INTEGRATION OF BERTH ALLOCATION AND STORAGE ALLOCATION PROBLEMS

by

Özgür Emre Sivrikaya

B.S., Industrial Engineering, Boğaziçi University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering

Boğaziçi University

2013

ACKNOWLEDGEMENTS

I wish to thank, first and foremost, my thesis supervisor Prof. Necati Aras for his valuable assistance in the preparation and completion of this study. His endless support and assistance help me to hurdle all the obstacles in the completion of this research work.

I am indebted to Assist. Prof. Zeki Caner Taşkın and Assist. Prof. Yavuz Boğaç Türkoğulları for their willingness to take part on my thesis committee and their valuable suggestions.

I am thankful to my friends Ahu Akdemir, Ayberk Çalık, Ali Burak Aktekin and Özdemir Küğcümen for their endless support and help.

Last but not least, I would like to thank to my parents and my sister for their never ending love and support.

ABSTRACT

INTEGRATION OF BERTH ALLOCATION AND STORAGE ALLOCATION PROBLEMS

Berth allocation and storage allocation problems deal with two very important steps in the optimization of the container terminal operations. Berth allocation problem concentrates on the allocation of berth segments to the vessels. The objective is to find the proper berthing time and position for a vessel while minimizing the penalty cost that incurs if the service of the vessel cannot be completed until its due time. Storage allocation problem allocates storage locations to containers considering the availability of the storage yard. The objective minimizes the total transportation distance of the containers and the number of reshuffling which can be defined as relocating a container to remove another one from the storage yard. As the berthing position of a vessel and the storage location of a container, which arrives or departs with that vessel, are related with each other, they have to be considered as an integrated problem. However, the integrated problem is an NP-hard problem and heuristic approaches are needed to solve the larger problem instances. This study details the development of an integrated mathematical model for the berth allocation and the storage allocation problems and offers different solution approaches for the larger real-life instances. The performances of the different solution methods for individual containers, in terms of the objective values and solution times, are compared. Grouping containers, its advantages and disadvantages on individual containers are also examined in the scope of this thesis.

ÖZET

BİRLEŞİK RIHTIM ATAMA VE KONTEYNER DEPOLAMA PROBLEMİ

Rıhtım atama problemi ve konteyner depolama problemi, konteyner terminal operasyonları eniyilemesinin iki önemli adımıdır. Rıhtım atama problemi, gemilerin yanaşacağı rıhtım bölümlerini belirler. Amaç, gemilere uygun bir rıhtım bölümü ve yanaşma zamanı belirlerken, gemilerin önceden belirlenmiş zamandan sonra rıhtımdan ayrılması durumunda oluşan cezayı enküçüklemektir. Konteyner depolama problemi, depo sahasının uygunluğunu gözeterek, konteynerelere uygun bir depolama alanı belirler. Amaç fonksiyonu, konteynerlerin toplam taşınma mesafesi ve elleçleme sayısını enküçükler. Elleçleme, bir konteynerin, altındaki diğer konteyner depo sahasından alınırken yerinden kaldırılmasıdır. Geminin yanaşacağı rıhtım bölümü ve konteynerlerin depolanacağı depolama alanı birbiriyle yakından ilişkili olduğu için, bu problemler birlikte ele alınmalıdır. Ancak birleşik problem NP-zor olduğu için, büyük problem örneklerinin çözümünde sezgisel yöntemlere ihtiyaç duyulmaktadır. Bu çalışmada, birleşik rıhtım atama ve konteyner depolama problemi için matematiksel model geliştirilmiş, büyük problemlerin çözümü için ise farklı çözüm yöntemleri önerilmiştir. Konteynerler için geliştirilen farklı çözüm yöntemlerinin performansları, amaç fonksiyonu değerleri ve problem çözüm süreleri göz önünde bulundurularak değerlendirilmiştir. Konteyner gruplama, avantajları ve dezavantajları da tez kapsamında çalışılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

B	Set of berth segments
c_{p1}^{v}	Penalty cost which will be paid when the service of vessel v cannot be completed before due time
c_{p2}^{v}	Penalty cost which will be paid when vessel v cannot berth upon arrival
c_s	Average reshuffling cost
c_t	Average transportation cost
d_{bk}	Distance between berth segment b and storage location k
E_{vqu}	One if vessel v covers time period-berth segment block (q, u), zero otherwise
h_{kt}	Maximum number of containers that can be stored at storage location k at time period t
I	Set of containers
J	Set of containers
K	Set of storage locations
l_v	Length of vessel v in terms of number of berth segments required
M	Big M
N_{ibk}	One if storage location k is allocated to container i which departs from berth segment b, zero otherwise
P_v^+	Positive deviation of berthing time of vessel v from arrival time
Q	Set of berth segments
r_{iv}^{a}	One if container i arrives within vessel v, zero otherwise
r_{iv}^{d}	One if container i departs within vessel v, zero otherwise
S_{ij}	One if container j needs to be reshuffled when container i is removed, zero otherwise
S_v^-	Positive deviation of departure time of vessel v from due time
S_v^+	Positive deviation of departure time of vessel v from due time
T	Set of time periods

t_a^v	Expected arrival time of vessel v
t_c	Average container handling time
t_d^v	Due time of vessel v
t_p^v	Processing time of vessel v
U	Set of time periods
V	Set of vessels
X_{ibk}	One if storage location k is allocated to container i which arrives at berth segment b, zero otherwise
Y_{btv}	One if vessel v berths at segment b at time period t, zero otherwise
Z_{ikt}	One if container i is in storage location k at time period t, zero otherwise
α_i	Arrival time of container i
β_i	Departure time of container i
γ_i	Number of containers in container group i
η	Maximum number of containers that can be stored at a storage location
$\Theta_{vv'}$	One if vessel v has to wait vessel v', zero otherwise

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
BAP	Berth Allocation Problem
CLM	Container Location Model
CS	Clustering Search
DBAP	Dynamic Berth Allocation Problem
ETA	Expected Time of Arrival
GA	Genetic Algorithm
HS	Harmony Search
MCBAP	Minimum Cost Berth Allocation Problem
MDVRPTW	Multi-Depot Vehicle Routing Problem with Time Windows
QCAP	Quay Crane Assignment Problem
QCSP	Quay Crane Scheduling Problem
SAP	Storage Allocation Problem
SBAP	Static Berth Allocation Problem

1. INTRODUCTION

Maritime transportation plays a vital role in international logistics due to its certain advantages over other transportation modes and containers are the main transportation tool in this mode of transportation. Containers are solid boxes which enables products to be transported and stacked in a cheaper and more secure way. Due to their global standards, containers ease transportation with the possibility of usage of the certain transfer vehicles and methods. Since containers are mostly made from steel, it is possible to use each container for several times and in several locations. Today, as noted by Lee *et al*. [1] usage of maritime transportation and containers are increasing day by day.

A container terminal is a transfer node between land and sea. Container terminals consist of quayside and storage yard. After a ship berths, quay cranes transfer containers from the ship to trucks which will carry them to storage yard. In storage yards, yard cranes remove cargo from the trucks and place them in the stacking area. As containerization is increasing, container movements in the terminals are getting more and more complex, which triggers a very important problem, i.e., managing container operations within the terminal.

Quayside and storage yard operations face different problems. To manage quayside operations efficiently we need to solve Berth Allocation Problem, Quay Crane Allocation Problem and Quay Crane Scheduling Problem. Storage yard operations can be handled efficiently by solving Truck Allocation Problem, Truck Scheduling Problem, Yard Crane Allocation Problem, Yard Crane Scheduling Problem and Storage Allocation Problem. There are studies on these problems but most of the studies concentrate on the above-mentioned problems separately because of the complicated nature of the problems.

Determining the berthing position of a vessel is the first step in the optimization of container terminal operations and Berth Allocation Problem (BAP) deals with the allocation of berth spaces. There are different BAP formulations in the literature. However, the main goal is to find a suitable berthing position for a vessel as soon as it arrives. Storage Allocation Problem (SAP) allocates storage spaces to containers considering the availability of the storage yard. Previous studies on SAP consider containers as batches instead of unique items as there are too many containers in a container yard to be dealt with. However, this consideration generates suboptimality as batches are treated like unique items and cannot be stored separately.

Independent of different formulations and assumptions, berthing position of a vessel and storage position of a container are related with each other and have to be considered as an integrated problem. Unfortunately, due to their complex nature, it is not possible to find an integrated solution easily. This study focuses on modeling these two problems in an integrated way. However, we can only obtain exact solutions for small problem sizes. As problem size increases, other solution methods should be used to obtain a solution in reasonable time.

Expected arrival times (ETA) of vessels are known up to fifteen days prior to their arrival, but these times are subject to change especially for container terminals built near inland seas like Marmara. Exact information on ETA for this kind of terminals is received when the arriving vessel leaves the previous port. Therefore, planning should be completed until the vessel arrives, which is not more than ten hours. We also determine our solution time as ten hours for that fact.

The rest of this paper is organized as follows. Chapter 2 presents a brief literature review on optimization problems in container terminals. Model assumptions, mathematical model and two solution methods are provided in Chapter 3. Chapter 4 will cover the computational results and three new solution approaches. Some important conclusions and possible future research suggestions are presented in Chapter 5.

2. LITERATURE REVIEW

Steenken *et al.* [2] give a comprehensive literature review about container terminal operations. Different problems in container terminals as well as main concepts (equipments, terminal systems etc.) of container transportation are also described in that study. Although a very detailed list of studies from an operational research viewpoint is presented, only a few of them offer solutions to integrated problems. Stahlbock and Voß [3] updated Steenken *et al.* [2] by mostly focusing on the researches published after Steenken *et al.* [2]. Unfortunately, most of the studies still do not focus on problems in an integrated way. Although there are a few studies in the literature review suggesting integrated solutions using simulation approaches, none of them focuses on the integration of BAP and SAP.

2.1. Berth Allocation Problem

BAP can be modeled for either continuous or discrete cases. In discrete case, berth is partitioned into a finite number of berth segments. Bierwirth and Meisel [4] give a detailed literature survey about planning the seaside operations. Since BAP is one of the main problems at container terminals, it is possible to find several studies on this subject including both discrete and continuous cases. There are also papers on the integration of BAP with other problems such as Quay Crane Assignment Problem (QCAP). However, integration of BAP and SAP cannot be found among the proposed studies.

A formulation for Dynamic Berth Allocation Problem (DBAP) and a lagrangean relaxation for this formulation is introduced by Imai *et al.* [5]. In contrast to their previous study in which they introduce Static Berth Allocation Problem (SBAP) which does not allow new vessel arrivals through planning horizon, DBAP formulation allows new vessel arrivals while the service is in progress. SBAP is inefficient as it requires all vessels have to be in the port and new vessel arrivals through the planning horizon have to be planned in the subsequent planning horizon.

Cordeau *et al.* [6] offer two mathematical models for the discrete case. The first formulation is the DBAP. In the second model, they consider berth segments as depots and vessels as customers and formulate the problem as a Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW). They can find optimal solutions for small size problems. However, for more realistic cases they need heuristic approach. Therefore, they develop a tabu search algorithm for the second formulation. Then, they extend their study by developing another tabu search algorithm for the continuous case of BAP. A column generation method for the MDVRPTW is suggested by Mauri *et al.* [7]. Their algorithm outperforms the tabu search offered by Cordeau *et al.* [6]. Oliveira *et al.* [8] introduce a new solution method of MDVRPTW. They propose a Clustering Search (CS) algorithm as an alternative solution method for the two studies Cordeau *et al.* [6] and Mauri *et al.* [7]. CS gives good solutions in reasonable time when compared to the previous studies.

Buhrkal *et al.* [9] discuss five different DBAP formulations and compare their computational performances. Two of these formulations are proposed by Imai *et al.* [10] and Cordeau *et al.* [6] and other two formulations are created by modifying these two models. The last formulation is defining DBAP as a generalized set partitioning problem. Buhrkal *et al.* [9] notes that the last formulation outperforms the others in terms of solution time.

Guan and Cheung [11] discuss two different problem representations, relative position and position assignment formulations. In the first formulation, relative positions of vessels on time period-berth segment block are determined whereas in the second formulation space covered on time period-berth segment by vessels are determined. We also use the second representation in our mathematical model.

Lee and Chen [12] propose a heuristic for the BAP. In the mathematical model, vessel's berthing positions are determined with respect to their preferred positions.

Each vessel determines its preferred position, which is considerably larger than their length. The model selects first the berth section among the preferred berth sections and then determines the exact berthing position within the borders of the selected berth section. Therefore the model can be considered as both a continuous and discrete case of BAP. Repositioning of vessels during service is also allowed.

Hansen *et al*. [13] introduce Minimum Cost Berth Allocation Problem (MCBAP). MCBAP minimizes total waiting and handling costs. Handling costs of vessels and rewarding earliness as well as penalizing lateness are also considered.

2.2. Storage Allocation Problem

Studies on container terminal operations generally consider problems separately. A genetic algorithm in order to solve a real-life storage allocation problem is developed by Bazzazi *et al*. [14]. Since most of the container traffic is due to import and export containers in the terminal studied on, only these kinds of containers are considered. The objective functions of the problem minimizes imbalance between workloads allocated to blocks.

Lee *et al*. [1] present two heuristics for SAP, a sequential method and a column generation method. To reduce congestion and reshuffling in the container terminal a consignment strategy is adopted where at each block only one vessel's containers can be stored. The study concentrates on minimizing the total number of yard cranes in each shift.

Han *et al*. [15] updates their previous study Lee *et al*. [1]. In the previous study, possible blocks are preassigned to certain vessels and storage blocks are allocated to vessels among these preassigned blocks. This study presents iterative improvement to solve SAP and determines storage blocks without considering any preassignments. The objective of the formulation is again minimizing the total number of yard cranes and the consignment strategy remains unchanged.

Moccia and Astorino [16] present a mathematical model for SAP considering containers as groups. The study considers reallocation of container groups through the planning horizon with respect to predetermined berthing plans of vessels. The objective function of the mathematical model is the minimization of transportation and handling costs of the container groups.

The mathematical model offered by Preston and Kozan [17] minimizes the maximum handling time of the containers. Both transportation times and set up times, which are defined as the times required to remove the containers from the stacks, constitue handling times of the containers. A genetic algorithm (GA) for the container location model (CLM) is also developed in the study.

Ayachi *et al.* [18] develop a harmony search (HS) algorithm for the storage allocation problem which minimizes total reshuffling in a storage yard. Unlike the previous study Ayachi *et al.* [19], where an HS algorithm is proposed for identical containers, different types of containers (reefer, open top etc.) of the same length are considered.

Although we deal with container storage, there are other materials to be stored at a seaside storage yard. Cordeau *et al.* [20] present a mathematical model for the SAP. Their problem deals with cars instead of containers. This study has common properties with our mathematical model. For example, positions of cars do not change during the storage period. Cars are grouped with respect to their properties and storage locations are allocated to these groups. The objective function of the model minimizes total handling time whereas we concentrate on minimizing transportation and reshuffling costs. Since cars cannot be stacked like containers, there is no reshuffling cost to be considered in the mathematical model.

2.3. Integrated Problems

Although SAP is one of the most important problems in container terminal operations, there is not enough study on integration of SAP with other problems. To the best of our knowledge, none of the studies focus on an integrated solution of BAP and SAP. Lee *et al.* [21] is the closest study to ours. The mathematical model determines the berthing container terminals for the vessels and the storage locations for the container groups. However, it does not concentrate on the exact berthing point for the vessels. Several container terminals are dealt with and by minimizing transportation cost of container groups the arrival and departure terminals for the container groups are determined. Reallocation of containers within or between the terminals through the storage duration is also allowed.

Lee *et al.* [22] propose a hybrid insertion algorithm for the solution of integrated yard truck scheduling and storage allocation problems by minimizing weighted sum of the cost of total travel time and total delay.

Cao *et al.* [23] study truck scheduling and storage allocation problems although only departing containers are considered. Mathematical model minimizes the makespan. Due to the NP-hardness of the problem a genetic algorithm and a greedy heuristic are proposed for the problem. Greedy heuristic outperforms the genetic algorithm on the basis of experiments carried out.

Meisel and Bierwirth [24] propose an integrated model of berth allocation and crane allocation problems. In contrast with previous researches which propose mathematical models with some unrealistic assumptions (e.g. fixed processing times of vessel), integrated problem is modeled with more realistic assumptions. Two heuristics are proposed to solve the model, which can obtain a berthing plan for the problem instances of size 20, 30 and 40 vessels.

3. MATHEMATICAL MODEL

In this chapter, we first will describe our mathematical model. Then we will discuss how the integrated model is divided into two separate models and solved sequentially. Finally, we introduce how we use rolling horizon procedure to solve our integrated model. In Chapter 4 we compare the performances of different solution methods.

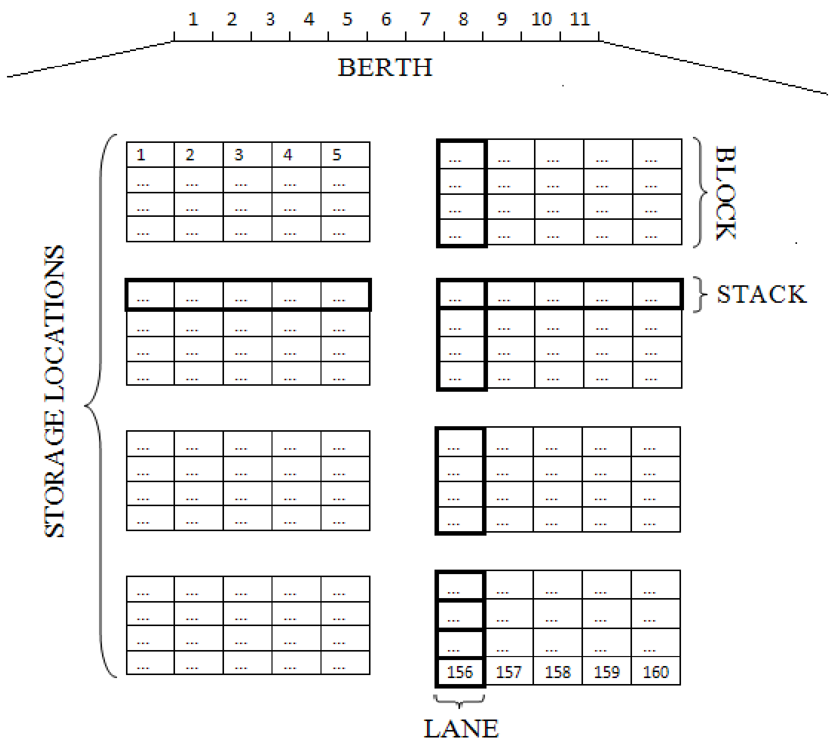

Figure 3.1. Layout of a container terminal.

A typical layout of a container terminal can be seen in Figure 3.1. Berths are divided into equal size berth segments and each vessel should be assigned to some of these berth segments. This assignment indicates that the vessel occupies the berth starting from that berth segment. Therefore, we should convert vessel lengths to the number of berth segments required to hold the vessel. For example, when a vessel of length 4 arrives to the port and berth segment 3 is assigned to this vessel, this vessel occupies the berth segments 3 to 6. Although we require a certain length for a berth segment, higher lengths (e.g. more than 5 20-ft container length, i.e., about 30 meters) may result inefficient usage of the berth.

There are storage locations in the storage area. A typical storage location (numbered from 1 to 160) has a stacking height of a 5-8 containers, which is called tier. A block is a group of containers that are stored side by side. A line is a group of blocks in the same direction. Each line has stacks and lanes. Usually the vertical group is called lane and the horizontal group is called stack.

In our problems, we will use the following sets:

- I, J : set of containers
- B, Q : set of berth segments
- K : set of storage locations
- T, U : set of time periods
- V : set of vessels

There will be two types of events that create a penalty cost. Although we know the expected arrival time of a vessel, it may not berth as soon as it arrives since there may not be enough berth segments available. In such a case, there will be a penalty cost proportional to the deviation from the service start time. Penalty cost also incurs if service of the vessel cannot be completed before the due time. We assume that the average container handling time, which is the time required to complete an operation for a container (i.e. loading and unloading times of a container) is known. The average container time is used to calculate the processing time (service completion time) of a vessel and together with the expected arrival time is used to calculate due time of the vessel.

In addition to penalty cost, there are two more costs: average reshuffling and transportation costs. Average reshuffling cost is the cost of removing the cargo from the truck and placing it to the storage location or moving the cargo from one storage location to another. This cost is due to vertical movement of the cargo. On the other hand, average transportation cost is due to horizontal movement of the container.

We know which containers arrive and depart within each vessel. Therefore, our mathematical model determines berthing point of vessels and storage locations of containers using this information. Finally, rectilinear distance between storage locations and berth segments is used as a parameter. All parameters used in the mathematical model are given below:

- c_s : reshuffling cost per container movement
- c_t : transportation cost per unit distance
- $c_{p_1}^v$: penalty cost that incurs if the service of vessel v cannot be completed before its due time
- $c_{p_2}^v$: penalty cost that incurs if vessel v cannot berth upon arrival
- l_v : length of vessel v in terms of number of berth segments
- t_c : average container handling time
- t_a^v : expected arrival time of vessel v
- $r_{iv}^a : \begin{cases} 1, & \text{if container } \ i \ \text{ arrives with vessel } \ v \\ 0, & \text{otherwise} \end{cases}$
- $r_{iv}^d : \begin{cases} 1, & \text{if container } \ i \ \text{ departs with vessel } \ v \\ 0, & \text{otherwise} \end{cases}$
- d_{bk} : rectilinear distance between berth segment b and storage location k
- t_p^v : processing time of vessel v $\left(t_c \left(\sum_{i\in I} r_{iv}^a \sum_{i\in I} r_{iv}^d\right)\right)$
- t_d^v : due time of vessel v $(t_d^v = t_a^v + t_p^v)$
- h_{kt} : max. number of containers that can be stored at storage location k in time period t
- M : big M

Since we determine both berthing position of a vessel and storage locations of the containers, we have vessel related decision variables and container related decision variables.

Consider a chart where the vertical axis represents berth segments whereas the horizontal axis represents time periods. The chart is divided into B*T blocks and each vessel should be assigned to some of the time period-berth segment blocks. We have the following decision variables for the vessels:

- $E_{vqu} = \begin{cases} 1, & \text{if vessel } v \text{ covers time period-berth segment block } (q,u) \\ 0, & \text{otherwise} \end{cases}$
- $Y_{btv} = \begin{cases} 1, & \text{if vessel } v \text{ berths at segment } b \text{ in time period } t \\ 0, & \text{otherwise} \end{cases}$

A simple representation of the berth allocation for vessels can be seen in Figure 3.2. For example, V1 berths at segment 5 in time period 1 and covers segments $5-10$ in time period 1. Hence, $Y_{5,1,1} = 1$ and $E_{1,5,1} = 1$.

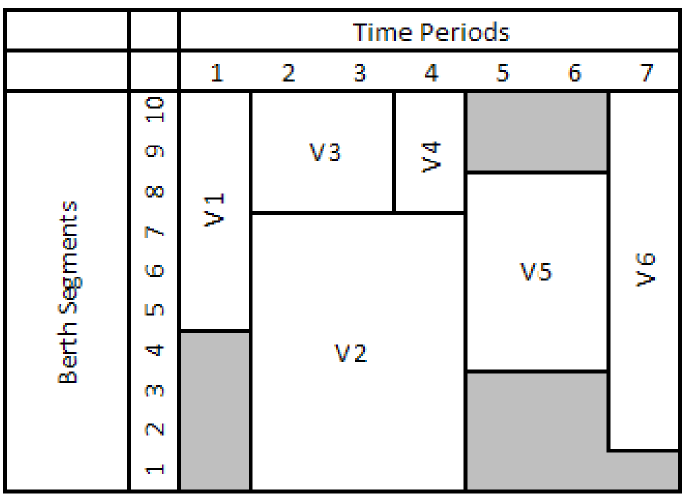

Figure 3.2. A simple representation of the vessel allocation.

We define two decision variables for the storage locations of the containers:

- $X_{ibk} =$
$$\begin{cases} 1, & \text{if storage location } k \text{ is allocated to container } i \text{ which arrives at segment } b \\ 0, & \text{otherwise} \end{cases}$$
- $N_{ibk} =$
$$\begin{cases} 1, & \text{if storage location } k \text{ is allocated to container } i \text{ which departs from segment } b \\ 0, & \text{otherwise} \end{cases}$$

Other decision variables defined in the mathematical model are as follows:

- $S_{ij} = \begin{cases} 1, & \text{if container } j \text{ needs to be reshuffled when container } i \text{ is removed} \\ 0, & \text{otherwise} \end{cases}$
- $Z_{ikt} = \begin{cases} 1, & \text{if container } i \text{ is in storage location } k \text{ in time period } t \\ 0, & \text{otherwise} \end{cases}$
- α_i = arrival time of container i to the berth
- β_i = departure time of container i from the berth
- S_v^+, S_v^- = deviations of departure time of vessel v from due time
- P_v^+ = deviations of berthing time of vessel v from arrival time

3.1. Integrated Model

Before continuing with the mathematical model, we want to introduce the assumptions of the integrated model:

- Only transit containers are considered. Export and import containers are not taken into account due to uncertainties in their arrival and departure times.
- Each container is assigned to a storage location and stays there until the departure time from the terminal.
- Storage yard is considered as a 2D space. When a container arrives its assigned location, it is placed on top of the stack.

- If a container is reshuffled to remove another container, it should be placed into its assigned position again.
- If container i arrives in time period t, container j departs in time period t and two containers are assigned to the same position, then container j should be removed first to avoid reshuffling.
- Arrival and departure times of the containers are the berthing time period of the vessel which will load or unload the related container. Loading or unloading process may last longer than a single time period but as we do not determine loading or unloading sequence of the containers, we do not know the exact arrival or departure time period of a container.
- Arrival and departure points of the containers are assumed to be the berthing points of the vessels (i.e. Y_{btv}) which load or unload them.

These assumptions are also valid for sequential and rolling horizon solution methods. The objective function (3.1) of the model minimizes the total cost. As mentioned before, there are four different cost components: two kinds of penalty cost, transportation cost and reshuffling cost. First component is multiplied with the total transportation distance (includes arrival and departure) of container i. The latter occurs if container j needs to be reshuffled due to container i. Remaining components are multiplied with related deviations. We define two subsets before continuing with the model:

- $\boldsymbol{T}_v = \{t_a^v, t_a^v + 1, \ldots, T - t_p^v + 1\}$
- $\boldsymbol{B}_v = \{1, 2, \ldots, B - l_v + 1\}$

The integrated problem can be formulated as follows:

$$\begin{aligned} \min \quad z = & c_t \left(\sum_{i \in I} \sum_{b \in B} \sum_{k \in K} (X_{ibk} + N_{ibk}) d_{bk} \right) + c_s \left(\sum_{i \in I} \sum_{j \in I (j \neq i)} S_{ij} \right) + \\ & \sum_{v \in V} c_{p_1}^v S_v^- + \sum_{v \in V} c_{p_2}^v P_v^+ \end{aligned} \tag{3.1}$$

s.t.

$$\sum_{b\in B_v}\sum_{t\in T_v} tY_{btv} + t_p^v + S_v^+ - S_v^- = t_d^v \qquad v \in \boldsymbol{V} \tag{3.2}$$

$$\sum_{b\in B_v}\sum_{t\in T_v} tY_{btv} = t_a^v + P_v^+ \qquad v \in \boldsymbol{V} \tag{3.3}$$

$$\sum_{b\in B}\sum_{k\in K} X_{ibk} = 1 \qquad i \in \boldsymbol{I} \tag{3.4}$$

$$\sum_{b\in B} X_{ibk} = \sum_{b\in B} N_{ibk} \qquad i \in \boldsymbol{I}, k \in \boldsymbol{K} \tag{3.5}$$

$$\sum_{k\in K} X_{ibk} = \sum_{t\in T_v} Y_{btv} \qquad i \in \boldsymbol{I}, b \in \boldsymbol{B}_v, v \in \boldsymbol{V} \text{such that } r_{iv}^a = 1 \tag{3.6}$$

$$\sum_{k\in K} N_{ibk} = \sum_{t\in T_v} Y_{btv} \qquad i \in \boldsymbol{I}, b \in \boldsymbol{B}_v, v \in \boldsymbol{V} \text{such that } r_{iv}^d = 1 \tag{3.7}$$

$$\sum_{i\in I} Z_{ikt} \le h_{kt} \qquad k \in \boldsymbol{K}, t \in \boldsymbol{T} \tag{3.8}$$

$$tZ_{ikt} \le \beta_i - 1 \qquad i \in \boldsymbol{I}, k \in \boldsymbol{K}, t \in \boldsymbol{T} \tag{3.9}$$

$$\sum_{b\in B} X_{ibk} + (1 - Z_{ikt})|T| \ge \frac{\alpha_i}{t} \qquad i \in \boldsymbol{I}, k \in \boldsymbol{K}, t \in \boldsymbol{T} \tag{3.10}$$

$$\sum_{k\in K}\sum_{t\in T} Z_{ikt} = \beta_i - \alpha_i \qquad i \in \boldsymbol{I} \tag{3.11}$$

$$\sum_{v\in V} E_{vbt} \le 1 \qquad b \in \boldsymbol{B}, t \in \boldsymbol{T} \tag{3.12}$$

$$\sum_{b\in B_v}\sum_{t\in T_v} Y_{btv} = 1 \qquad v \in \boldsymbol{V} \tag{3.13}$$

$$\sum_{u=t}^{t+t_p^v-1}\sum_{q=b}^{b+l_v-1} E_{vqu} - t_p^v l_v - M(Y_{btv} - 1) \ge 0 \qquad v \in \boldsymbol{V}, b \in \boldsymbol{B}_v, t \in \boldsymbol{T}_v \text{ s.t. } t \ge t_a^v \tag{3.14}$$

$$\sum_{v\in V}\sum_{b\in B_v}\sum_{t\in T_v} tY_{btv} r_{iv}^a = \alpha_i \qquad i \in \boldsymbol{I} \tag{3.15}$$

$$\sum_{v\in V}\sum_{b\in B_v}\sum_{t\in T_v} tY_{btv} r_{iv}^d = \beta_i \qquad i \in \boldsymbol{I} \tag{3.16}$$

$$Z_{ik,t-1} + Z_{jkt} + Z_{jk,t-1} - 3 - tZ_{ikt} + \frac{\alpha_j - \alpha_i}{|T|} \leq S_{ij} \qquad k \in \boldsymbol{K}, i \in \boldsymbol{I}, j \in \boldsymbol{I}, t \in \boldsymbol{T} \tag{3.17}$$

$$X_{ibk}, E_{vqu}, Y_{btv}, S_{ij}, Z_{ikt} \in \{0, 1\} \tag{3.18}$$

$$S_v^+, S_v^-, P_v^+, \alpha_i, \beta_i \geq 0 \tag{3.19}$$

Constraints (3.2) define the relationship between departure time and due time of vessel v. If a vessel cannot be berthed as soon as it arrives, then its service cannot be completed until its due time, since due time of a vessel is calculated using expected arrival time of the vessel. Therefore, a penalty cost incurs in this case for not completing the service until due time. The amount of the penalty cost is the deviation of the completion time from due time. If service is completed until due time, then S_v^- and thus the penalty becomes zero.

Constraints (3.3) define the relationship between arrival time and berthing time of vessel v. Similar to constraints (3.2), if vessel v berths as soon as it arrives, then $T_v^- = 0$ and no penalty cost incurs. Constraints (3.4) imply that every container should be assigned to a storage location and a berth segment. Berth segment assignment of the containers are also satisfied by constraints (3.6) and (3.7).

Containers should stay at the same storage location from arrival to departure. It is possible to remove container i to reach container j but after removing container j it is assumed that container i is placed at the same storage slot. Constraints (3.5) guarantee that each container stays at the same slot. As mentioned before, constraints (3.6) and (3.7) imply that the arriving and departing points of container i are consistent with the arriving and departing points of the vessel v which carries container i. Constraints (3.8) restrict the total number of containers assigned to storage location k in time period t. Constraints (3.9) ensure that if container i leaves storage yard in time period t, then decision variable Z_{ikt} should be equal to zero for the time periods $t, t+1, t+2, \ldots$.

Constraints (3.10) guarantee that if $\sum_{b \in B} X_{ibk} = 0$, then $Z_{ikt} = 0$ since the ratio α_i/t is always greater than zero. When $\sum_{b \in B} X_{ibk} = 1$ and container has not arrived yet, then the ratio α_i/t is larger than one. In that case Z_{ikt} should always be equal to zero. When $\sum_{b \in B} X_{ibk} = 1$ and container has arrived, then the ratio α_i/t is smaller than one and Z_{ikt} may be either one or zero. Constraints (3.11) limit the total number of Z_{ikt} variables which can be assigned to one. Therefore, constraints (3.9) - (3.11) determine correct values for Z_{ikt}.

Constraints (3.12) imply that each time period-berth segment block can be covered by at most one vessel and constraints (3.13) satisfy that all vessels should be assigned to a berthing position.

Constraints (3.14) ensure that time period-berth segment blocks covered by each vessel should be adjacent. This is to satisfy that service should not be preemptive. Arrival and departure times of containers are defined by constraints (3.15) and (3.16). As mentioned before, service may last longer than a single time period, but we assume that all containers are in the storage yard between the berthing times of the vessels in which they arrive and depart.

Constraints (3.17) define S_{ij}. If container j needs to be reshuffled when container i leaves, then S_{ij} becomes one. Otherwise, S_{ij} becomes zero. Since the objective function aims at minimizing cost, S_{ij} is one if and only if $Z_{ik,t-1}$, Z_{jkt} and $Z_{jk,t-1}$ equal to one, Z_{ikt} equals to zero and arrival time of container i is smaller than the arrival time of container j. Unless $Z_{ik,t-1}$, Z_{jkt} and $Z_{jk,t-1}$ are all one, then summation of these decision variables becomes less than three and S_{ij} may take zero value. Similarly, if Z_{ikt} is one, then due to the tZ_{ikt} component of the left-hand side, S_{ij} may take zero value. If these four decision variables take required values, summation of left hand side except $(\alpha_j - \alpha_i)/M$ becomes zero. If container j arrives later than container i, $(\alpha_j - \alpha_i)/M$ takes a value between zero and one which forces S_{ij} to take a value of one.

Constraints (3.18) are restrictions on binary variables and constraints (3.19) are nonnegativity constraints.

3.2. Sequential Models

As will be mentioned in Chapter 4, it is not always possible to obtain exact results for the integrated models in reasonable time. For the more realistic problem instances generated (i.e. ten vessels and a thousand containers), optimal results cannot be obtained in ten hours of execution time. Therefore, sequential models are needed not only for the comparison with the results of the integrated model but also to obtain a feasible solution that may be used a starting solution for the integrated model in larger problems.

We divide the integrated model into two different models, one is to obtain a berth plan (BAP) and the other is to determine storage locations of containers (SAP). BAP does not consider storage locations of the containers. It only minimizes penalty costs which occurs if either service does not start as soon as a vessel arrives or service cannot be completed until the due time of a vessel. Therefore, we expect that the total penalty costs of BAP should be less than or equal to the integrated model. However, total transportation distance and reshuffling may increase.

After we determine the berthing plan, we solve SAP to determine storage locations of the containers. SAP minimizes total transportation and reshuffling costs. Decision variables of BAP formulation (i.e., $E_{vqu}, Y_{btv}, S_v^+, S_v^-, P_v^+$) become parameters in the SAP formulation where $X_{ibk}, N_{ibk}, S_{ij}, Z_{ikt}, \alpha_i$ and β_i are the decision variables.

The sequential formulation is as follows:

BAP:

$$\min \quad z_1 = \sum_{v \in V} c^v_{p_1} S^-_v + \sum_{v \in V} c^v_{p_2} P^+_v \tag{3.20}$$

s.t. Constraints $(3.2) - (3.3), (3.12) - (3.14)$

$$E_{vqu}, Y_{btv} \in \{0, 1\} \tag{3.21}$$

$$S^+_v, S^-_v, P^+_v \geq 0 \tag{3.22}$$

SAP:

$$\min \quad z_2 = c_t \left(\sum_{i \in I} \sum_{b \in B} \sum_{k \in K} (X_{ibk} + N_{ibk}) d_{bk} \right) + c_s \left(\sum_{i \in I} \sum_{j \in I (j \neq i)} S_{ij} \right) \tag{3.23}$$

s.t. Constraints $(3.4) - (3.11), (3.15) - (3.17)$

$$X_{ibk}, S_{ij}, Z_{ikt} \in \{0, 1\} \tag{3.24}$$

$$\alpha_i, \beta_i \geq 0 \tag{3.25}$$

The objective function of BAP and SAP are the last two components and first two components of the integrated model, respectively. We compare the performances of sequential formulations in Chapter 4. Performance analysis includes comparison of the objective values and the solution times.

Decomposition allows us to solve large-sized instances. However, although we may increase the number of vessels in BAP, SAP allows only a small increase in the total number of containers. The latter is the main restriction in the integrated formulation. Therefore, including container related constraints in SAP limits the size of the problem instances which can be solved in a reasonable amount of time. In such a case, we offer another solution method, which is introduced in Section 3.3.

As explained in Chapter 4, our mathematical model works fine with real life cases. Even though we cannot solve big problems with our model, we generated our inputs (arriving times of vessels, transportation and reshuffling costs, penalty costs for vessels, maximum number of containers that can be stored at a storage location in a certain time period) with respect to real life data. Container terminal layout is compatible with the real container terminals. In real life, container terminals always have a possibility to meet some extreme cases.

For example, vessel v, which takes container i may arrive before vessel v', which brings that container. In real life problems, there are two options for such a case: vessel v either waits vessel v' or departs without taking container i. Container i have to be loaded to another vessel if the first option realizes. Therefore, we have to update r^d_{iv} value for container i. Since the first option enables us to change vessels, we can load container i to one of the vessels that arrives after vessel v'. In such a case, the sequential method works fine.

However, if the second option realizes, then vessel v, in which container i departs with, has to until vessel v' arrives. In such a case, vessel v can wait either at the berth or at open sea. If we use the integrated model, as we have a constraint indicating that the departure times of all containers, which leave with the same vessel should be equal to the departure time of the vessel. However, if we use the sequential method to solve the problem such a case leads to infeasibility as we do not have any parameters for the definition of the precedence relations between the vessels. Therefore, we define another parameter for the BAP model of the sequential method.

$$\Theta_{vv'} : \begin{cases} 1, & \text{if vessel } \; v \; \text{ has to wait vessel for } \; v' \\ 0, & \text{otherwise} \end{cases}$$

We also add another constraint to our BAP formulation.

$$\sum_{q=t}^{T} qE_{vqu} \geq tY_{btv'}\Theta_{vv'} \qquad \forall k \in K, \forall t \in T \qquad (3.26)$$

Constraints (3.26) restricts E_{vqu} if there is a precedence relationship between vessels v and v'. If vessel v has to wait vessel v', then vessel v has to be berthed anytime after vessel v' arrives.

3.3. Rolling Horizon Method

Although sequential models can be used to solve medium-sized problems, if we increase the problem size, i.e., number of containers, problems may not be solved in a reasonable time. Therefore, we define another time concept in the rolling horizon method. $|T|$ is still total number of time periods. However, we need time segments which are defined as h. Total number of time segments, $|H|$ is our planning horizon. We divide our problems into $|H|$ different problems and solve them using the integrated model. However, we do not have the full information of time horizon which leads to suboptimality.

We may continue with an example to introduce rolling horizon method. We have six vessels which have a berth plan given in Figure 3.2. If we use rolling horizon method, the new berth plan looks like the one given in Figure 3.3.

We divide the whole problem into six different problems. First, we solve the first time segment using the integrated model. From Figure 3.3, we may see that Vessel 1, Vessel 2 and Vessel 3 should be included in the first problem. Therefore we have a problem of three vessels. Integrated model determines $E_{vqu}, Y_{btv}, S_v^+, S_v^-, P_v^+$ for these vessels. We also determine X_{ibk}, Z_{ikt} and α_i for the containers which arrive within these vessels and N_{ibk} and β_i for the containers which depart within these vessels.

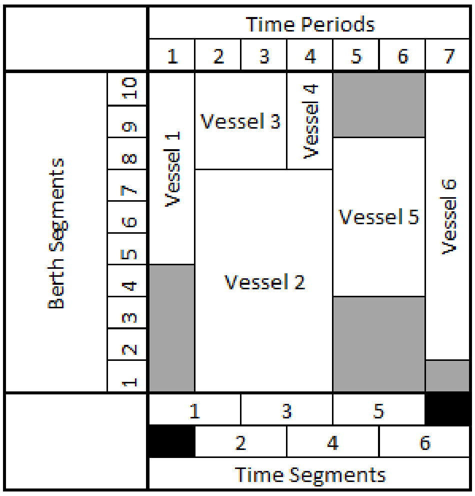

Figure 3.3. Berth plan in rolling horizon method.

However, Vessel 2 and Vessel 3 arrive in time period two. Therefore, our second problem recalculates $E_{vqu}, Y_{btv}, S_v^+, S_v^-, P_v^+$ values for these two vessels. This time Vessel 1 is also included, but $E_{vqu}, Y_{btv}, S_v^+, S_v^-, P_v^+$ values are determined by the first problem's solution and these values of Vessel 1 will be taken as parameters. We continue until we reach the last problem. Our sixth problem is a six-vessel problem. $E_{vqu}, Y_{btv}, S_v^+, S_v^-, P_v^+$ values for Vessel 1 - Vessel 5 are taken as parameters. $E_{vqu}, Y_{btv}, S_v^+, S_v^-, P_v^+$ are decision variables for only Vessel 6. In other words, although we consider two time periods in each problem, we also include previously determined plans. For example, problem 3 plans time periods 3 and 4, but takes the information of time periods 1 and 2 as parameters. Using this method, the objective function of problem 6 is our total cost for our planning horizon. This method also eases the calculations of S_{ij}. Since we do not know the berth plans of the future time periods, we cannot determine exact S_{ij} values. We can only consider expected departure and arrival times of the containers to prepare storage plans. Therefore, S_{ij} values, which are determined before the problem 6, are not exact but estimated values. In problem 6, we recalculate S_{ij} values with respect to exact departure and arrival times.

We should include the variables in the problems with similar method. For example, if container i arrives with Vessel 1 and departs with Vessel 5, then exact values of X_{ibk}, α_i and N_{ibk}, β_i are determined in problem 1 and problem 5, respectively. Z_{ikt} values are also determined when both α_i and β_i are determined.

4. COMPUTATIONAL RESULTS

In this chapter, we compare the performances of different solution methods introduced before. To do this, we generate different groups of problem instances. As mentioned earlier, solution times are limited to 10 hours. We implement our mathematical model into GAMS version 23.6 and use two different solvers, Gurobi version 4.5.1 and Cplex version 12.3.

The first set of problem instances is given in Table 4.1. This group includes 15 instances of the largest size that can be solved optimally in ten hours. These problem instances are also solved by the sequential method and the rolling horizon method using Gurobi and Cplex. The rolling horizon method has two options: 2-Time Horizons (2H) and 3-Time Horizons (3H). We have time segments of length two time periods in 2-Time Horizons problems and three time periods in 3-Time Horizon problems. As the length of time segments goes to the number of time periods of the original problem, solution also converges to the optimal solution.

The largest problem instance that can be solved optimally consists of 120 containers, 4 vessels, 10 berth segments, 30 storage locations and 40 time periods. The results of the different methods can be found in Table 4.2 and Table 4.3 while solution times of the problems are in Table 4.4 and Table 4.5. The comparison of the objective values of different solution methods can be found in Figure 4.1 and comparison of solution times can be found in Figure 4.2 − Figure 4.4.

According to the first instance group, all solution methods except 2H gives very close results. For the sequential method, since we divide the overall problem into BAP and SAP, solution times are considerably better except P7-Gurobi, which is slightly worse than the optimal objective value. The performance of the rolling horizon method varies for this instance group. 3H results are close to the optimal solutions as expected but 2H does not give good results for even very small instances. Since we do not

Table 4.1. The first set of instances.

	# of Containers	# of Vessels	# of Berth Segments	# of Storage Locations	# of Time Periods
P1	30	3	10	30	10
P2	30	3	10	30	20
P3	30	3	10	30	40
P4	30	3	10	60	10
P5	30	3	10	60	20
P6	30	3	10	60	40
P7	60	4	10	30	10
P8	60	4	10	30	20
P9	60	4	10	30	40
P10	60	4	10	60	10
P11	60	4	10	60	20
P12	60	4	10	60	40
P13	120	4	10	30	10
P14	120	4	10	30	20
P15	120	4	10	30	40

Table 4.2. Results of the first set of instances - 1.

	Solver	Optimal	Sequential	2H	3H
P1	Cplex	7752	7752	8144	7907
P1	Gurobi	7752	7752	8144	7907
P2	Cplex	7752	7752	8703	7985
P2	Gurobi	7752	7752	8703	7985

Table 4.3. Results of the first set of instances - 2.

	Solver	**Optimal**	**Sequential**	**2H**	**3H**
P3	Cplex	7,752	7,752	8,703	7,985
P3	Gurobi	7,752	7,752	8,703	7,985
P4	Cplex	7,740	7,740	8,112	7,866
P4	Gurobi	7,740	7,740	8,112	7,866
P5	Cplex	7,740	7,740	8,336	8,336
P5	Gurobi	7,740	7,740	8,336	8,336
P6	Cplex	7,740	7,740	8,336	8,336
P6	Gurobi	7,740	7,740	8,336	8,336
P7	Cplex	17,928	18,287	18,126	17,986
P7	Gurobi	17,928	18,287	18,126	17,986
P8	Cplex	17,928	18,287	18,654	18,469
P8	Gurobi	17,928	18,287	18,654	18,469
P9	Cplex	17,928	18,287	20,707	18,824
P9	Gurobi	17,928	18,287	20,707	18,824
P10	Cplex	18,922	20,625	21,244	20,625
P10	Gurobi	18,922	20,625	21,244	20,625
P11	Cplex	18,922	20,625	21,891	20,651
P11	Gurobi	18,922	20,625	21,891	20,651
P12	Cplex	18,922	20,625	22,073	21,022
P12	Gurobi	18,922	20,625	22,073	21,022
P13	Cplex	22,152	23,703	22,419	22,152
P13	Gurobi	22,152	23,703	22,419	22,152
P14	Cplex	22,152	23,703	24,634	23,957
P14	Gurobi	22,152	23,703	24,634	23,957
P15	Cplex	22,152	23,703	23,312	23,312
P15	Gurobi	22,152	23,703	23,312	23,312

Table 4.4. Solution times of the first set of instances - 1.

	Solver	**Optimal**	**Sequential**	**2H**	**3H**
P1	Cplex	708	448	435	452
P1	Gurobi	395	370	359	402
P2	Cplex	2,785	1,549	1,472	1,672
P2	Gurobi	1,715	1,567	1,473	1,517
P3	Cplex	5,012	2,811	2,933	3,246
P3	Gurobi	3,224	2,624	2,412	2,602
P4	Cplex	1,354	1,275	1,199	1,376
P4	Gurobi	797	637	535	562
P5	Cplex	3,102	2,451	1,489	1,695
P5	Gurobi	1,907	1,261	1,046	1,267
P6	Cplex	4,579	3,088	2,826	3,245
P6	Gurobi	3,622	2,259	2,065	2,197
P7	Cplex	1,059	952	914	923
P7	Gurobi	513	549	478	497
P8	Cplex	2,003	1,382	1,202	1,214
P8	Gurobi	1,488	1,042	812	853
P9	Cplex	4,543	3,907	3,204	3,236
P9	Gurobi	2,655	1,991	1,812	1,903
P10	Cplex	3,919	3,253	3,090	3,289
P10	Gurobi	2,987	2,509	2,434	2,531
P11	Cplex	8,012	7,531	6,703	6,971
P11	Gurobi	6,256	4,504	3,829	3,867

Table 4.5. Solution times of the first set of instances - 2.

	Solver	Optimal	Sequential	2H	3H
P12	Cplex	14,923	10,148	9,031	9,512
P12	Gurobi	12,678	9,762	8,493	9,442
P13	Cplex	16,798	11,458	9,094	10,475
P13	Gurobi	12,986	8,917	8,407	8,591
P14	Cplex	24,331	17,955	14,807	17,680
P14	Gurobi	21,643	14,968	12,554	15,700
P15	Cplex	33,412	27,398	21,370	24,798
P15	Gurobi	29,521	20,074	17,063	18,404

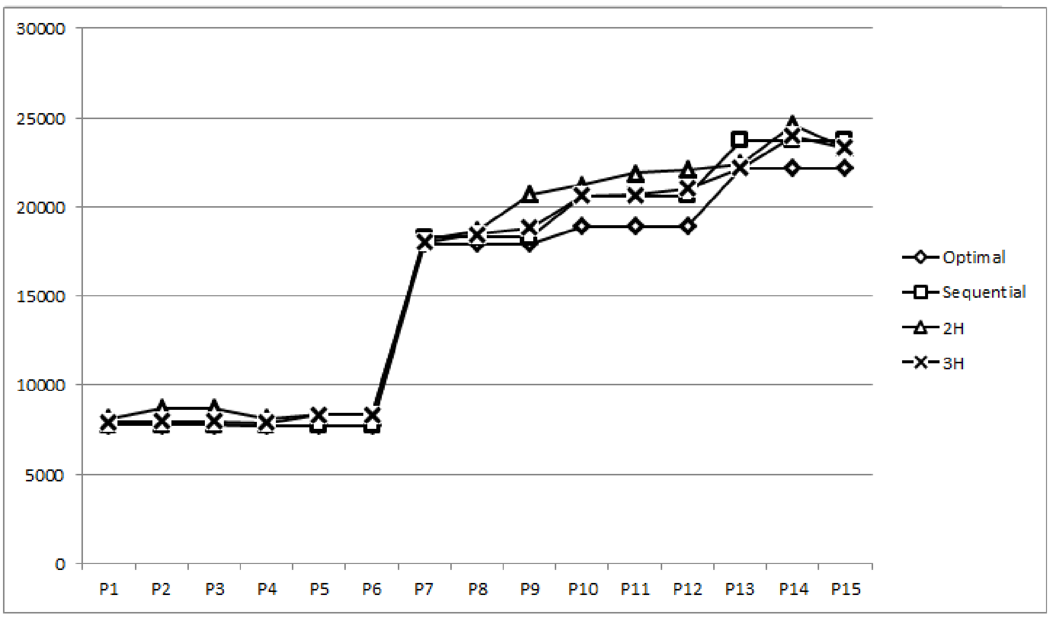

Figure 4.1. Objective values of the first set of instances.

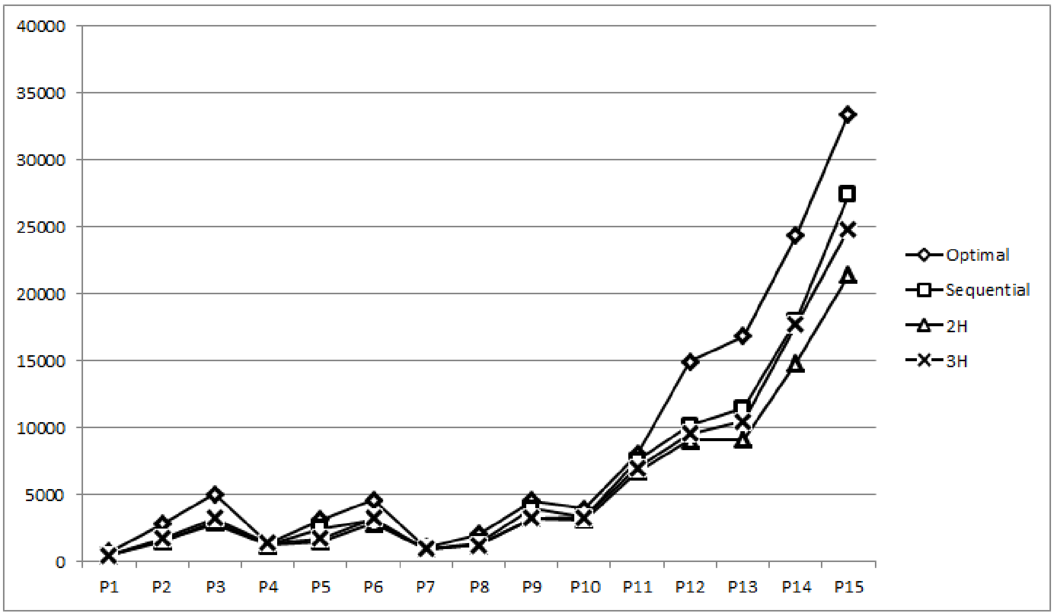

Figure 4.2. Solution times of the first set of instances with Cplex.

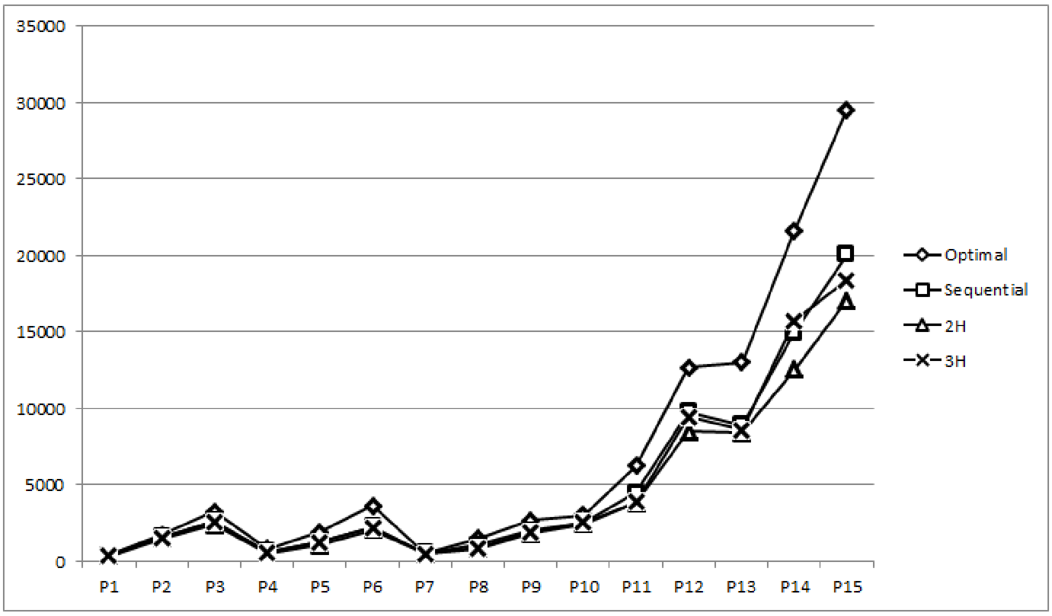

Figure 4.3. Solution times of the first set of instances with Gurobi.

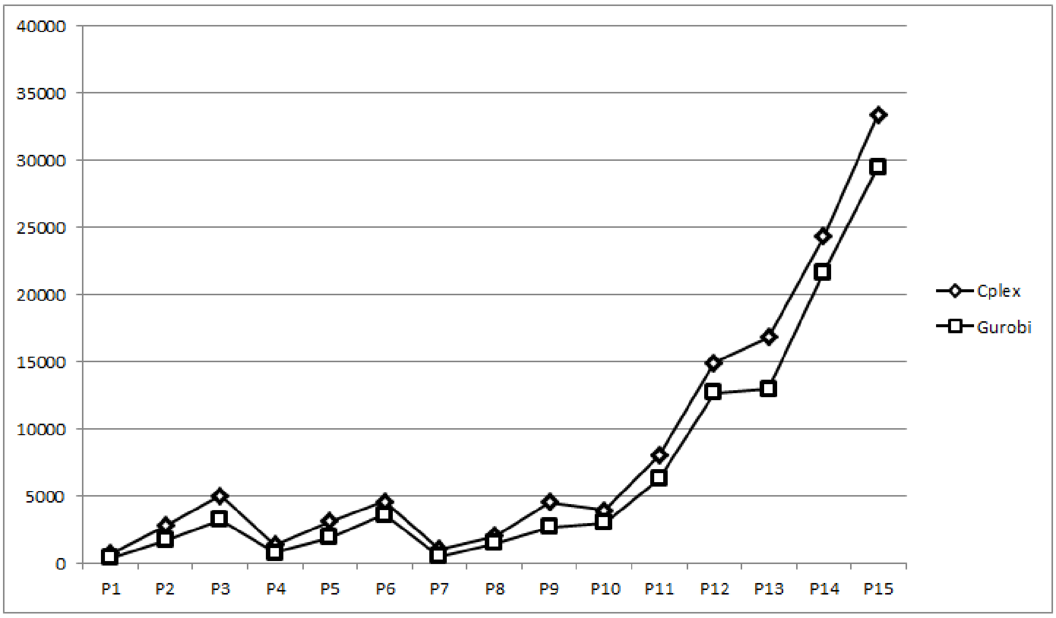

Figure 4.4. Comparison of optimal solution times of the first set of instances with different solvers.

have the information about the containers which arrive in the upcoming time periods, containers of the current time period are allocated to the nearest available storage location and this results, either reshuffling or longer transportation distance for upcoming containers. The rolling horizon method sometimes gives the same objective values for 2H and 3H, like P5 and P6. There are two many time periods for limited number of vessels for these problem instances and there are more than 3 time periods between two vessels. Therefore our objective value stays same for these problems.

If we compare our results, for the first set of instances, it turns out that the sequential method and the 3H method outperform other solution methods. This is an expected result for this set. As the number of vessels and containers are small in these problems, berthing position does not affect the objective function considerably. Therefore, the sequential method gives good results for all problem instances. 3H gives better results especially when the number of time periods are small. As the number of time periods increases, performance of the 3H method becomes poorer.

If we look at solver performances, Gurobi gives better solution times when compared to Cplex.

As sequential and 3H rolling horizon methods are good alternatives for optimal solution for the first set of instances, another set of instances are generated. This set is for busy period. In other words, vessel traffic is heavy in contrast to the first set of instances which is for idle period. Now we will compare the results for busy period.

Since optimal results for larger instances cannot be solved in less than ten hours, we decrease the number of time periods to increase the number of vessels. We also increase the lengths of vessels and decrease the penalty cost of the vessels to increase the planning options for BAP and compare the results of sequential and optimal results better. A final change we make is to reduce the number of containers that can be serviced in one period. Vessels need to be berthed for more than one period with this update. This change is done not only for the comparison of the performances of different solution methods, but also for the verification of the model if vessels need to stay berthed for more than one period. Another set of problem instances will be generated for other extreme cases later. Problem instances of the second set can be found in Table 4.6.

Finding an optimal solution especially for P22, P25 and P28 are not possible with the previous vessel lengths. Therefore, we decrease the vessel lengths for these instances while keeping them the same for other instances. We do not change transportation cost, reshuffling cost and distances between berth segments and storage locations. As is the case with the first set, there can be at most five containers at a storage location in a certain time period, but if a container leaves in time period t and another container arrives in the same period, then storage location k can be allocated to both containers even if the total number of containers exceeds five with the arriving container, since we assume that leaving containers are removed from storage locations first.

Results of the second set of instances can be found in Table 4.7, Table 4.8 and

Table 4.6. Second set of instances.

	# of Containers	**# of Vessels**	**# of Berth Segments**	**# of Storage Locations**	**# of Time Periods**
P16	30	6	10	30	5
P17	30	6	10	30	10
P18	30	6	10	30	20
P19	30	6	10	60	5
P20	30	6	10	60	10
P21	30	6	10	60	20
P22	60	8	10	30	5
P23	60	8	10	30	10
P24	60	8	10	30	20
P25	60	8	10	60	5
P26	60	8	10	60	10
P27	60	8	10	60	20
P28	120	8	10	30	5
P29	120	8	10	30	10
P30	120	8	10	30	20

Figure 4.5.

Table 4.7. Results of the second set of instances - 1.

	Solver	Optimal	Sequential	2H	3H
P16	Cplex	7,740	7,740	9,365	8,514
P16	Gurobi	7,740	7,740	9,365	8,514
P17	Cplex	7,782	7,902	7,946	7,812
P17	Gurobi	7,782	7,902	7,946	7,812
P18	Cplex	7,782	7,902	8,012	7,856
P18	Gurobi	7,782	7,902	8,012	7,856
P19	Cplex	17,928	17,928	20,707	18,824
P19	Gurobi	17,928	17,928	20,707	18,824
P20	Cplex	14,079	16,016	15,622	14,312
P20	Gurobi	14,079	16,016	15,622	14,312
P21	Cplex	13,812	15,971	16,766	15,604
P21	Gurobi	13,812	15,971	16,766	15,604
P22	Cplex	11,790	12,252	12,320	11,902
P22	Gurobi	11,790	12,252	12,320	11,902
P23	Cplex	12,402	13,506	14,042	12,976
P23	Gurobi	12,402	13,506	14,042	12,976
P24	Cplex	16,587	17,102	19,646	17,254
P24	Gurobi	16,587	17,012	19,646	17,254
P25	Cplex	11,879	12,532	12,364	11,879
P25	Gurobi	11,879	12,532	12,364	11,879
P26	Cplex	26,331	31,110	31,560	28,475
P26	Gurobi	26,331	31,110	31,560	28,475
P27	Cplex	14,032	18,442	19,236	16,554
P27	Gurobi	14,032	18,442	19,236	16,554

Comparison of solution times of the second set of instances can be found in Table

Table 4.8. Results of the second set of instances - 2.

	Solver	Optimal	Sequential	2H	3H
P28	Cplex	26,012	28,002	28,953	26,104
P28	Gurobi	26,012	28,002	28,953	26,104
P29	Cplex	22,541	25,199	26,457	24,874
P29	Gurobi	22,541	25,199	26,457	24,874
P30	Cplex	33,126	38,182	42,325	37,216
P30	Gurobi	33,126	38,182	42,325	37,216

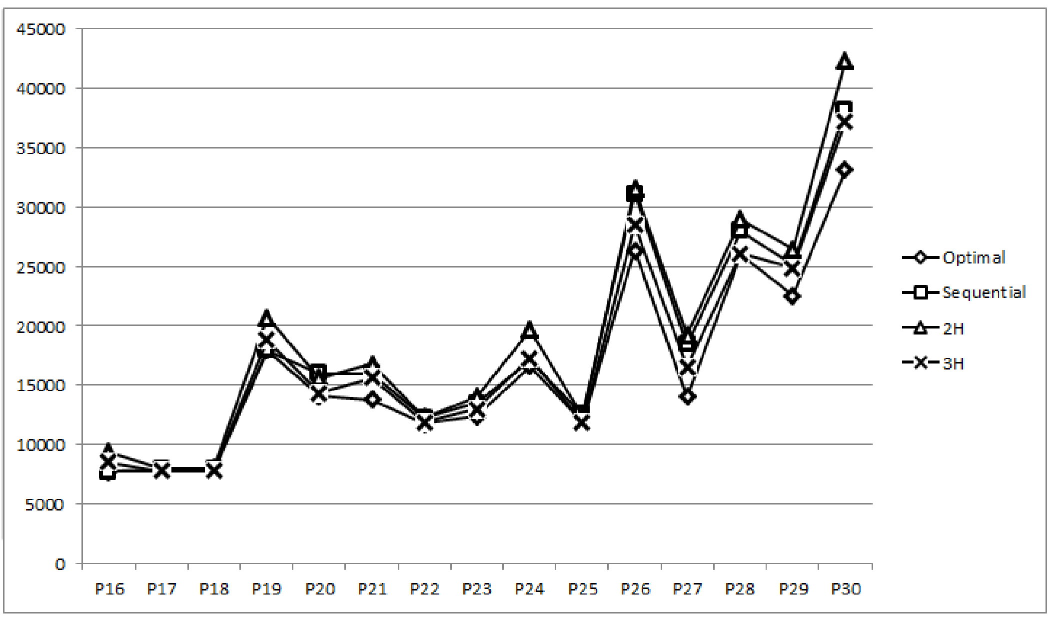

Figure 4.5. Objective values of the second set of instances.

4.9, Table 4.10 and Figure 4.6 - Figure 4.8.

Table 4.9. Solution times of the second set of instances - 1.

	Solver	Optimal	Sequential	2H	3H
P16	Cplex	24	18	19	21
P16	Gurobi	22	16	15	19
P17	Cplex	60	30	42	39
P17	Gurobi	56	45	44	56
P18	Cplex	181	100	112	136
P18	Gurobi	195	185	157	169
P19	Cplex	74	62	42	67
P19	Gurobi	61	52	34	36
P20	Cplex	175	66	46	164
P20	Gurobi	112	224	86	98
P21	Cplex	426	280	232	327
P21	Gurobi	487	903	198	241
P22	Cplex	647	424	316	349
P22	Gurobi	591	324	302	392
P23	Cplex	1,012	678	616	894
P23	Gurobi	999	581	564	743
P24	Cplex	2,567	1,818	1,975	2,345
P24	Gurobi	2,221	1,578	1,684	1,957
P25	Cplex	1,926	1,654	1,522	1,698
P25	Gurobi	1,888	1,623	1,597	1,754
P26	Cplex	3,504	3,122	2,933	3,433
P26	Gurobi	3,311	2,745	2,600	3,065
P27	Cplex	8,844	7,676	7,027	8,244
P27	Gurobi	7,913	6,845	6,214	7,541

The objective values obtained by the sequential method is not as good as the 3H

Table 4.10. Solution times of the second set of instances - 2.

	Solver	**Optimal**	**Sequential**	**2H**	**3H**
P28	Cplex	6,037	5,780	5,475	5,936
P28	Gurobi	5,743	4,214	4,070	4,963
P29	Cplex	11,987	8,541	8,863	9,675
P29	Gurobi	10,365	7,987	7,635	8,763
P30	Cplex	21,546	15,242	14,963	18,542
P30	Gurobi	19,642	14,233	13,587	17,253

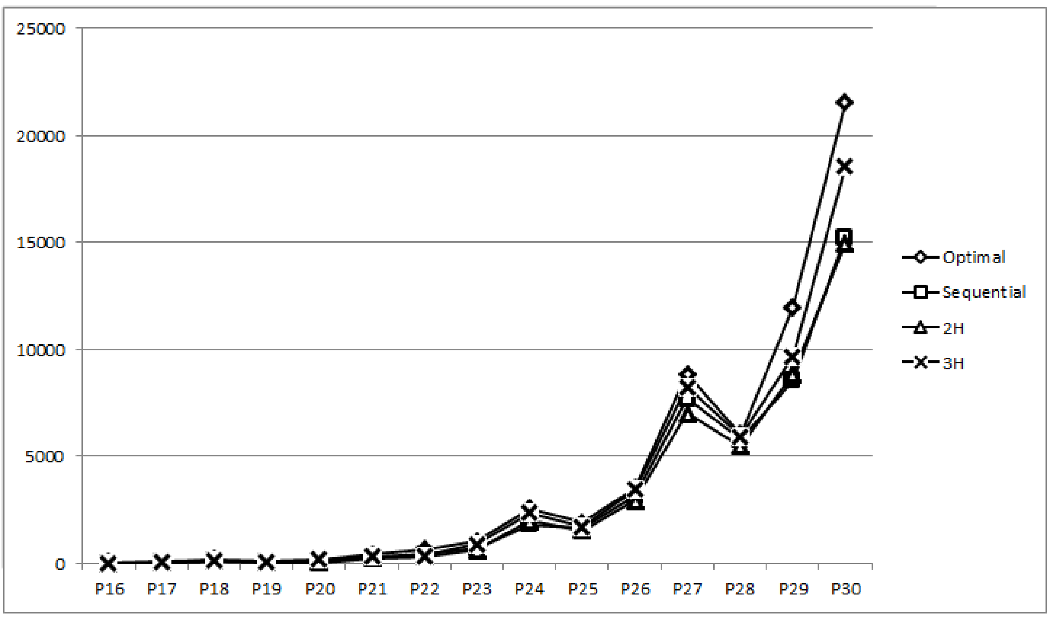

Figure 4.6. Solution times of the second set of instances with Cplex.

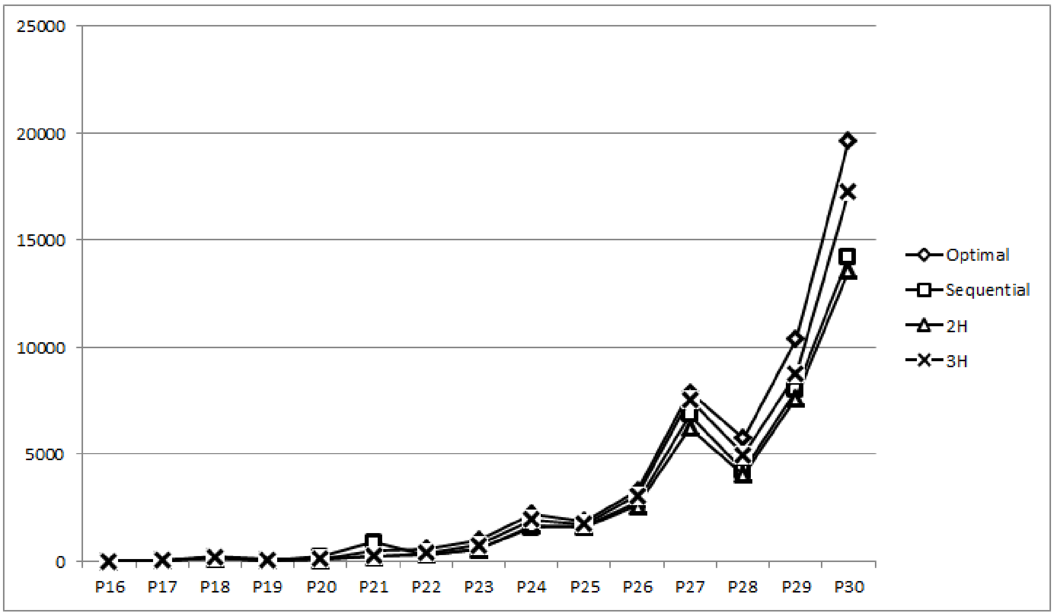

Figure 4.7. Solution times of the second set of instances with Gurobi.

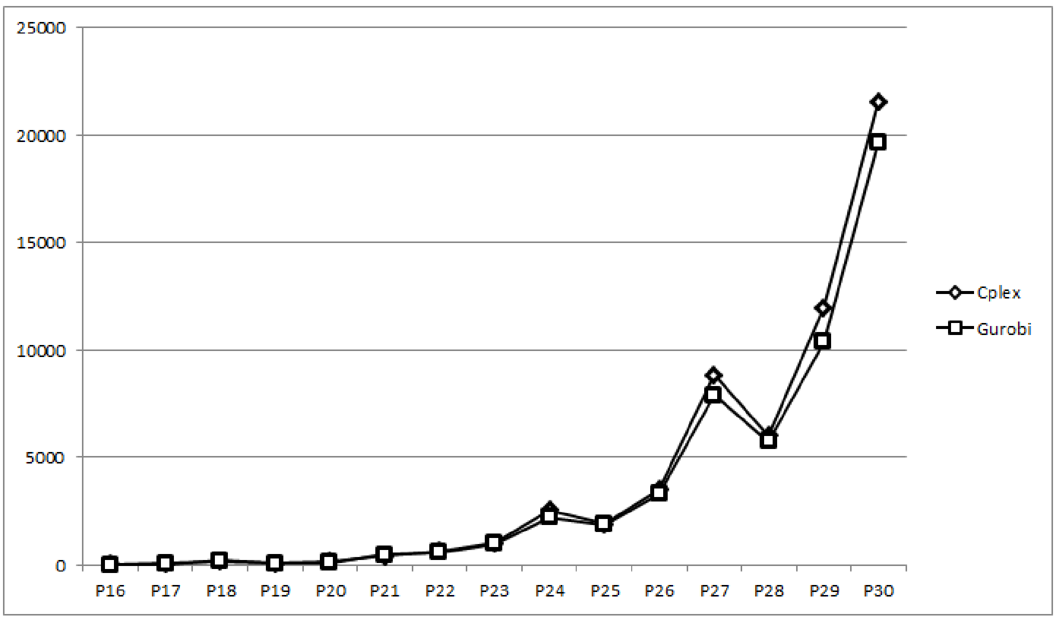

Figure 4.8. Comparison of the optimal solution times of the second set of instances with different solvers.

method in this set of instances. Since the number of time periods is very small, 3H results are close to optimal solution and as we increase the number of time periods, the performance of 3H becomes worse. This can also be noticed from the performance of the 2H method. 2H results are the worst results when compared to others. Therefore, we cannot use rolling horizon method in real life instances as there will be considerably more time periods. We need to use longer time segments instead of two or three and as we increase the length of time segments, solution times also increase.

Solution times of the methods are closer when compared to first set of instances. However, they are still unacceptable in real life problems as we need to run our planning procedure as soon as we receive new information. When it comes to solver performances, Gurobi is still better than Cplex. We will talk about another approach to decrease solution times after we introduce another set of instances.

One of the biggest disadvantage of rolling horizon method occurs when two vessels arrive in the same time period. Consider problem instances in Table 4.11. In these instances, Vessel 1 and Vessel 2 arrive in time periods 1 and 2, respectively, and Vessel 3 and Vessel 4 arrive in time period 5. Therefore, in the rolling horizon method, containers which will leave with Vessels 4 and 5 are considered as they will leave in the same time period. As a result, there will be no reshuffling cost. On the other hand, the summation of the lengths of these vessels are longer than the length of the berth and their service cannot be done in the same time period. This results in a high reshuffling cost and rolling horizon method's performance is poor for this case. Results of these problem instances can be found in Table 4.12.

Storage plan of P31 can be found in Figure 4.10. In this problem, C8 and C19 arrives with Vessel 1 and depart with Vessel 3, whereas C11 arrives with Vessel 1 and departs with Vessel 4 and C16 arrives with Vessel 2 and departs with Vessel 4. Therefore, C11 and C16 needs to be reshuffled while removing C8 and C19.

Table 4.11. Third set of instances.

	# of Containers	# of Vessels	# of Berth Segments	# of Storage Locations	# of Time Periods
P31	30	4	5	20	10
P32	60	4	5	20	10
P33	90	4	5	20	10

Table 4.12. Results of the third set of instances.

	Solver	**Optimal**	**2H**	**3H**
P31	Gurobi	8,514	9,365	8,604
P32	Gurobi	16,920	19,427	18,240
P33	Gurobi	22,017	26,678	24,982

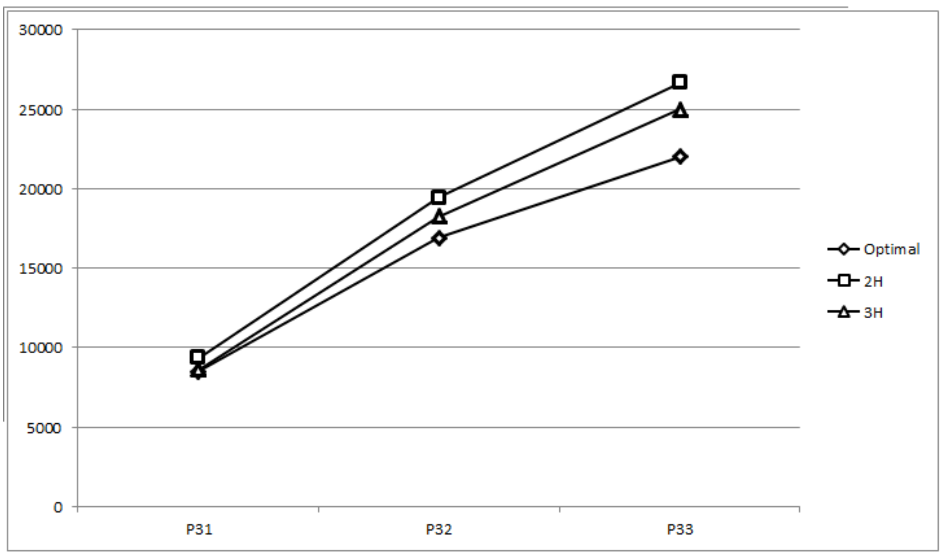

Figure 4.9. Objective values of the third set of instances.

c2	c7	c24	
c27	c26	c20	c14
c25	c3	c16	c5
c15	c11	c30	c12
c29	c8	c19	c13
k1	k2	k3	k4

Figure 4.10. Storage plan of P31.

4.1. Grouping Containers

Although sequential and rolling horizon methods make it possible to increase the problem size, we cannot still solve the real life instances with these methods. Therefore, we need to develop another approach to increase the number of containers dealt with. In this section we group containers and consider all containers in the same group as one item. Since this allows us to increase the problem size, if the results of this approach is acceptable we can use this method in medium-sized problems.

Our batches consist of at most five containers, as in each storage location there can be at most five containers in certain time period. Containers, which arrive and depart with the same vessel are in the same group. If the number of such containers is more than five and multiple of five, then all of our groups are complete groups. If the number of containers is not the multiple of five, then we have some incomplete groups.

We will introduce two methods to plan container groups. First, we use both complete and incomplete groups in our mathematical model.

$$\min \quad z = \gamma_i \left(c_t \left(\sum_{i \in I} \sum_{b \in B} \sum_{k \in K} (X_{ibk} + N_{ibk}) d_{bk} \right) + c_s \left(\sum_{i \in I} \sum_{j \in I (j \neq i)} S_{ij} \right) \right) + \sum_{v \in V} c^v_{p_1} S^-_v + \sum_{v \in V} c^v_{p_2} P^+_v \tag{4.1}$$

s.t.

$$\text{Constraints } (3.2) - (3.7), (3.9) - (3.19)$$

$$\sum_{i \in I} \gamma_i Z_{ikt} \leq h_{kt} \quad k \in \boldsymbol{K}, t \in \boldsymbol{T} \tag{4.2}$$

Now, set i represents the container groups not containers. All decision variables and parameters, which are related with the individual containers in our original mathematical model, are valid for the container groups in this changed model.

Constraints (4.2) indicate that the maximum number of containers at a storage location cannot exceed h_{kt}. Although our mathematical model can solve considerably larger problem instances with this update, since we do not allow to break the groups, it might result in some inefficiencies. This method has also one advantage regarding with the calculations of the reshuffling costs, since we use the container groups we do not need to check individual containers. Therefore, we decrease the solution time owing to the calculations of the number of reshuffling. We update the reshuffling cost part from the objective function 4.3.

Our second approach is solving the whole problem in two steps. As there are more containers in a complete group, we first use our integrated model to plan only complete groups. In this problem, our storage yard becomes a 2D place instead of a 3D place. We have an other advantage because of grouping, since we only plan complete groups, there is no reshuffling cost in the first step. Then after the first step, we break the incomplete groups into the individual containers and plan afterwards. We use our original integrated model in the second step, but since some storage locations are already full, we do not allow the containers to be allocated to these locations by setting their Z_{ikt} values to zero. Since the vessels are already planned, Y_{btv} and E_{vqu} values are input for the second step. Additionally, since α_i and β_i values are determined with respect to vessels' berthing times, these values are also determined in the first step.

In the first step, we determine the service time of a vessel considering total number of containers arrives and departs with that vessel. Therefore, we do not have any infeasibility regarding to the service times of the vessels in the second step.

Mathematical model of the first step is similar with the previous approach. The only difference is, we do not have γ_i in this model as all groups have the same number of containers. Instead, we use a constant η which equals to the maximum number of containers that can be stored at a storage location.

$$\min \quad z = \eta c_t \left(\sum_{i\in I} \sum_{b\in B} \sum_{k\in K} (X_{ibk} + N_{ibk}) d_{bk} \right) + \sum_{v\in V} c^v_{p_1} S^-_v + \sum_{v\in V} c^v_{p_2} P^+_v \tag{4.3}$$

$$\text{s.t. Constraints } (3.2) - (3.7), (3.9) - (3.16), (3.19), (4.2)$$

$$X_{ibk}, E_{vqu}, Y_{btv}, Z_{ikt} \in \{0, 1\} \tag{4.4}$$

Results of the two solution approaches can be found in Table 4.13 and Figure 4.11. As it can be seen from the results, our new approaches give very good results when compared to the optimal results. It is an expected result as optimal solution also tries to group containers since grouping gives the least reshuffling cost. Additionally we are dealing with the smaller problem instances and grouping does not affect our optimal berthing and storage plan.

As a third approach, we updated first problem. We kept complete groups in the problem, but we broke our incomplete groups and plan them individually. If we analyze our previous results, optimal results can be reached with this consideration and it is definitely meaningful to group containers if it is possible to build a complete group. When we solved all 30 problem instances with the third approach, we reached optimal results. Solution times of the all third approach can be obtained from the Table 4.14, Table 4.15 and Figure 4.12. As Gurobi always gave the better results before, we solved the problem instances with Gurobi and did not use the Cplex.

As it can be obtained from the solution times, grouping decreases solution times considerably. Therefore, it let us increase problem sizes and we generated another

Table 4.13. Results of the two solution approaches.

	Complete and Incomplete Groups	**Complete Groups and Individual Containers**		**Complete and Incomplete Groups**	**Complete Groups and Individual Containers**
P1	7,752	7,752	P16	7,740	7,740
P2	7,752	7,752	P17	7,782	7,782
P3	7,752	7,752	P18	7,782	7,782
P4	7,740	7,740	P19	17,928	17,928
P5	7,740	7,740	P20	14,079	14,079
P6	7,740	7,740	P21	13,812	13,812
P7	18,024	17,928	P22	11,830	11,790
P8	18,024	17,928	P23	12,402	12,402
P9	18,024	17,928	P24	16,624	16,587
P10	18,922	18,922	P25	11,879	11,879
P11	18,922	18,922	P26	26,331	26,331
P12	18,922	18,922	P27	14,032	14,032
P13	22,234	22,152	P28	26,124	26,012
P14	22,234	22,152	P29	22,647	22,541
P15	22,234	22,152	P30	33,246	33,126

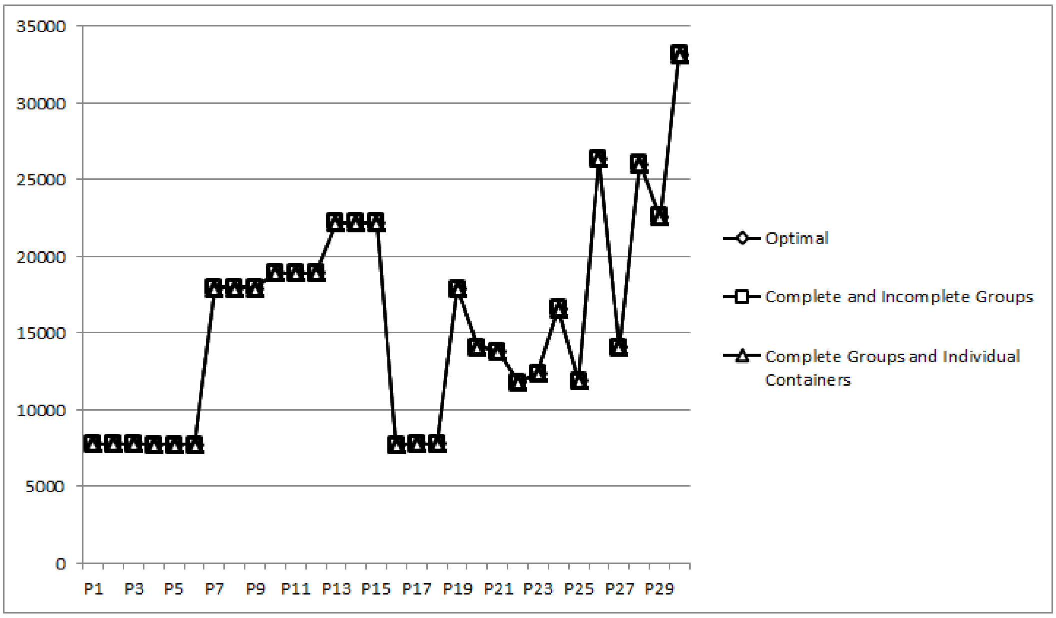

Figure 4.11. Comparison of the new solution approaches with optimal values.

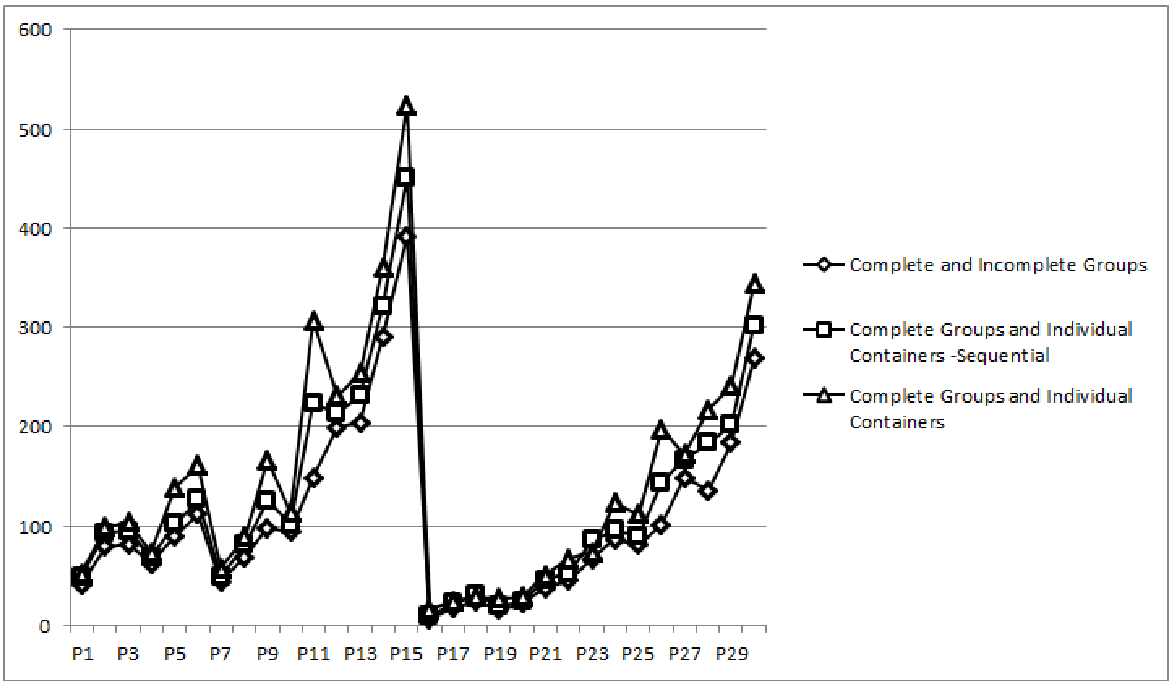

Figure 4.12. Solution times of the three solution approaches.

Table 4.14. Solution times of the three solution approaches - 1.

Problem Instances	**Complete and Incomplete Groups**	**Complete Groups and Individual Containers Sequential**	**Complete Groups and Individual Containers**
P1	41	49	52
P2	79	93	99
P3	82	95	104
P4	62	69	74
P5	90	102	139
P6	112	127	161
P7	44	49	57
P8	69	81	89
P9	98	126	167
P10	94	101	114
P11	149	224	307
P12	199	213	231
P13	204	232	254
P14	291	322	361
P15	392	451	524
P16	7	9	16
P17	17	22	25
P18	25	30	29
P19	16	20	27
P20	22	24	29
P21	38	46	51
P22	45	52	67
P23	67	86	74
P24	87	96	123

Table 4.15. Solution times of the three solution approaches - 2.

Problem Instances	**Complete and Incomplete Groups**	**Complete Groups and Individual Containers Sequential**	**Complete Groups and Individual Containers**
P25	81	90	112
P26	101	144	198
P27	149	167	172
P28	136	184	217
P29	184	202	241
P30	269	302	344

instance group. In this group, time periods are considered as two hours and parameters are updated with respect to this. Forth set of instances can be found in Table 4.16.

Objective values of this set can be found in Table 4.17, Table 4.18 and Figure 4.13. Solution times of the forth set of instances can be found in Table 4.19, Table 4.20 and Figure 4.14.

As it can be obtained from the results, last problem instance which plans 1000 containers, 20 vessels and 80 time periods can represent a week of a medium-sized container terminal. Solution time of this instance is about four hours and this also makes sense. As 20 vessels arrive in 80 time periods, then we expect that we update our vessel information every four time periods. Therefore, 4 hours can be thought as a proper solution time for planning 20 vessels in 80 time periods.

When we look at the results, all three solution approaches has very close results. However, if there are many incomplete groups in a problem, then not breaking the incomplete groups may lead to infeasibilities. Therefore, even Complete and Incomplete

Table 4.16. Forth set of instances.

	# of Containers	# of Vessels	# of Berth Segments	# of Storage Locations	# of Time Periods
P34	200	4	10	60	20
P35	200	4	10	60	40
P36	200	4	10	60	80
P37	250	4	10	60	20
P38	250	4	10	60	40
P39	250	4	10	60	80
P40	250	5	10	60	20
P41	250	5	10	60	40
P42	250	5	10	60	80
P43	250	5	10	120	40
P44	250	5	10	120	80
P45	500	5	10	120	40
P46	500	5	10	120	80
P47	500	6	10	120	40
P48	500	6	10	120	80
P49	500	8	10	120	40
P50	500	8	10	120	80
P51	500	10	10	120	40
P52	500	10	10	120	80
P53	500	10	10	240	40
P54	500	10	10	240	80
P55	1000	10	10	240	40
P56	1000	10	10	240	80
P57	1000	15	10	240	40
P58	1000	15	10	240	80
P59	1000	20	10	240	40
P60	1000	20	10	240	80

Table 4.17. Results of the forth set of instances - 1.

Problem Instances	Complete and Incomplete Groups	Complete Groups and Individual Containers Sequential	Complete Groups and Individual Containers
P34	12,458	12,361	12,361
P35	14,278	14,278	14,278
P36	16,874	16,657	16,657
P37	11,079	10,965	10,965
P38	14,598	14,402	14,402
P39	13,254	13,198	13,198
P40	17,099	16,844	16,844
P41	15,352	15,214	15,214
P42	16,522	16,497	16,497
P43	17,456	17,369	17,369
P44	13,647	13,517	13,517
P45	21,602	21,247	21,247
P46	22,773	22,631	22,603
P47	24,063	23,851	23,851
P48	24,965	24,790	24,763
P49	19,969	19,900	19,812
P50	26,098	25,754	25,689
P51	23,185	22,982	22,982
P52	21,548	21,412	21,386
P53	25,142	24,604	24,604
P54	28,546	28,471	28,403
P55	34,012	33,521	33,412

Table 4.18. Results of the forth set of instances - 2.

Problem Instances	Complete and Incomplete Groups	Complete Groups and Individual Containers Sequential	Complete Groups and Individual Containers
P56	36,887	36,452	36,396
P57	42,158	41,012	40,850
P58	39,458	39,295	39,202
P59	51,268	50,856	50,714
P60	54,089	53,549	53,321

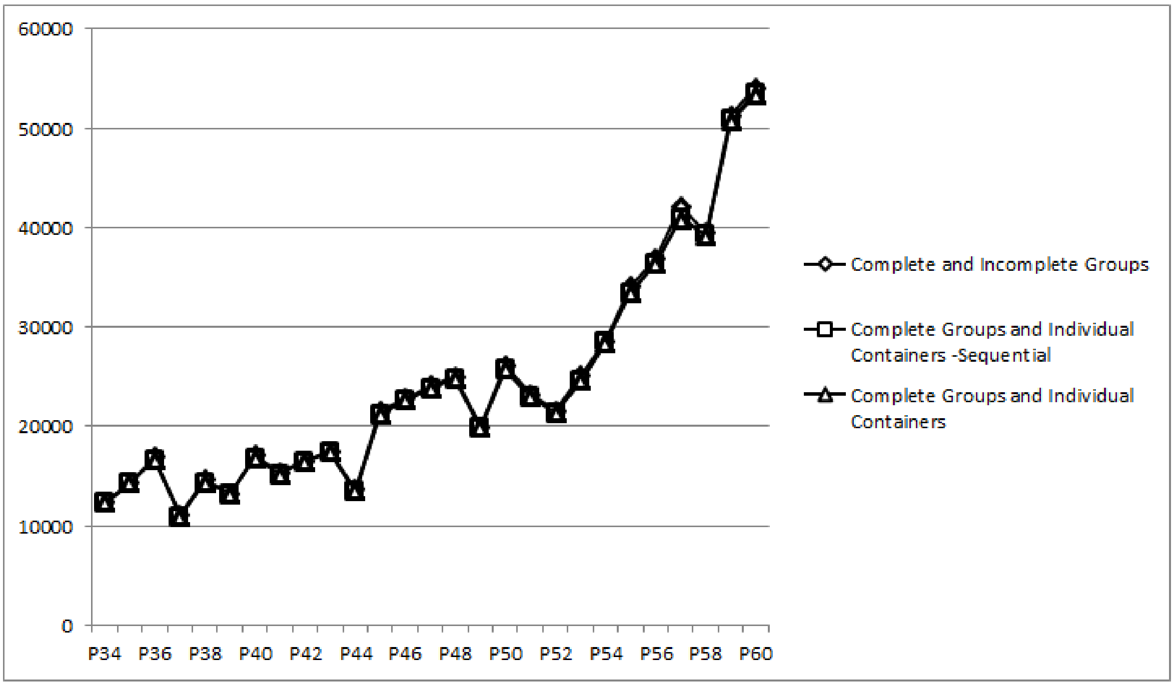

Figure 4.13. Objective values of the forth set of instances.

Table 4.19. Solution times of the forth set of instances - 1.

Problem Instances	**Complete and Incomplete Groups**	**Complete Groups and Individual Containers Sequential**	**Complete Groups and Individual Containers**
P34	423	473	510
P35	596	634	703
P36	654	683	752
P37	452	514	544
P38	606	680	715
P39	802	860	911
P40	481	532	579
P41	662	704	769
P42	841	904	957
P43	902	961	1,004
P44	1,241	1,361	1,405
P45	1,942	2,110	2,196
P46	2,358	2,567	2,704
P47	2,119	2,341	2,583
P48	2,841	3,156	3,475
P49	2,443	2,697	2,936
P50	3,204	3,524	3,869
P51	2,781	3,060	3,337
P52	3,698	4,003	4,299
P53	4,832	5,214	5,622
P54	5,903	6,485	6,711
P55	8,765	9,403	10,217

Table 4.20. Solution times of the forth set of instances - 2.

Problem Instances	Complete and Incomplete Groups	Complete Groups and Individual Containers Sequential	Complete Groups and Individual Containers
P56	10,736	11,326	11,967
P57	9,216	10,145	10,821
P58	13,156	14,263	15,087
P59	11,067	12,114	12,953
P60	14,552	16,241	17,165

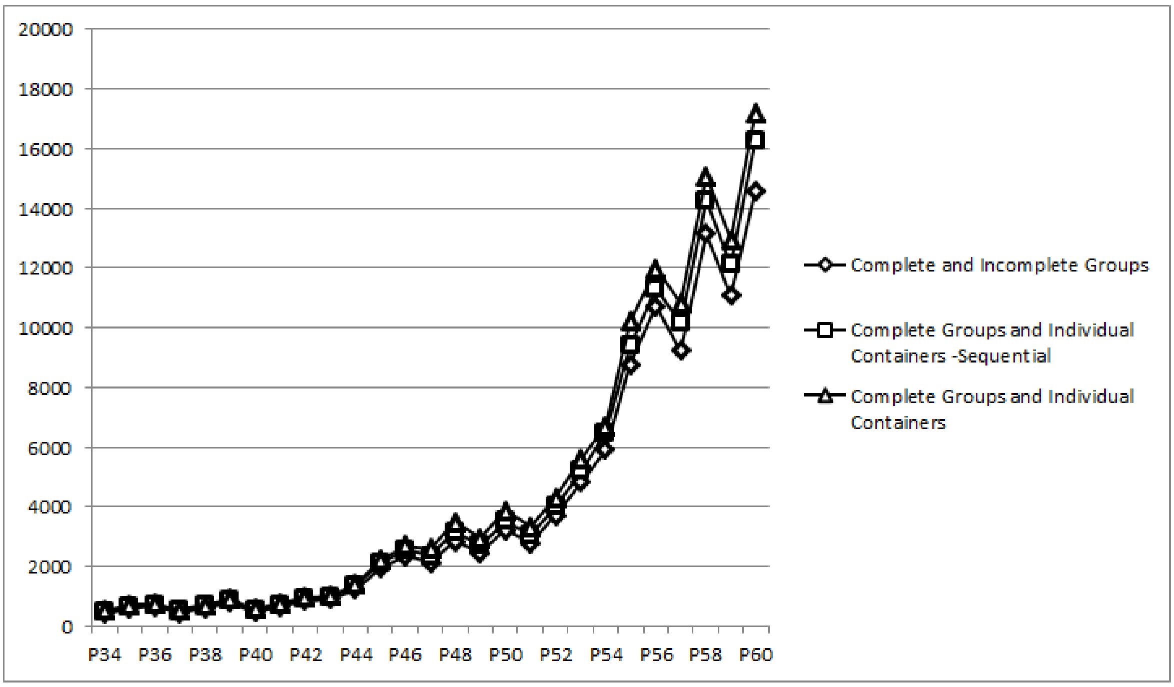

Figure 4.14. Solution times of the forth set of instances.

Groups outperforms in the forth set of instances, for larger problem instances we need to use the other approaches. As we also obtained from the previous problem instances, grouping has an advantage over non-grouping. Main reason behind this is that, by grouping we eliminate most of the reshuffling cost. Our original model follows a similar logic to decrease the reshuffling cost and when we examine the optimal non-grouping results, we can see that containers which arrive and departs with the same vessel are stacked together.

5. CONCLUSION

In this thesis we develop a mathematical model which plans the berthing times and the positions of the vessels and the storage locations of the containers in a container terminal. The model considers containers as individuals. Therefore, it is not possible to solve the real life problems in reasonable time with this model.

We use different approaches to decrease the solution time. First, we decompose our integrated model into two different models, one of which plans the berthing times and the positions of the vessels whereas other model determines the storage location of the containers. Our second approach is the rolling horizon method. We divide the time into segments and solve each time segment separately. However, the results of the rolling horizon method is not very good as our time segments have short lengths. Increasing the lengths of the time segments improves the performance. As our optimal model can be considered as a one-berth-segment rolling horizon method, objective values get better as we increase the lengths of time segments. However, this also increases the solution time. The sequential method has similar solution time problem, although it gives acceptable results for the small problem instances, we still cannot solve larger real life instances in reasonable time with the sequential method.

To solve larger problem instances, we try to group the containers. Our grouping logic is simple, containers, which arrive and depart with the same vessel belong to the same container group. As it is not possible to stack more than certain amount of containers at a storage location in a certain time period, we limit the group sizes with the maximum number of containers allowed at a storage location and define these groups as complete groups. If a group has less containers than the maximum containers allowed, then we call this group as an incomplete group. Grouping gives good results in better solution times when compared to the non-grouping results. Therefore, we increase the problem sizes and solve the problems with three grouping approaches. First, we use both complete and incomplete groups. Second approach is to break the incomplete groups and to plan these containers individually. Our third approach is to solve the second approach sequentially. We first plan the complete groups and then at the second step, we plan individual containers. Although solving the complete and the incomplete groups together outperforms in our problem instances, it may lead some infeasibilities in the larger problems. Therefore, although we can use this approach in the middle and the small-sized container terminals, we need to use other two approaches in the larger container terminals.

A set of equations is given by Kim [25] to calculate estimated number of reshuffling when departure times of the containers are extremely uncertain (i.e. when the containers departs with trucks instead of vessels or trains). Since we deal with the transit containers in this study, estimated departure times of the containers are known. However, as a future work we may consider extending our study by adding stochasticity to our problem. In such a case, these kinds of methods are needed.

REFERENCES

1. Lee, L. H., E. P. Chew, K. C. Tan and Y. Han, "An Optimization Model for Storage Yard Management in Transhipment Hubs", *OR Spectrum*, Vol. 28, pp. 539–561, 2006.

2. Steenken, D., S. Voß and R. Stahlbock, "Container Terminal Operation and Operations Research – A Classifcation and Literature Review", *OR Spectrum*, Vol. 26, pp. 3–49, 2004.

3. Stahlbock, R. and S. Voß, "Operations Research at Container Terminals: A Literature Update", *OR Spectrum*, Vol. 30, pp. 1–52, 2007.

4. Bierwirth, C. and F. Meisel, "A Survey of Berth Allocation and Quay Crane Scheduling Problems in Container Terminals", *European Journal of Operational Research*, Vol. 202, No.3, pp. 615–627, 2010.

5. Imai, A., K. Nagaiwa and C. W. Tat, "Efficient Planning of Berth Allocation for Container Terminals in Asia", *Journal of Advanced Transportation*, Vol. 31, pp. 75–94, 1997.

6. Cordeau, J. F., G. Laporte, P. Legatoyand and L. Moccia, "Models and Tabu Search Heuristics for the Berth Allocation Problem", *Transportation Science*, Vol. 39, pp. 526–538, 2005.

7. Mauri, G. R., A. C. M. Oliveira and L. A. N. Lorena, "A Hybrid Column Generation Approach for the Berth Allocation Problem", *Proceedings of the 8th European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 110–122, 2008.

8. de Oliveira, R. M., G. R. Mauri and L. A. N. Lorena, "Clustering Search for the Berth Allocation Problem", *Expert Systems with Applications*, Vol. 39, pp. 5499–

5505, 2012.

9. Buhrkal, K., 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*, Vol. 47, No.4, pp. 461–473, 2011.

10. Imai, A., E. Nishimura and S. Papadimitriou, "The Dynamic Berth Allocation Problem for a Container Port", *Transportation Research Part B*, Vol. 35, pp. 401–417, 2001.

11. Guan, Y. and R. K. Cheung, "The Berth Allocation Problem:Models and Solution Methods", *OR Spectrum*, Vol. 26, pp. 75–92, 2004.

12. Lee, Y. and C. Y. Chen, "An Optimization Heuristic for the Berth Scheduling Problem", *European Journal of Operational Research*, Vol. 196, No.2, pp. 500–508, 2008.

13. Hansen, P., C. Oğuz and N. Mladenovića, "Variable Neighborhood Search for Minimum Cost Berth Allocation", *European Journal of Operational Research*, Vol. 191, No.3, pp. 636–649, 2008.

14. Bazzazi, M., N. Safaei and N. Javadian, "A Genetic Algorithm to Solve the Storage Space Allocation Problem in a Container Terminal", *Computers and Industrial Engineering*, Vol. 56, pp. 44–52, 2009.

15. Han, Y., L. H. Lee, E. P. Chew and K. C. Tan, "A Yard Storage Strategy for Minimizing Traffic Congestion in a Marine Container Transshipment Hub", *OR Spectrum*, Vol. 30, pp. 697–720, 2008.

16. Moccia, L. and A. Astorino, "The Group Allocation Problem in a Transshipment Container Terminal", *Proceedings of World Conference on Transport Research*, 2007.

17. Preston, P. and E. Kozan, “An Approach to Determine Storage Locations of Containers at Seaport Terminals”, *Computers and Operations Research*, Vol. 28, pp. 983–995, 2001.

18. Ayachi, I., R. Kammarti, M. Ksouri and P. Borne, “Harmony Search to Solve the Container Storage Problem with Different Container Types”, *International Journal of Computer Applications*, Vol. 48, pp. 26–32, 2012.

19. Ayachi, I., R. Kammarti, M. Ksouri and P. Borne, “Harmony Search Algorithm for the Container Storage Problem”, *8th International Conference of Modeling and Simulation - MOSIM*, 2010.

20. Cordeau, J. F., G. Laporte, L. Moccia and G. Sorrentino, “Optimizing Yard Assignment in an Automotive Transshipment Terminal”, *European Journal of Operational Research*, Vol. 215, pp. 149–160, 2011.

21. Lee, D. H., J. G. Jin and J. H. Chen, “Terminal and Yard Allocation Problem for a Container Transshipment Hub with Multiple Terminals”, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 48, pp. 516–528, 2012.

22. Lee, D. H., J. X. Cao, Q. Shi and J. H. Chen, “A Heuristic Algorithm for Yard Truck Scheduling and Storage Allocation Problems”, *Transportation Research Part E*, Vol. 45, pp. 810–820, 2009.

23. Cao, J., Q. Shi and D. H. Lee, “A Decision Support Method for Truck Scheduling and Storage Allocation Problem at Container”, *Tsinghua Science and Technology*, Vol. 34/67, pp. 211–216, 2008.

24. Meisel, F. and C. Bierwirth, “Heuristics for the Integration of Crane Productivity in the Berth Allocation Problem”, *Transportation Research Part E*, Vol. 45, No.3, pp. 196–209, 2008.

25. Kim, K. H., “Evaluation of the Number of Rehandles in Container Yards”, *Com-*

puters and Industrial Engineering, Vol. 32, pp. 701–711, 1997.