# Simultaneous Berth Allocation and Quay Crane Assignment Problem

by

Senem Sancar

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, 2014

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Senem Sancar

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:

Assoc. Prof. Emre Alper Yıldırım

Prof. Deniz Özdemir

Prof. Serpil Sayın

Prof. Fikri Karaesmen

Prof. Çağatay Başdoğan

Date:

## ABSTRACT

Effective container terminal management is becoming more important since sea transportation constitutes over 80% of all global freight transportation by volume. Berth allocation and quay crane assignment constitute two important operations whose enhancements lead to improved service levels at container terminals. In this thesis, we address the problem of reducing the service time of vessels and improving quay crane assignments. We develop a bi-objective integer linear programming model to solve berth allocation (BAP) and quay crane assignment (QCAP) problems simultaneously. We assume a discrete berth structure and dynamic vessel arrivals. The first objective of our model is to minimize the total service time of vessels (waiting plus handling times). The second objective is to minimize the number of quay crane setups. We conduct computational experiments to evaluate the effectiveness of the model by using real data obtained from the Port of Izmir Container Terminal. For both objectives, the solutions obtained from our model are considerably better than the solutions implemented at the terminal. We obtain Pareto optimal solution sets in an attempt to examine the interplay between the two objective functions.

# ÖZETÇE

Uluslararası ticaretin % 80'i deniz ulaşımı yoluyla yapılmakta olduğundan etkin konteyner terminali yönetimi limanlar için oldukça yüksek önem arz etmektedir. Rıhtım atama ve vinç atama işlemlerinde kaydedilen iyileştirmeler limanlardaki hizmet kalitesinin iyileştirilmesinde önemli rol oynamaktadır. Bu tezde limana gelen gemilerin hizmet alma sürelerini azaltma ve vinç atamalarını iyileştirme problemleri ele alınmaktadır. Rıhtım atama ve vinç atama problemlerini eş zamanlı çözmek icin çok amaçlı bir tamsayılı doğrusal eniyileme modeli geliştirilmiştir. Ele alınan problemde rıhtımların parçalı yapıda oldukları ve gemilerin dinamik şekilde geldikleri varsayılmıştır. Modelimizin ilk amacı gemilerin toplam hizmet alma süresini (limanda bekleme ve hizmet alma sürelerinin toplamı) enküçüklemek iken ikinci amacı ise toplam vinç kurulum sayılarını enküçüklemektir. Modelimizin etkinliğini test etmek için İzmir Limanı'ndan gerçek veriler alınmış ve kullanılmıştır. Ortaya çıkan Pareto en iyi sonuçlar kümesi iki hedefin arasındaki etkileşimi görmeyi sağlamaktadır. Modelden elde edilen çözümler, limanda uygulananlara kıyasla iki amaç fonksiyonu için de önemli iyileştirmeler sağlamıştır.

## ACKNOWLEDGMENTS

The project was supported in part by TÜBİTAK (The Scientific and Technological Research Council of Turkey) Grant Number 112M865. We gratefully acknowledge this support.

# TABLE OF CONTENTS

## LIST OF TABLES

# LIST OF FIGURES

# NOMENCLATURE

| | | |
|---|---|---|
| $P1$ | : | minimization of service time |
| $CP1$ | : | minimization of service time subject to a crane setup constraint |
| $P2$ | : | minimization of number of crane setups |
| $CP2$ | : | minimization of number of crane setups subject to a service time constraint |

# Chapter 1

# INTRODUCTION

Sea transportation is the safest and the most cost effective way of transportation compared with other types of transportations. It plays a vital role in container transportation as over 80% of all international trade by volume is done via sea transport [1]. Therefore, container terminals are crucial players in global logistics. As the demand for container transportation increases, there is a need for improved terminal operations at container terminal ports. In order to meet the increasing demand and provide better services for incoming vessels, container terminals may choose to construct new berths and increase the number of terminal equipment by doing sizeable investments or by effective berth planning and effective use of terminal equipment. Improvement of operational efficiency is much more cost effective than adding new equipment to the port. Therefore, the improvement of the operational efficiency of container terminals is crucial in container terminal management.

## 1.1 Container Terminal Operations

The operations of a container terminal can be categorized as seaside operations, yard operations, and landside operations. The relationships between these operations are shown in Figure 1.1.

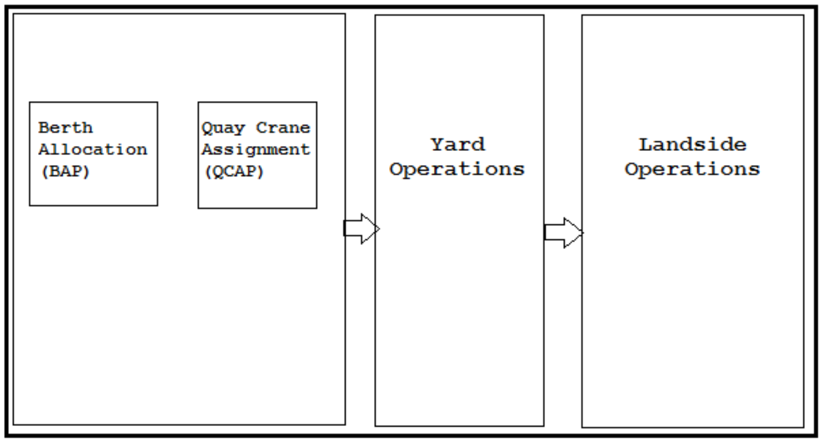


Figure 1.1: Container terminal operations

The process starts with seaside operations. After a vessels arrives at the port, it is assigned to a suitable berth. Then, cranes are assigned to the vessel located at a berth. Seaside operations continue with the yard operations, the assignment process of yard cranes to vessels in order to transport the containers handled by quay cranes to the required places in the port. The last stage consists of landside operations, which deals with the movement of containers within the port [8]. Berth allocation and crane assignment are two important seaside terminal operations whose increased efficiency also increases the operating capacity of the container terminal.

## *1.2 Decision Problems*

Berth allocation problem (BAP) is the allocation of quay space and assignment of berthing time for incoming vessels. There are two types of berth structures used for BAP, continuous and discrete [14]. In a discrete berth structure, the quayside is partitioned into a fixed number of berths. For this structure, no more than one vessel can be assigned to a specific berth at any given time. In a continuous berth structure, the quayside is not partitioned into berths and the vessels can be assigned to suitable locations along the quayside as long as the total lengths of vessels do not exceed the length of the quayside. Berth allocation problem with a discrete berth structure is called BAPD, while BAP with a continuous berth structure is referred to as BAPC.

Vessel arrivals are classified as static and dynamic [14]. For the static arrival of vessels, it is assumed that vessels are already present at the port and are available to be assigned to berths. On the other hand, in a dynamic arrival scheme, future arrival times of vessels are given and a vessel therefore cannot be assigned to any berth before its arrival time.

The second seaside operation, quay crane assignment problem (QCAP), is the assignment of quay cranes to vessels at berths. The quay cranes can be either portable or static. The static cranes can only move on railways along the quayside while the portable ones can move independently between berths.

The movement of quay cranes is a slow and time consuming task. Whenever a quay crane is assigned to a vessel located at a berth, the crane needs to be moved towards the vessel and positioned properly before container handling operations can begin. We therefore refer to this operation as a crane setup.

## 1.3 Motivation and Contributions

For container terminals, it is vital to meet the demand of customers in a timely and cost-effective manner. In order to improve the service level, service times of vessels can be reduced by an effective berth allocation and crane assignment procedure. In order to reduce operating costs, crane utilizations can be improved by an effective scheduling scheme. Our motivation in this thesis is to increase the efficiency of container terminals.

Berth assignment and crane scheduling must be handled together in order to take into consideration the joint effect of these operations on the service level and terminal utilization. In this thesis, we develop a bi-objective integer linear programming model with the objectives of minimizing total service time of all vessels and minimizing the number of crane setups. The service time of a vessel is defined as the sum of the waiting time of the vessel at the port and the container handling time after the vessel is assigned to a berth. In our model, we assume a discrete berth structure and dynamic vessel arrivals.

We test our model using real data obtained from the Port of Izmir Container Terminal. Our computational results reveal that the solutions obtained from our model yield considerable improvements over the solutions implemented at the port for both objectives.

## 1.4 Organization

This thesis is organized as follows: In Chapter 2, we review the literature on the berth allocation problem, quay crane allocation problem, and the integrated approach. In Chapter 3, we give a detailed definition of our problem and present our bi-objective integer linear programming model. Computational results are presented and discussed in Chapter 4. Chapter 5 concludes the thesis.

# Chapter 2

# LITERATURE REVIEW

Container terminal management has gained utmost importance over the recent centuries. Increasing container vessel traffic results in higher demand for efficient terminal operations. Several related operations are performed at container terminals. Berth allocation, quay crane assignment and quay crane scheduling are considered to be the most important operations as their proper management may lead to improved service levels at a reduced cost.

## *2.1 Berth Allocation Problem*

Berth Allocation Problem (BAP) is described as the allocation of berth spaces along a quayside for incoming vessels. Vessels arrive at the port and they need to be assigned to berths for the service (unloading/loading of containers) to start. The objective is to improve the quality of service by reducing the service time of vessels. Considering the berth layout, BAP can be broken down into two classes: the discrete BAP and the continuous BAP.

### *2.1.1 Discrete Berth Allocation Problem*

In a discrete berth structure, the quayside is partitioned into berth areas. In a discrete layout, one berth can serve at most one vessel at a time. Considering the arrival times of vessels, two types of BAP are introduced in [11], the static and the dynamic BAP. In the static BAP, all the vessels are assumed to be located at the port and can immediately be assigned to berths. In the dynamic version of BAP, fixed arrival times are given for each vessel and berth assignment of a vessel cannot start before its arrival time [11].

We first review the literature on the discrete BAP. Imai et al. [10] address BAP for commercial container terminals with static arrival of vessels. In most of the container terminals, berth allocation is performed on a first-come-first-served (FCFS) basis. In order to increase the throughput of container terminals, Imai et al. [10] state that vessel-to-berth assignments can be improved by using a different scheme than FCFS. However, the use of a different scheme may result in customer dissatisfaction due the order of service. In order to increase berth performance and decrease customer dissatisfaction with the order of service, they propose a bi-objective nonlinear optimization problem. They develop a heuristic method to find a non-inferior set of solutions with respect to the aforementioned two objectives.

Imai et al. [11] extend the static version of BAP to the dynamic setting. Similar to [10], the main objective is to minimize the sum of service times of all vessels. To solve this problem, they propose a mixed integer programming model. They use a subgradient method to solve the Lagrangian relaxation.

Imai et al. [12] study a modified version of the discrete BAP with dynamic arrivals by adding certain service priorities for incoming vessels.

Theofanis et al. [20] develop a genetic algorithm based heuristic method to solve the discrete BAP with dynamic arrivals. The main objective of their study is to minimize the total weighted service time taking into account different service priorities. They present an MILP (mixed integer linear programming) model, which is an extension of the MILP model in [12].

Imai et al. [13] address the discrete BAP with static arrivals at multi-user container terminals with limited quay capacities. Vessels are typically served at container terminals. However, in container terminals with limited quay capacities, when the expected waiting time of a vessel exceeds the time limit, it is assigned to an external terminal. An integer linear programming model is proposed with the objective of minimizing the total service time of vessels at the external terminal. They develop a genetic algorithm based heuristic method to solve this problem.

Hansen et al. [7] study the discrete BAP with dynamic arrivals. They propose a mixed linear integer programming model. Their model takes the weights of vessels as priority criteria. Vessel handling times and handling costs depend on the berth that they are assigned to. They develop a heuristic method to solve this problem.

Nishimura et al. [18] extend the dynamic version of BAP studied by Imai et al. [11] by adding multi-water depth constraints to the problem. These constraints ensure that vessels can only be served at berths with suitable water depths. They propose a genetic algorithm to solve this problem.

Xu et al. [22] study continuous BAP with additional water-depth constraints under the assumptions of both static and dynamic arrivals. They aim to minimize the total weighted service time of the vessels, where the weights correspond to the unit waiting costs of vessels. As a solution methodology, they propose efficient heuristic methods for both of the problems.

### *2.1.2 Continuous Berth Allocation Problems*

In a continuous berth layout, there is no partition along a quayside. Vessels can be located anywhere along quaysides [4]. Having a continuous berth structure generally provides more flexibility and may lead to higher container terminal utilization. However, the continuous BAP is in general more difficult than the discrete BAP.

Lim [17] studies the continuous BAP with dynamic arrivals. The objective is to minimize the maximum amount of quay space used at any time under the assumption that once a vessel is assigned to a quay space, it will not be moved until its departure time. He develops a heuristic method to solve this problem.

Imai at al. [14] study the continuous BAP with the objective of minimizing the total service time under the assumption that handling times of vessels depend on their berthing positions along the quayside. They propose a heuristic method to find a solution.

## 2.2 Quay Crane Allocation Problem

Quay crane allocation problem (QCAP) deals with the assignment of quay cranes to vessels. Quay crane movements are limited in container terminals, as they usually move on tracks along the quayside. Furthermore, static quay cranes are only allowed to move along a railway and their relative positions are fixed. Quay cranes constitute important container terminal equipment since they are usually limited in number and incorrect assignment of quay cranes may result in reduced operational efficiency of container terminals. In addition, ineffective quay crane assignments may lead to unnecessary crane movements and reduced handling capacity, which may lead to increased costs and reduced service levels.

Dazango [5] proposes a heuristic method to efficiently schedule cranes by minimizing the total cost of delay in waiting times of vessels for the berth assignments.

The quay crane dynamic assignment problem is introduced by Liang et al. [15]. They propose a multi-objective optimization model. The first objective aims to minimize the total container handling times, the waiting times and delay for every vessel and the second objective minimizes the number of crane movements. To solve this problem, the authors propose a multi-objective hybrid genetic algorithm method.

Zeng et al. [24] consider the integrated handling operations of quay cranes, yard trailers, and yard cranes. Quay cranes load/unload containers onto/from vessels and yard trailers move the containers handled by quay cranes from quayside to storage yard or vice versa. Zeng et al. [24] consider the multi-crane oriented crane scheduling problem, in which yard trailers and quay cranes are considered as integrated terminal equipment with the objective of minimizing the total service time.

## 2.3 Integrated BAP and QCAP Approach

Improving berth allocation and quay crane scheduling are important for terminal operations as they may lead to improved service levels and reduced operating costs. Reducing service times of vessels increases both the terminal utilization and customer satisfaction. In many studies, berth allocation and quay crane assignment problems are considered separately. However, berth allocation and quay crane assignment jointly determine the service time of vessels.

Liang et al [16] consider BAP and QCAP simultaneously. They propose a model under the assumptions of a discrete berth structure and dynamic vessel arrivals. They impose a limit on the maximum number of quay cranes that can be assigned to each vessel and handling times of vessels depend on the number of quay cranes assigned to them. Different from our study, they take into consideration the service order of vessels and define priorities for each vessel by considering the first-come first-served (FCFS) principle. They use identical quay cranes and present an integer linear programming model with the objective of minimizing the service times of vessels. They propose hybrid approaches that combine genetic algorithm with a heuristic method to find an approximate solution for this problem.

Hu [8] also considers BAP and QCAP as integrated problems and develops an integer linear programming model to find the optimal assignment of quay cranes with minimum quay crane movements. In this study, he first solves the continuous BAP with dynamic arrivals and then uses the computed solution of BAP as an input to QCAP.

Zeng et al. [23] also consider BAP and QCAP as integrated problems. They remark that the planned schedules of berth allocation and quay crane assignment may fail due to unexpected events such as bad weather conditions. They develop an algorithm and propose two schedules for quay crane assignments and berth allocations. One of them is used as a backup plan against unexpected events.

An MILP (mixed integer linear programing) model for the integrated BAP and QCAP is presented by Raa et al. [19]. In their model, they consider vessel priorities, berthing positions, and handling times of vessels. They discretize the time horizon for quay crane assignments. The model uses a single objective, which consists of three components. The first component deals with the handling times of vessels, which includes penalties for vessel handling delays. The priorities may be given for vessels according to their level of importance. The second component of the objective function deals with the berthing location of vessels and includes a penalty for the deviation of vessels from their preferred berthing positions. The third component of their objective function reduces the number of quay crane movements by penalizing the changes in the number of cranes assigned to a vessel during its service time. The increase in the number of quay cranes working on a vessel results with penalty costs whereas the opposite situation is not charged with a penalty cost. They aim to minimize the sum of all three components. The model proposed by Raa et al. [19] can solve real life instances in a reasonable amount of time.

Imai et al. [9] develop a model for the simultaneous berth-quay crane allocation problem with the objective of total service time minimization. The formulation of berth allocation and quay crane assignment problem is partitioned into two parts as BAP and QCAP. For BAP, they assume a discrete berth structure and dynamic arrival of vessels. They develop separate models for BAP and QCAP. In order to propose an integrated formulation for BAP and CAP, they add constraints of QCAP to the model developed for BAP. They propose a heuristic method to find an approximate solution.

Türkoğulları et al. [21] address the continuous BAP with dynamic arrivals. They formulate an integer linear programming model for the integrated berth allocation and quay crane assignment problems (BACAP). They extend their study by adding crane scheduling and they propose a mixed integer linear programming model (MILP) for the integrated berth allocation and quay crane assignment and scheduling problems (BACASP). While the BACAP can be solved for large instances in a reasonable amount of times, the BACASP can only be solved for relatively small instances. In order to solve the BACASP for larger instances, they propose a cutting plane algorithm. Their algorithm generates cuts and adds them to the model for solving BACAP. Therefore, an optimal solution of BACASP can be obtained from an optimal solution of BACAP.

## 2.4 Our Focus

In the literature, most of the studies consider berth allocation and quay crane assignment problems independently from each other. However, the service time of a vessel, which is the sum of waiting and handling times, is closely related to both berth assignments and quay crane allocations. The waiting times are reduced by minimizing the time that vessels wait to be assigned to berths and handling times are reduced by making sure that an appropriate berth and a sufficient number of cranes are available to serve each vessel.

In our model, we deal with berth allocation and quay crane assignment problems simultaneously. We propose a bi-objective integer linear programming model to solve the simultaneous berth allocation and quay crane assignment problem. The first objective of our model is to minimize the total service time of incoming vessels. The second objective minimizes the number of quay crane setups. For these two objectives, we find the set of Pareto optimal solutions in an attempt to reveal the trade-off between the two objectives.

In an attempt to reduce the solution time of our model, we introduce valid inequalities that impose lower bounds on the minimum number of periods a vessel needs to stay at a berth until all of its containers are handled. These valid inequalities often result in improved solution times.

Our focus in this thesis is on computing exact solutions for the simultaneous berth allocation and quay crane assignment problem. Different from previous studies in the literature, we consider two different but related objectives. Our approach enables us to quantify the trade-off between the total service time of vessels and the number of quay crane setups.

Our computational results on the real data obtained from the Port of İzmir Container Terminal reveal that the solutions obtained from our model reduce the total service time by 59% and the number of crane setups by 43% on average.

Chapter 3

# PROBLEM DEFINITION AND OPTIMIZATION MODEL

## 3.1 Problem Definition

In this study, we consider the berth allocation and quay crane assignment problems as an integrated problem. Each incoming vessel is assigned to a berth and then cranes are assigned to each vessel at a berth. We adopt a bi-objective approach. The first objective of the model is to minimize the total service time, which consists of handling and waiting times. The second objective of the model is to minimize the number of quay crane setups. In order for a crane to serve a vessel, the crane needs to be moved and properly positioned next to the vessel. Therefore, each crane assignment to a vessel requires an initial setup.

Another objective of our study is to understand how these two objectives are related to each other. For this purpose, we develop a bi-objective integer linear programming model for the simultaneous berth allocation and quay crane assignment problem under the following assumptions:

- (A0) The planning horizon is divided into a finite number of periods.
- (A1) Each vessel should be served.
- (A2) Each vessel should be assigned to exactly one berth.
- (A3) A vessel cannot get service before its arrival.
- (A4) A vessel cannot leave the berth before all of its containers are handled.

- (A5) At most one vessel can be assigned to a berth in any time period.
- (A6) There is a limit on the maximum number of quay cranes that can be assigned to a vessel.
- (A7) There are two types of cranes.
    1. Non-portable cranes move on a railway along the quayside and the order of cranes cannot be changed (for instance, if cranes are lined up as crane 1, crane 2, crane 3 etc., crane 1 cannot move ahead of crane 2).
    2. Portable cranes can move independently among berths.
- (A8) Each crane may have a different handling rate.
- (A9) Berths may have different and specified lengths.
- (A10) Vessels may have different lengths.
- (A11) A vessel can only be assigned to a berth if the length of the vessel does not exceed the length of the berth.
- (A12) All vessels have to be served by the end of the planning horizon.
- (A13) When a vessel is assigned to a berth, it immediately starts to get service.
- (A14) The service of a vessel ends when there are no containers left to be handled.

## 3.2 Optimization Model

In this section, we present our integer linear programming model. We start by defining our parameters, sets, and decision variables.

**Definitions**

$I$ : number of vessels

$J$: number of cranes

$K$: number of berths

$T$: number of periods

**Sets**

$V = \{1, \ldots, I\}$ (set of vessels)

$C = \{1, \ldots, J\}$ (set of cranes)

$B = \{1, \ldots, K\}$ (set of berths)

$P = \{1, \ldots, T\}$ (set of time periods)

**Parameters**

$l_i$ : vessel length including safety margin for vessel $i$ , $i \in V$

$Q_k$ : length of berth $k$, $k \in B$

$a_i$ : arrival time of vessel $i$ , $i \in V$

$N0_i$ : number of containers that require handling for vessel $i$, $i \in V$

$R_j$ : container handling rate of crane $j$ per time period, $j \in C$

$U$ : maximum number of cranes that can be assigned to a vessel in any period.

$nc$ : number of non-portable cranes (we assume that non-portable cranes are labeled $1, \ldots, nc$ and portable cranes are labeled $nc + 1, \ldots, J$.)

**Decision Variables**

$y_{ijtk}$ : 1 if vessel $i$ is served by crane $j$ at berth $k$ during time period $t$, $i \in V, j \in C, t \in P, k \in B$

$x_{ijt}$ : 1 if crane $j$ is assigned to vessel $i$ during time period $t$, $i \in V, j \in C, t \in P$

$BV_{itk}$ : 1 if vessel $i$ is assigned to berth $k$ during time period $t$, $i \in V, t \in P, k \in B$

$\delta_{ik}$ : 1 if vessel $i$ is assigned to berth $k$, $i \in V, k \in B$

$YH_{it}$ : 1 if vessel $i$ is assigned to any berth during time period $t$, $i \in V, t \in P$

$p_{it}$ : 1 if at the beginning of time period $t$ there exist containers that still need to be handled for vessel $i$, $i \in V, t \in P$

$N_{it}$ : number of containers remaining to be handled for vessel $i$ at the beginning of time period $t$, $i \in V, t = 1, ..., (T+1)$

$\alpha_i$ : number of periods vessel $i$ spends at the port, $i \in V$

$CR_{ijt}$: 1 if crane $j$ starts working on vessel $i$ at time period $t$, $i \in V, j \in C, t \in P$

**Mathematical Model**

$f_1(service\ time) : \min \sum_{i \in V} \alpha_i$

$f_2(crane\ setup) : \min \sum_{i \in V} \sum_{j \in C} \sum_{t \in P: t \geq a_i} CR_{ijt}$

subject to

$$\alpha_i \geq (t - a_i + 1) \,.\, p_{it} \qquad i \in V, t \in P \qquad (1)$$

$$p_{it} \leq N_{it} \qquad i \in V, t \in P \qquad (2)$$

$$N_{it} \leq N0_i \,.\, p_{it} \qquad i \in V, t \in P \qquad (3)$$

$$BV_{i,(t+1),k} \geq BV_{itk} + p_{i(t+1)} - 1 \qquad i \in V, t = 1, .., (T-1), k \in B \qquad (4)$$

$$BV_{itk} \leq \delta_{ik} \qquad i \in V, t = a_i, .., T, k \in B \qquad (5)$$

$$BV_{itk} = 0 \qquad i \in V, t = 1, .., (a_i - 1), k \in B \qquad (6)$$

$$\sum_{k \in B} \delta_{ik} = 1 \qquad i \in V \qquad (7)$$

$$\sum_{k \in B} BV_{itk} = YH_{it} \qquad i \in V, t \in P \qquad (8)$$

$$N_{it} - \sum_{j \in C} R_j \,.\, x_{ijt} \leq N_{i,(t+1)} \qquad i \in V, t = a_i, \ldots, (T-1) \qquad (9)$$

$$\sum_{i \in V} BV_{itk} \leq 1 \qquad t \in P, k \in B \qquad (10)$$

$$l_i \,.\, \delta_{ik} \leq Q_k \qquad i \in V, k \in B \qquad (11)$$

$$\sum_{i \in V} x_{ijt} \leq 1 \qquad j \in C, t \in P \qquad (12)$$

$$\sum_{k \in B} y_{ijtk} = x_{ijt} \qquad i \in V, j \in C, t \in P \qquad (13)$$

$$\sum_{j \in C} x_{ijt} \leq U \,.\, YH_{it} \qquad i \in V, t \in P \qquad (14)$$

$$\sum_{j \in C} y_{ijtk} \leq U \,.\, BV_{itk} \qquad i \in V, t \in P, k \in B \qquad (15)$$

$$\sum_{j \in C} \sum_{(t \in P : t \geq a_i)} \sum_{k \in B} y_{ijtk} \geq 1 \qquad i \in V \qquad (16)$$

$$N_{it} = N0i \qquad i \in V, t = 1, ..a_i \qquad (17)$$

$$N_{i,(T+1)} \leq 0 \qquad i \in V \qquad (18)$$

$$\sum_{i \in V} \sum_{j' \in C:(j' \geq (j+1))} \sum_{k' \in B:(k' \leq (k-1))} yij'tk' \leq I \,.\, (nc - j) \,.\, (k-1) \,.\, (1 - \sum_{i \in V} y_{ijtk}) \qquad j = 1, ..nc, t \in P, k \in B \qquad (19)$$

$$CR_{ijt} = 0 \qquad i \in V, j \in C, t = 1, ..(a_i - 1) \qquad (20)$$

$$CR_{ijt} \geq x_{ijt} - x_{i,j,(t-1)} \qquad i \in V, j \in C, t = 2, ..T \qquad (21)$$

$$CR_{ij1} \geq x_{ij1} \qquad i \in V, j \in C \qquad (22)$$

$$YH_{it} \leq p_{it} \qquad i \in V, t = a_i, ..T \qquad (23)$$

$$\sum_{j \in C} x_{ijt} \geq YH_{it} \qquad i \in V, t \in P \qquad (24)$$

$$y_{ijtk} \in \{0,1\} \qquad i \in V, j \in C, t \in P, k \in B \tag{25}$$

$$x_{ijt} \in \{0,1\} \qquad i \in V, j \in C, t \in P \tag{26}$$

$$BV_{itk} \in \{0,1\} \qquad i \in V, t \in P, k \in B \tag{27}$$

$$\delta_{ik} \in \{0,1\} \qquad i \in V, k \in B \tag{28}$$

$$YH_{it} \in \{0,1\} \qquad i \in V, t \in P \tag{29}$$

$$p_{it} \in \{0,1\} \qquad i \in V, t \in P \tag{30}$$

$$CR_{ijt} \in \{0,1\} \qquad i \in V, j \in C, t \in P \tag{31}$$

$$N_{it} \qquad i \in V, t \in P \tag{32}$$

$$\alpha_i \in \mathbb{Z} \qquad i \in V \tag{33}$$

The general process for a vessel is as follows. The vessel arrives at the port and starts to get service after it is assigned to a berth and a crane is allocated to it. When all containers are handled, the vessel leaves the port. The time interval between the arrival and departure of a vessel is defined as the service time of a vessel. The first objective $f_1$ minimizes the total service time of all vessels. When a crane is assigned to a vessel for the first time, this process requires a setup. The second objective $f_2$ in our model minimizes the number of crane setups.

Constraint set (1) is used to compute the service time for each vessel. Constraint sets (2) and (3) relate the variables $p_{it}$ and $N_{it}$. The constraints (4) ensure that vessels cannot switch berths while they are served. The constraints (5) through (7) imply that each vessel can be assigned to exactly one berth after its arrival time. Constraint set (8) relates the variables $BV_{itk}$ and $YH_{it}$. For each time period, constraint set (9) sets the relationship between the number of containers that are yet to be handled for each vessel and the number of cranes which are assigned to that vessel by considering the container handling rates of those cranes. Constraint set (10) ensures that, during any time period, there cannot be more than one vessel assigned to a berth. Constraint set (11) assures that each vessel should be assigned a suitable berth, i.e., the length of a vessel cannot exceed the length of the berth that it is assigned to. In constraint set (12), it is guaranteed that a crane cannot serve more than one vessel in the same period. Constraint set (13) relates the variables $x_{ijt}$ and $y_{ijtk}$. Constraint sets (14) and (15) guarantee that at most a given number ($U$) of cranes can be assigned to each vessel in each period. Constraint set (16) guarantees that, for each vessel, there should be at least one crane assignment. Constraint set (17) initializes the number of containers that needs to be handled for each vessel. Constraint set (18) ensures that all of the containers on each vessel are handled before the end of the time period. By constraint set (19), it is stated that, for the assignment of cranes to berths, the order of non-portable cranes should be respected in each period. For instance, if crane $j$ serves a vessel at berth $k$ in a time period, then no other crane with a larger label than $j$ (e.g., cranes $j+1$, $j+2$,...), can serve any vessel at any berth with a smaller label than $k$ in the same time period (e.g., berths $k-1$, $k-2$,...). Constraint set (20) prevents crane to vessel assignments before the arrival of vessels. Constraint sets (21) and (22) ensure the connection between the variables $CR_{ijt}$ and $x_{ijt}$. Constraint set (23) relates the variables $YH_{it}$ and $p_{it}$. By constraint set (24), if a vessel is assigned to a berth, at least one crane must be present at the berth to serve the vessel. The constraints (25) through (33) specify the range of each decision variable.

### 3.3 Valid Inequalities

In the previous section, we presented our bi-objective integer linear programming model for the simultaneous berth allocation and quay crane assignment problem. In an attempt to reduce the solution time of the model, we consider adding valid inequalities to our model. Our valid inequalities impose lower bounds on the minimum number of periods required for a vessel to stay at a berth until all of its containers are handled. Therefore, we refer to these valid inequalities as "minimum period constraints".

The maximum number of quay cranes that can be assigned to a vessel is $U$. Therefore, to calculate the minimum number of period requirements for each vessel, we consider the largest number of quay cranes with the largest handling rates. We sort the quay cranes in non-increasing order of their container handling rates, and choose the largest $U$ of them. Then, we divide the total number of containers to be handled for a vessel by the sum of container handling rates of chosen cranes, and calculate the minimum number of time periods.

We now explain this procedure in more detail. The first step is to sort quay cranes in non-increasing order of their container handling rates. Here, $R_j$ represents the container handling rate of crane $j$.

By relabeling the cranes (if necessary), we may assume that

$$R_1 \geq R_2 \geq ... \geq R_{J-1} \geq R_J.$$

The second step is to calculate the minimum number of time periods each vessel needs to spend at a berth. We denote by $m_i$ the minimum number of time periods that vessel $i$ needs to spend at a berth. The required minimum number of time periods is computed by dividing the total number of containers to be handled for vessel $i$ by the sum of the container handling rates of the maximum allowable number of quay cranes:

$$m_i = \left\lceil \frac{N0_i}{\sum_{j=1}^{U} R_j} \right\rceil, \qquad i \in V.$$

The minimum period constraint is therefore given by

$$\sum_{t:t \geq a_i} YH_{it} \geq m_i, \qquad i \in V. \tag{3.1}$$

By imposing minimum time requirement for each vessel, we aim to strengthen our formulation. We discuss the effects of these valid inequalities on the solution time in the next chapter.

# Chapter 4

# COMPUTATIONAL EXPERIMENTS

In this thesis, we use real data obtained from the Port of Izmir Container Terminal. The computational experiments have been conducted to evaluate the effectiveness of the proposed model in comparison with solutions implemented at the port.

## *4.1 Port of Izmir Container Terminal*

The Port of Izmir Container Terminal is the second biggest container terminal in Turkey after the Port of Mersin Container Terminal. It plays an important role for the container transportation in Turkey since the port is connected to the rest of the country via railroad and highway networks [3]. After the arrival of vessels at the container terminal, each one is assigned to a suitable berth. After the assignment of vessels to the berths, cranes are assigned to vessels.

The map of the Port of Izmir Container Terminal is shown in Figure 4.1. The port consists of continuous quaysides, and quay cranes move along these quaysides to serve the vessels. The container terminal has seven quay cranes, five of which are static and two are portable. The static quay cranes move on a railway along the quayside. The order of static cranes cannot be changed whereas the portable cranes can move independently among the quaysides.

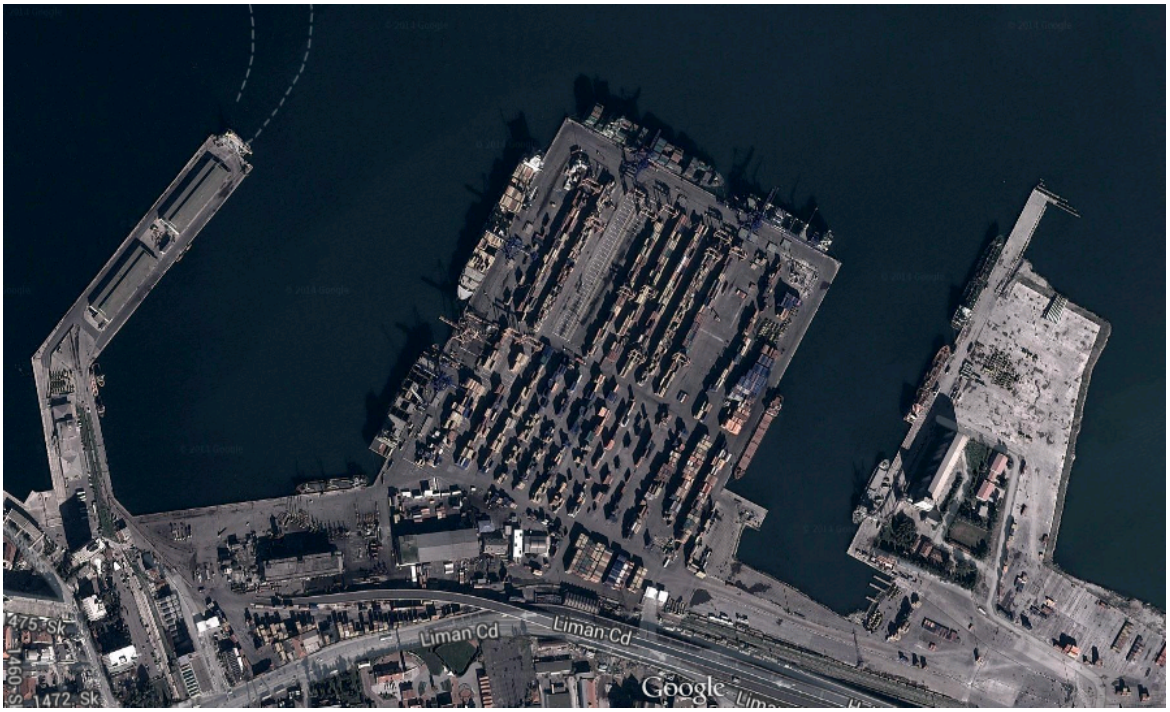


Figure 4.1: Port of Izmir Container Terminal

### *4.1.1 Berth Structure*

The total length of the port equals 3.3 kilometers [3]. The Port of Izmir Container Terminal has a continuous berth structure. As seen in Figure 4.1, there are six non-partitioned quaysides and four of them can be used by container vessels. Incoming vessels are assigned to quaysides according to their lengths and the availability of quaysides. More than one vessel can be assigned to the same quayside as long as the total length (including safety margins) of vessels does not exceed the length of the quayside.

### *4.1.2 Quay Crane Information*

There are seven non-identical quay cranes (QC) in the container terminal. The handling rates of QC 1, QC 3, QC 4,QC 5 and QC 7 are 15 TEU per hour and those of QC 2 and QC6 are 14 TEU per hour. TEU (twenty-feet equivalent unit) is an inexact unit of cargo capacity often used to describe the capacity of container vessels [2]. Two of the quay cranes are portable and can move among the berths, while the remaining five quay cranes are assumed to be static. The static quay cranes are labeled $1, \ldots, 5$ and the portable cranes are labeled 6 and 7.

#### *4.1.3 Time Periods*

At the container terminal, each day is divided into three shifts; the first 8 hours of day (between 00:00 and 08:00) is defined as Shift 1, between 08:00 and 16:00 is defined as Shift 2 and the last 8 hours (16:00 to 00:00) is defined as Shift 3.

#### *4.1.4 Incoming Vessel Information*

The data obtained from the container terminal includes names, lengths, arrival times, and container information of vessels. The data spans one year and the average number of vessel arrivals per week is about 21.

## 4.2 Data Adaptation

The data set is composed of two parts. The first part consists of the names of vessels, lengths of vessels, information about the berths they were assigned to, their times of arrivals and departures, and the total number of containers handled for each vessel. The second part of the data includes daily information that gives vessel to berth assignments in each shift together with their assignment times.

### *4.2.1 Time Period Designation*

In order to determine the length of time periods to use in the model, we analyzed the data. The data set includes aggregated information for each shift. For instance, there are instances when a crane works on two different vessels during the same shift. In contrast, we assume that each crane can be assigned to exactly one vessel in each time period. Therefore, we decided to divide each shift used by the container terminal into two 4-hour time periods. By using 4-hour time periods, a crane that was assigned to two different vessels in a shift can now be assumed to work on one of those vessels for one period and on the other one for the other period.

### *4.2.2 Planning Horizon*

For this study, we have chosen the four busiest weeks of the container terminal. We generated four data sets by using information of these four weeks. We decided to use one week as our planning horizon due to increased uncertainties in the arrival times of vessels for longer durations. Note that arrival times of vessels are given as an input to our model.

In these data sets, one time period consists of 4 hours. Therefore, there are 6 periods per day and at least 42 periods per week. We determined the exact number of periods based on the real life solution. The actual length of the planning horizon is determined by the departure time of the last vessel from the port.

### *4.2.3 Conversion of Continuous Berth Structure to Discrete Berth Structure*

The container terminal has a continuous berth structure. However, in our model, we assume a discrete berth structure. Therefore, we needed to modify the data.

We devised a berth partitioning algorithm to convert the continuous berth structure to an appropriate discrete berth structure. Berth conversion is done according to the vessel-berth assignment data obtained from the Port of Izmir. The berth conversion algorithm applied to each of the four quaysides is presented below:

**Berth Conversion Algorithm**

Initializations

$Q$ : number of quaysides

$T$ : number of time periods

$V$ : number of vessels

$M_{tk}$ : set of vessels assigned to quayside $k$ at time period $t$, t= 1,...,$T$, k=1,...,$Q$

$B_k$ : number of berths at quayside $k$, $k \in Q$

$l_{jk}$ : the length of the $j^{th}$ berth at quayside $k$, $k \in Q$

Algorithm

**step 0:**

for $k = 1$ to $Q$

**step 1:**

$B_k = max\ \{\ |M_{tk}|\ \}, \qquad t = 1, \ldots, T$

**step 2:**

for $j = 1$ to $B_k$

$l_{jk}$ = max {length of the $j^{th}$ longest vessel in $M_{tk}$: $|M_{tk}| \geq j$ } , $t = 1, \ldots, T$

end for

end for

The algorithm is illustrated on a small example. In this example, there is a single quayside and vessels are assigned to this quayside. The planning horizon consists of six periods. The time periods are given in the first column and the lengths of vessels (in non-increasing order) that are at the quayside at each time period are given in the second set of columns of Table 4.1.

Table 4.1: Example of Berth Conversion

| Time Period | vessel 1 | vessel 2 | vessel 3 | vessel 4 | vessel 5 | vessel 6 |
|---|---|---|---|---|---|---|
| 1 | 210 | 180 | 95 | 80 | -- | -- |
| 2 | 210 | 120 | -- | -- | -- | -- |
| 3 | -- | -- | -- | -- | -- | |
| 4 | 250 | 160 | 110 | -- | -- | -- |
| 5 | 190 | 155 | 100 | 90 | 80 | 50 |
| 6 | 220 | 160 | 100 | 80 | -- | -- |

For each quayside, we perform the following steps:

- The first step is to determine the number of vessels that are assigned to the quayside at any time period. The number of vessels in the busiest period determines the number of berths. In Example 4.1, it is seen that the busiest period is Period 5 and the number of berths defined for this quayside is therefore equal to 6.

- In the second step, we determine the length of each berth. For each time period $t$ $(t = 1, \ldots, T)$, we examine the length of the $j^{th}$ longest vessel to determine the length of the $j^{th}$ berth. $(j = 1, \ldots, B_k)$ and the length of the longest such vessel determines the length of the berth. In the example given in Table 4.1, the length of Berth 1 is given by the length of the longest vessel, which is equal to 250. The length of Berth 2 equals 180, which is the length of the second longest vessels among all periods that have at least two vessels. In a similar manner, the lengths of Berths 3, 4, and 5 equal 110, 90 and 50, respectively.

The main idea behind the berth conversion algorithm is to define the smallest number of berths and to determine the length of each berth in such a way that the real-life solution can be converted into a feasible solution for the resulting discrete structure. In conclusion, converting the continuous structure to a discrete enabled us to use the container terminal data in our model.

## *4.3 Data Sets*

We analyzed the data obtained from the container terminal by counting the number of vessels that arrive at the port on a yearly, monthly, and daily basis. To make a fair comparison and to demonstrate the effectiveness of our model, we aimed to solve the problem for the busiest weeks of the container terminal. We generated four data sets obtained from the data of the four busiest weeks. We henceforth refer to those four data sets as Data Set 1, 2, 3 and 4 respectively.

Each data set consists of the number of vessels, the lengths of the vessels (with safety margins), their arrival times (in periods), and the container information of the vessels.

The number of berths can vary for each data set since the algorithm may compute a different number of berths on different data sets.

There are seven quay cranes (QC). The handling rates of quay cranes are not identical. Quay cranes 1, 3, 4, 5, and 7 each has a handling rate of 60 TEU/period, whereas quay cranes 2 and 6 each has a handling rate of 56 TEU/period. Quay cranes 6 and 7 are assumed to be portable and move independently among the berths, whereas the remaining five cranes are assumed to be static. The maximum allowable number of quay cranes assigned to a vessel in a period is equal to four.

In each data set, one period consists of 4 hours. The number of time periods may vary for different data sets depending on the departure time of the last vessel.

Table 4.2 gives general information about each of the four data sets.

Table 4.2: Summary of Data Sets

| Data Set | # of Vessels | # of Periods | # of Berths |
|---|---|---|---|
| 1 | 22 | 48 | 5 |
| 2 | 25 | 50 | 4 |
| 3 | 26 | 46 | 7 |
| 4 | 23 | 50 | 8 |

Tables 4.3 through 4.10 present all the detailed information about the Data Sets 1 through 4, respectively.

Table 4.3: Vessel Information for Data Set 1 (22 Vessels and 48 Time Periods)

| Vessel | Length(including safety margin) | Arrival Time | # of Containers |
|---|---|---|---|
| 1 | 208 | 2 | 364 |
| 2 | 184 | 3 | 181 |
| 3 | 207 | 4 | 176 |
| 4 | 167 | 8 | 468 |
| 5 | 148 | 10 | 281 |
| 6 | 125 | 10 | 222 |
| 7 | 149 | 10 | 268 |
| 8 | 184 | 12 | 295 |
| 9 | 122 | 15 | 82 |
| 10 | 216 | 16 | 350 |
| 11 | 222 | 21 | 310 |
| 12 | 157 | 22 | 443 |
| 13 | 260 | 26 | 283 |
| 14 | 166 | 30 | 263 |
| 15 | 222 | 30 | 617 |
| 16 | 200 | 31 | 353 |
| 17 | 213 | 35 | 309 |
| 18 | 155 | 39 | 237 |
| 19 | 156 | 39 | 286 |
| 20 | 149 | 42 | 215 |
| 21 | 216 | 42 | 173 |
| 22 | 169 | 42 | 480 |

Table 4.4: Berth Information for Data Set 1 (5 Berths)

| | Berth 1 | Berth 2 | Berth 3 | Berth 4 | Berth 5 |
|---|---|---|---|---|---|
| Lengths of Berths (m) : | 222 | 125 | 260 | 207 | 184 |

Table 4.5: Vessel Information for Data Set 2 (25 Vessels and 50 Time Periods)

| Vessel | Length(including safety margin) | Arrival Time | # of Containers |
|---|---|---|---|
| 1 | 133 | 3 | 80 |
| 2 | 184 | 6 | 480 |
| 3 | 195 | 7 | 495 |
| 4 | 125 | 6 | 180 |
| 5 | 180 | 6 | 418 |
| 6 | 168 | 12 | 354 |
| 7 | 149 | 14 | 158 |
| 8 | 243 | 18 | 671 |
| 9 | 184 | 15 | 608 |
| 10 | 207 | 25 | 237 |
| 11 | 149 | 23 | 400 |
| 12 | 122 | 24 | 71 |
| 13 | 223 | 31 | 274 |
| 14 | 148 | 26 | 342 |
| 15 | 173 | 26 | 341 |
| 16 | 134 | 27 | 289 |
| 17 | 101 | 28 | 81 |
| 18 | 184 | 31 | 676 |
| 19 | 177 | 33 | 756 |
| 20 | 244 | 34 | 1531 |
| 21 | 149 | 34 | 294 |
| 22 | 216 | 32 | 631 |
| 23 | 101 | 39 | 82 |
| 24 | 155 | 42 | 378 |
| 25 | 184 | 42 | 651 |

Table 4.6: Berth Information for Data Set 2 (4 Berths)

| | Berth 1 | Berth 2 | Berth 3 | Berth 4 |
|---|---|---|---|---|
| Lengths of Berths (m) : | 244 | 184 | 243 | 184 |

Table 4.7: Vessel Information for Data Set 3 (26 Vessels and 46 Time Periods)

| Vessel | Length(including safety margin) | Arrival Time | # of Containers |
|---|---|---|---|
| 1 | 222 | 2 | 249 |
| 2 | 155 | 5 | 595 |
| 3 | 180 | 6 | 412 |
| 4 | 101 | 7 | 282 |
| 5 | 149 | 9 | 307 |
| 6 | 184 | 9 | 414 |
| 7 | 294 | 11 | 650 |
| 8 | 172 | 10 | 396 |
| 9 | 148 | 16 | 177 |
| 10 | 122 | 20 | 197 |
| 11 | 125 | 18 | 322 |
| 12 | 117 | 20 | 46 |
| 13 | 202 | 22 | 674 |
| 14 | 156 | 23 | 11 |
| 15 | 176 | 21 | 344 |
| 16 | 207 | 21 | 408 |
| 17 | 122 | 28 | 39 |
| 18 | 123 | 29 | 139 |
| 19 | 265 | 30 | 905 |
| 20 | 101 | 32 | 283 |
| 21 | 223 | 33 | 377 |
| 22 | 149 | 36 | 662 |
| 23 | 180 | 30 | 366 |
| 24 | 188 | 43 | 1205 |
| 25 | 166 | 38 | 280 |
| 26 | 217 | 42 | 282 |

Table 4.8: Berth Information for Data Set 3 (7 Berths)

| Lengths of | Berth 1 | Berth 2 | Berth 3 | Berth 4 | Berth 5 | Berth 6 | Berth 7 |
|---|---|---|---|---|---|---|---|
| Berths (m) : | 294 | 149 | 223 | 166 | 122 | 184 | 156 |

Table 4.9: Vessel information for Data Set 4 (23 Vessels and 50 Time Periods)

| Vessel | Length(including safety margin) | Arrival Time | # of Containers |
|---|---|---|---|
| 1 | 185 | 2 | 302 |
| 2 | 155 | 5 | 636 |
| 3 | 117 | 8 | 117 |
| 4 | 101 | 12 | 50 |
| 5 | 149 | 9 | 221 |
| 6 | 180 | 11 | 404 |
| 7 | 101 | 10 | 113 |
| 8 | 150 | 12 | 152 |
| 9 | 173 | 16 | 490 |
| 10 | 101 | 17 | 244 |
| 11 | 217 | 20 | 176 |
| 12 | 201 | 21 | 580 |
| 13 | 124 | 23 | 263 |
| 14 | 117 | 24 | 225 |
| 15 | 121 | 24 | 27 |
| 16 | 265 | 28 | 1624 |
| 17 | 175 | 24 | 292 |
| 18 | 222 | 26 | 205 |
| 19 | 187 | 33 | 1100 |
| 20 | 208 | 32 | 682 |
| 21 | 155 | 27 | 404 |
| 22 | 140 | 36 | 238 |
| 23 | 167 | 38 | 300 |

Table 4.10: Berth Information for Data Set 4 (7 Berths)

| Length of | Berth 1 | Berth 2 | Berth 3 | Berth 4 | Berth 5 | Berth 6 | Berth 7 | Berth 8 |
|---|---|---|---|---|---|---|---|---|
| Berths (m): | 265 | 180 | 101 | 222 | 208 | 117 | 175 | 155 |

### 4.4 Computer Specifications

The computational experiments were performed on a workstation that consists of 24 processors (Xeon E5-2640 Core 12 with 2.50 GHz) and 48 GB RAM. The operating system of the machine is Linux. The optimization model was coded on OPL and IBM ILOG CPLEX Optimizer version 12.5 was used to solve the MILP problems.

### 4.5 Computational Results

The computational experiments were performed to evaluate the efficiency and the advantages of the model over the real life solution implemented at the container terminal. In this study, we define and solve four different problems. In the first problem, we aim to minimize the sum of service times of all vessels, referred to as $P1$. The second problem is the minimization of total service time of vessels subject to a constraint on the number of crane setups, referred to as $CP1$. In the third problem, called $P2$, we aim to minimize the number of crane setups. The fourth problem, denoted by $CP2$, is concerned with the minimization of the number of crane setups subject to a constraint on the total service times of all vessels.

We solve each of these four problems on each data set and compare the solutions obtained from our model to the real life solutions implemented at the container terminal. The table of comparisons is given at the end of the explanations of each data set. More specifically, the results from our model and the solution implemented at the container terminal are presented in Tables 4.11, 4.12, 4.13, and 4.14 for Data Sets 1, 2, 3, and, 4, respectively.

We defined a 6-day (518.400 sec) time limit to the CPLEX solver. For each problem and each data set, we report the best feasible solution computed within the time limit. The percentages next to the solutions (if any) denote the gap information upon termination. The entries '-' indicate that no feasible solution was found by the solver within the time limit.

#### *4.5.1 Results for $P1$*

In $P1$, we aim to minimize the total service time. Comparing our solutions with the implemented solutions at the container terminal, we observe that we have an improvement of 65% for Data Set 1, 52% for Data Set 2, 56% for Data Set 3, and 63% for Data Set 4.

#### *4.5.2 Results for $CP1$*

In $CP1$, our objective is to minimize the total service time of vessels subject to a constraint on the total number of crane setups. For each data set, we minimize the total service time of vessels subject to the constraint that total number of crane setups does not exceed the total number of crane setups for the solution implemented at the port. Our main objective was to measure the extent of improvement in the total service time of vessels while having the same performance measure on the number of crane setups. Our computational results reveal that we improve the total service time by about 42% for Data Set 1, and and by 53% for Data Set 4. The solver could not compute feasible solutions for Data Sets 2 and 3 in the 6-day (518.400 sec) time limit.

#### *4.5.3 Results for $P2$*

In $P2$, our objective is to minimize the total number of crane setups. For $P2$, in most of the instances, the solver is terminated due to the time limit. However, using the best feasible solutions computed by the solver, we still have 49% improvement over the container terminal's solution for Data Set 1, 24 % improvement for Data Set 2, 45% improvement for Data Set 3, and 53% improvement for Data Set 4.

### *4.5.4 Results for $CP2$*

In CP2, our goal is to minimize the number of crane setups subject to the constraint that the total service time of vessels does not exceed the total service time of the solution implemented at the port. Once again, we are interested in the extent of improvement in the number of care setups while having the same performance measure on the total service time of vessels. For Data Set 1, we improved the number of quay crane usages by about 42%, for Data Set 2 by 40%, for Data Set 3 by 41%, for Data Set 4 by 52%.

Figures A.2, A.6, A.10, and A.14 display the graphs of the real life solutions implemented at the Port of Izmir Container Terminal. Figures A.3, A.7, A.11, and A.15 give a graphical representation of the optimal solutions for the problem of $P1$. Figures A.4, A.8, A.12, and A.16 display solutions for the problem of $CP2$. Figures A.5, A.9, A.13, and A.17 illustrate solutions for the problem of $P2$.

Table 4.11: Data Set 1

| **Data Set 1** | Total Service Time | Number of Crane Setups | Time |
|---|---|---|---|
| Container Terminal | 114 | 45 | - |
| $P1$: min obj1 | 40 | 84 | 15.09 sec |
| $CP1$: min obj1 (obj2 $\leq$ 45) | 65 (6.29%) | 45 | 518402.36 sec |
| $P2$: min obj2 | 224 | 23 | 95193.33 sec |
| $CP2$: min obj2 (obj1 $\leq$ 114) | 114 | 26 (18.57%) | 518426.60 sec |

Table 4.12: Data Set 2

| **Data Set 2** | Total Service Time | Number of Crane Setups | Time |
|---|---|---|---|
| Container Terminal | 138 | 61 | - |
| $P1$: min obj1 | 66 | 99 | 1055.38 sec |
| $CP1$: min obj1 (obj2 $\leq$ 61) | - | - | - |
| $P2$: min obj2 | 231 | 46 (51.28%) | 518425.95 sec |
| $CP2$: min obj2 (obj1 $\leq$ 138) | 138 | 36 (33.98%) | 518432.23 sec |

Table 4.13: Data Set 3

| **Data Set 3** | Total Service Time | Number of Crane Setups | Time |
|---|---|---|---|
| Container Terminal | 148 | 58 | - |
| $P1$: min obj1 | 65 | 98 | 76.16 sec |
| $CP1$: min obj1 (obj2 $\leq$ 58) | - | - | - |
| $P2$: min obj 2 | 255 | 32 (35.37%) | 518401.78 sec |
| $CP2$: min obj2 (obj1 $\leq$ 148) | 148 | 34 (35.34%) | 518482.32 sec |

Table 4.14: Data Set 4

| **Data Set 4** | Total Service Time | Number of Crane Setups | Time |
|---|---|---|---|
| Container Terminal | 156 | 53 | - |
| $P1$: min obj1 | 57 | 84 | 262.26 sec |
| $CP1$: min obj1 (obj2 $\leq$ 53) | 72 (6.40%) | 53 | 518401.30 sec |
| $P2$: min obj2 | 378 | 25 | 240166.51 sec |
| $CP2$: min obj2 (obj1 $\leq$ 156) | 156 | 25 (30.34%) | 518429.57 sec |

### *4.5.5 Discussion*

When we analyze the results of our experiments on each of the four data sets, it is clear that our model has significant improvements other real-life solutions implemented at the terminal. For each problem and for each data set, the percentage of improvements varies. On average, we obtained 59% improvement on the total service time and we obtained 43% improvement on the number of crane setups. We obtained a 47% improvement on the total service time and 44% improvement on the number of crane setups while achieving the same performance on the other objective as the solution implemented at the port. Note that the improvement in the number of crane setups in the constrained problem seems to be better due to the quality of the best solutions the solver could compute within the time limit.

### *4.5.6 Pareto Optimal Solution Set*

As expected, our results illustrate that there is a negative correlation between the number of quay crane setups and the total service time of vessels. We compute the Pareto optimal solution set in order to examine the interplay between the two objective functions. For bi-objective optimization problems, the Pareto optimal set is a subset of the set of feasible points of solutions for which one of the objective function values cannot be improved without worsening the other one.

We use the epsilon constraint method to compute the set of Pareto optimal solutions. The epsilon constraint method is a well-known method for solving multi-criteria optimization problems [6]. For bi-objective problems, the epsilon constraint method minimizes one of the objectives whereas the other one is imposed as a constraint.

We outline the epsilon constraint method below:

### Epsilon Constraint Method

**step 0:** min obj2 $= M$

**step 1:** min obj1 $= a$

**step 2:** max obj1$= b$

**step 3:** $\epsilon = a$

**step 4:**

min obj2

subject to

obj1 $\leq \epsilon$

$\epsilon = \epsilon + 1$

Continue **step 4** until obj2 reaches its minimum value $M$ or obj1 reaches its maximum value $b$.

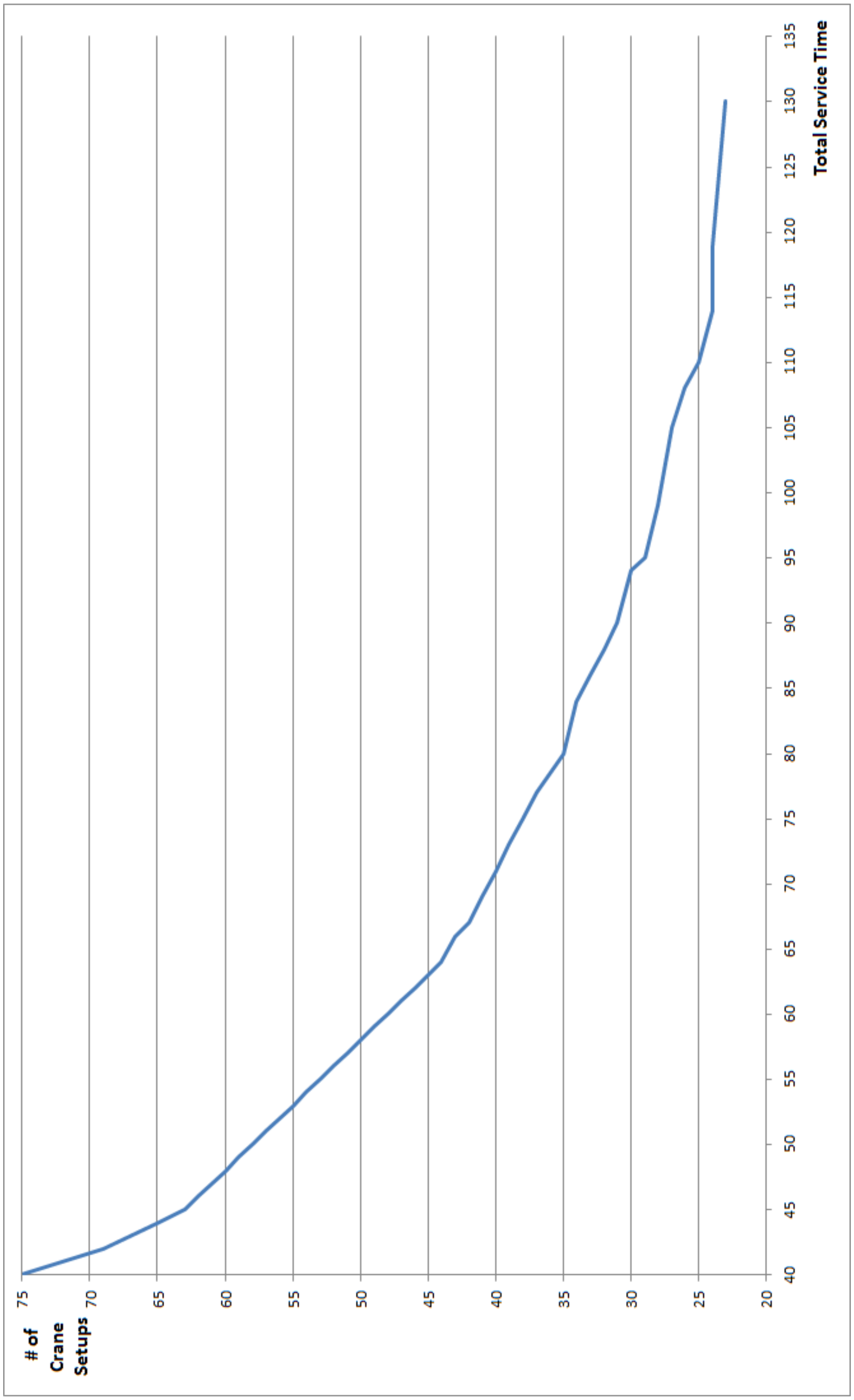


Figure 4.2: Pareto Optimal Solution

For our problem, we chose minimizing the number of crane setups as our original objective and added the total service time requirement as a constraint. First, we solved both of the problems $P1$ and $P2$ for different data sets and we analyzed the solution times. We observed that the problem of $CP2$ (min obj2 s.t. obj1 $\leq$ some value) is relatively easier to solve than $CP1$ (min obj1 s.t. obj2 $\leq$ some value) . In order to find the Pareto optimal solution set, we decided to solve the problem $CP2$. To reach a solution for each problem we increased the minimum service time constraint (obj1) by one unit and calculated the corresponding number of minimum crane setups required. The process ended when the number of crane setups reached its minimum value. For the problem $CP2$ until obj1 increases from 40 to 59, the computed obj2 values are optimal. Beyond that threshold, the results obtained for obj2 are the best feasible solutions computed by CPLEX within a 2-hour time limit. The results are presented in Table 4.15.

Table 4.15: Pareto Optimal Solution Set

| Solution ID | $f_1$: Total Service Time (periods) | $f_2$: Number of Crane Setups | GAP |
|---|---|---|---|
| 1 | 40 | 75 | 0 % |
| 2 | 41 | 72 | 0 % |
| 3 | 42 | 69 | 0 % |
| 4 | 43 | 67 | 0 % |
| 5 | 44 | 65 | 0 % |
| 6 | 45 | 63 | 0 % |
| 7 | 46 | 62 | 0 % |
| 8 | 47 | 61 | 0 % |
| 9 | 48 | 60 | 0 % |
| 10 | 49 | 59 | 0 % |
| 11 | 50 | 58 | 0 % |
| 12 | 51 | 57 | 0 % |
| 13 | 52 | 56 | 0 % |
| 14 | 53 | 55 | 0 % |
| 15 | 54 | 54 | 0 % |
| 16 | 55 | 53 | 0 % |
| 17 | 56 | 52 | 5.86 % |
| 18 | 57 | 51 | 5.26 % |
| 19 | 58 | 50 | 0 % |
| 20 | 59 | 49 | 0 % |
| 21 | 60 | 48 | 2.96 % |
| 22 | 61 | 47 | 0 % |
| 23 | 62 | 46 | 2.95 % |
| 24 | 63 | 45 | 6.36 % |
| 25 | 64 | 44 | 11.03 % |
| 26 | 66 | 43 | 8.12 % |
| 27 | 67 | 42 | 11.78 % |
| 28 | 69 | 41 | 11.43 % |
| 29 | 71 | 40 | 6.79 % |
| 30 | 73 | 39 | 14.80 % |
| 31 | 75 | 38 | 14.54 % |
| 32 | 77 | 37 | 15.53 % |
| 33 | 80 | 35 | 17.55 % |
| 34 | 84 | 34 | 21.40 % |
| 30 | 86 | 33 | 21.64 % |
| 31 | 88 | 32 | 20.59 % |
| 32 | 90 | 31 | 21.68 % |
| 33 | 94 | 30 | 26.44 % |
| 34 | 95 | 29 | 20.67 % |
| 35 | 99 | 28 | 19.10 % |
| 36 | 105 | 27 | 25.31 % |
| 37 | 108 | 26 | 21.80 % |
| 38 | 110 | 25 | 19.10 % |
| 39 | 119 | 24 | 18.80 % |
| 40 | 130 | 23 | 19.20 % |

**Note : Solutions with zero gap information are optimal solutions.**

### *4.5.7 Analysis of Valid Inequalities*

In this section, we analyze the effects of valid inequalities on the solution times of each problem on each data set. After the addition of minimum period constraints, the performance of our model on problems $P1$, $CP1$, $P2$, and $CP2$ for each data set is compared with the performance without the valid inequalities. The comparison of results for problems $P1$, $CP1$, $P2$ and $CP2$ are presented for Data Set 1 in Tables 4.16, 4.17 4.19 and 4.18, for Data Set 2 in Tables 4.20, 4.21, 4.23 and 4.22, for Data Set 3 in Tables 4.24, 4.25, 4.27 and 4.26 and for Data Set 4 in Tables 4.28, 4.29, 4.31 and 4.30, respectively.

In each of these tables, the second column gives statistics about the model without valid inequalities whereas the third column presents the results after the addition of valid inequalities. The objective function values of the problems, optimality gap information, and the solution times of problems are given in the first three rows. The number of constraints, variables, binary variables, non-zeros, and simplex iterations are also given in the following rows.

We solved the problems $P1$, $CP1$, $P2$, and $CP2$ under a time limit of 518.400 sec (6 days). The solutions given with 0% gap represent optimal solutions. The solutions which have positive gap indicate the best feasible solutions computed by the solver under the time limit.

Table 4.16: Computational Results for **P1** for Data Set 1

| | | without valid inequality | after ”min period constraint” |
|---|---|---|---|
| Objective 1 | | 40 | 40 |
| Total time | | 15.09 sec | 10.38 sec |
| GAP | | 0% | 0% |
| Reduced MIP has | constraints | 13488 | 13510 |
| | variables | 16046 | 16046 |
| | nonzeros | 109233 | 109778 |
| | binaries | 15511 | 15511 |
| Iterations | | 4579 | 3910 |
| Objective 2 | | 84 | 84 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.17: Computational Results for **CP1** for Data Set 1

| | | without valid inequality | after ”min period constraint” |
|---|---|---|---|
| Objective 1 | | 65 | 63 |
| Given objective 2 | | 45 | 45 |
| GAP | | 6.29% | 7.54% |
| Total time | | 518402.36 sec | 518402.56 sec |
| Reduced MIP has | constraints | 16913 | 19635 |
| | variables | 24053 | 24053 |
| | nonzeros | 141616 | 142205 |
| | binaries | 23468 | 23468 |
| Iterations | | 258063671 | 166106466 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.18: Computational Results for **P2** for Data Set 1

| | | without valid inequality | after "min period constraint" |
|---|---|---|---|
| Objective 2 | | 23 | 23 |
| GAP | | 0% | 0% |
| Total time | | 95193.33sec | 130747.62 sec |
| Reduced MIP has | constraints | 19049 | 19271 |
| | variables | 24031 | 24031 |
| | nonzeros | 13667 | 136956 |
| | binaries | 23468 | 23468 |
| Iterations | | 21026606 | 38137235 |
| Objective 1 | | 179 | 160 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.19: Computational Results for **CP2** for Data Set 1

| | | without valid inequality | after "min period constraint" |
|---|---|---|---|
| Objective 2 | | 26 | 26 |
| Given objective 1 | | 114 | 114 |
| GAP | | 18.57% | 26.00% |
| Total time | | 518426.60 sec | 518434.98 sec |
| Reduced MIP has | constraints | 19613 | 19635 |
| | variables | 24053 | 24053 |
| | nonzeros | 137515 | 138104 |
| | binaries | 23468 | 23468 |
| Iterations | | 100921570 | 117057315 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.20: Computational Results for **P1** for Data Set 2

| | | without valid inequality | after ”min period constraint” |
|---|---|---|---|
| Objective 1 | | 66 | 66 |
| Total time | | 1055.38 sec | 891.37 sec |
| GAP | | 0% | 0% |
| Reduced MIP has | constraints | 16982 | 17007 |
| | variables | 21088 | 21088 |
| | nonzeros | 132573 | 133264 |
| | binaries | 20413 | 20413 |
| Iterations | | 3048065 | 2741852 |
| Objective 2 | | 99 | 99 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.21: Computational Results for **CP1** for Data Set 2

| | | without valid inequality | after "min period constraint" |
|---|---|---|---|
| Objective 1 | | - | 83 |
| Given objective 2 | | - | 61 |
| GAP | | - | 10.84 % |
| Total time | | - | 518401.93 sec |
| Reduced MIP has | constraints | - | 21364 |
| | variables | - | 25444 |
| | nonzeros | - | 150851 |
| | binaries | - | 24769 |
| Iterations | | - | 144259800 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.22: Computational Results for **P2** for Data Set 2

| | | without valid inequality | after "min period constraint" |
|---|---|---|---|
| Objective 2 | | 46 | 44 |
| GAP | | 51.28 % | 41.45% |
| Total time | | 518425.95 sec | 518401.44 sec |
| Reduced MIP has | constraints | 20688 | 20713 |
| | variables | 25419 | 25419 |
| | nonzeros | 144341 | 145032 |
| | binaries | 24769 | 24769 |
| Iterations | | 87981666 | 101766212 |
| Objective 1 | | 231 | 233 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.23: Computational Results for **CP2** for Data Set 2

| | | without valid inequality | after ”min period constraint” |
|---|---|---|---|
| Objective 2 | | 36 | 37 |
| GAP | | 33.98% | 38.89% |
| Given objective 1 | | 138 | 138 |
| Total time | | 518433.48 sec | 518409.76 sec |
| Reduced MIP has | constraints | 21339 | 21364 |
| | variables | 25444 | 25444 |
| | nonzeros | 145666 | 146357 |
| | binaries | 24769 | 24769 |
| Iterations | | 107110056 | 126473394 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.24: Computational Results for **P1** for Data Set 3

| | | without valid inequality | after "min period constraint" |
|---|---|---|---|
| Objective 1 | | 65 | 65 |
| GAP | | 0% | 0% |
| Total time | | 76.16 sec | 82.92 sec |
| Reduced MIP has | constraints | 18111 | 18137 |
| | variables | 22646 | 22646 |
| | nonzeros | 195805 | 196477 |
| | binaries | 21992 | 21992 |
| Iterations | | 126515 | 206438 |
| Objective 2 | | 98 | 98 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.25: Computational Results for **CP1** for Data Set 3

| | | without valid inequality | after "min period constraint" |
|---|---|---|---|
| Objective 1 | | - | - |
| Given objective 2 | | - | - |
| GAP | | - | - |
| Total time | | - | - |
| Reduced MIP has | constraints | - | - |
| | variables | - | - |
| | nonzeros | - | - |
| | binaries | - | - |
| Iterations | | - | - |
| Time limit | | 518400 sec | 518400 sec |

Table 4.26: Computational Results for **P2** for Data Set 3

| | | without valid inequality | after "min period constraint" |
|---|---|---|---|
| Objective 2 | | 32 | 34 |
| GAP | | 35.37 % | 34.94% |
| Total time | | 518401.78 sec | 518413.65 sec |
| Reduced MIP has | constraints | 21337 | 21363 |
| | variables | 26474 | 26474 |
| | nonzeros | 206111 | 206783 |
| | binaries | 25846 | 25846 |
| Iterations | | 194225102 | 122770266 |
| Objective 1 | | 255 | 240 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.27: Computational Results for **CP2** for Data Set 3

| | | without valid inequality | after ”min period constraint” |
|---|---|---|---|
| Objective 2 | | 34 | 33 |
| Given objective 1 | | 148 | 148 |
| GAP | | 35.34% | 34.57% |
| Total time | | 518482.32 sec | 518327.64 sec |
| Reduced MIP has | constraints | 21966 | 21992 |
| | variables | 26500 | 26500 |
| | nonzeros | 207393 | 208065 |
| | binaries | 25846 | 25846 |
| Iterations | | 108382431 | 88003451 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.28: Computational Results for **P1** for Data Set 4

| | | without valid inequality | after "min period constraint" |
|---|---|---|---|
| Objective 1 | | 57 | 57 |
| GAP | | 0% | 0% |
| Total time | | 262.26 sec | 112.42 sec |
| Reduced MIP has | constraints | 22047 | 16868 |
| | variables | 28926 | 21440 |
| | nonzeros | 270579 | 200432 |
| | binaries | 28227 | 20910 |
| Iterations | | 406993 | 144190 |
| Objective 2 | | 84 | 84 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.29: Computational Results for **CP1** for Data Set 4

| | | without valid inequality | after "min period constraint" |
|---|---|---|---|
| Objective 1 | | 72 | 72 |
| Given objective 2 | | 53 | 53 |
| GAP | | 6.40% | 8.32% |
| Total time | | 518401.30 sec | 518433.38 sec |
| Reduced MIP has | constraints | 26294 | 26317 |
| | variables | 33172 | 33172 |
| | nonzeros | 287702 | 288417 |
| | binaries | 32473 | 32473 |
| Iterations | | 320375053 | 260109626 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.30: Computational Results for **P2** for Data Set 4

| | | without valid inequality | after "min period constraint" |
|---|---|---|---|
| Objective 2 | | 25 | 25 |
| GAP | | 0 % | 6.82 % |
| Total time | | 240166.51 sec | 518401.87 sec |
| Reduced MIP has | constraints | 25617 | 25460 |
| | variables | 33149 | 33149 |
| | nonzeros | 281965 | 282680 |
| | binaries | 32473 | 32473 |
| Iterations | | 34320475 | 57323801 |
| Objective 1 | | 231 | 317 |
| Time limit | | 518400 sec | 518400 sec |

Table 4.31: Computational Results for **CP2** for Data Set 4

| | | without valid inequality | after "min period constraint" |
|---|---|---|---|
| Objective 2 | | 25 | 27 |
| Given objective 1 | | 156 | 156 |
| GAP | | 30.34% | 43.15% |
| Total time | | 518429.57 sec | 518432.32 sec |
| Reduced MIP has | constraints | 26294 | 26317 |
| | variables | 33172 | 33172 |
| | nonzeros | 283340 | 284055 |
| | binaries | 32473 | 32473 |
| Iterations | | 78881267 | 76092504 |
| Time limit | | 518400 sec | 518400 sec |

### 4.5.8 Discussion After the Addition of Valid Inequalities

When we analyze the effect of valid inequalities on the solution time of the model for the problems $P1$, $P2$, and $CP2$, we observe that valid inequalities have different effects on different data sets. While the addition of valid inequalities may reduce the solution time of one problem for a data set, it may increase the solution time of the same problem for another data set.

Tables 4.32, 4.33, 4.34 and 4.35 present solution times of $P1$, $P2$, $CP1$ and $CP2$ for Data Sets 1, 2, 3 and 4.

Table 4.32: Solution Times for P1

| $P1$ | CPU Time before Valid Inequalities | CPU Time after Valid Inequalities |
|---|---|---|
| Data Set 1 | 15.09 sec | 10.38 sec |
| Data Set 2 | 1055.38 sec | 891.37 sec |
| Data Set 3 | 76.16 sec | 82.92 sec |
| Data Set 4 | 262.26 sec | 112.42 sec |

For $P1$, we observed that the addition of valid inequalities improves the solution time for Data Sets 1, 2 and 4. However, the solution time for Data Set 3 increases.

Table 4.33: Solution Times for P2

| $P2$ | CPU Time before Valid Inequalities | CPU Time after Valid Inequalities |
|---|---|---|
| Data Set 1 | 95193.33sec (0%) | 130747.62 sec (0 %) |
| Data Set 2 | 518425.95 sec (51.28%) | 518401.44 sec (41.45%) |
| Data Set 3 | 518401.78 sec (35.37%) | 518413.65 sec (34.94%) |
| Data Set 4 | 240166.51sec (0%) | 518401.87 sec (6.82 %) |

For $P2$, both versions of the model were solved to optimality only on Data Set 1. The addition of valid inequalities worsens the solution time of our model. On Data Sets 2 and 3, the solver could not verify optimality of the best incumbent solution in the time limit for both versions. On these instances, we therefore compare the optimality gaps. The addition of valid inequalities reduced the optimality gap of results for Data Set 2 and 3. Finally, on Data Set 4, the solver computed an optimal solution using the model without the valid inequalities. The addition of valid inequalities leads to a positive optimality gap on this data set.

Table 4.34: Solution Times for CP1

| $CP1$ | CPU Time before Valid Inequalities | CPU Time after Valid Inequalities |
|---|---|---|
| Data Set 1 | 518402.36 sec (6.29%) | 518402.56 sec (7.54%) |
| Data Set 2 | - | 518401.93 sec (10.84%) |
| Data Set 3 | - | - |
| Data Set 4 | 518401.30 sec (6.40%) | 518433.38 sec (8.32%) |

For $CP1$, the addition of valid inequalities leads to larger optimality gaps on Data Sets 1 and 4. On Data Set 3, the solver could not compute a feasible solution using either version of the model. On Data Set 2, the solver could not compute a feasible solution in the time limit using the original model. On the other hand, with the addition of valid inequalities, we managed to obtain a feasible solution on this data set.

For $CP2$, the addition of valid inequalities slightly reduced the optimality gap on Data Set 3. However, the optimality gaps were larger on Data Sets 1, 2 and 4 after adding the valid inequalities.

We therefore recommend the addition of valid inequalities for the problem $P1$. The valid inequalities usually improve the solution time of the problem. However, for the

Table 4.35: Solution Times for CP2

| $CP2$ | CPU Time before Valid Inequalities | CPU Time after Valid Inequalities |
|---|---|---|
| Data Set 1 | 518426.60 sec (18.57 %) | 518434.89 sec (26.00 %) |
| Data Set 2 | 518433.48 sec (33.98 %) | 518409.76 sec (38.89 %) |
| Data Set 3 | 518482.32 sec (35.34 %) | 518327.64 sec (34.57 %) |
| Data Set 4 | 518429.57 sec (30.34 %) | 518432.32 sec (43.15%) |

other three problems $P2, CP1$, and $CP2$, valid inequalities did not generally lead to an improved performance.

# Chapter 5

# CONCLUSION

Container terminals should simultaneously improve the service levels and cost efficiency. The cheapest and the most efficient way of achieving this goal is to increase the efficiency of container terminal operations. Berth allocation and quay crane assignment are considered as two significant container terminal operations, whose effective management helps to increase the efficiency of the container terminal. In this thesis, we aimed to increase the efficiency of container terminals by simultaneously considering these two important container terminal operations.

Considering the berth allocation problem and quay crane assignment problem as an integrated problem, we proposed a bi-objective integer linear programing model. The first objective is to minimize the total service time of incoming vessels. The second objective is to minimize the number of crane setups. We examined the interplay between the two objective functions. By presenting this interaction to the terminal management, we provide them with the option of choosing the best combination of service level and number of crane setups.

In this thesis, we described four problems: $P1$, $CP1$, $P2$ and $CP2$. $P1$ minimizes the total service time, $P2$ minimizes the number of crane setups, $CP1$ minimizes the total service time subject to a constraint on the number of crane setups. In contrast, $CP2$ minimizes the number of crane setups subject to a service time constraint.

The data obtained from the Port of Izmir Container Terminal was used to test our model and the results of our model were compared with the real life solutions. Our model has significant improvements on real life solutions implemented at the container terminal. We demonstrated these improvements with computational experiments which we conducted using different periods during the year. We can conclude that our model can be applied on real-life instances to increase the efficiency of container terminal operations.

We defined a 6-day (518.400 sec) time limit for the solver and we obtained optimal solutions for the problem $P1$ for all data sets. However, the problem $P2$ turned out to be harder than $P1$. We obtained optimal solutions only on two instances. $CP1$ is the hardest problem when it is compared with the others. In spite of a 6-day time limit, our model could compute feasible solutions on only two of the four instances and could not compute any feasible solution on another set of two instances for the problem $CP1$. The solver could only compute feasible solutions for the problem $CP2$ for all data sets.

In this study, we focused on exact solutions. We consider developing a heuristic method to obtain approximate solutions for larger instances as future work. We believe that a sequential solution approach may be a reasonable alternative, in which the berth allocation problem is solved first and the solution is given as an input to the quay crane assignment problem.

# Appendix A

## *A.1 Brief Explanations about the Figures*

The solutions of the problems are presented in the following figures. In these charts, the "x-axis" represents the berths, while the "y-axis represents the 4-hour time intervals (time periods). Vessels are denoted by their labels and are represented by different colors. Berth and quay crane allocations for vessels are shown inside the cells in the figures.

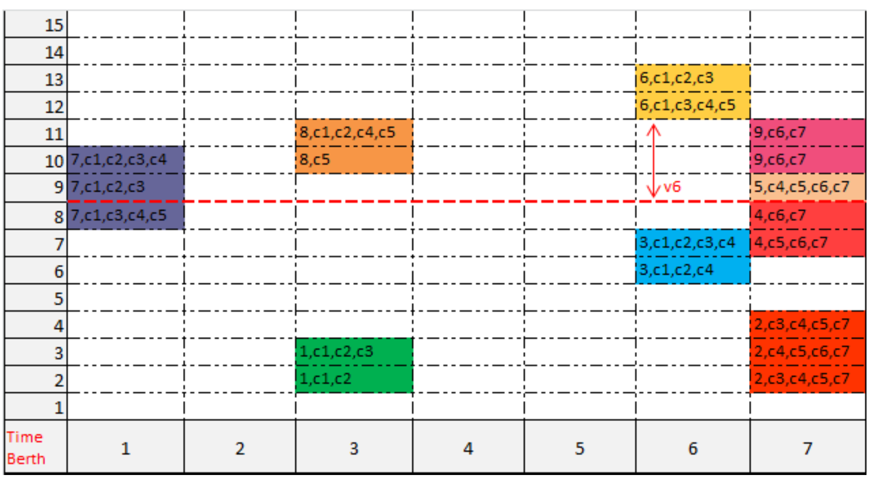


Figure A.1: Example Solution Figure

In Figure A.1, we illustrate a small example. For instance, the green rectangles represent Vessel 1. Vessel 1 is served at Berth 3 during time period 2 by Cranes 1 and 2. Crane 3 joins the other two cranes to serve Vessel 1 at time period 3. Since the berth assignment process is done according to available berths and available cranes, vessels may have to wait for berth assignment for two different reasons: either there is no available crane or no available berth for the incoming vessel. The red dotted-lines and arrows in the figure indicate the waiting times of vessels. For instance, Vessel 6 arrives at time period 9 and has to wait for berth assignment for three periods because all cranes serve other vessels in period 9. It is assigned to Berth 6 and starts to get service at time period 12. All containers for vessel 6 are handled by the end of period 13.

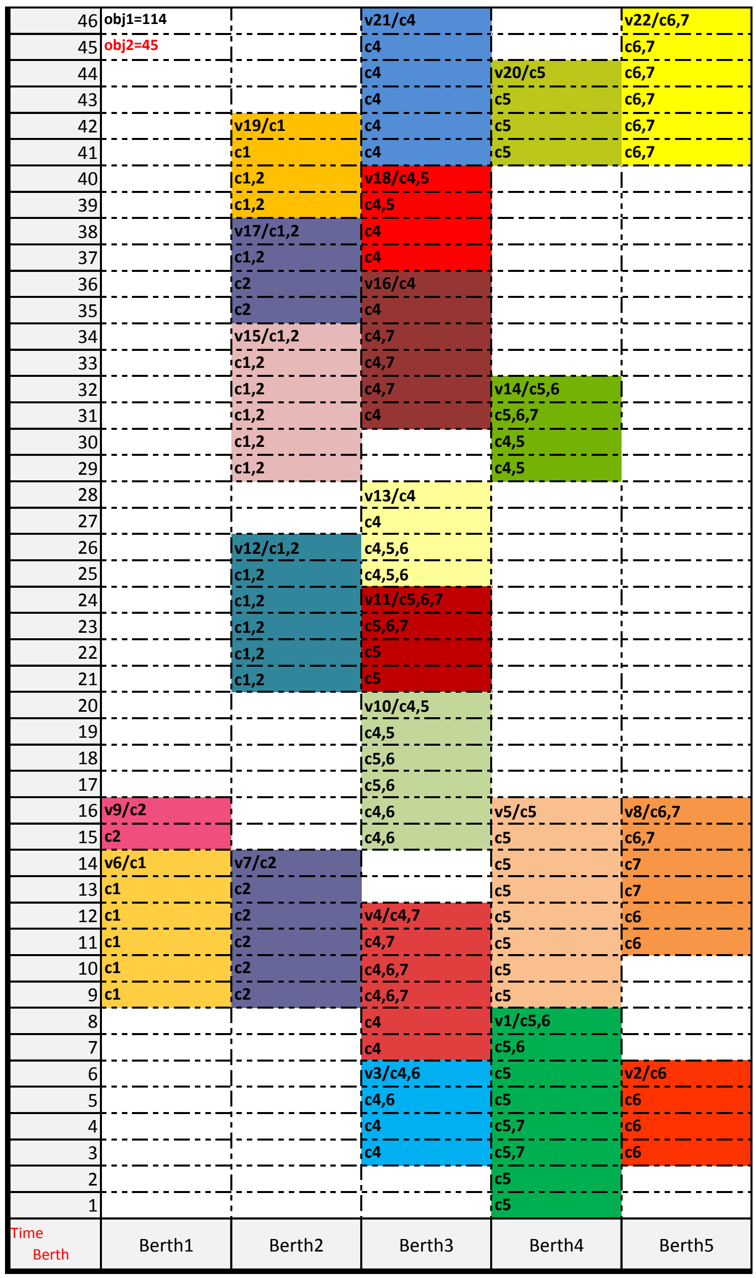


Figure A.2: Solution of Port of Izmir Container Terminal for Data Set 1

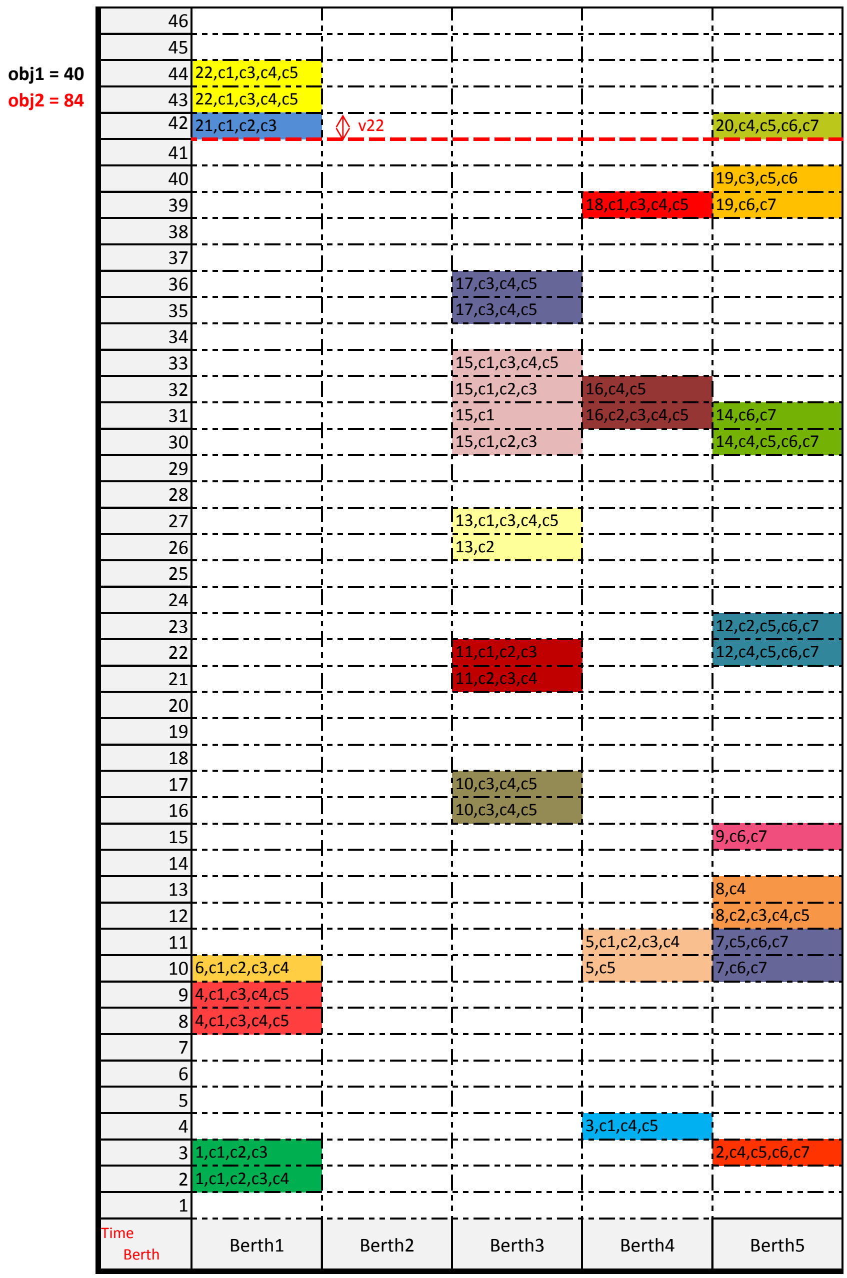


Figure A.3: Solution of P1 (Data Set1)

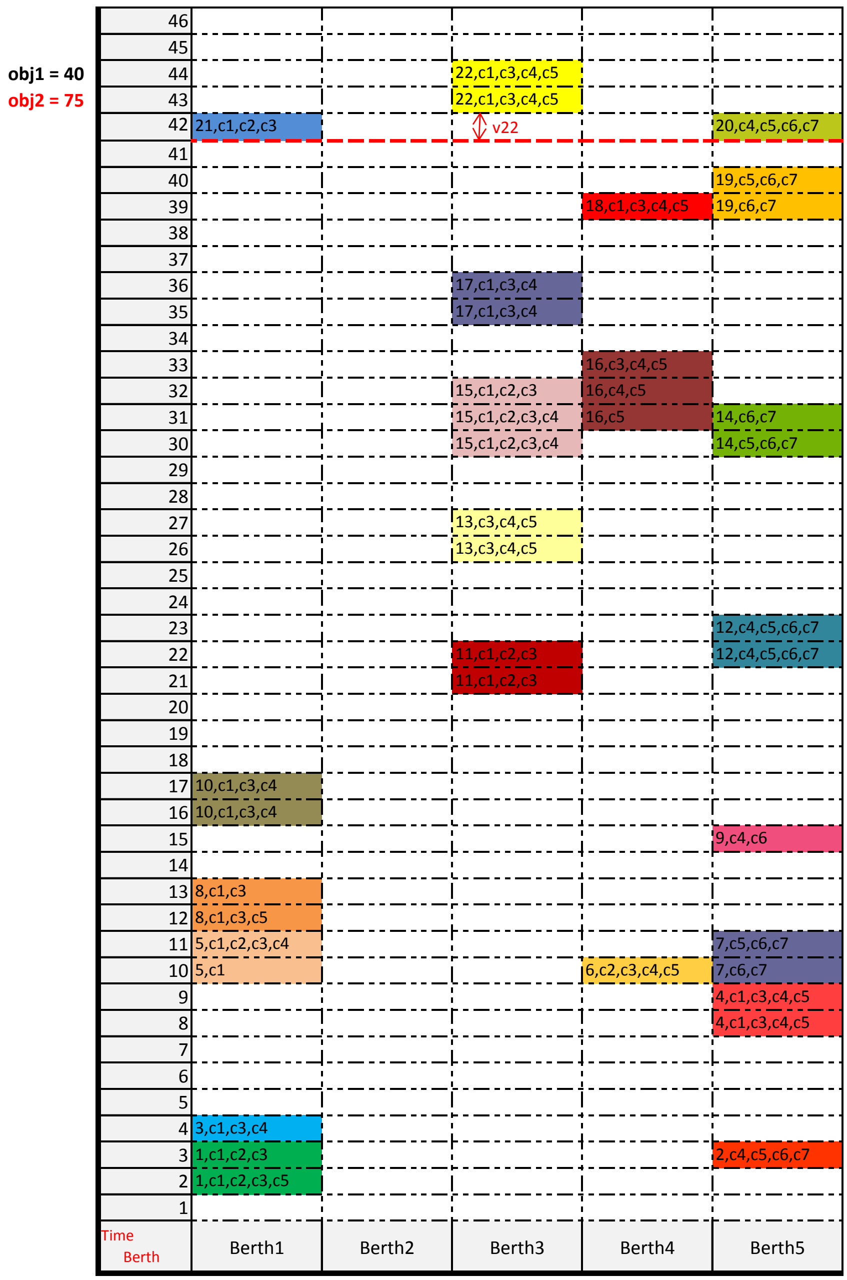


Figure A.4: Solution of CP2 (Data Set1)

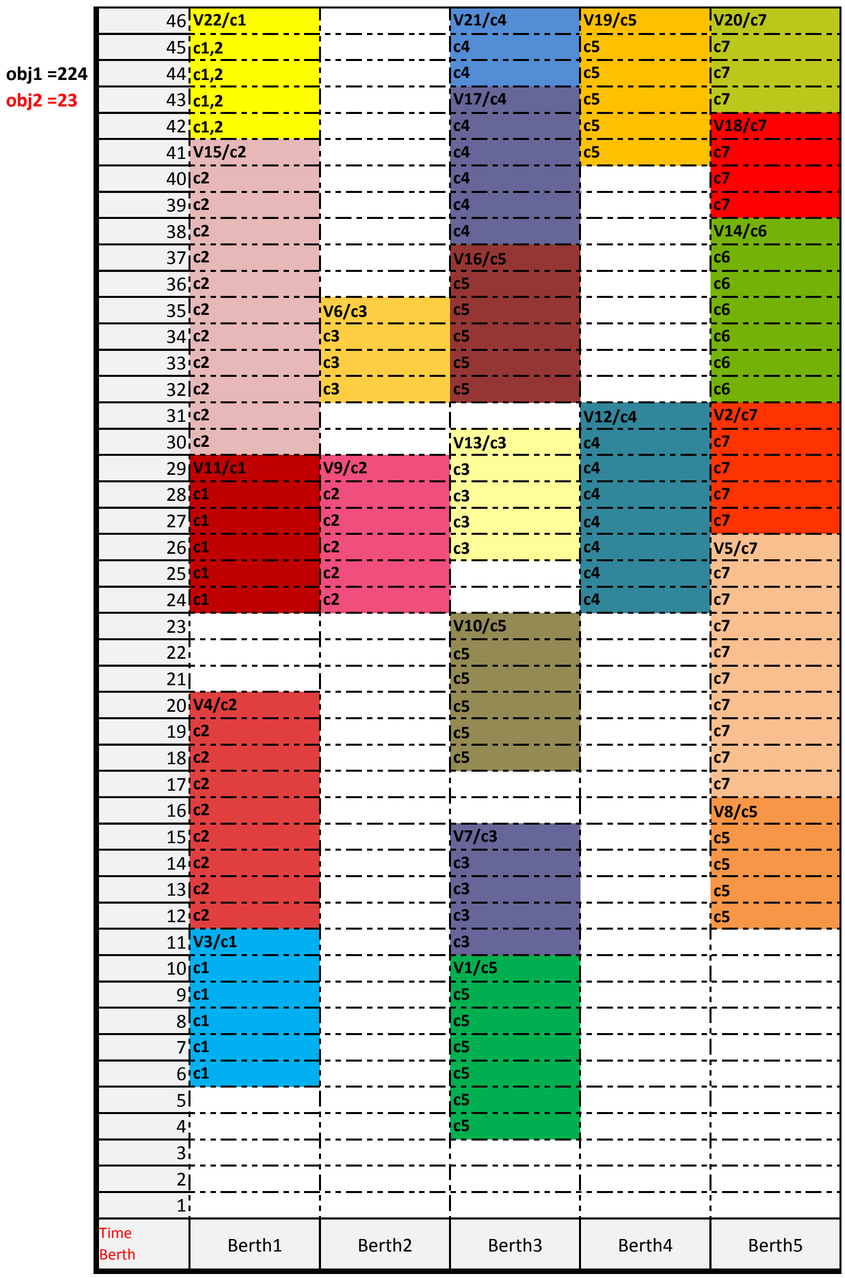


Figure A.5: Solution of P2 (Data Set1)

| | | | | |
|---|---|---|---|---|
| 50 | **obj1 = 138** | | 25,c4,c6 | |
| 49 | **obj2 = 61** | | 25,c4,c6 | |
| 48 | | 20,c3,c6,c7 | 25,c3,c4 | |
| 47 | | 20,c3,c6,c7 | 25,c3,c4 | |
| 46 | 24,c4,c5,c6,c7 | 20,c3 | 25,c4,c7 | |
| 45 | 24,c5,c6,c7 | 20,c3 | 25,c4,c7 | |
| 44 | | 20,c3,c6 | 25,c4,c5 | |
| 43 | | 20,c3,c6 | 25,c4,c5 | |
| 42 | | 20,c3,c6 | 25,c4 | 22,c5,c7 |
| 41 | | 20,c3,c6 | | 22,c4,c5,c7 |
| 40 | 19,c1,c7 | 20,c3,c6 | 23,c4 | 22,c5,c7 |
| 39 | 19,c1,c7 | 20,c3,c6 | 23,c4 | 22,c5,c7 |
| 38 | 19,c1,c7 | 20,c2,c3 | 21,c6 | 22,c4,c5 |
| 37 | 19,c1,c7 | 20,c2,c3 | 21,c6 | 22,c4,c5 |
| 36 | 19,c1,c7 | 20,c2,c3,c6 | 21,c4,c5 | 22,c5 |
| 35 | 19,c1,c7 | 20,c2,c3,c6 | 21,c4,c5 | |
| 34 | 19,c1,c7 | 20,c2,c3,c6 | 21,c4 | 18,c5 |
| 33 | 19,c1,c7 | 20,c2,c3,c6 | | 18,c4,c5 |
| 32 | | 14,c2 | | 18,c4,c5,c6 |
| 31 | | 14,c2 | | 18,c4,c5,c6 |
| 30 | 16,c1,c7 | 14,c2 | | 18,c5 |
| 29 | 16,c1,c7 | 14,c2 | | 18,c5 |
| 28 | 16,c1 | 14,c2,c3 | 15,c4 | 13,c5,c6,c7 |
| 27 | 16,c1 | 14,c2,c3 | 15,c4,c6 | 13,c5,c7 |
| 26 | 11,c1,c2 | 14,c3 | 15,c4,c6 | 13,c5,c7 |
| 25 | 11,c1,c2 | 14,c3 | 15,c4,c6 | 13,c5,c7 |
| 24 | 11,c1 | 17,c3 | 12,c4 | |
| 23 | 11,c1 | 17,c3 | 12,c4 | |
| 22 | 11,c1,c2 | 17,c3 | 8,c4,c5,c7 | |
| 21 | 11,c1,c2 | 10,c3 | 8,c4,c5,c7 | |
| 20 | 9,c1,c2 | 10,c3,c6 | 8,c4,c5,c7 | |
| 19 | 9,c1,c2 | 10,c3,c6 | 8,c4,c5,c7 | |
| 18 | 9,c1,c2,c3 | | 8,c4,c5,c7 | |
| 17 | 9,c1,c2,c3 | | 8,c4,c5,c7 | |
| 16 | 9,c1,c2 | | 7,c4 | 6,c5,c6,c7 |
| 15 | 9,c1,c2 | | 7,c4 | 6,c5,c6,c7 |
| 14 | | 5,c2,c3 | 7,c4,c5 | 6,c6,c7 |
| 13 | | 5,c2,c3 | | 6,c4,c5 |
| 12 | 3,c1 | 5,c2 | | 6,c4,c5 |
| 11 | 3,c1 | 5,c2 | | |
| 10 | 3,c1 | 5,c2 | | 2,c4,c5,c6 |
| 9 | 3,c1 | 5,c2 | | 2,c4,c5,c6 |
| 8 | 3,c1,c7 | 5,c2 | 4,c4 | 2,c5,c6 |
| 7 | 3,c1,c7 | 5,c2 | 4,c4 | 2,c5,c6 |
| 6 | 3,c7 | 5,c2 | 4,c4 | 2,c5 |
| 5 | 1,c2 | | 4,c4 | 2,c5 |
| 4 | 1,c2 | | | |
| 3 | 1,c2 | | | |
| 2 | | | | |
| 1 | | | | |
| Time Berth | 1 | 2 | 3 | 4 |

Figure A.6: Solution for Port of Izmir Container Terminal for Data Set 2

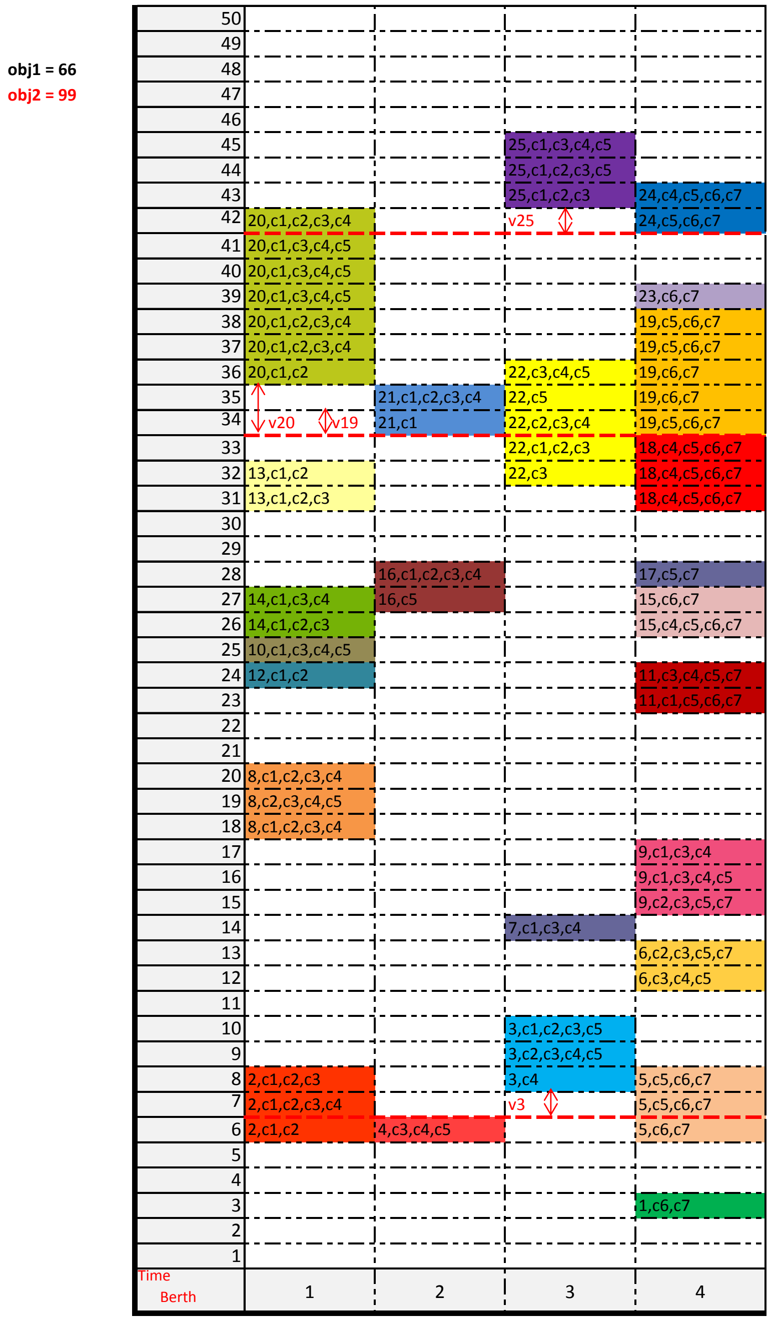


Figure A.7: Solution of P1 (Data Set 2)

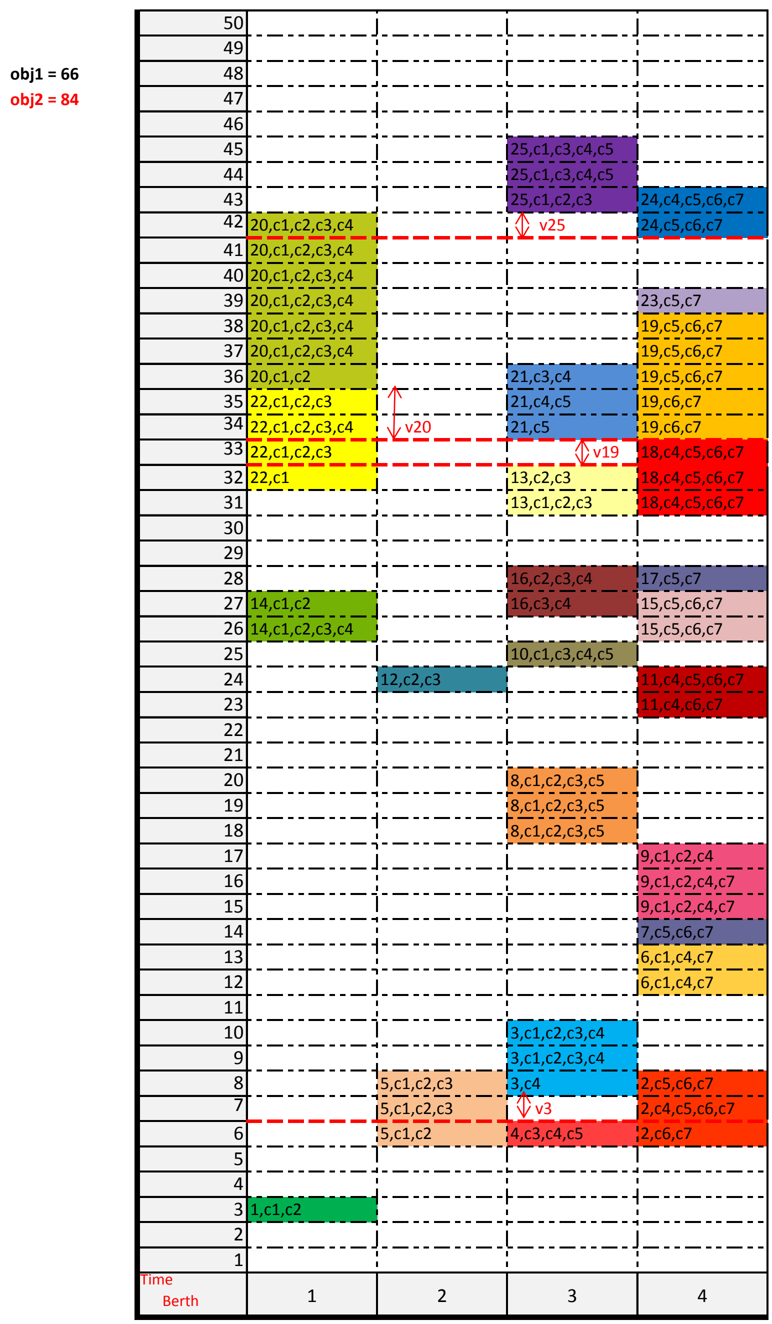


Figure A.8: Solution of CP2 (Data Set 2)

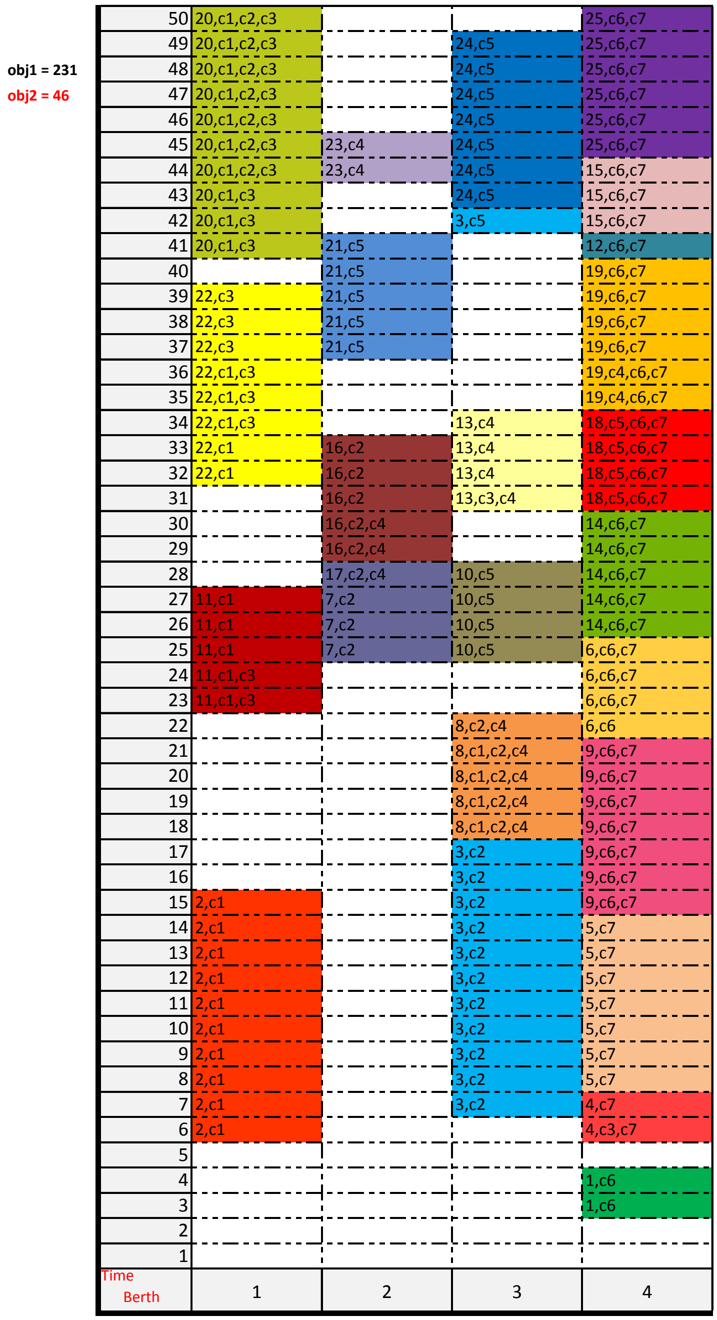


Figure A.9: Solution of P2 (Data Set 2)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 48 | obj1 = 148 | | | | | | |
| 47 | obj2 = 58 | | | | | | |
| 46 | | | | | 24,c5,c6,c7 | | |
| 45 | | | | | 24,c5,c6,c7 | | |
| 44 | | 26,c2,c3 | | | 24,c4,c5,c7 | | |
| 43 | | 26,c2,c3 | | | 24,c4,c5,c7 | | |
| 42 | | 26,c2,c3 | | | 24,c4,c5,c7 | | |
| 41 | 22,c2 | 26,c3 | | | 24,c4,c5,c7 | | |
| 40 | 22,c2 | | 25,c6,c7 | | 24,c4,c5 | | |
| 39 | 22,c2 | | 25,c6,c7 | | 24,c4,c5 | | |
| 38 | 22,c2,c3 | | 25,c4,c7 | | 24,c5,c6 | | |
| 37 | 22,c2,c3 | | 25,c4,c7 | | 24,c5,c6 | | |
| 36 | 22,c1,c2 | | 21,c4,c6,c7 | | 20,c5 | | |
| 35 | 22,c1,c2 | | 21,c4,c6,c7 | | 20,c5 | | |
| 34 | | | 21,c4 | | 20,c5 | | |
| 33 | | | 21,c4 | | 20,c5 | | |
| 32 | 19,c2,c3,c6,c7 | | 21,c4 | | 20,c5 | | |
| 31 | 19,c2,c3,c6,c7 | | 23,c4 | | 20,c5 | | |
| 30 | 19,c2,c3,c6,c7 | | 23,c4 | 18,c5 | | | |
| 29 | 19,c2,c3,c6,c7 | | 23,c4 | 18,c5 | | | |
| 28 | | 13,c2,c3,c6,c7 | 23,c4, | 18,c5 | | | |
| 27 | | 13,c2,c3,c6,c7 | 23,c4 | | 17,c5 | | |
| 26 | | 13,c2,c3,c6 | 23,c4 | | 11,c5 | | 15,c6,c7 |
| 25 | | 13,c2,c3,c6 | 23,c4 | | 11,c5 | | 15,c6,c7 |
| 24 | 10,c2 | 13,c3 | 16,c4,c6 | | 11,c5 | | 15,c7 |
| 23 | 10,c2 | 13,c3 | 16,c4,c6 | | 11,c5 | | 15,c7 |
| 22 | 10,c2 | 13,c3 | 16,c4,c6 | | 11,c5 | | 15,c7 |
| 21 | 10,c2 | 13,c3 | 16,c4,c6 | | 11,c5 | | 15,c7 |
| 20 | 10,c2 | 12,c3 | 9,c4 | | 11,c5 | 14,c7 | 15,c6 |
| 19 | 10,c2 | 12,c3 | 9,c4 | | 11,c5 | 14,c7 | 15,c6 |
| 18 | 7,c2,c3 | | 9,c4 | | 11,c5 | | |
| 17 | 7,c2,c3 | | 9,c4,c5 | | | | |
| 16 | 7,c2,c3 | | 9,c4 | | | 6,c6,c7 | |
| 15 | 7,c2,c3 | | 9,c4 | | 8,c5,c6,c7 | 6 | |
| 14 | 7,c2,c3 | | 5,c4 | | 8,c5 | 6,c6,c7 | |
| 13 | 7,c2,c3 | | 5,c4 | | 8,c5 | 6,c6,c7 | |
| 12 | 7,c2,c3 | | 5,c4 | | 8,c5 | 6,c7 | |
| 11 | 7,c2,c3 | | 5,c4 | | 8,c5 | 6,c7 | |
| 10 | 7,c2,c3 | | 5,c4 | | 8,c5 | 6,c7 | |
| 9 | 7,c2,c3 | | 5,c4,c5 | | | 6,c7 | |
| 8 | 2,c2,c3,c7 | | | | 3,c4,c5 | | |
| 7 | 2,c2,c3,c7 | | | | 3,c4,c5 | | |
| 6 | 2,c2,c3,c7 | | | | 3,c4,c5 | | 4,c6 |
| 5 | 2,c2,c3,c7 | | | | 3,c4,c5 | | 4,c6 |
| 4 | 2,c2,c3 | | 1,c4,c5,c6 | | | | 4 |
| 3 | 2,c2,c3 | | 1,c4,c5 | | | | 4,c6 |
| 2 | | | 1,c4 | | | | 4,c6 |
| 1 | | | 1,c4 | | | | 4,c6 |
| Time Berth | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Figure A.10: Solution of Port of Izmir Container Terminal for Data Set 3)

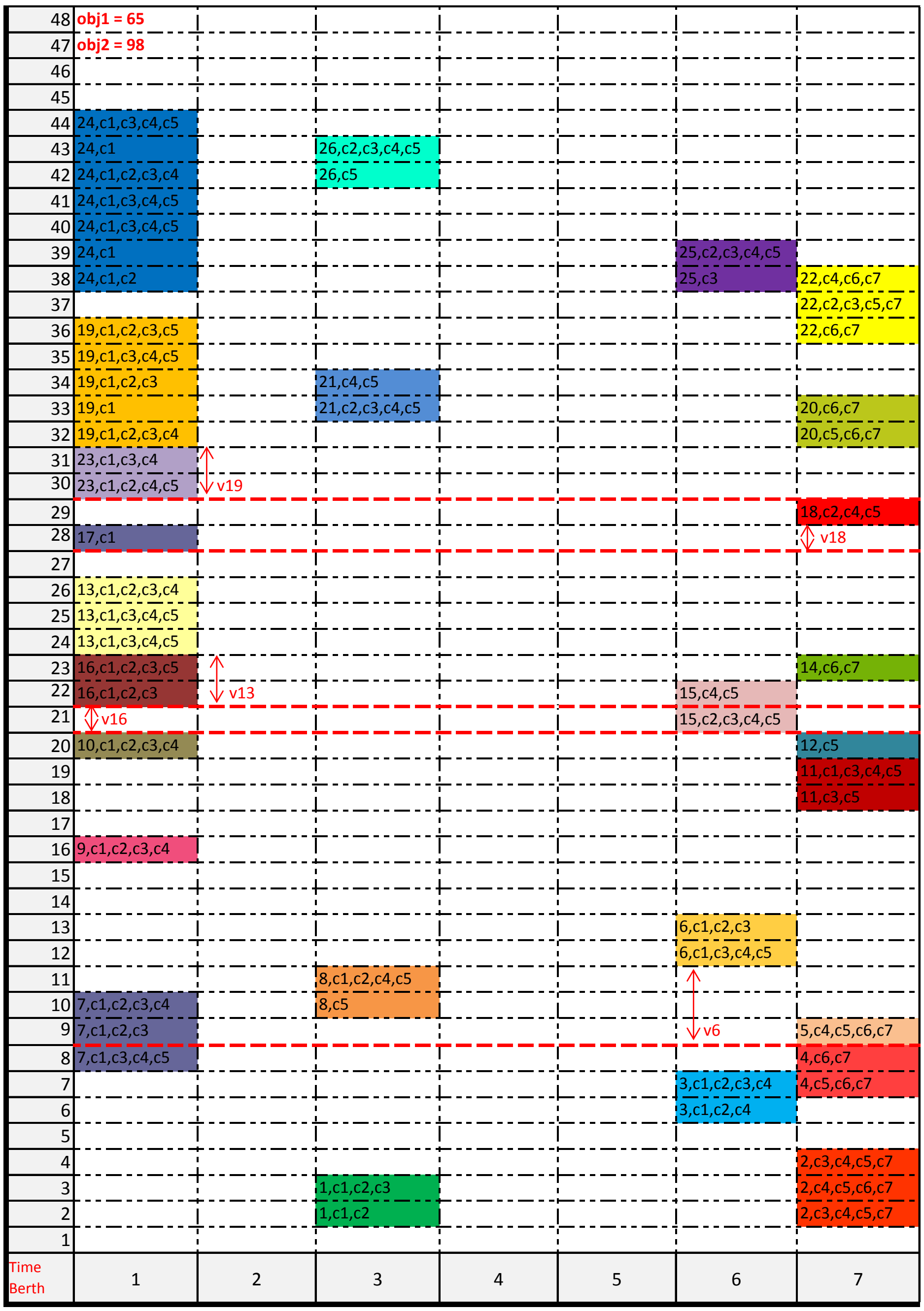


Figure A.11: Solution of P1 (Data Set 3)

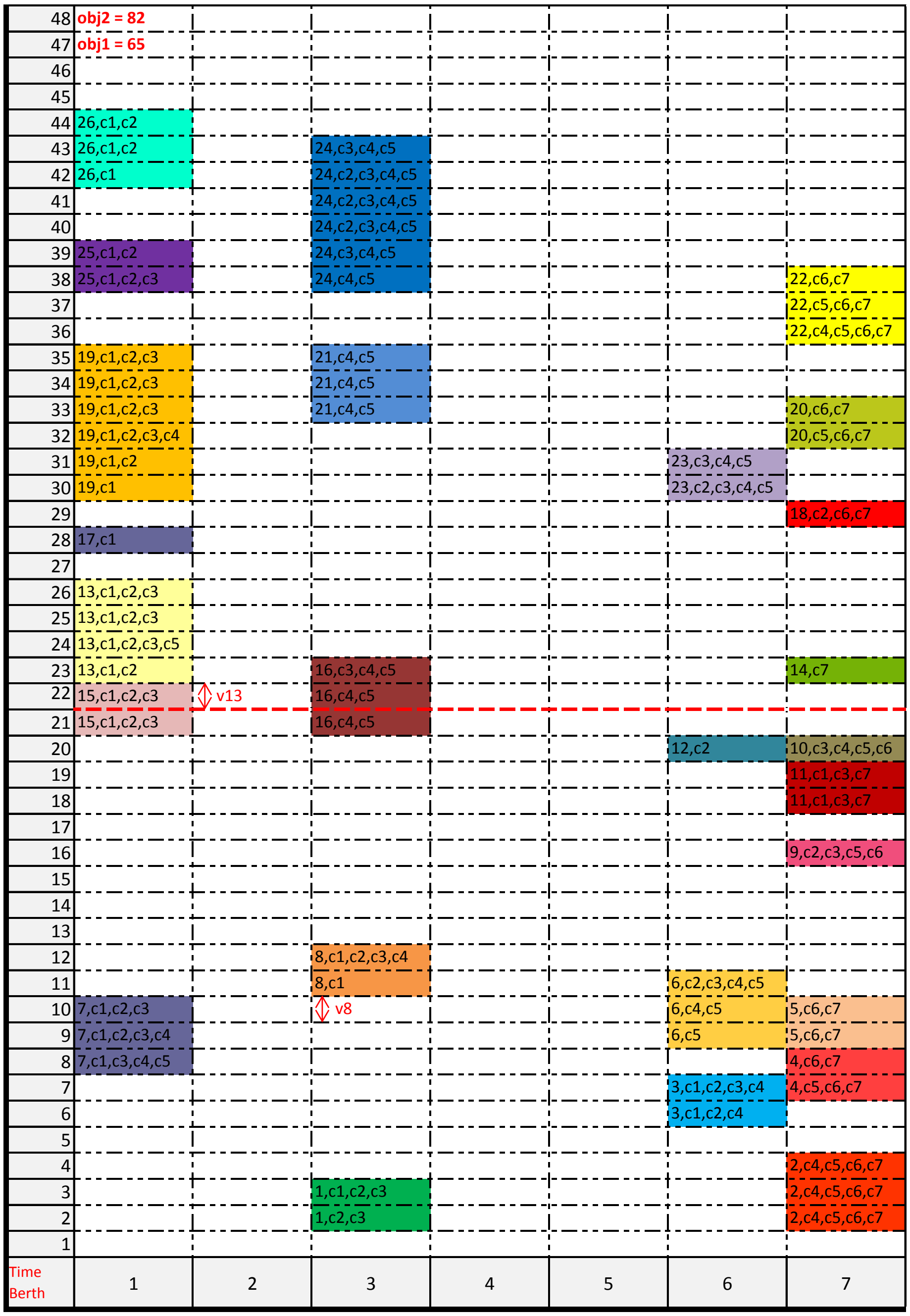


Figure A.12: Solution of CP2 (Data Set 3)

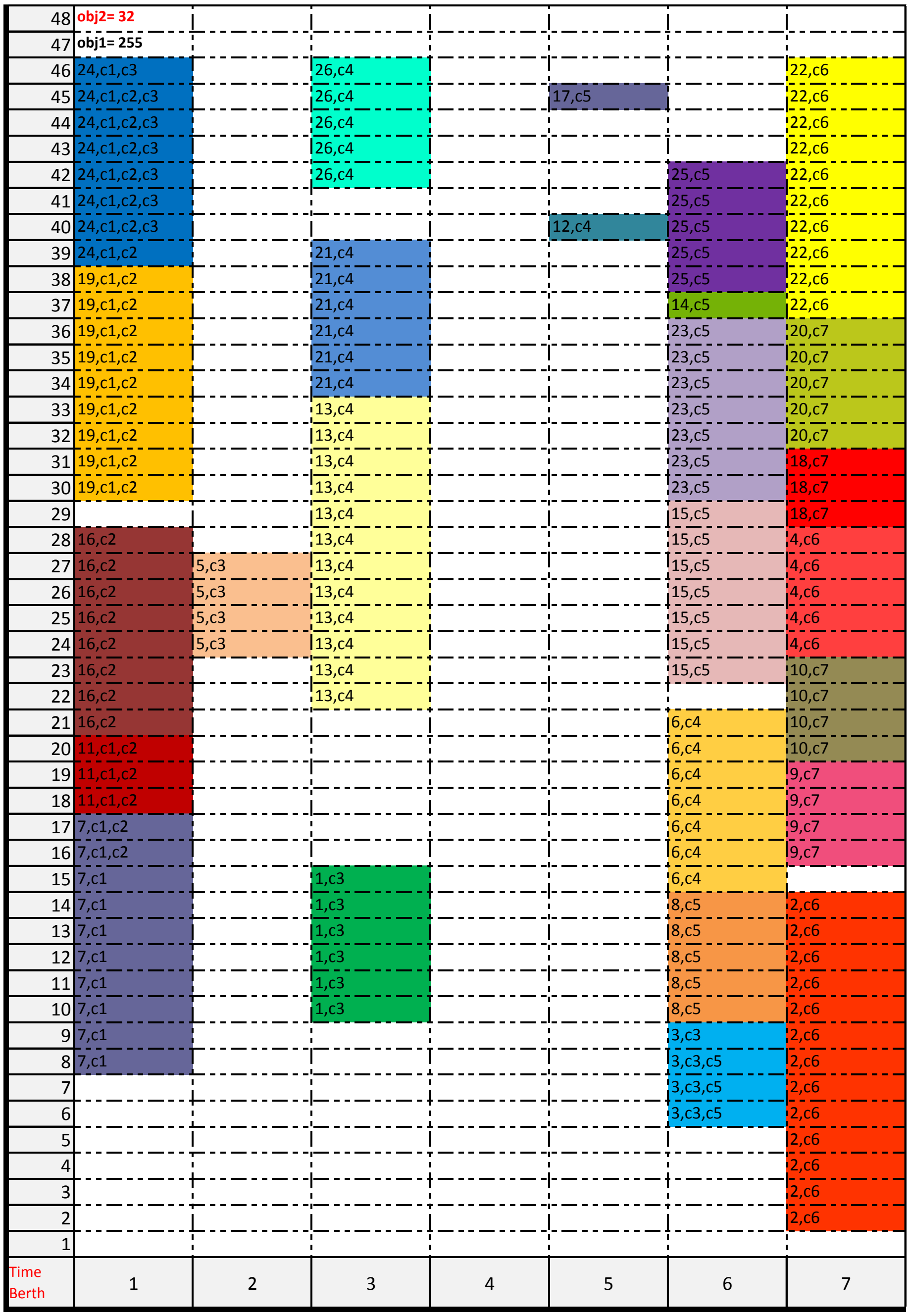


Figure A.13: Solution of P2 (Data Set 3)


Figure A.14: Solution of Port of Izmir Container Terminal for Data Set 4

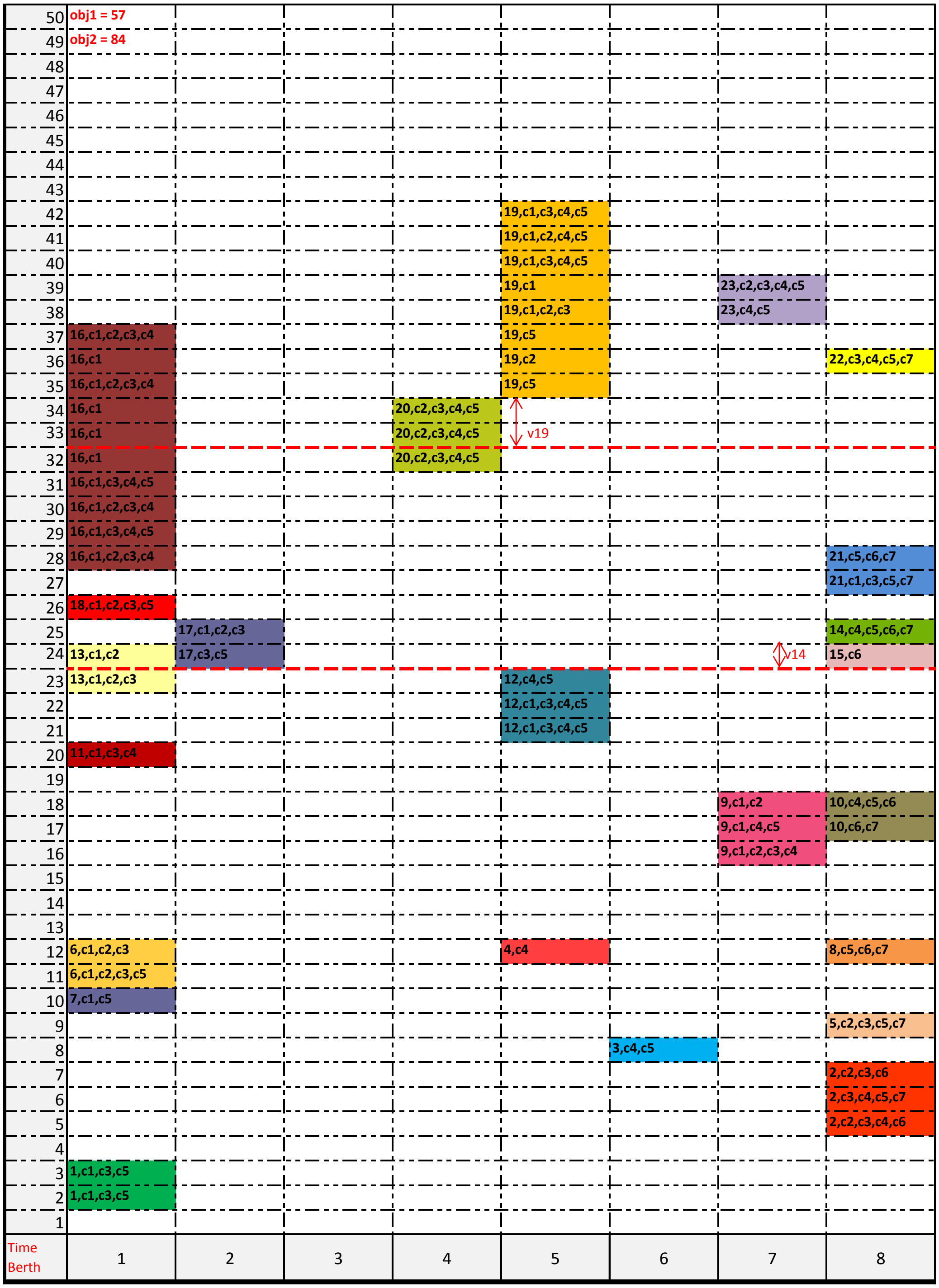


Figure A.15: Solution of P1 (Data Set 4)

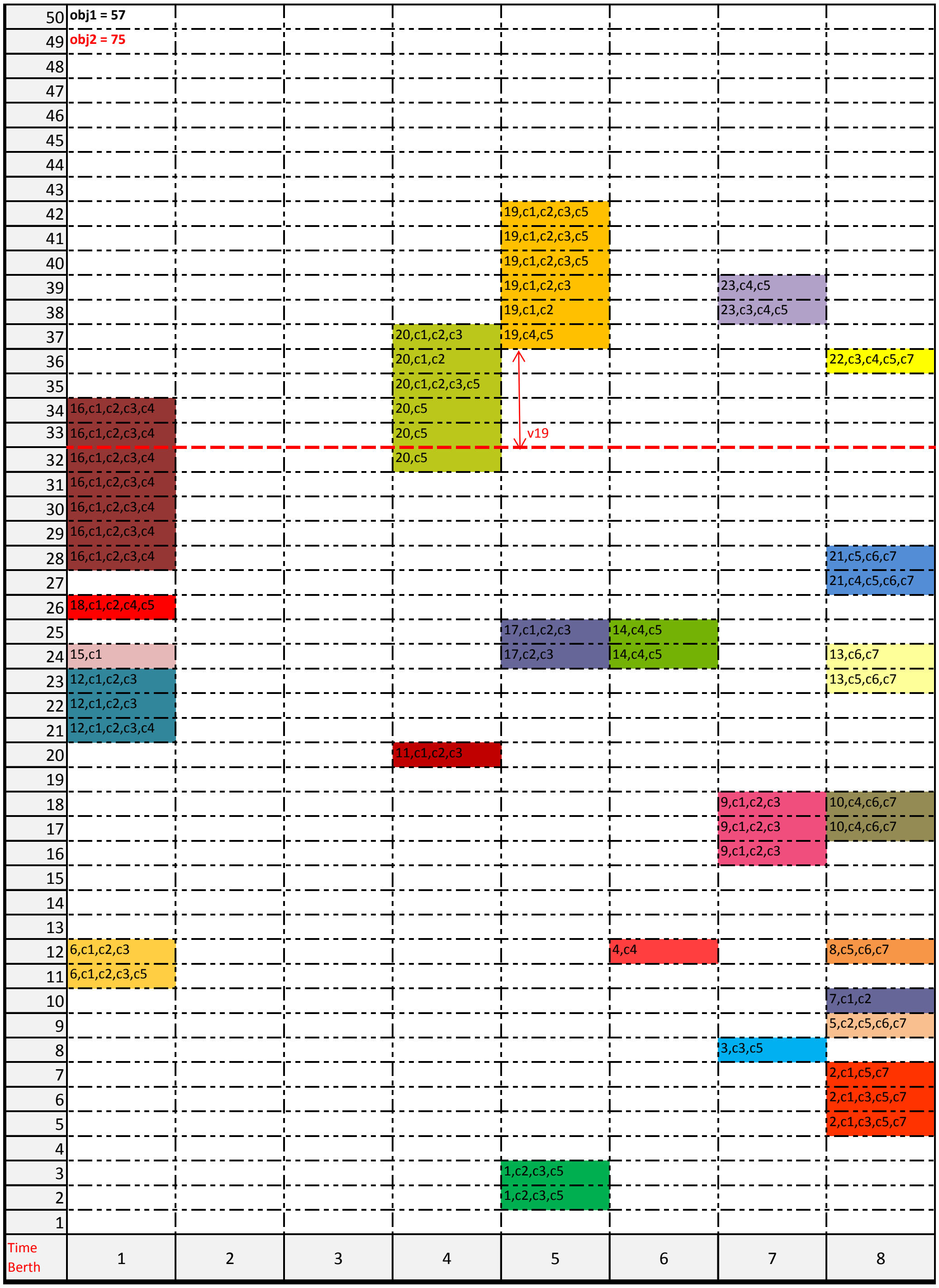


Figure A.16: Solution of CP2 (Data Set 4)

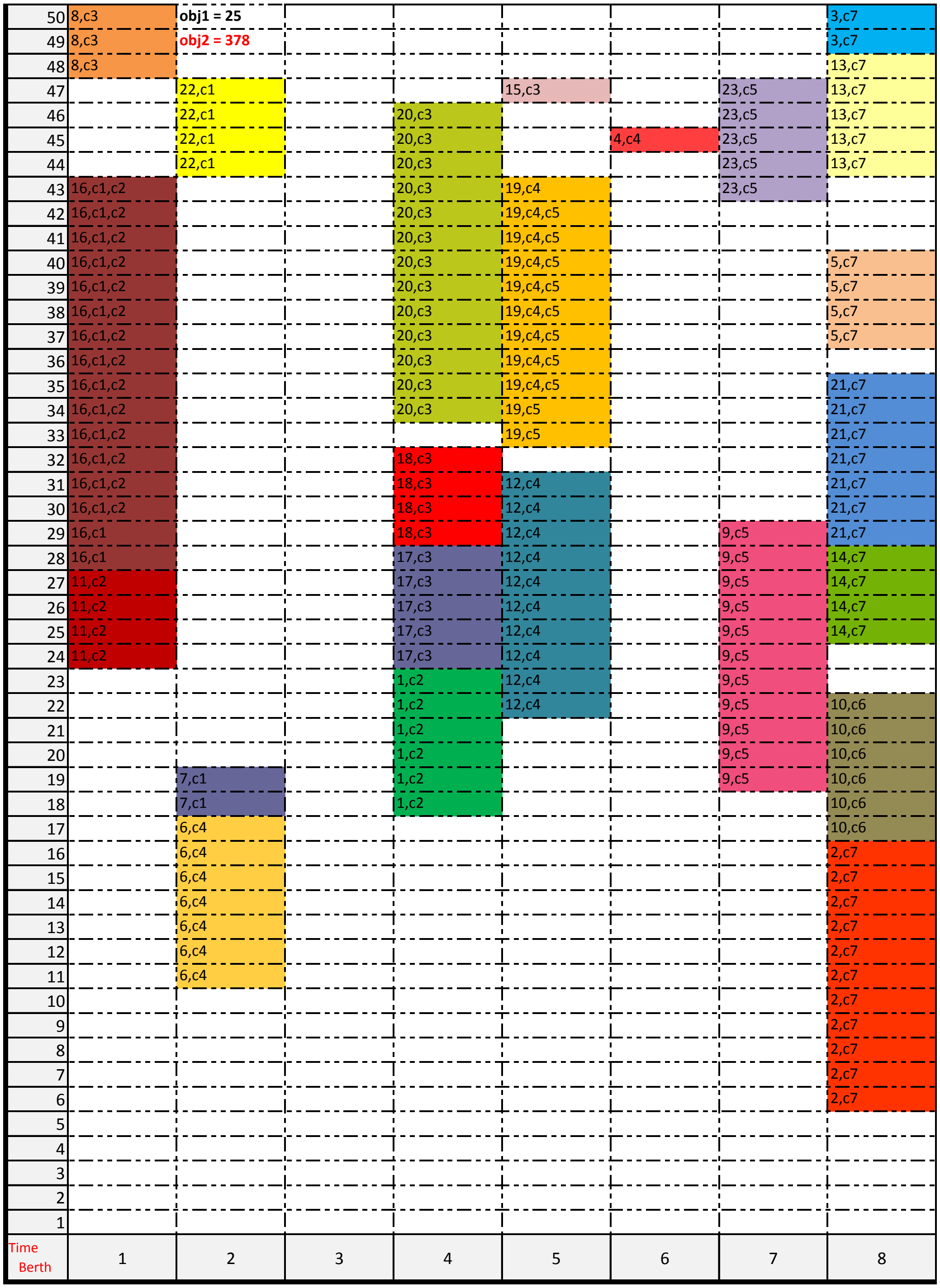


Figure A.17: Solution of P2 (Data Set 4)

# BIBLIOGRAPHY

[1] *International Trade.* Retrieved 7, June, 2014, from http://www.un.org/en/development/desa/policy/wesp/wesp_archive/2012chap2.pdf.

[2] *Twenty-Foot Equivalent Unit.* Retrieved 4, June, 2014, from http://en.wikipedia.org/wiki/Twenty-foot_equivalent_unit.

[3] *WPS - Port of Izmir Port Commerce.* Retrieved 7, June, 2014, from http://www.worldportsource.com/ports/commerce/TUR_Port_of_Izmir_2061.php.

[4] Christian Bierwirth and Frank Meisel. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3):615–627, May 2010.

[5] Carlos F. Dazango. The crane scheduling problem. *Transportation Research Part B: Methodological*, 23(3):159–175, September 1989.

[6] Matthias Ehrgott. *Multicriteria Optimization*, volume 39. Springer, second edition, February 2005.

[7] Pierre Hansen, Ceyda Oğuz, and Nenad Mladenović. Variable neighborhood search for minimum cost berth allocation. *European Journal of Operational Research*, 191(3):636–649, December 2008.

[8] Zhi-Hua Hu. A Quay Crane Assignment Approach for Berth Allocation Problem in Container Terminal. *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, pages 17–21, April 2010.

[9] Akio Imai, Hsieh Chia Chen, Etsuko Nishimura, and Stratos Papadimitriou. The simultaneous berth and quay crane allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, 44(5):900–920, September 2008.

[10] Akio Imai, Ken'Ichiro Nagaiwa, and Chan Weng Tat. Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation*, 31(1):75–94, June 1997.

[11] Akio Imai, Etsuko Nishimura, and Stratos Papadimitriou. The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, 35(4):401–417, May 2001.

[12] Akio Imai, Etsuko Nishimura, and Stratos Papadimitriou. Berth allocation with service priority. *Transportation Research Part B: Methodological*, 37(5):437–457, June 2003.

[13] Akio Imai, Etsuko Nishimura, and Stratos Papadimitriou. Berthing ships at a multi-user container terminal with a limited quay capacity. *Transportation Research Part E: Logistics and Transportation Review*, 44(1):136–151, January 2008.

[14] Akio Imai, Xin Sun, Etsuko Nishimura, and Stratos Papadimitriou. Berth allocation in a container port: using a continuous location space approach. *Transportation Research Part B: Methodological*, 39(3):199–221, March 2005.

[15] Chengji Liang, Jianquan Guo, and Yang Yang. Multi-objective hybrid genetic algorithm for quay crane dynamic assignment in berth allocation planning. *Journal of Intelligent Manufacturing*, 22(3):471–479, August 2009.

[16] Chengji Liang, Youfang Huang, and Yang Yang. A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Computers & Industrial Engineering*, 56(3):1021–1028, April 2009.

[17] Andrew Lim. The berth planning problem 1. *Operations Research Letters*, 22(22):105–110, 1998.

[18] Etsuko Nishimura, Akio Imai, and Stratos Papadimitriou. Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research*, 131(2):282–292, June 2001.

[19] Birger Raa, Wout Dullaert, and Rowan Van Schaeren. An enriched model for the integrated berth allocation and quay crane assignment problem. *Expert Systems with Applications*, 38(11):14136–14147, May 2011.

[20] Sotirios Theofanis, Maria Boile, and Mihalis Golias. An optimization based genetic algorithm heuristic for the berth allocation problem. *2007 IEEE Congress on Evolutionary Computation*, pages 4439–4445, September 2007.

[21] Yavuz B Türkoğulları, Z. Caner Taşkın, Necati Aras, and Kurban Altınel. Optimal Berth Allocation and Time-Invariant Quay Crane Assignment in Container Terminals. *European Journal of Operational Research*, 235(1):88–101, May 2014.

[22] Dongsheng Xu, Chung-Lun Li, and Joseph Y.-T. Leung. Berth allocation with time-dependent physical limitations on vessels. *European Journal of Operational Research*, 216(1):47–56, January 2012.

[23] Qingcheng Zeng, Zhongzhen Yang, and Xiangpei Hu. Disruption recovery model for berth and quay crane scheduling in container terminals. *Engineering Optimization*, 43(9):967–983, September 2011.

[24] Qingcheng Zeng, Zhongzhen Yang, and Luyuan Lai. Models and algorithms for multi-crane oriented scheduling method in container terminals. *Transport Policy*, 16(5):271–278, September 2009.