
16_1_Unit_Noon_5	0.5351	0.4632	18_1_Unit_Noon_5	0.4698	0.2574
16_1_Unit_Uniform_16	0.4806	0.1543	18_1_Unit_Uniform_16	0.465	0.2574
16_1_Unit_Uniform_5	0.4151	0.1543	18_1_Unit_Uniform_5	0.5304	0.2574
set 1: $M = 5, v_j \sim U(0, 2)$			set 2: $M = 5, v_j \sim U(0, 1.5)$		

Table 4.14: Summary results of t-test for the difference of running times of two sets, which are different in parameter v_j

Item	Value
H_0	$\mu_D = 0$
α	0.05
n	20
t	2.08
A	$[-t_{\alpha/2,n-1}, t_{\alpha/2,n-1}]$
$[-t_{0.025,19}, t_{0.025,19}]$	$[-2.093, 2.093]$
acceptance region	$t \in A$
rejection region	$t \notin A$

Table 4.15: Results of gap value for two sets, which are different in parameter v_j

	set 1 ($M=5, v_j \sim U(0,2)$)			set 2 ($M=5, v_j \sim U(0,1.5)$)		
	$Gap_{(GA2,PM)}$			$Gap_{(GA2,PM)}$		
Scenario	avg	min	max	avg	min	max
16_1_2c_Noon_3c	3.58%	1.53%	8.48%	1.17%	0.00%	2.68%
16_1_2c_Uniform_3c	6.22%	0.00%	14.28%	2.50%	0.00%	10.32%
16_1_Uniform_Noon_16	6.77%	2.62%	13.00%	5.66%	0.00%	20.74%
16_1_Uniform_Noon_5	6.62%	2.31%	15.27%	1.71%	0.00%	4.37%
16_1_Uniform_Uniform_16	9.44%	3.22%	16.01%	3.68%	0.00%	8.21%
16_1_Uniform_Uniform_5	3.56%	0.15%	11.34%	0.54%	0.00%	1.62%
16_1_Unit_Noon_16	0.63%	0.00%	1.56%	0.63%	0.00%	1.56%
16_1_Unit_Noon_5	1.61%	0.84%	2.93%	1.61%	0.84%	2.93%
16_1_Unit_Uniform_16	1.32%	0.00%	2.07%	1.32%	0.00%	2.07%
16_1_Unit_Uniform_5	0.55%	0.00%	2.26%	0.55%	0.00%	2.26%
18_1_2c_Noon_3c	4.51%	0.65%	9.41%	1.81%	0.00%	4.58%
18_1_2c_Uniform_3c	4.47%	0.00%	18.05%	1.73%	0.00%	3.35%
18_1_Uniform_Noon_16	8.22%	2.16%	13.35%	3.29%	0.00%	10.10%
18_1_Uniform_Noon_5	4.74%	1.63%	9.08%	2.49%	0.49%	5.76%
18_1_Uniform_Uniform_16	9.71%	1.48%	20.04%	4.64%	1.89%	9.85%
18_1_Uniform_Uniform_5	6.42%	1.91%	10.79%	4.52%	1.07%	10.51%
18_1_Unit_Noon_16	1.50%	0.33%	2.90%	1.50%	0.33%	2.90%
18_1_Unit_Noon_5	2.56%	0.18%	5.33%	2.56%	0.18%	5.33%
18_1_Unit_Uniform_16	1.79%	0.04%	4.24%	1.79%	0.04%	4.24%
18_1_Unit_Uniform_5	1.36%	0.00%	3.22%	1.36%	0.00%	3.22%

Table 4.16: Summary results of t-test for the difference of gap values of two sets, which are different in parameter v_j

Item	Value
H_0	$\mu_D \geq 0$
α	0.05
n	20
t	2.19
A	$[-t_{\alpha,n-1}, +\infty)$
$[-t_{0.05,19}, +\infty)$	$[-1.729, +\infty)$
acceptance region	$t \in A$
rejection region	$t \notin A$

Chapter 5

SUMMARY AND FUTURE RESEARCH

5.1 Summary

This research provided a comprehensive analysis of berth planning problems in dry bulk terminals. This study focuses on the partitioned version of Berth Allocation Problem (BAP). We present a mixed integer linear programming model to minimize the sum of arriving vessels' completion times. The model, which is based on sequence-variables, reflects allocating vessels to berth positions subject to tidal constraints. We have also developed three additional constraints and add them the model to make it computationally more tractable.

To better understand the performance of models, we test the model on instances generated based on the real data of a dry bulk terminal. The mathematical models are solved by IBM ILOG CPLEX 12.5.1. Furthermore, we develop two heuristic algorithms that enable us to obtain near optimal solutions to the problem within a short computational time for large-size instances.

To reflect the importance of some parameters in the proposed model, we applied sensitivity analysis. We also compared the solutions obtained from the partitioned model with the solutions obtained from the continuous model proposed in [30]. All parameters used in the models are the same and fixed. According to the results of this comparison, we concluded that both models are strong, effective and applicable depending on the planning horizon.

5.2 Future Research

Despite the accomplishments, this study is subject to limitations like any other research. Some suggestions for future work include the following:

In this research it is assumed that the lengths of partitioned sections are static and fixed at the beginning of the time period. However, they can be dynamic with respect to the length of vessels assigned to them. Thus, future research may consider this factor and build a more practical model for berth operations.

Berth allocation problems are dynamic in real world. Many unpredictable causes may lead to changes in operating conditions. For example, a vessel may arrive earlier or later than its expected arrival time due to an unexpected change in its departure time from the previous port. Changes may also occur during the service time of vessels. Hence, designing a dynamic algorithm for solving the BAP is necessary. This research considers a static BAP in which all input data are generated as constant parameters. For future research, operational information, such as vessels' arrival times or vessels' handling times may not be known ahead of time and hence may change dynamically over time. The model can also be extended to multi-objective cases.

In this study we design two simple heuristic algorithms. They are shown to be efficient and helpful to get near optimal solutions as the upper bound values for the proposed model. For the future study, more effective heuristics can be developed to produce a solution in a reasonable time frame for solving the problem. In particular, adopting some meta-heuristics (e.g., genetic algorithm) for the problem can be also of interest, especially for solving large-size instances.

BIBLIOGRAPHY

[1] K. Buhrkal, S. Zuglian, S. Ropke, J. Larsen, and R. Lusby, "Models for the discrete berth allocation problem: a computational comparison," Transportation Research Part E: Logistics and Transportation Review **47**, 461–473 (2011).

[2] J. Hiltermann, G. Lodewijks, J. C. Rijsenbrij, D. L. Schott, J. A. Dekkers, Y. Pang *et al.*, "Reducing the power consumption of troughed belt conveyors by speed control," (2009).

[3] K. Lai and K. Shih, "A study of container berth allocation," Journal of advanced transportation **26**, 45–60 (1992).

[4] A. Imai, K. Nagaiwa, and C. W. Tat, "Efficient planning of berth allocation for container terminals in asia," Journal of Advanced Transportation **31**, 75–94 (1997).

[5] A. Imai, E. Nishimura, and S. Papadimitriou, "The dynamic berth allocation problem for a container port," Transportation Research Part B: Methodological **35**, 401–417 (2001).

[6] A. Imai, E. Nishimura, and S. Papadimitriou, "Corrigendum to" the dynamic berth allocation problem for a container port"[transportation research part b 35 (2001) 401-417]," Transportation Research Part B: Methodological **39**, 197 (2005).

[7] A. Imai, E. Nishimura, and S. Papadimitriou, "Berth allocation with service priority," Transportation Research Part B: Methodological **37**, 437–457 (2003).

[8] M. F. Monaco and M. Sammarra, "The berth allocation problem: a strong formulation solved by a lagrangean approach," Transportation Science **41**, 265–280 (2007).

[9] P. Hansen, C. Oğuz, and N. Mladenović, "Variable neighborhood search for minimum cost berth allocation," European Journal of Operational Research **191**, 636–649 (2008).

[10] J.-F. Cordeau, G. Laporte, P. Legato, and L. Moccia, "Models and tabu search heuristics for the berth-allocation problem," Transportation science **39**, 526–538 (2005).

[11] D. Xu, C.-L. Li, and J. Y.-T. Leung, "Berth allocation with time-dependent physical limitations on vessels," European Journal of Operational Research **216**, 47–56 (2012).

[12] L. L. Lorenzoni, H. Ahonen, and A. G. de Alvarenga, "A multi-mode resource-constrained scheduling problem in the context of port operations," Computers & Industrial Engineering **50**, 55–65 (2006).

[13] V. H. Barros, T. S. Costa, A. C. Oliveira, and L. A. Lorena, "Model and heuristic for berth allocation in tidal bulk ports with stock level constraints," Computers & Industrial Engineering **60**, 606–613 (2011).

[14] K. H. Kim and K. C. Moon, "Berth scheduling by simulated annealing," Transportation Research Part B: Methodological **37**, 541–560 (2003).

[15] C.-l. Li, X. Cai, and C.-y. Lee, "Scheduling with multiple-job-on-one-processor pattern," IIE transactions **30**, 433–445 (1998).

[16] Y. Guan and R. K. Cheung, "The berth allocation problem: models and solution methods," Or Spectrum **26**, 75–92 (2004).

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

[18] C. Cheong, K. Tan, D. Liu, and C. Lin, "Multi-objective and prioritized berth allocation in container ports," Annals of Operations Research **180**, 63–103 (2010).

[19] R. Moorthy and C.-P. Teo, "Berth management in container terminal: the template design problem," OR spectrum **28**, 495–518 (2006).

[20] A. Imai, J.-T. Zhang, E. Nishimura, and S. Papadimitriou, "The berth allocation problem with service time and delay time objectives," Maritime Economics & Logistics **9**, 269–290 (2007).

[21] A. Imai, E. Nishimura, M. Hattori, and S. Papadimitriou, "Berth allocation at indented berths for mega-containerships," European Journal of Operational Research **179**, 579–593 (2007).

[22] N. Umang, M. Bierlaire, and I. Vacca, "Exact and heuristic methods to solve the berth allocation problem in bulk ports," Transportation Research Part E: Logistics and Transportation Review **54**, 14–31 (2013).

[23] Y.-M. Park and K. H. Kim, "A scheduling method for berth and quay cranes," in "Container Terminals and Automated Transport Systems," (OR Spectrum, 2003), pp. 1–23.

[24] J.-F. Cordeau, M. Gaudioso, G. Laporte, and L. Moccia, "The service allocation problem at the gioia tauro maritime terminal," European Journal of Operational Research **176**, 1167–1184 (2007).

[25] C. Liang, Y. Huang, and Y. Yang, "A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning," Computers & Industrial Engineering **56**, 1021–1028 (2009).

[26] C. Liang, L. Lin, and J. Jo, "Multiobjective hybrid genetic algorithm for quay crane scheduling in berth allocation planning," International Journal of Manufacturing Technology and Management **16**, 127–146 (2009).

[27] F. Meisel and C. Bierwirth, “Heuristics for the integration of crane productivity in the berth allocation problem,” Transportation Research Part E: Logistics and Transportation Review **45**, 196–209 (2009).

[28] J. Blazewicz, T. E. Cheng, M. Machowiak, and C. Oguz, “Berth and quay crane allocation: a moldable task scheduling model,” Journal of the Operational Research Society **62**, 1189–1197 (2011).

[29] T. Robenek, N. Umang, M. Bierlaire, and S. Ropke, “A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports,” European Journal of Operational Research **235**, 399–411 (2014).

[30] A. T. Ernst, C. Oguz, G. Singh, and G. Taherkhani, “Mathematical models for the berth allocation problem in dry bulk terminal,” under review for Journal of Scheduling (2015).