

DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**SOLVING DYNAMIC BERTH ALLOCATION
AND QUAY CRANE SCHEDULING PROBLEM
SIMULTANEOUSLY BY USING SIMULATION
OPTIMIZATION**

by
Gökçeççek TAŞOĞLU

October, 2019
İZMİR

SOLVING DYNAMIC BERTH ALLOCATION AND QUAY CRANE SCHEDULING PROBLEM SIMULTANEOUSLY BY USING SIMULATION OPTIMIZATION

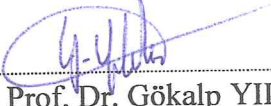
**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of
Philosophy in Industrial Engineering, Industrial Engineering Program**

**by
Gökçeçecek TAŞOĞLU**

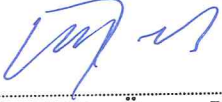
**October, 2019
İZMİR**

Ph.D. THESIS EXAMINATION RESULT FORM

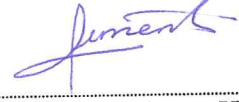
We have read the thesis entitled “SOLVING DYNAMIC BERTH ALLOCATION AND QUAY CRANE SCHEDULING PROBLEM SIMULTANEOUSLY BY USING SIMULATION OPTIMIZATION” completed by GÖKÇEÇİÇEK TAŞOĞLU under supervision of ASSOC. PROF. DR. GÖKALP YILDIZ and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.


Assoc. Prof. Dr. Gökalp YILDIZ

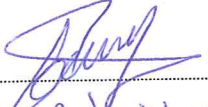
Supervisor


Assoc. Prof. Dr. Özcan KILINÇCI

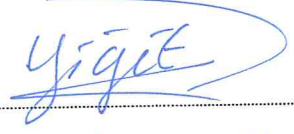
Thesis Committee Member


Assoc. Prof. Dr. Çimen K. ÇETİN

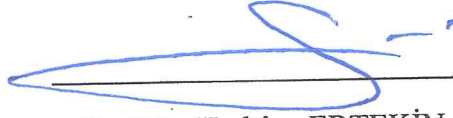
Thesis Committee Member


Assoc. Prof. Dr. Banu Y. Ekren

Examining Committee Member


Prof. Dr. Yiğit KAZANÇOĞLU

Examining Committee Member


Prof. Dr. Kadriye ERTEKİN

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my supervisor Assoc. Prof. Dr. Gökcalp YILDIZ for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis and whenever I lost my motivation, he encouraged me to carry on and complete my Ph.D. study. I am very glad to have worked with him.

Besides my supervisor, I would like to thank the rest of my thesis committee: Prof. Dr. Soner ESMER, Assoc. Prof. Dr. Özcan KILINÇCI and Assoc. Prof. Dr. Çimen KARATAŞ ÇETİN for their insightful comments and encouragement, but also for the hard question which incited me to widen my research from various perspectives.

I also take this opportunity to thank the staff in the Department of Industrial Engineering at Manisa Celal Bayar University for their support and understanding throughout my Ph.D. study.

Very special thanks would be given to my family; my mother Müzeyyen TUNA, my father Tümer TUNA and my brother Mehmet Göker TUNA as they have always supported me throughout my all education life and during the process of researching and writing this thesis. I am very lucky and grateful to have such a family.

At last but not most, I would like to thank my husband Mustafa TAŞOĞLU for his continuous support, understanding, patience and love. Although it was very challenging to complete this thesis with a little baby, he was always next to me and cared our little daughter Zeynep.

Finally, I dedicated this thesis to my little daughter Zeynep. Because this thesis wouldn't have been possible without her love.

Gökçeçişek TAŞOĞLU

SOLVING DYNAMIC BERTH ALLOCATION AND QUAY CRANE SCHEDULING PROBLEM SIMULTANEOUSLY BY USING SIMULATION OPTIMIZATION

ABSTRACT

This dissertation proposes a simulation optimization-based solution approach for the integrated Berth Allocation and Quay Crane Scheduling Problems (BAP&QCSP) with multi-quay hybrid berth layout, dynamic arrivals of vessels, stochastic handling times and non-crossing constraints of quay cranes. This solution approach integrates the Simulated Annealing (SA) based search procedure with the simulation model of a real-world inspired container terminal.

The development of this solution approach consists of four phases; In the first phase, a conflict free quay crane scheduling algorithm for stochastic handling times is proposed. In the second phase, a general parametric simulation model which represents the seaside operations of a typical container terminal is built by including most of the realistic features of a port. Then, in the third phase, a simulated annealing-based simulation optimization procedure is proposed to minimize the completion time for handling operations of the last vessel in a container terminal (*i.e.*, makespan). In order to validate the proposed simulation optimization procedure, in the fourth phase, randomly generated ten small-sized deterministic problems are solved by both mathematical programming model and the proposed simulation optimization procedure. After validating our approach, the proposed simulation optimization procedure is applied to three different real-world inspired sample container terminals. The results revealed that the proposed simulation optimization procedure can be used to solve large-sized real-port BAPs and QCSPs for the objective of minimizing makespan. Furthermore, the output of this simulation optimization procedure gives the decision maker the estimated berthing/unberthing times, berthing locations and quay crane schedules for each vessel for the most probable scenario.

Keywords: Berth allocation problem, quay crane scheduling problem, multi-quay hybrid berth layout, simulation optimization, non-crossing constraint of quay cranes



DİNAMİK RIHTIM ATAMA VE RIHTIM VİNCİ ÇİZELGELEME PROBLEMİNİN BENZETİM OPTİMİZASYONU YOLUYLA EŞ ZAMANLI ÇÖZÜMÜ

ÖZ

Bu çalışmada, çoklu melez rıhtım yerleşimine, dinamik gemi gelişlerine, stokastik işlem sürelerine ve rıhtım vinçlerinin çakışma kısıtlarına sahip, entegre edilmiş yanaşma yeri atama ve vinç çizelgeleme problemleri için simülasyon optimizasyon tabanlı bir çözüm yaklaşımı önerilmektedir. Bu çözüm yaklaşımı, tavlama benzetimi tabanlı bir arama prosedürü ile gerçek hayattan esinlenmiş bir konteyner rıhtımını temsil eden simülasyon modelini entegre etmektedir.

Bu çözüm yaklaşımının tasarlanması dört aşamadan oluşmaktadır: İlk aşamada, stokastik yükleme/boşaltma sürelerini göz önünde bulunduran, çakışmasız (konumsal kısıtlamalardan arındırılmış) bir rıhtım vinci çizelgeleme algoritması önerilmektedir. İkinci aşamada, bir limanın gerçeğe uygun birçok özelliğini taşıyan tipik bir konteyner terminalindeki kıyı operasyonlarını temsil eden, genel ve parametrik bir simülasyon modeli oluşturulmuştur. Daha sonra, üçüncü aşamada, konteyner terminalinde işlem gören en son geminin yükleme/boşaltma işlemlerinin tamamlanma zamanını enküçükleyen, tavlama benzetimi tabanlı simülasyon optimizasyon prosedürü geliştirilmiştir. Önerilen simülasyon optimizasyonu prosedürünü doğrulamak için, dördüncü aşamada, rastgele türetilmiş on adet küçük boyutlu, deterministik problem, matematiksel model ve önerilen simülasyon optimizasyonu prosedürünün her ikisiyle de çözülmüştür. Çözüm yaklaşımının doğrulaması yapıldıktan sonra, simülasyon optimizasyon prosedürü, gerçek hayattan esinlenilmiş üç farklı konteyner rıhtımı örneğine uygulanmıştır. Sonuçlar, önerilen çözüm yaklaşımının, büyük boyutlu gerçek konteyner terminallerindeki rıhtım yeri yanaşma ve vinç çizelgeleme problemlerini, tamamlanma süresini en küçükmek amacıyla etkin bir şekilde çözebileceğini göstermiştir. Buna ek olarak, bu çözüm yaklaşımının çıktısı, karar vericiye, her gemi için tahmini yanaşma zamanı, tamamlanma zamanı, yanaşma yeri ve rıhtım vinci

izelgesi bilgisi gibi bilgileri ieren gerekleŖebilecek en olası senaryoyu sunabilmektedir.

Anahtar kelimeler: YanaŖma yeri tahsisi problemi, rıhtım vinci izelgeleme problemi, oklu melez rıhtım yerleŖimi, simlasyon optimizasyon, rıhtım vinlerinin konumsal hareket kısıtlamaları



CONTENTS

	Page
Ph.D. THESIS EXAMINATION RESULT FORM	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZ	vi
LIST OF FIGURES	x
LIST OF TABLES	xii
 CHAPTER ONE - INTRODUCTION	 1
1.1 Motivation	1
1.2 Original Contributions	2
1.3 Organization of the Dissertation	5
 CHAPTER TWO - LITERATURE REVIEW	 6
2.1 Overview of Container Terminal Operations	6
2.2 Overview of Seaside Problems	8
2.2.1 Berth Allocation Problem	8
2.2.2 Quay Crane Assignment & Scheduling Problems	17
2.2.3 Integrated Berth Allocation, Quay Crane Assignment and Scheduling Problems	22
 CHAPTER THREE - THE PROPOSED SIMULTANEOUS SOLUTION APPROACH FOR MULTI-QUAY DYNAMIC BERTH ALLOCATION AND QUAY CRANE SCHEDULING PROBLEM	 35
3.1 Description of the System	35
3.2 Development of Conflict Free Quay Crane Scheduling Algorithm	39
3.3 Development of the Simulation Model	43
3.4 Development of Simulation Optimization-based Solution Approach	50

3.4.1 Initial Solution Generation	54
3.4.2 Neighborhood Solution Generation	56
3.4.3 Determination of SA Parameter Values using Taguchi Design	57
3.5 Validation of the Proposed Solution Approach.....	60
 CHAPTER FOUR - IMPLEMENTATION OF THE PROPOSED SOLUTION APPROACH	 70
4.1 Representation and Solution of Sample Problem 1	74
4.2 Representation and Solution of Sample Problem 2.....	82
4.3 Representation and Solution of Sample Problem 3.....	91
 CHAPTER FIVE - CONCLUSIONS	 102
 REFERENCES.....	 106
 APPENDICES	 117

LIST OF FIGURES

	Page
Figure 2.1 Operation areas of a container terminal and flow of transport	7
Figure 2.2 Planning problems for container terminals.....	8
Figure 2.3 Representation of two-dimensional solution for the BAP.....	9
Figure 2.4 Berth and quay relationship-discrete layout	10
Figure 2.5 Berth and quay relationship-continuous layout	10
Figure 2.6 Berth and quay relationship-hybrid layout	11
Figure 2.7 A drawing of QCs working on a vessel	18
Figure 3.1 Representation of a sample quayside.....	36
Figure 3.2 Representation of a conflict area between two consecutive QCs	39
Figure 3.3 The first part of the CFQCSA.....	41
Figure 3.4 The second part of the CFQCSA.....	41
Figure 3.5 Search procedure	43
Figure 3.6 General simulation procedure.....	46
Figure 3.7 Part of simulation procedure 1.....	47
Figure 3.8 Simulation procedure 1.....	48
Figure 3.9 Simulation procedure 2.....	49
Figure 3.10 Simulation procedure 3.....	49
Figure 3.11 The procedure of SA algorithm	51
Figure 3.12 Flow chart of the proposed solution approach.....	53
Figure 3.13 Objective value of the candidate solution at each iteration	58
Figure 3.14 Main effect plots for mean objective values (makespan)	59
Figure 3.15 Convergence graph of the proposed solution approach.....	68
Figure 3.16 Berth space-time diagram for the optimum solution	68
Figure 3.17 Berth space-time diagram for the solution of the proposed approach....	69
Figure 4.1 Representation of the quayside for the sample problem 1.....	76
Figure 4.2 Berth space-time diagram for the solution of the proposed methodology for sample problem 1	77
Figure 4.3 Convergence graph of the sample problem 1	78
Figure 4.4 Berth space-time diagram for the solution of the sample problem 1 when it is not crane-constrained.....	80

Figure 4.5 Berth space-time diagram for the solution of the sample problem 1 when it is not crane-constrained with only vessel numbers.....	81
Figure 4.6 Representation of the quayside for the sample problem 2.....	84
Figure 4.7 Berth space-time diagram for the solution of the proposed methodology for the sample problem 2	85
Figure 4.8 Convergence graph of the sample problem 2	86
Figure 4.9 Berth space-time diagram for the solution of the sample problem 2 when it is not crane-constrained.....	89
Figure 4.10 Berth space-time diagram for the solution of the sample problem 2 when it is not crane-constrained with only vessel numbers.....	90
Figure 4.11 Representation of the quayside for the sample problem 3.....	93
Figure 4.12 Berth space-time diagram for the solution of the proposed methodology for sample problem 3	94
Figure 4.13 Berth space-time diagram for the solution of the proposed methodology for sample problem 3 with only vessel numbers.....	95
Figure 4.14 Convergence graph of the sample problem 3	96
Figure 4.15 Berth space-time diagram for the solution of the sample problem 3 when it is not crane-constrained	100
Figure 4.16 Berth space-time diagram for the solution of the sample problem 3 when it is not crane-constrained with only vessel numbers.....	101

LIST OF TABLES

	Page
Table 2.1 Summary of the studies which are related to integrated BAP&QCAP&QCSP	31
Table 3.1 The notation required for solution approach.....	38
Table 3.2 Auxiliary variables used in the simulation procedures	50
Table 3.3 The sample solution string	54
Table 3.4 Berth allocation plan for the sample solution string	54
Table 3.5 Assignment of the first vessel	55
Table 3.6 Assignment of the second vessel	55
Table 3.7 Final assignment of random initial solution.....	56
Table 3.8 Random selection of vessels for swapping	56
Table 3.9 Candidate solution generated by swapping.....	56
Table 3.10 Random selection of vessel for moving	57
Table 3.11 Candidate solution generated by moving.....	57
Table 3.12 Factor levels	58
Table 3.13 L9 orthogonal array based on experimental design employed in this study	59
Table 3.14 The number of vessels and the lengths of vessels in terms of ship-bays. 65	
Table 3.15 The computational results for the problem instances.....	65
Table 3.16 The input data for problem instance 5	67
Table 4.1 Berthing locations, average berthing times, departure times and crane assignment information of the vessels (sample problem 1).....	79
Table 4.2 Berthing locations, average berthing times, departure times and crane assignment information of the vessels (sample problem 2).....	87
Table 4.3 Berthing locations, berthing times, departure times and crane assignment information of the vessels (sample problem 3)	97

CHAPTER ONE

INTRODUCTION

In this chapter, the motivation of this dissertation is given, novel aspects of the study and contributions to the literature are presented and the organization of the dissertation is outlined.

1.1 Motivation

Nowadays, both containerization and globalization have raised the importance of maritime transportation. Maritime transportation has become an important ring of the global supply chains and this leads a tough competition between container terminals. To succeed in this competitive environment, container terminals should reduce their operational costs by efficiently utilizing resources such as human, berths, quay cranes, yards and various equipment. Considering these resources, the operations in a container terminal can be broken down into three functional systems: seaside operations, yard operations and land-side operations. Among these three functional systems, to attract new customers and to improve customer satisfaction, container terminals should primarily focus on efficient management of seaside operations. Moreover, they should provide high level of service and should increase port throughput by reducing the required time for seaside operations.

There are three typical problems in seaside operations of a container terminal: The first one is the Berth Allocation Problem (BAP) which relates to determining the berthing times and berthing locations of incoming vessels. The second one is the Quay Crane Assignment Problem (QCAP) which relates to determining the number of QCs assigned to each vessel. The last one is the Quay Crane Scheduling Problem (QCSP) which relates to determining the sequence of handling tasks that each QC will perform on each incoming vessel. Since the quay crane schedules directly affect the total handling time of a vessel, the QCAP and the QCSP are strongly interrelated with the BAP. Thus, solving the BAP, the QCAP and the QCSP simultaneously is more realistic. In addition, in most of the papers related to the BAP, the quayside is assumed

to be a single line. However, in today's ports, most container terminals consist of multiple quay sections which are separated by discontinuities such as curves or sharp corners. These discontinuities are caused by geographic structure or the artificial layout design of the terminal. Furthermore, there are some uncertainties in handling times due to the factors such as weather conditions, lack of information, maintenance and reliability of equipment. This stochasticity complicates preparing a conflict-free quay crane schedule beforehand.

The aforementioned characteristics and dynamics of the problems with real-life constraints in seaside operations at a container terminal are main source of motivation in this dissertation. The main objective of this dissertation is to propose a simulation optimization-based solution approach for solving integrated Dynamic Multi-Quay BAP (DMQBAP) and QCSP with stochastic handling times and non-crossing constraints of quay cranes.

1.2 Original Contributions

Managing multiple, simultaneous terminal operations together with all the associated handling equipment and vehicles is a challenging task. In addition, real-time solutions with changing conditions and uncertainties of the container terminal operations make efficiently management of the terminals even more challenging. Thus, Operational Research (OR) has become increasingly important in recent years for the management of container terminals due to the complex nature of multiple operations running simultaneously and a considerable number of studies have been conducted using OR methods in operational planning at container terminals. Most of these studies especially focus on problems of seaside operations: BAP, QCAP and QCSP.

In the literature, there are numerous studies which handle the BAP, the QCAP and the QCSP independently. These problems have been solved under some assumptions using prospective arrival and loading/unloading times, in order to ignore the interrelation between the berth allocation and quay crane assignment/scheduling

decisions. However, since the number of QCs assigned to a vessel and the operation schedule of the QCs on the vessel directly affects the handling time of the corresponding vessel, the simultaneous solution of the BAP, QCAP and QCSP gives more realistic results. In this context, in recent years, integration of the BAP, QCAP and QCSP started to attract researchers' attention.

When the relevant literature is reviewed, the gaps on the integrated berth allocation and quay crane scheduling problem can be stated as follows: There are not so many existing studies on solving the Multi-Quay Hybrid Berth Allocation & Quay Crane Scheduling Problems simultaneously, especially ones that focus on the solutions obtained from the simulation optimization methodology which also integrates numerous real-life constraints in the solution of this problem (See Table 2.1).

In order to fill the gaps on the integrated BAP and QCSP in the literature, in this dissertation, a simulation optimization-based solution approach is proposed for solving integrated Dynamic Multi-Quay BAP (DMQBAP) and QCSP with stochastic handling times and non-crossing constraints of quay cranes. The development of the solution approach consists of four phases: 1) Developing a Conflict-Free Quay Crane Scheduling Algorithm (CFQCSA). 2) Developing a general parametric simulation model which represents the seaside operations of a typical container terminal by including most of the realistic features of a port. 3) Developing a Simulated Annealing (SA) based simulation optimization procedure to minimize the completion time of the last vessel in a port. 4) Validation of the proposed simulation optimization procedure.

In the first phase, an algorithm is proposed in order to obtain a conflict-free quay crane schedule assuming that identical QCs are located on a rail and bypassing each other is not allowed. In addition to this, the handling times are stochastic and QCs can only move on their specified areas which may overlap. In the second phase, the general parametric simulation model is developed by using ARENA 14.0 Simulation Software. This model simultaneously handles the DMQBAP and the conflict-free QCSP with stochastic handling times based on the proposed algorithm developed in the first phase. In the third phase, an SA-based simulation optimization procedure is

developed in MATLAB and it is integrated with the simulation model which is developed in the second phase in order to minimize the makespan. In this procedure, the SA algorithm finds a candidate solution which gives vessel-to-berth assignments and this solution is passed to the simulation model as input. Then, the simulation model is executed for estimating the makespan of this berth allocation plan and passes the estimated makespan to the SA algorithm as fitness value. This process continues until the termination criterion is met. In the fourth phase, the proposed simulation optimization procedure is validated for deterministic handling times with respect to an existing Mixed-Integer Programming (MIP) model in literature, which solves the integrated BAP and QCSP, in order to demonstrate how the proposed procedure converges to the minimum makespan value. Afterwards three different sample cases which have most of the realistic features of a port are generated and solved by the procedures proposed in the preceding phases. The results show that the proposed solution approach is efficient in solving simultaneous BAP and QCSP considering the stochastic and dynamic nature of the seaside operations with real-life constraints and it provides the decision makers with a detailed berth and quay crane schedules in a reasonable time.

The main contributions of the solution approach proposed in this dissertation can be summarized as follows:

- It integrates deeply the BAP and the QCSP with multi-quay hybrid berth layout, dynamic arrivals of vessels, coverage area and non-crossing constraints of QCs and stochastic handling times, for the first time.
- It provides a general parametric simulation model that can be adapted easily to different container terminal configurations.
- It integrates the SA-based search procedure with the developed simulation model to suggest the decision-maker the estimated berthing/unberthing times, berthing locations and quay crane schedules for each vessel for the most probable scenario in order to inform incoming vessels before their arrivals.

1.3 Organization of the Dissertation

This dissertation consists of five chapters. The introduction is given in the first chapter and the remainder is organized as follows:

In the second chapter, a general overview of the container terminal operations is given. Moreover, problems encountered in seaside operations have been classified as BAP, QCAP & QCSP, integrated BAP & QCSP. Also, a brief literature review for each problem type is presented.

In the third chapter, first, a real-world inspired container terminal which the solution approach handles is described. Afterwards, the four phases of the development of the solution approach are explained in detail: Development of a QC scheduling algorithm with stochastic loading/unloading times and non-crossing constraint of QCs, development of the simulation model which represents seaside operations of a real-world inspired container terminal and development of a SA-based simulation optimization procedure are explained, respectively. Furthermore, the validation of the proposed solution approach is presented as the fourth phase of the solution approach. The selected MIP model for the validation of the proposed solution approach is described. Also, different deterministic problem instances are solved by both MIP model and the proposed solution approach to show how the proposed approach converges to the optimum solution.

In the fourth chapter, implementation of the proposed solution approach on three different real-world inspired container terminals is presented. Using the hypothetical data sets, numerical experiments are conducted.

In the last chapter, concluding remarks and future research directions for the integrated BAP and QCSP are highlighted.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview of Container Terminal Operations

Although there exists container terminals which are different in size, function and layout, container terminals basically comprise of the same three sub-systems:

- Seaside operations
- Yard operations
- Land-side operations.

Between these sub-systems, there exists two interfaces which ensure transshipment of the containers in the container terminal (see Figure 2.1). The first one is quayside which is located between the seaside operations and yard operations. At the quayside, unloading and loading operations of containers are performed. When a vessel arrives at the port, it waits for its own berthing time to allocate the corresponding berthing location along the quayside with respect to a pre-determined berth plan. Afterwards, quay cranes which have been assigned to the vessel perform loading/unloading operations of containers according to a given schedule. Mostly, unloading operations of the containers precede the loading operations.

Unloaded (import) containers are located to the yard where they are stored temporarily and picked up by the specific cranes or lifting equipment and distributed to the respective stocks in the yard. In some container terminals, there exists sheds or empty depots in the yard for additional logistics services. If sheds or empty depots exist, additional moves are performed; these moves include the transportation between empty stock, packing area, import and export container stocks.

The second interface within the container terminal is hinterland that links the yard operations to land-side operations and where containers are loaded and unloaded

on/off trucks and trains. In the hinterland, the import containers are loaded on trucks and trains to follow their destination route.

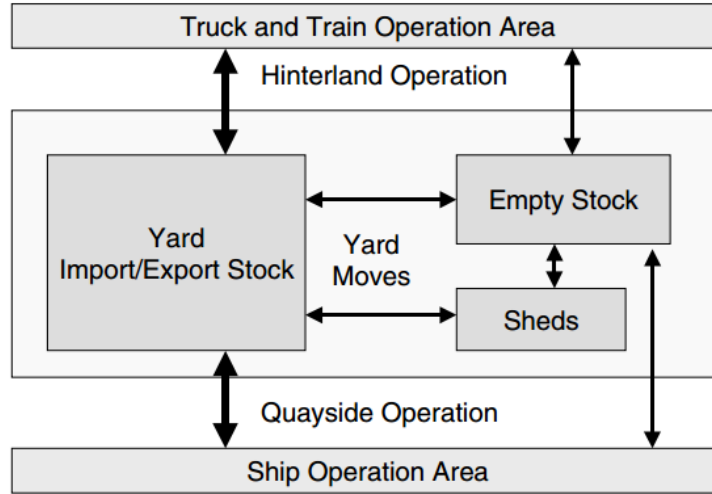


Figure 2.1 Operation areas of a container terminal and flow of transport (Steenken, Voß & Stahlbock, 2005)

The chain of operations for export containers can be described as follows: After arrival at the terminal by truck or train, the container is identified and registered with its major data (*e.g.*, content, destination, outbound vessel and shipping line), picked up by internal transportation equipment, and distributed to storage area in the yard. The operations necessary to handle an export container are performed in the reverse order of an import container (Günther & Kim, 2006).

Considering the transshipment process of a container, Bierwirth & Meisel (2010) classify the planning problems at a container terminal as strategic and operational planning problems. Also, they have summarized the relations of these planning problems with respect to seaside, yard and land-side operations (See Figure 2.2).

As shown in Figure 2.2, BAP, QCAP and QCSP are classified as operational planning problems of seaside operations and these problems are strongly interrelated with each other. Moreover, berth layout directly impacts the solution of the BAP and workforce planning of the container terminal is affected by quay crane assignment decisions.

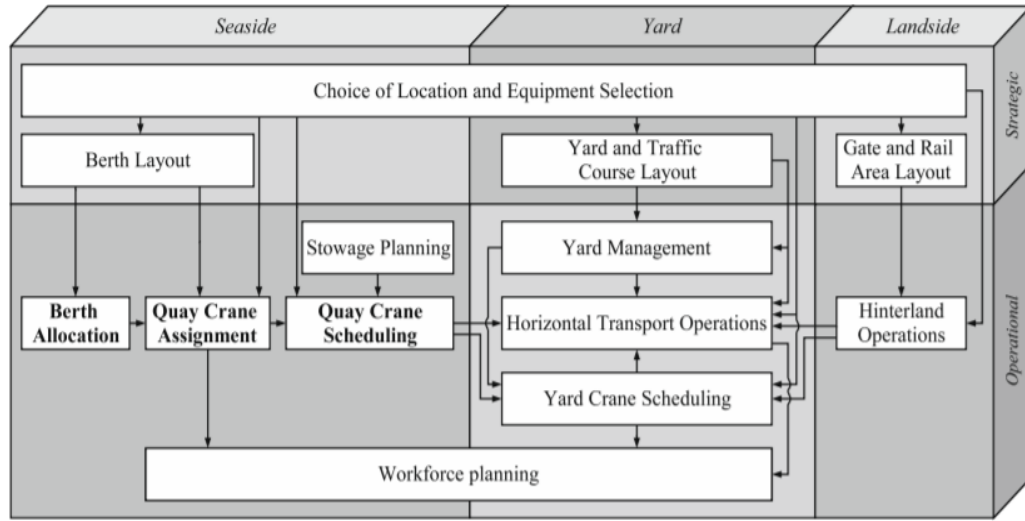


Figure 2.2 Planning problems for container terminals (Bierwirth & Meisel, 2010)

2.2 Overview of Seaside Problems

The major problems encountered in operations of container terminals have been comprehensively reviewed by Vis & Koster (2003), Steenken et al. (2005) and Stahlbock & Voß (2008). In addition, Bierwirth & Meisel (2010, 2015) can be referred for detailed classification of the papers which are related to the seaside operations of a container terminal. They have reviewed and classified the papers according to basic features of BAP, QCAP and QCSP. Moreover, they used the “Integration Type” concept for the first time in order to classify simultaneous solution approaches for BAP, QCAP and QCSP.

2.2.1 Berth Allocation Problem

The BAP relates to determine an optimal schedule to sequence and assign vessels to berthing locations along a quay. While solving the BAP, container terminal managers face two interrelated decisions: where and when the vessels should berth. Primary objectives of the BAP are maximizing berth utilization, minimizing total service time of all vessels in the planning period and minimizing the completion time of handling operations of all vessels (*i.e.*, makespan).

Before the arrival of a vessel, the required berthing space has to be allocated taking the prospective time that the vessel spends in the terminal into account. Also, for each vessel additional data like the vessel's length including safety margin, its draft, the expected time of arrival and the estimated handling time can be given (Bierwirth & Meisel, 2009).

The problem can be represented in a two-dimensional space (See Figure 2.3), where vessels are rectangles whose dimensions are length, including safety margin, and handling time. These rectangles must be placed in the decision space without overlapping each other and a set of constraints must be satisfied at the same time.

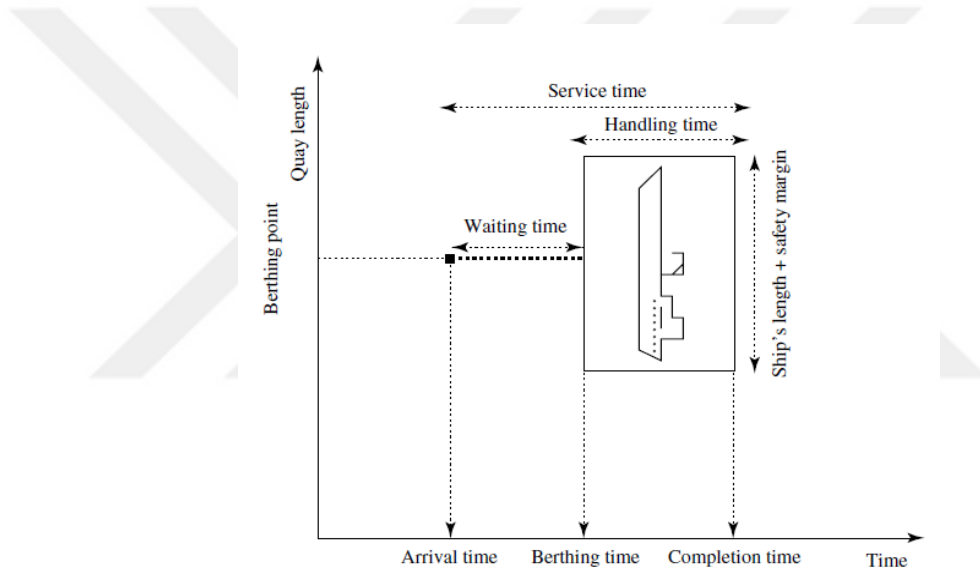


Figure 2.3 Representation of two-dimensional solution for the BAP (Cordeau, Laporte, Legato & Moccia, 2005)

According to Imai, Sun, Nishimura & Papadimitriou (2005) the following cases are represented as the layout constraint for the berth allocation problem:

- **Discrete Layout:** In discrete berth layout, the quayside is assumed to be partitioned into a number of discrete berths with different length and depth (See Figure 2.4). Each berth is allowed to be allocated by only one vessel at a time, in accordance with the vessel's length and draft. The discrete BAP can be regarded as parallel machine scheduling problem with unrelated machines where job and a machine correspond to a vessel and a berth, respectively.



Figure 2.4 Berth and quay relationship-discrete layout (Bierwirth & Meisel, 2010)

- **Continuous Layout:** In continuous berth layout, the quayside is assumed to be a single continuous line, vessels can berth anywhere along the quay within the boundaries (See Figure 2.5). Although the continuous berth layout increase the berth utilization and productivity by ensuring flexibility for the berth allocation, solving continuous berth allocation problem is more complicated than discrete one.

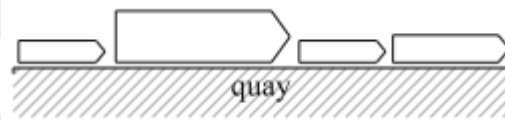


Figure 2.5 Berth and quay relationship-continuous layout (Bierwirth & Meisel, 2010)

- **Hybrid Layout:** In hybrid berth layout, the quayside is assumed to be partitioned into discrete berths, but large vessels may occupy more than one berth (Figure 2.6a) or one berth can be shared by more than one small vessels (Figure 2.6b). Also, an intended berth can be observed if two opposing berth exist and intended berths can be used to serve large vessels from both sides (Figure 2.6c).

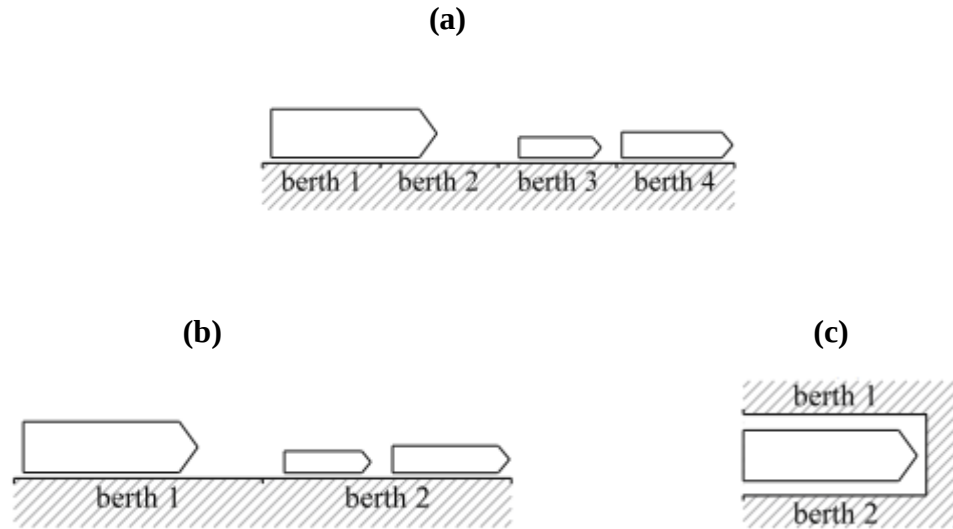


Figure 2.6 Berth and quay relationship-hybrid layout (Bierwirth & Meisel, 2010)

According to Imai, Nishimura & Papadimitriou (2001), if the arrivals of the vessels are considered in the BAP, they are classified in two categories:

- **Static arrival:** In case of static arrivals, all vessels are assumed to be at the port and they are waiting to be serviced.
- **Dynamic arrival:** In case of dynamic arrivals, vessels arrive at the port at different times and they are not allowed to berth before their arrival times. These arrival times can be either deterministic or stochastic. If the arrival times are deterministic, they are known in advance, otherwise arrival times are said to be stochastic.

In the following, the studies conducted between the years 2001-2019, about solving the BAP are summarized.

Imai et al. (2001) propose an algorithm to solve dynamic berth allocation problem. They propose a Lagrangian Relaxation-based heuristic method to obtain a near optimal solution.

Imai, Nishimura & Papadimitriou (2003) address a dynamic berth allocation problem with discrete berth layout. They modify the proposed dynamic berth allocation model considering the different service priorities of incoming vessels and develop Genetic Algorithm (GA) based heuristic to solve the non-linear model that they have proposed before (Imai et al., 2001).

Kim & Moon (2003) propose a mixed integer linear programming (MILP) model for berth allocation problem and they apply simulated annealing algorithm to the problem to find near optimal solution.

Imai et al. (2005) also develop a heuristic algorithm in order to solve the berth allocation problem with continuous berth layout for efficient allocation of the quayside. They make the same assumptions as previously made in Imai et al. (2001) and handle the problem as a Cutting Stock Problem.

Cordeau et al. (2005) propose a Tabu Search heuristic for the dynamic BAP in two versions with both discrete and continuous location cases. They propose a new model for the discrete berth allocation problem. A Tabu Search-based heuristic is developed to solve the discrete berth allocation problem and is extended to solve the continuous berth allocation problem. The objective of the models is the minimization of the total service time for all vessels which is the time elapsed between the arrival and completion of handling operations of the vessel.

Imai, Nishimura, Hattori & Papadimitriou (2007) propose a new integer linear programming model and extend the model for berth allocation problem with intended berths in order to serve mega-containerships and feeders more efficiently. They solve the model by GA- based heuristic method.

Monaco & Sammarra (2007) formulate the discrete BAP as dynamic scheduling problem. They develop Lagrangean heuristic algorithm and present computational results of their formulation.

Wang & Lim (2007) propose a mixed integer programming model which minimizes the total allocation cost. They transform the berth allocation problem into a multi-stage decision making procedure. Then, they propose stochastic Beam Search for solving the BAP. They carry out computational experiments using real-life data to demonstrate the efficiency of the proposed stochastic beam search algorithm.

Hansen, Oguz & Mladenovic (2008) introduce new MILP for static, dynamic berth allocation problem with priorities. Since only small-sized problems can be solved exactly by using existing optimization software, they develop a Variable Neighborhood Search heuristic to obtain an approximate solution for the minimum cost berth allocation problem that they introduced.

Imai, Nishimura, Hattori & Papadimitriou (2008) address a variation of the berth allocation problem at multi-user terminals. In their study, it is assumed that the vessels which would normally be served at the terminal but their expected waiting time exceeds the time limit, are assigned to an external terminal. The objective of the problem is to minimize the total service time of ships at the external terminal. A GA-based heuristic is developed and various numerical experiments are performed. Results of the experiments show that the heuristic developed performs well in reducing external terminal usage.

Lee & Chen (2009), develop a neighborhood-based search heuristic method to solve berth allocation problem which is dynamic in nature. In their study, the quay is assumed to be continuous line and vessels have to be berthed within one of preferred berthing locations previously determined. They also consider some practical constraints such as safety margin between vessels, first-come-first-served rule of the vessels and vessel shifting.

Cheong, Tan, Liu & Lin (2010) propose a Multiobjective Evolutionary Algorithm (MOEA), combined with a local search heuristic, a hybrid solution decoding scheme, and an optimal berth insertion procedure. They solve the berth allocation problem so as to minimize makespan, waiting time and degree of deviation from a predetermined

priority schedule. Proposed hybrid solution decoding scheme ensures hybridization of both discrete and continuous BAPs and berth allocation decision is made in accordance with hybrid berth layout.

Lee, Chen & Cao (2010) solve the continuous and dynamic BAP to minimize the total weighted flow time. They propose an efficient method which determines the berthing location for the next vessel. Then, they develop two versions of Greedy Randomized Adaptive Search Procedure (GRASP) to find near optimal solutions. To demonstrate the effectiveness of the proposed GRASPs, small and large sized numerical experiments are performed and compared with CPLEX and stochastic Beam Search, respectively.

Arango, Cortes, Munuzuri & Onieva (2011) handle the problems associated with allocating berths for containerhips in the port of Seville. This paper addresses the berth allocation planning problems using simulation and optimization with ARENA software. A mathematical model is proposed and heuristic procedure based on genetic algorithm is developed to solve non-linear problems. The objective of the study is to minimize the total service time for each ship and considers a first-come-first-served allocation strategy.

Buhrkal, Zuglian, Ropke, Larsen & Lusby (2011) consider the dynamic discrete berth allocation problem. They review different three models for the dynamic discrete berth allocation problem that proposed by Imai et al. (2001), Monaco & Sammarra (2007) and Cordeau et al. (2005). They select one model and improve its performance. Through extensive numerical experiments, they compare all models from a computational perspective and demonstrate that General Set Partitioning model outperforms all other reviewed models.

Du, Chen, Quan, Long & Funk (2011) propose a model on berth allocation considering fuel consumption and overcome the nonlinear complexity by casting it as a mixed integer second order cone programming model. Furthermore, they conduct the vessel emission calculation with the widely-used emission factors. Besides, vessel

emissions in mooring periods are also analyzed through a post-optimization phase on waiting time. Experimental results demonstrate that the new berth allocation strategy, reflected by the proposed model, was competent to significantly reduce fuel consumption and vessel emissions, while simultaneously retaining the service level of the terminal.

Xu, Li & Leung (2012) consider both static and dynamic cases of a berth allocation problem in container terminals in which the assignments of vessels to berths is limited by water depth and tidal condition. They model the problem as a parallel-machine scheduling problem with inclusive processing set restrictions. They analyze the computational complexity and develop efficient heuristics for these two cases. Computational experiments are conducted to test the effectiveness of the heuristics and to evaluate the benefits of taking tidal condition into consideration when making berth allocation decisions.

Karafa, Golias, Ivey, Saharidis & Leonardos (2013) formulate the berth allocation problem with stochastic vessel handling times as a biobjective problem that minimizes the total service time for all vessels and minimize the service start and finish time risk for all vessels. An Evolutionary Algorithm-based heuristic and a simulation-based Pareto Front Pruning algorithm are proposed to solve the resulting problem. Computational examples show that the proposed approach provides solutions superior to the ones where the expected value of the vessel handling times was used.

Lin & Ting (2014) focus on both discrete and continuous cases of the dynamic berth allocation problem. They propose SA approaches to solve the dynamic berth allocation problems. The proposed SA approaches are tested with numerical instances for different sizes from the literature.

Ting, Wu & Chou (2014) propose a Particle Swarm Optimization (PSO) algorithm to solve the discrete and dynamic berth allocation problem. They test their algorithm with two set of benchmark instances from the literature and demonstrate that their proposed PSO algorithm outperforms.

Hu (2015a) emphasizes the daytime preference of the container terminals since carrying out the loading /unloading operations in daytime increases working safety and energy saving degrees. The proposed dynamic discrete berth allocation model which previously developed by Cordeau et al. (2005) is extended. Multi-objective GA is addressed to solve the proposed bi-objective model.

Zhen (2015) proposes two formulations for solving tactical berth allocation problem: A stochastic programming formulation and a robust formulation. Also, the analytic relationship between two formulations are examined and solution methods are suggested for the models. Validation of the models are performed by numerical experiments to show the effectiveness of the formulations.

Lalla-Ruiz, Expósito-Izquierdo, Melián-Batista & Moreno-Vega (2016) address the BAP with time-dependent restrictions. They aim to assign all incoming vessels to the berthing positions considering the water depth and tidal constraints. They propose a mathematical model based on the generalized set partitioning model and compare their model with a model which is given in related literature (Xu et al., 2012).

Mauri, Ribeiro, Lorena & Laporte (2016) propose an Adaptive Large Neighborhood Search heuristic to solve both discrete and continuous berth allocation problems. Numerical experiments are conducted for various set of problem instances. Results indicate that proposed heuristic provides good solutions in low computational time.

Umang, Bierlaire & Erera (2017) study on the hybrid berth allocation problem in real time assuming that actual arrival times and handling times of vessels may deviate from their expected values and this variation causes disruption in a pre-determined berth plan and make it infeasible. They propose an optimization-based recovery algorithm based on the Set Partitioning and a Smart Greedy Algorithm to reassign vessel in case of disruption.

Xiang, Liu & Miao (2017) study on the berth allocation problem under uncertainty. They propose a bi-objective robust berth allocation formulation which considers economic performance and customer satisfaction. They develop an Adaptive Grey Wolf Optimizer Algorithm and effectiveness of the developed algorithm and model are verified by numerical experiments.

Lin, Ting & Wu (2018) aim to develop SA algorithms to solve the dynamic, continuous berth allocation problem whose objective is to minimize the total weighted service time and the deviation cost from vessels' preferred position.

Song, Zhang, Liu, & Chu (2019) study a tactical berth allocation problem under the consideration of time-varying water depths when making the vessel berthing decisions for the jumbo container vessels. They propose an integer programming model, a set partitioning formulation and an improved Differential Evolutionary Algorithm for resolution of the problem.

2.2.2 Quay Crane Assignment & Scheduling Problems

Quay crane assignment and scheduling problems are directly influence the effectiveness of the seaside operations as well as the competitive advantages of the container terminal.

The QCAP relates to determine the number of QCs assigned to a vessel for handling loading/unloading operations. The problem is classified as time-variant and time-invariant assignment of QCs. In the time-variant assignment problems which increase the QC utilization, the number of QCs assigned to a vessel can change until the completion of loading/unloading operations of the vessel. In case of time-invariant assignment, the number of QCs assigned to a vessel is not allowed to change. When the related literature about QCAP is considered, the all QCAP are integrated with BAP and the integrated QCAP and BAP are reviewed in Section 2.2.3.

The QCSP relates to find a schedule for the QCs with respect to a given objective function which is mostly minimization of the handling time of a vessel. The schedule assigns a particular QC to a loading/unloading task and determine a starting time for handling the task for each QC. Also, the QCSP determines operation sequence of the QCs assigned to a vessel. Figure 2.7 provides a sketch of QCs working on a vessel.

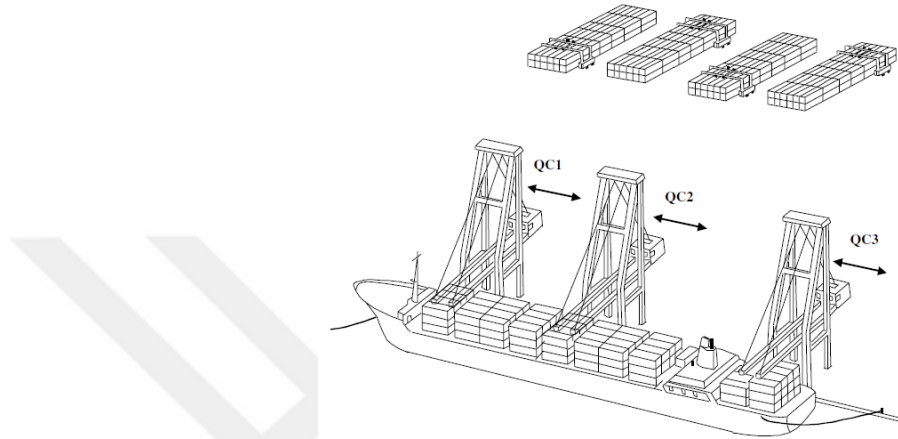


Figure 2.7 A drawing of QCs working on a vessel (Kim & Park, 2004)

In the following, the studies about solving the quay crane assignment and scheduling problems are summarized.

Scheduling problems with interference constraint of quay cranes in ports, rail terminals and warehouses and production environments are reviewed in detail by Boysen, Briskorn & Meisel (2017).

Liu, Wan & Wang (2006), study the quay crane scheduling problem at a container terminal to minimize the maximum relative tardiness of vessels with different arrival times. The problem is formulated as a mixed integer linear programming model. Due to the difficulty of solving such a model directly, a heuristic approach which decomposes the whole problem into two levels is proposed. Computational experiments are carried out on a large set of problems to evaluate the performance of the heuristic approach against a lower bound.

Sammarra, Cordeau, Laporte, & Monaco (2007) propose a Tabu Search metaheuristic for the QCSP, the problem of scheduling a fixed number of quay cranes in order to load and unload containers into and from a ship. The objective function of the problem is minimization of completion time. Precedence and non-simultaneity constraints between tasks are taken into account. The QCSP is decomposed into a routing problem and a scheduling problem. The routing problem is solved by a Tabu Search heuristic, while a Local Search technique is used to generate the solution of the scheduling problem. The effectiveness of their algorithm is assessed by comparing it to a branch-and-cut algorithm and to a GRASP.

Canonaco, Legato, Mazza & Musmanno (2008) aim to solve a real problem of logistics at a maritime container terminal by developing a complex queuing network model. It is described by an event graph-based discrete-event simulation and validated against real data. The developed simulator includes several features that were designed to support some specific management decisions, such as dynamic allocation of SCs (straddle carrier) to a specific berth crane, planning and scheduling of cranes and real-time tracking of actual completion of loading/unloading operations at any given hold. A panel-like, manager-friendly framework is implemented to easily perform some what-if experiments for optimum seeking by simulation.

Lee, Wang & Miao (2008) study the quay crane scheduling problem to determine a handling sequence of holds for quay cranes assigned to a container vessel considering interference between quay cranes. They provide a mixed integer programming model for the considered quay crane scheduling problem that is NP-complete in nature. A genetic algorithm is proposed to obtain near optimal solutions. Computational experiments are conducted to examine the proposed model and solution algorithm. The computational results show that the proposed genetic algorithm is effective and efficient in solving the considered quay crane scheduling problem.

Bierwirth & Meisel (2009) present a revised optimization model for the scheduling of quay cranes and proposed a heuristic solution procedure. For this problem, different models and algorithms are proposed. To provide a correct treatment of crane

interference, a revised MIP model is developed. A new solution heuristic is proposed and it is compared with a MIP solver and other methods.

Moghaddam, Makui, Salahi, Bazzazi & Taheri (2009) present a novel, mixed-integer programming (MIP) model for the quay crane (QC) scheduling and assignment problem in a container terminal. Obtaining an optimal solution for this type of complex, large-sized problem in reasonable computational time by using traditional approaches and optimization tools is extremely difficult. Thus, a GA is proposed to solve the above-mentioned QCSAP for the real-world situations. Further, the efficiency of the proposed GA is compared against the LINGO software package in terms of computational times for small sized problems.

Meisel & Bierwirth (2011), present a new platform for the comparison of models and algorithms with application to quay crane scheduling. They provide a scalable, reproducible, and unbiased test environment, which reflects many important features of crane scheduling that are met in practice as well as in research. In total, they generated 400 test instances and solved these instances by a powerful heuristic, which can serve as a benchmark for future research in the field.

Kavesghar, Huynh & Rahimian (2012) address the quay crane scheduling problem (QCSP). They aim to improve the efficiency of the GA search by using an initial solution based on the S-LOAD rule developed by Sammarra et al. (2007), using a new approach for defining the chromosomes to reduce the number of decision variables, and using new procedures for calculating tighter lower and upper bounds for the decision variables. The effectiveness of the developed GA is also tested using several benchmark instances proposed by Meisel & Bierwirth (2011).

Legato, Trunfio & Meisel (2012) provide a rich model for quay crane scheduling that covers important issues of practical relevance like crane-individual service rates, ready times and due dates for cranes, safety requirements, and precedence relations among container groups. For solving the problem, they employ a branch-and-bound scheme.

Lu, Han, Xi & Erera (2012) construct a QC schedule for an entire vessel with the objective of minimizing the makespan of all container loading/unloading tasks. Considering QC moving time and interference constraints, the concept of contiguous bay operations is proposed and a heuristic is developed to generate QC schedules with this feature. Using the heuristic method, they obtain effective and efficient results.

Chung & Choy (2012), propose a modified genetic algorithm to deal with the problem of determining the service sequence of ship bays in a vessel of each quay crane. The proposed algorithm is tested by solving a set of well-known benchmarking problems and comparing the results obtained with other well-known existing algorithms.

Nguyen, Zhang, Johnston & Tan (2013) develop a new priority-based schedule construction procedure to generate quay crane schedules. From this procedure, two new hybrid evolutionary computation methods based on GA and Genetic Programming (GP) are developed. A local search heuristic is also developed to improve the solutions obtained by GA and GP. The experimental results have shown that the proposed methods are very effective and efficient as compared to the existing methods in the literature.

Al-Dhaheri & Diabat (2015), introduce a new formulation for QCSP which includes real-life circumstances such as bidirectional crane schedules and the ability of cranes to move between bays.

Al-Dhaheri, Jebali & Diabat (2016a), propose a stochastic MIP model to represent the QCSP which takes into account the dynamics and uncertainty factors in handling process of containers. They develop a simulation-based GA to find near optimum solution to the problem.

Al-Dhaheri, Jebali & Diabat (2016b), propose a MIP model for the QCSP considering the various practical constraints such as preemption, non-crossing, safety margin, QC traveling time, initial positions of QCs and vessel stability. To solve the

problem, they develop GA which gives a near-optimal solution in a reasonable computational time.

Al-Dhareri & Diabat (2017), develop a Lagrangian relaxation-based algorithm that decomposes the QCSP by ship, which is solved efficiently as a single ship case.

A. Zhang, W. Zhang, Y. Chen, G. Chen & X. Chen (2017) handle the QCSP with non-crossing constraint of QCs. They propose a Selected Partition-based Algorithm (SPA) for m quay cranes and perform computational experiments to examine the partition-based algorithms. The results show that SPA performs even worse than existing partition-based algorithms, nevertheless, the combination of SPA and Greedy Partition-based Algorithm (GPA) is effective and efficient enough for solving the QCSP.

Msakni, Diabat, Rabadi, Al-Salem & Kotachi (2018) handle the QCSP in container terminals where a task is defined as handling of a single container and subject to different technical constraints including QCs' safety margins, non-crossing, initial positions and nonzero traveling times. Two exact methods are proposed to provide optimal schedules.

Kasm & Diabat (2019), consider the non-crossing and safety margin constraints while solving QCSP and improve the recent novel formulation proposed by Al-Dhareri & Diabat (2017).

2.2.3 Integrated Berth Allocation, Quay Crane Assignment and Scheduling Problems

Considerable studies have been done on the BAP, the QCAP and the QCSP. In the former studies, the berthing positions and berthing time for incoming vessels are determined first, then the assignments of quay cranes to each vessel and operation sequence of the assigned QCs are determined. According to the relationships between

the BAP, the QCAP and the QCSP in the solution procedure, the existing approaches can be classified into two categories: independent approach and integrated approach.

The independent approach means solving the BAP, the QCAP and the QCSP separately. It is clear that the number of QCs assigned to a vessel and operation sequence of the QCs influence the vessel handling time directly. Therefore, in the integrated approach, the QCAP and the QCSP are integrated with the BAP to calculate the vessel handling time and this approach generates more efficient schedules than the independent one. In summary, the independent approach focuses on the performance of the BAP, the QCAP and the QCSP individually, while the integrated approach focuses on the internal connection between them. Simultaneous solution of the BAP, the QCAP and the QCSP includes both factors in the integrated approach.

So far, the BAP, the QCAP and the QCSP have been investigated independently. In this part, only the papers which dealt with the integration and simultaneous solution of BAP, QCAP and QCSP have been investigated. In the light of the classification scheme that Bierwirth & Meisel (2010, 2015) proposed, the papers which deal with the integration and simultaneous solution of BAP, QCAP and QCSP are evaluated and summarized in Table 2.1 according to the following characteristics of the problems to emphasize the novelties of this dissertation:

- **Integration Type:** In the literature, different integration concepts are observed while solving BAP, QCAP and QCSP simultaneously and these concepts are classified in three categories: Sequential integration, deep integration and functional integration. In the sequential integration, BAP, QCAP and QCSP are solved sequentially. However, existing interrelations between the problems are ignored in sequential integration. Since the number of QCs assigned to a vessel and service order of the QCs influence the handling times of vessels directly, two more integration concepts have been developed: In the deep integration, BAP, QCAP and QCSP can be unified in a single model. However, deep integration of the problems is challenging. In functional integration, the problem is divided into sub-levels. An integration mechanism for these sub-

problems is constructed where a feed-back loop conveys the outcome of one problem to another problem. In order to improve the solution, the feed-back mechanism is employed iteratively until a termination criterion is met.

- **Berth Layout:** Berth layout can be either discrete, continuous or hybrid. In discrete berth layout, the quayside is partitioned into discrete berths and only one vessel can be served at each single berth. In continuous berth layout, the vessel can berth anywhere along the quay. Although, continuous berth layout increases the berth utilization, it makes the BAP more difficult to solve. In case of hybrid layout, the quay is partitioned into discrete berths, but vessels can occupy more than one berth or one berth can be shared by more than one vessel.
- **Loading/unloading times:** Loading/unloading times of a vessel can be either deterministic or stochastic. Deterministic loading/unloading times are known in advance while stochastic loading/unloading times are random due to the uncertainties emerging from the disruptions.
- **Vessel arrival:** In the BAP, arrival of the vessels can be either static or dynamic. If the arrival of the vessels is static, it is assumed that all vessels have been arrived at the port at the same time and waiting for being serviced. In case of dynamic arrivals, vessels arrive at the port at different times and these arrival times can be either deterministic or stochastic. If the arrival times are deterministic, they are known in advance, otherwise arrival times are said to be stochastic.
- **Task Attribute:** Bierwirth & Meisel (2015) describe the task attribute as the aggregation of a vessel's containers into crane tasks and classified the aggregation in four ways: A task can be defined as loading/unloading operations of all containers at bay of a vessel (*bay*), single container groups of a bay (*group*), container stacks of a bay (*stack*) or a single container (*container*).
- **Non-Crossing constraint of QCs:** Since the QCs are rail-mounted, they cannot pass each other. This movement restriction which is called as non-crossing constraint of QCs directly affects the loading/unloading times.
- **Coverage area of QCs:** The area that each QC may operate on is limited and QC can only move and operate on its specified area which is called as coverage

area. Coverage areas of the QCs may be overlapped and there may be conflict if the QCs move in opposite directions on the overlapped area.

- **Assignment of QCs:** In the QC assignment process, if the number of QCs assigned to a vessel is constant during the loading/unloading process, time-invariant QC assignment has been made. Otherwise, the time-variant QC assignment is subject.
- **Methodology and objective function:** Since the BAP, QCAP and QCSP are generally optimization problems that have many conflicting constraints, various operations research methods and techniques have been applied. These solution techniques and objective functions of the problems are also presented in Table 2.1.

In the following, the papers which are related to simultaneous solution of BAP, QCAP and QCSP have been reviewed and they are grouped with respect to the integration type of the problems.

Park & Kim (2005) raise the idea of integrating the BAP, the QCAP and the QCSP sequentially for the first time. A MIP model and a Lagrangian-based heuristic method are proposed to solve the BAP and the QCAP simultaneously in their paper. The QC schedules are determined subsequently. Correcher & Alvarez-Valdes (2017) and Agra & Oliveira (2018) also integrate the same problems sequentially. Correcher & Alvarez-Valdes (2017) propose a biased random-key GA for solving a variant of BAP and QCAP and its extension as QCSP. The objective function is minimization of the overall cost including waiting cost, delay cost, deviation cost and exceeding horizon cost incurred by the terminal. Agra & Oliveira (2018) handle a real-case of an integrated BAP, QCAP and QCSP considering a set of heterogeneous QCs. They present a flexible mathematical model for the problem and propose heuristic methods to solve the model.

Numerous studies focus on deep integration of the different combinations of BAP, QCAP and QCSP. Imai, Chen, Nishimura & Papadimitriou (2008) apply GA to obtain a berth allocation plan. Then, a maximum flow based algorithm is applied to determine

the fitness value of the chromosome obtained from the GA in order to obtain an approximate solution for the integrated BAP and QCSP. Meisel & Bierwirth (2009) emphasize the productivities of QCs while solving the BAP and the QCAP simultaneously. They propose a construction heuristic, a local-refinement procedure, squeaky wheel optimization and tabu search to solve these problems. Liang, Huang, & Yang (2009) deal with the simultaneous solution of BAP and QCAP. They propose a formulation to present the problems. Afterwards, a hybrid GA which combines the GA with a heuristic method is applied to obtain an approximate solution. Chang, Jiang, Yan & He (2010) suggest an integration model of BAP and QCAP considering the energy consumption of QCs. They propose a mathematical model which minimizes the total deviation between the actual and the best berthing locations of vessels, total penalty of delayed berthing/departure times and total energy consumption of QCs. They also propose a hybrid parallel GA to obtain an approximate solution. Coverage area of QCs are considered by Zhang, Zheng, Zhang, Shi & Armstrong (2010). They address a MIP model to represent the simultaneous BAP and QCAP and apply sub-gradient optimization technique to solve the problem. Raa, Dullaert & Schaeren (2011) develop a MIP model for the integrated BAP and QCAP considering many real-life features such as priorities of vessels, preferred berthing locations and handling time considerations. They propose a hybrid-heuristic solution procedure for the MIP model. Ursavas (2014) presents a decision support system for optimizing seaside operations at a container terminal. In the study, the BAP, the QCAP and the QCSP are deeply integrated. Berthing positions of vessels, number of QCs assigned to vessels and QCs operating on the vessels are determined dynamically. Also, beside the rail-mounted QCs, rubber-tired QCs are employed for handling operations of the vessels. Hu (2015b) integrates the continuous BAP with QCAP considering the efficient use of QCs assigned to a vessel. Some real-world considerations are also taken into account such as penalties arising due to the early arrivals and departure delays. The problem is formulated as a MIP model. Rolling-horizon and neighborhood search heuristics are applied to solve the problem. Han, Gong & Jo (2015) propose a two-phased model for the BAP and the QCAP. In the first phase, they establish a new continuous berth allocation model considering the coverage area of QCs. In the second phase, they suggest a multi-objective programming model for QCAP and apply particle swarm

optimization for solving the two-phased model. Shang, Cao & Ren (2016) introduce deep integration model of the BAP and the QCAP with QC setup times. However, they consider the uncertainties that may occur in crane productivities. To handle these uncertainties, a robust optimization model is constructed based on the deterministic model. A GA and an insertion heuristic algorithm is applied to obtain near optimal solutions. Giallombardo, Moccia, Salani & Vacca (2010) integrate the BAP and the QCAP at the tactical level. They present two mathematical formulations: mixed integer quadratic program and mixed integer linear program. To solve the problem, they develop a heuristic algorithm which combines Tabu Search and mathematical programming methods. Vacca, Salani & Bierlaire (2012) present an exact algorithm for the integrated planning of berth allocation and quay crane assignment. They design and implement an exact branch and price algorithm to provide good quality bounds and to find optimum solutions to the problem. With the same aim, Lalla-Ruiz, González-Velarde, Melián-Batista & Moreno-Vega (2014) propose a biased random key genetic algorithm which is applicable for efficiently solving the integrated BAP and QCAP.

Functional integration of the problems become more popular in recent years. Lee & Wang (2010) integrate discrete BAP and QCSP functionally. They divide the problem into two parts. The first part determines the berth allocation scheme. In the second part, an approximation algorithm is adopted in order to determine the schedule of ship bay-quay crane assignments with non-crossing constraints. They apply GA to find a near optimal solution which minimizes the makespan. Song, Cherrett & Guan (2012) investigate a bi-level programming approach for simultaneously solving BAP and QCSP. In the upper level, the BAP is formulated and solved by GA. In the lower level, the QCSP is solved using branch and bound algorithm. Working efficiencies of QCs which are determined by the lower level of the problem are used for solving BAP. Thus, the BAP and the QCSP are integrated functionally. Yang, Wang & Li (2012) introduce the BAP and the QCAP as a coupling problem. They consider the interactions between the problems and propose a loop-based algorithm. In this functional algorithm, solution of BAP gives QCAP a temporary berth allocation plan whereas QCAP calculates and feeds back more accurate handling time of a vessel to

improve the efficiencies of berths and QCs. Lalla-Ruiz, Izquierdo, Batista & Moreno-Vega (2013) propose a solution framework which integrates the tactical BAP and QCSP functionally. They apply Variable Neighborhood Search heuristic to get a solution which minimizes the makespan of all vessels. Meisel & Bierwirth (2012) propose an integrated framework of the BAP, the QCAP and the QCSP. This functional integrated framework consists of three phases. In phase I, they estimate the productivity rates of QCs based on the stowage plans of vessels. In phase II, they use the productivity rates to make berthing decisions and to assign QCs to vessels. In phase III, they determine the detailed crane schedule. Another functional integration approach is introduced by Karam & Eltawil (2016). According to this approach, presumed handling times are used to initiate the integration mechanism and to obtain a berth schedule. Using the berth schedule as input, quay crane assignments are made considering the non-crossing constraint. If a feasible quay crane schedule is obtained, new handling times are calculated and sent back to the berth schedule. In order to improve the solution, this feedback which integrates the BAP and the QCSP continues until a stable state is reached. Türkoğulları, Taşkın, Aras & Altınel (2014) present an integer programming model to solve the BAP and the QCAP simultaneously. Afterwards, they extend the model by considering the quay crane schedule. These two models are functionally integrated by a post-processing algorithm and cutting plane algorithm is proposed to solve the problem. Türkoğulları, Taşkın, Aras & Altınel (2016) propose cutting plane algorithm to solve the BAP, the QCAP and the QCSP with setup times of QCs and extend their previous study (Türkoğulları et al., 2014) for the case of time-invariant QC assignment.

When the relevant literature is considered, there are only two studies (Han, Lu & Xi, 2010; Peng-Fei & Kang, 2008) that integrate the BAP and the QCAP, and there is only one study (Legato, Mazza & Gulli, 2014) that integrates the BAP and the QCSP considering the uncertainties in handling times of vessels. Since the first two studies do not consider the QCSP, they ignore modelling the detailed quay crane position and movement, and they do not consider interference constraint of QCs. To demonstrate the relative importance of our proposed approach in regard to these studies, the following theoretical comparison is made.

Peng-Fei & Kang (2008) deal with the discrete BAP and QCAP simultaneously using mathematical programming approach. They propose a stochastic integer programming model that minimizes the average waiting time of vessels at the container terminal and develop a GA for its solution. They use expected values for arrival and loading/unloading times to calculate the expected handling times of vessels, and the scheduling of QCs is not considered.

Han et al. (2010) integrate the discrete BAP and QCAP by formulating a robust nonlinear MIP model due to uncertainties in arrival and handling times of vessels which arrive dynamically with different service priorities and due times. They apply a simulation-based GA to determine berthing positions of vessels and the number of QCs assigned to each vessel simultaneously. For simplification they ignore modelling the detailed quay crane position and movement and they do not consider interference constraint of QCs. Moreover, the model does not include the detailed container stowage plan and QC working schedule on each bay to get the vessel's exact handling time. Thus, they state that their problem presents an upper bound for the integrated BAP and QCAP for the stochastic case.

Legato et al. (2014) is the most similar study to ours. Uncertainties in handling times are also considered by Legato et al. (2014) and they propose a simulation optimization procedure for functional integration of the continuous case of BAP and QCSP in the same framework. They partition the integrated problem into two levels as tactical and operational. They aim to tune the obtained tactical solution at the operational level. Beam search and SA based simulation optimization methods are applied to solve the BAP at each level. In order to evaluate the solution, they use discrete-event simulator and embed the current algorithm used by the management of container terminal for scheduling QCs into the simulator. However, there is no given information about how the QCSP is solved. In our study, hybrid berth layout and deep integration are considered whereas their study assumes the continuous berth layout and functional integration. Moreover, in our proposed simulation model, since the handling time of a vessel is determined after completing the loading/unloading operations, there is no possibility to estimate the loading/unloading time of a ship-bay as in real life.

None of the aforementioned studies except Legato et al. (2014), under stochastic circumstances, propose an algorithm which assigns QCs to the specific loading/unloading tasks and sequences them. The aim of our proposed approach is to provide an estimated berth allocation plan with a detailed QC schedule aside from minimizing the given objective function. However, the relevant studies only focus on the objective function value and they do not suggest the decision-maker a detailed plan of seaside operations.

Aside from the studies which integrate the problems related to seaside operations given above, a rigorous investigation of the existing approaches for the QCSP is also done. To the best of authors' knowledge, there are only two studies (Al-Dhaheri et al., 2016a; Legato, Mazza & Trunfio, 2010) which solve the QCSP with stochastic handling times. These studies solve the stochastic QCSP for only a single vessel. On the other hand, the proposed algorithm in this paper solves the QCSP with stochastic handling times considering dynamic arrivals of multiple vessels, which makes the solution of the QCSP more challenging. Legato et al. (2010) are the first which propose simulation based optimization methodology to handle the QCSP considering both the seaside operations and handling operations of containers between the quayside and stacking yard. An SA algorithm is used to find the schedule that minimizes the expected value of the weighted sum of vessel makespan and the sum of QCs' completion times. Al-Dhareri et al. (2016a) also integrate the yard operations with the QCSP claiming that they are interrelated. Different from the algorithm that Legato et al. (2010) propose, they employ two Quay Crane Scheduling Procedures to obtain more effective solutions: Left to Right Scheduling and Right to Left Scheduling procedures are used and they are integrated with the simulation model using Monte-Carlo Sampling. In our proposed approach, the impact of the transfer operation between quayside and yard on the QCSP is not taken into account unlike Legato et al. (2010) and Al-Dhareri et al. (2016a).

Table 2.1 Summary of the studies which are related to integrated BAP&QCAP&QCSP

Reference	Integration	Integration Type	Berth Layout	Handling Times	Arrival	Task Attribute
Park&Kim (2005)	BAP&QCAP&QCSP	Sequential	Continuous	Deterministic	Static	NA
Imai et al. (2008)	BAP& QCSP	Deep	Discrete	Deterministic	Dynamic/Deterministic	NA
Peng-Fe &Kang (2008)	BAP&QCAP	Deep	Discrete	Stochastic	Dynamic/Stochastic	NA
Liang et al. (2009)	BAP &QCAP	Deep	Discrete	Deterministic	Dynamic/Deterministic	NA
Meisel&Bierwirth (2009)	BAP&QCAP	Deep	Continuous	Deterministic	Dynamic/Deterministic	NA
Chang et al. (2010)	BAP&QCAP	Deep	Continuous	Deterministic	Dynamic/Deterministic	NA
Giallambardo et al. (2010)	BAP&QCAP	Deep	Discrete	Deterministic	Dynamic/Deterministic	NA
Han et al. (2010)	BAP&QCAP	Deep	Discrete	Stochastic	Dynamic/Stochastic	NA
Lee&Wang (2010)	BAP&QCSP	Functional	Discrete	Deterministic	Dynamic/Deterministic	Ship-bay
Zhang et al. (2010)	BAP&QCAP	Deep	Continuous	Deterministic	Dynamic/Deterministic	NA
Raa et al. (2011)	BAP&QCAP	Deep	Continuous	Deterministic	Dynamic/Deterministic	NA
Song et al. (2012)	BAP&QCSP	Functional	Discrete	Deterministic	Static	Group
Vacca et al. (2012)	BAP&QCAP	Deep	Discrete	Deterministic	Static	NA
Yang et al. (2012)	BAP&QCAP	Functional	Continuous	Deterministic	Dynamic/Deterministic	NA
Lalla Ruiz et al. (2013)	BAP&QCAP	Functional	Discrete	Deterministic	Dynamic/Deterministic	Group
Meisel&Bierwirth (2013)	BAP&QCAP&QCSP	Functional	Continuous	Deterministic	Dynamic/Deterministic	Ship-bay
Lalla Ruiz et al. (2014)	BAP&QCAP	Deep	Discrete	Deterministic	Dynamic/Deterministic	NA
Legato et al. (2014)	BAP&QCSP	Functional	Continuous	Stochastic	Dynamic/Stochastic	NA
Turkogullari et al. (2014)	BAP&QCAP&QCSP	Functional	Continuous	Deterministic	Dynamic/Deterministic	NA
Ursavaş (2014)	BAP&QCAP&QCSP	Deep	Discrete	Deterministic	Dynamic/Deterministic	NA
Hu (2015)	BAP&QCAP	Deep	Continuous	Deterministic	Dynamic/Deterministic	NA
Han et al. (2015)	BAP&QCAP&QCSP	Deep	Continuous	Deterministic	Dynamic/Deterministic	NA
Karam&Eltawil (2016)	BAP&QCSP	Functional	Continuous	Deterministic	Dynamic/Deterministic	NA
Shang et al. (2016)	BAP&QCAP	Deep	Continuous	Deterministic	Dynamic/Deterministic	NA
Türkoğulları et al. (2016)	BAP&QCAP&QCSP	Functional	Continuous	Deterministic	Dynamic/Deterministic	NA
Correcher&Alvarez-Valdes (2017)	BAP&QCAP&QCSP	Sequential	Continuous	Deterministic	Dynamic/Deterministic	NA
Agra & Oliveira (2018)	BAP& QCAP&QCSP	Sequential	Continuous	Deterministic	Dynamic/Deterministic	NA
This study	BAP & QCSP	Deep	Hybrid	Stochastic	Dynamic/Deterministic	Ship-bay

Table 2.1 continues

Reference	Non-Crossing	Coverage Ranges of QCs	Assignment of QCs	Methodology
Park&Kim (2005)	✓	X	Time-variant	MIP and heuristic method based on Lagrangean relaxation
Imai et al. (2008)	✓	X	Time-invariant	GA and Maximum Flow Problem-based algorithm
Peng-Fei&Kang (2008)	X	X	Time-invariant	Stochastic integer programming model and GA
Liang et al. (2009)	✓	X	Time-invariant	MIP and hybrid GA
Meisel&Bierwirth (2009)	✓	X	Time-variant	Squeaky wheel optimization & Tabu search
Chang et al. (2010)	✓	X	Time-invariant	Hybrid parallel GA
Giallambardo et al. (2010)	X	X	Time-variant	MILP & MIQP & Tabu search
Han et al. (2010)	X	X	Time-variant	MIP and simulation-based GA
Lee&Wang (2010)	✓	✓	Time-invariant	MIP and GA
Zhang et al. (2010)	✓	✓	Time-variant	MIP and Sub-gradient optimization technique
Raa et al. (2011)	✓	X	Time-variant	MIP
Song et al. (2012)	✓	X	Time-variant	GA and branch and bound
Vacca et al. (2012)	✓	X	Time-variant	Branch and price algorithm
Yang et al. (2012)	✓	X	Time-invariant	Evolutionary algorithm with nested-loop
Lalla Ruiz et al. (2013)	✓	X	Time-variant	Variable neighborhood search heuristic
Meisel&Bierwirth (2013)	✓	X	Time-variant	Hybrid heuristic
Lalla Ruiz et al. (2014)	✓	X	Time-variant	Biased random key genetic algorithm
Legato et al. (2014)	✓	X	Time-variant	Simulation optimization
Turkogullari et al. (2014)	✓	X	Time-invariant	Cutting plane algorithm
Ursavaş (2014)	✓	✓	Time-variant	Bi-objective integer programming & Branch and cut algorithm
Hu (2015)	X	X	Time-variant	MILP
Han et al. (2015)	X	✓	Time-variant	PSO & multi-objective programming
Karam&Eltawil (2016)	✓	✓	Time-variant	MILP
Shang et al. (2016)	✓	X	Time-variant	GA & Insertion heuristic algorithms
Türkoğulları et al. (2016)	✓	X	Time-variant	Cutting plane algorithm
Correcher&Alvarez-Valdes (2017)	✓	X	Time-invariant	Biased random-key genetic algorithm
Agra&Oliveira (2018)	✓	X	Time-variant	MILP
This study	✓	✓	Time-variant	SA based simulation optimization

Table 2.1 continues

Reference	Objective Function
Park&Kim (2005)	Makespan of all vessels (Min)
Imai et al. (2008)	Total service time (Min)
Peng-Fei & Kang (2008)	Average waiting time (Min)
Liang et al. (2009)	Total handling time, waiting time and delay time (Min)
Meisel&Bierwirth (2009)	Total service cost of all vessels
Chang et al. (2010)	Total energy consumption of QCs + total penalty of delayed departure times + total deviation between the actual and best berthing locations of vessels (Min)
Giallambardo et al. (2010)	Housekeeping costs arising from the transshipment flows between ships (Min)
Han et al. (2010)	Total service time + weighted tardiness (Min)
Lee&Wang (2010)	Makespan of handling all vessels (Min)
Zhang et al. (2010)	Total weighted handling cost (Min)
Raa et al. (2011)	Total handling time of vessels + deviation from the preferred berthing location + change in the number of cranes assigned to a vessel during its service (Min)
Song et al. (2012)	Makespan of all vessels + handling costs of all vessels (Min)
Vacca et al. (2012)	Total value of selected QC profiles+housekeeping cost generated by berth allocation plan (Min)
Yang et al. (2012)	Average service times of all vessels + average number of quay crane shift operations per shift (Min)
Lalla Ruiz et al. (2013)	Makespan of all vessels (Min)
Meisel&Bierwirth (2013)	Makespan of all vessels (Min)
Lalla Ruiz (2014)	Housekeeping costs arising from the transshipment flows between ships (Min)
Legato et al. (2014)	Total cost of deviation from the best berthing location (Min)
Turkogullari et al. (2014)	Total cost of deviation from the best berthing location+ the cost of berthing later than the arrival time +the cost of departing than the due time (Min)
Ursavaş (2014)	Total service time + labor cost (Min)
Hu (2015)	Total service cost of all vessels (Min)
Han et al. (2015)	Total cost of deviation from the best berthing location+ total service time of vessels
Karam&Eltawil (2016)	Total handling cost + average handling time per vessel (Min)
Shang et al. (2016)	Total weighted handling time and waiting time of all vessels (Min)
Türkoğulları et al. (2016)	Total cost of deviation from the best berthing location (Min)
Correcher&Alvarez-Valdes (2017)	Overall cost which includes waiting, delay, deviation and exceeding horizon costs (Min)
Agra&Oliveira (2018)	Total service completion time of all vessels (Min)
This study	Makespan of handling all vessels (Min)

In this study, 27 papers have been reviewed with respect to the given characteristics of the integrated BAP, QCAP and QCSP. According to the summary of the review given in Table 1, the following conclusions can be drawn:

- Since integrating the QCSP deeply with other problems is more difficult, only 3 out of 27 papers consider this type of integration.
- Berth layout is continuous in 16 papers and discrete in 11 papers. None of the reviewed papers has considered the hybrid berth layout in the integrated problems.
- In only 3 out of 27 papers, the handling times are assumed to be stochastic whereas the rest of them assume deterministic handling times.
- Only 2 of the papers select the task attribute as “ship-bay”. In the rest of the papers, there is no information about how they have classified the handling operations of the containers as a crane task.
- Coverage areas of QCs have been addressed in 5 out of 27 papers.
- Time-variant QC assignment has been addressed in 19 out of 27 papers.
- In most of the papers, arrivals of the vessels are dynamic and deterministic and non-crossing constraints of QCs are taken into consideration.

In the last row of Table 2.1, the characteristics of the considered problem in this study are shown. Considering the related papers in the literature, this dissertation is novel since it handles the following features in a single problem:

- The multi-quay container terminal with the hybrid berth layout configuration is considered.
- There is a set of vessels with deterministic/dynamic arrival times.
- The crane task is classified as ship-bay.
- QCs are assigned to ship-bays dynamically under the coverage area and non-crossing constraints.
- Ship-bay handling times are stochastic.

CHAPTER THREE

THE PROPOSED SIMULTANEOUS SOLUTION APPROACH FOR MULTI-QUAY DYNAMIC BERTH ALLOCATION AND QUAY CRANE SCHEDULING PROBLEM

3.1 Description of the System

The considered system in this dissertation is a container terminal which comprises quay sections with different lengths. These quay sections are separated from each other via natural or artificial discontinuities. Each quay section is partitioned into discrete berths with different lengths and water depths. The assumptions of this system are as follows:

- Whole quayside consists of berth sections and each berth section contains at most one QC.
- All of the incoming vessels are assumed to be longitudinally divided into bays with respect to their length.
- The length of a single berth section, the length of a ship-bay and the width of a QC are assumed to be equal.
- All of the QCs are identical and each QC has initial position on the quayside.
- All of the berth sections, bays and QCs are indexed in ascending order from left to the right.
- Container handling operations at each bay are defined as a task.

A sample container terminal with 4 quay sections, 8 berths, 70 berth sections and 3 vessels being serviced is depicted in Figure 3.1. Since a single berth section is defined as the smallest unit of a container terminal, each berth consists of different number of berth sections. Similarly, it is assumed that berths form quay sections and quay sections form whole quayside. For example, as it can be seen in the illustrated container terminal, Quay Section 2 is comprised of 3 berths: Berth 3, Berth 4, Berth 5 which have 9, 15 and 6 berth sections, respectively.

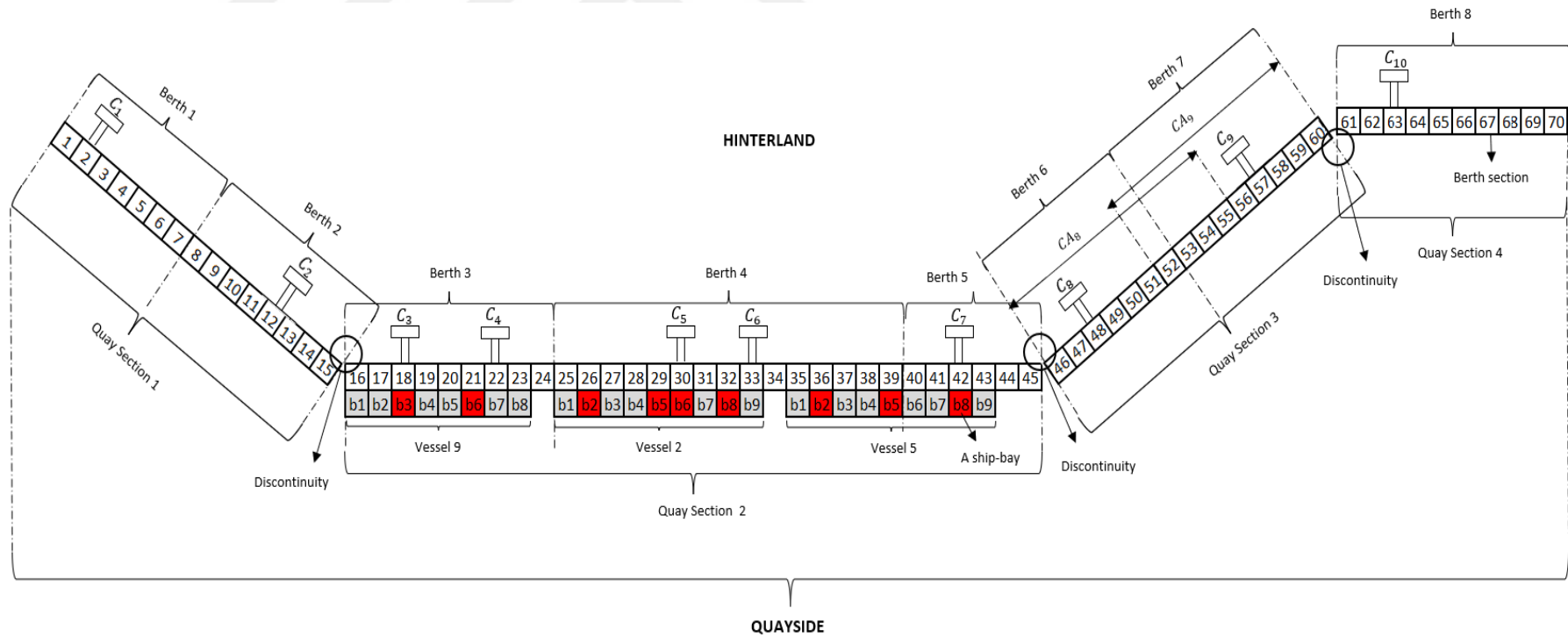


Figure 3.1 Representation of a sample quayside

A sample case illustrated in Figure 3.1 shows that Berth 3 is occupied by Vessel 9, whereas Berth 4 is occupied by both Vessel 2 and Vessel 5. On the other hand, Berth 4 and Berth 5 are both occupied by Vessel 5. As it can be seen in Figure 3.1, there may be discontinuities between two consecutive berths. Due to these discontinuities, it is not physically possible to allocate two consecutive berths simultaneously to a single vessel. If there is no discontinuity between two adjacent berths, a single vessel can be assigned to both berths simultaneously under the condition that the draft of the vessel is smaller than the depths of berths. Allowing simultaneous allocation of berths, a discrete berth allocation plan can be converted into a hybrid berth allocation plan. According to the hybrid berth allocation plan configuration, some vessels may occupy two consecutive berths or one berth may be occupied by more than one vessel simultaneously. In this case, turnaround times of vessels are reduced, waiting times of vessels are decreased and utilization of the quayside is increased.

The notation which is used in representing the solution approach for the problem is given in Table 3.1.

Table 3.1 The notation required for solution approach

Indices & Parameters	Definition
B	The total number of berths
V	The total number of vessels
S	The total number of quay sections
N	The total number of quay cranes operating at the quayside
Q	The total length of quayside in terms of berth sections
i	Index for the berths ($i=1,...,B$)
j	Index for the service order of a vessel ($j=1,...,nv_i$)
s	Index for the quay sections ($s=1,...,S$)
n	Index for the quay cranes ($n=1,...,N$)
$NT_{V_{ij}}$	The total number of tasks on Vessel V_{ij}
r	The handling task number on Vessel V_{ij} ($r=1,...,NT_{V_{ij}}$)
$length_{V_{ij}}$	The length of Vessel V_{ij}
$draft_{V_{ij}}$	The draft of Vessel V_{ij}
$T_{V_{ij}r}$	The r^{th} task on Vessel V_{ij}
$Z_{V_{ij}r}$	The bay number of the r^{th} task on Vessel V_{ij}
$depth_i$	The water depth of Berth i
bs_i	The total number of berth sections at Berth i
S_i	The starting berth section number of Berth i
E_i	The ending berth section number of the Berth i
C_n	The quay crane with number n
FL_n	The final location of C_n
UB_n	The ending berth section of the coverage area of C_n
LB_n	The starting berth section of the coverage area of C_n
Decision variables	Definition
V_{ij}	The vessel number assigned to Berth i with Service Order j
nv_i	The total number of vessels assigned to Berth i
$BT_{V_{ij}}$	The berthing time of Vessel V_{ij}
$VS_{V_{ij}}$	The first berth section number occupied by Vessel V_{ij}
$SC_{nTV_{ij}r}$	The starting time of the r^{th} task of Vessel V_{ij} performed by C_n
Related expressions	Definition
$VE_{V_{ij}} = VS_{V_{ij}} + length_{V_{ij}} - 1$	The last berth section occupied by Vessel V_{ij}
$TL_{V_{ij}r} = VS_{V_{ij}} + Z_{V_{ij}r} - 1$	The location of the r^{th} task on Vessel V_{ij} on the quayside
$E_i = S_i + bs_i - 1$	
$Q = \sum_{i=1}^B bs_i$	
$V = \sum_{i=1}^B nv_i$	

3.2 Development of Conflict Free Quay Crane Scheduling Algorithm

The area in which each QC can operate is restricted. This area is called as Coverage Area (CA) of a QC. The whole quayside should be covered by at least one QC. CAs of the consecutive QCs may overlap or not. In case CAs do not overlap, each QC can operate in its dedicated area without interrupting the operations of consecutive QCs. Since QCs are located on a rail along the quayside in practice and bypassing each other is not allowed, a conflict may occur when CAs overlap as it is depicted in Figure 3.2. This movement restriction on QC which is known as non-crossing constraint directly affects the handling time of a vessel and it makes it difficult to solve dynamic BAP and QCSP simultaneously. For this reason, possible conflicts should be considered before deciding on the sequence of the handling tasks on vessels that each QC will perform. Besides, dynamic behavior of the seaside operations and stochastic handling times make the solution of problem even more difficult.

Due to the dynamic and stochastic nature of the problem, starting and finishing times of handling tasks of QCs cannot be known exactly in advance. Therefore, in order to prevent any QC conflict, some conditions should be evaluated before a QC assignment to a handling task is made.

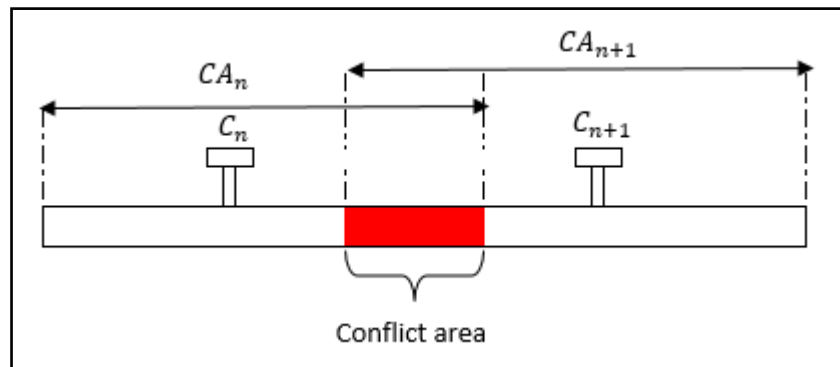


Figure 3.2 Representation of a conflict area between two consecutive QCs

The proposed CFQCSA considers the above-mentioned constraint under the following assumptions:

- All of the QCs are identical.
- Handling operations of a QC at each bay is defined as a task.
- Each task should be completed by only one QC, thus there is no preemption.
- Each QC can occupy only one berth section.
- Handling times are stochastic and the number of containers at a bay is generated randomly.
- Travel time of a QC between two bays depends on the distance between bays and velocity of QC.

An illustrative representation of a quay section with three vessels being handled by five QCs is given in Figure 3.1. In the figure, Vessel 9, Vessel 2 and Vessel 5 have berthed at their specified locations. Bays which need handling operations on each vessel are determined randomly and they are indicated by red squares. The berth section where a QC should be located to complete the task is defined as Task Location ($TL_{V_{ij}r}$). For example, since the second vessel assigned to Berth 4 is Vessel 5, $TL_{V_5 2}$ refers to the location of the second handling task on Vessel 5 and it is set to 39. This means that a crane should move to the 39th berth section on the quayside for the handling task on the 5th ship-bay of Vessel 5.

In the following, a CFQCSA is developed in order to obtain a dynamic conflict-free quay crane schedule for handling all incoming vessels in accordance with a given berth allocation plan. The algorithm is comprised of two parts; the first part of the CFQCSA is triggered whenever each incoming vessel V_{ij} has arrived at its berthing location. For each task r of V_{ij} , the $TL_{V_{ij}r}$ values are calculated and the tasks are assumed to be inserted into a virtual Task Queue (*TaskQ*) in which they are waiting for being handled as it is depicted in Figure 3.3. Also, a signal code is sent to activate the second part of the algorithm which enables the QCs to select a task from the *TaskQ* in order to construct a conflict-free QC schedule on each vessel.

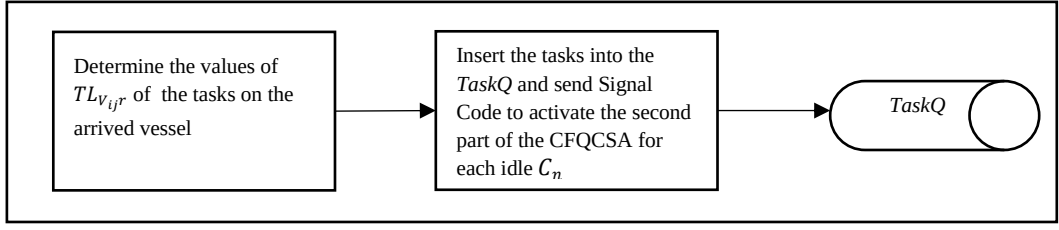


Figure 3.3 The first part of the CFQCSA

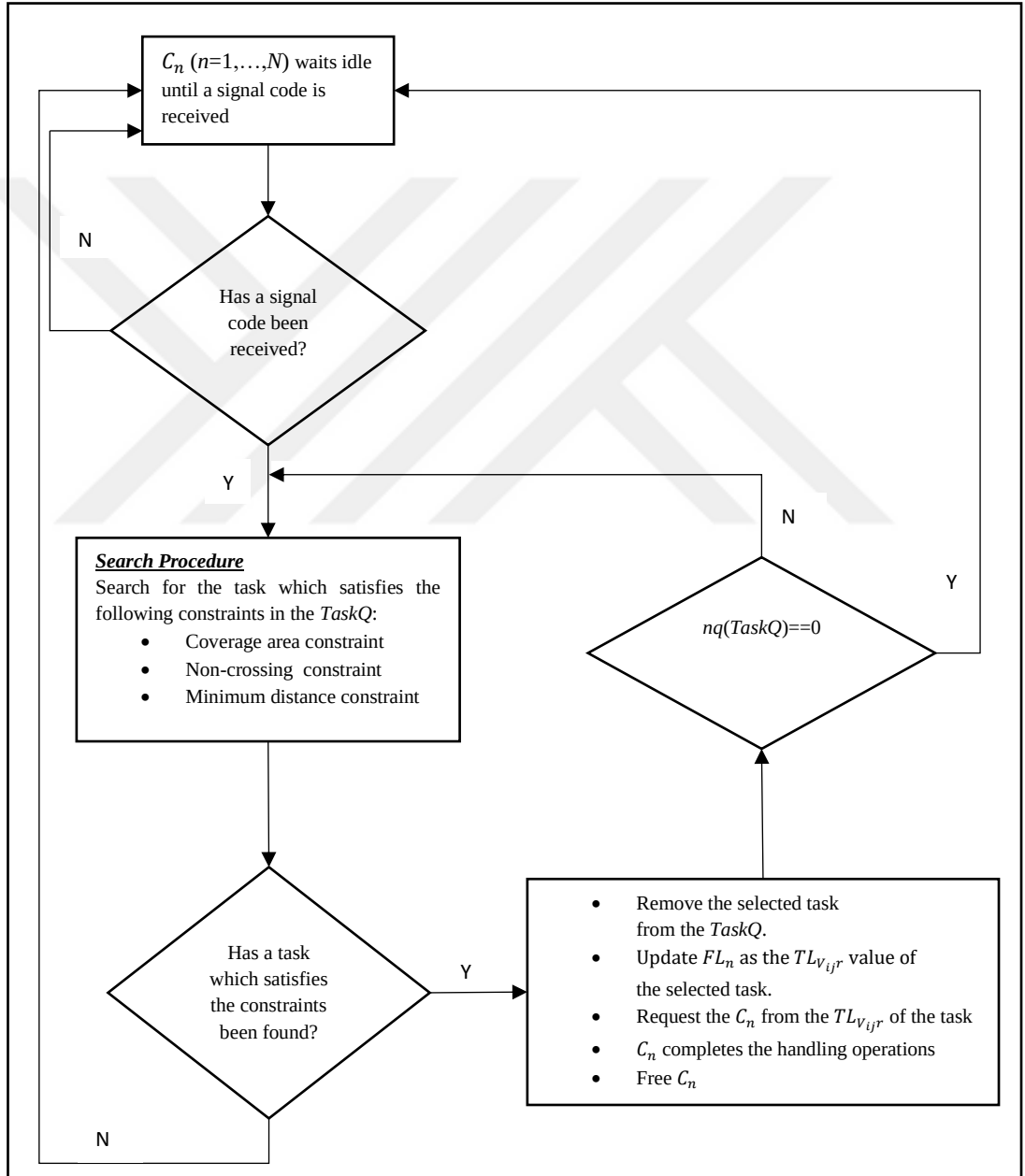


Figure 3.4 The second part of the CFQCSA

The second part of the CFQCSA is triggered by the signal code which is sent from the first part of the CFQCSA as it is depicted in Figure 3.4. As soon as the signal code has been received, all idle QCs attempt to find a handling task in the *TaskQ*. While the QCs are searching for a task, the following constraints must be satisfied respectively;

- Coverage area constraint
- Non-Crossing constraint
- Minimum distance constraint

The first constraint implies that the n^{th} QC can only serve those tasks in the *TaskQ*, which are located between the boundaries of its coverage area. Whenever a task satisfies the coverage area constraint, considering the non-crossing movement restriction of the n^{th} QC, it is looked whether its location is between the destination/final locations of the $(n - 1)^{th}$ QC and the $(n + 1)^{th}$ QC, which are referred to as FL_{n-1} and FL_{n+1} , respectively. These destination/final locations should have been updated to the location of the last task which was handled by the related QC since the state of QCs are not known with certainty in advance due to the stochastic handling times. Among the tasks that satisfy the first two constraints, the one with the minimum distance to the current location of the n^{th} QC is selected and the final location of the n^{th} QC (FL_n) is updated as the location of this selected task. Subsequently, the n^{th} QC moves to the selected task's location, completes the handling operations and if the *TaskQ* is not empty, this search procedure is repeated again by the n^{th} QC. If none of the tasks satisfy all the constraints in the search procedure, the n^{th} QC remains idle until another signal code is received. It must be noted that this search procedure is implemented for all of the idle QCs in crane number order. The pseudo code of the search procedure is given in Figure 3.5. The final locations of all QCs are set to the initial locations of C_n on the quayside at time zero.

```

1. Initialize  $distance \leftarrow M$ ,  $min\_distance \leftarrow M$ ,  $rank\_of\_the\_selected\_task \leftarrow 0$ 
2. For  $k=1:nq(TaskQ)$  (where  $nq(TaskQ)$  is the number of tasks in  $TaskQ$ )
3.  $Location \leftarrow TL_{V_{ij}r}$  value of the  $k^{th}$  task in the  $TaskQ$ 
4.   If  $LB_n \leq Location \leq UB_n$ 
5.     If  $n == 1$ 
6.       If  $Location < FL_{n+1}$ 
7.          $distance \leftarrow |FL_n - Location|$ 
8.       End If
9.     Else If  $n == N$ 
10.      If  $FL_{n-1} < Location$ 
11.         $distance \leftarrow |FL_n - Location|$ 
12.      End If
13.    Else If  $FL_{n-1} < Location < FL_{n+1}$ 
14.       $distance \leftarrow |FL_n - Location|$ 
15.    Else
16.      End If
17.    End If
18.    If  $distance < min\_distance$ 
19.       $min\_distance \leftarrow distance$ 
20.       $rank\_of\_the\_selected\_task \leftarrow k$ 
21.       $Selected\_location \leftarrow Location$ 
22.    End If
23.  $k \leftarrow k+1$ 
24. End For
25. If  $rank\_of\_the\_selected\_task \neq 0$ 
26.    $FL_n \leftarrow Selected\_location$ 
27. Else
28.   Wait for another signal code
29. End If

```

Figure 3.5 Search procedure

3.3 Development of the Simulation Model

In order to evaluate any given candidate berth allocation plan, a general parametric simulation model which handles the Dynamic Multi-Quay BAP and QCSP simultaneously is proposed. Considering the given berth allocation plan, the simulation model determines conflict-free quay crane schedule on the vessels based on the CFQCSA proposed in the previous section and calculates different performance measures such as utilizations of berths and cranes, waiting time of the vessels, maximum completion time of the vessels (makespan) and total turnaround time of all vessels.

The simulation model is developed by using simulation software ARENA 14.0 and it is a parametric model so that any given different quay, berth and QC configurations can be easily adapted. Thus, the simulation model in its current form is applicable as a decision support system for managing the container handling operations of the different container terminals.

The simulation model consists of nested procedures which interact with each other. When the necessary conditions are satisfied, the procedures are triggered. The flow of the general procedure is given in Figure 3.6 and Figure 3.7. The general procedure uses a candidate feasible discrete berth allocation plan of all vessels as an input and whenever the simulation model is run, this procedure is executed for each Berth i in berth number order.

The general procedure starts with the first or j^{th} vessel assigned to Berth i , namely Vessel V_{ij} . First, the arrival of the vessel is checked. If the vessel has arrived and is waiting in the queue for the berth to which it has been assigned, the vessel is taken from the queue VQ_i and Simulation Procedure 2 is activated. Firstly, this procedure searches for a berthing location for Vessel V_{ij} within the boundaries of Berth i . If there is not enough space for this vessel to fit in Berth i , the discontinuity between Berth i and Berth $i+1$ and water depth constraints are checked. If there is no discontinuity between Berth i and Berth $i+1$, and water depth constraint is satisfied, the berthing location for V_{ij} is searched again in a way that the vessel will be placed exceeding the right boundary of Berth i and covering some left part from Berth $i+1$. In other words, Simulation Procedure 2 is activated again by updating the search boundaries. If no berthing location can be found for Vessel V_{ij} , it is inserted into $LIST_i$. When the berthing location can be found for Vessel V_{ij} , the corresponding berth sections are allocated to Vessel V_{ij} . Afterwards, Simulation Procedure 1 is activated again for the next vessel assigned to Berth i and Simulation Procedure 3 is activated for the current Vessel V_{ij} simultaneously.

In Simulation Procedure 3, CFQCSA is executed and the conflict-free quay crane schedule for the handling operations of the vessel is obtained. When the handling operations of the vessel are completed and the vessel releases the occupied quay sections, the corresponding $LIST_i$ is checked. If there is a vessel waiting in $LIST_i$, that vessel is deleted from $LIST_i$ and Simulation Procedure 1 is activated for this vessel.

The general procedure continues until nv_i vessels have been handled at the container terminal for all i . The pseudo codes of Simulation Procedures 1, 2 and 3 are given in Figures 3.8, Figure 3.9 and Figure 3.10, respectively. Also, auxiliary variables used in the algorithms are given in Table 3.2. ARENA model of the simulation model is given in Appendix A1.

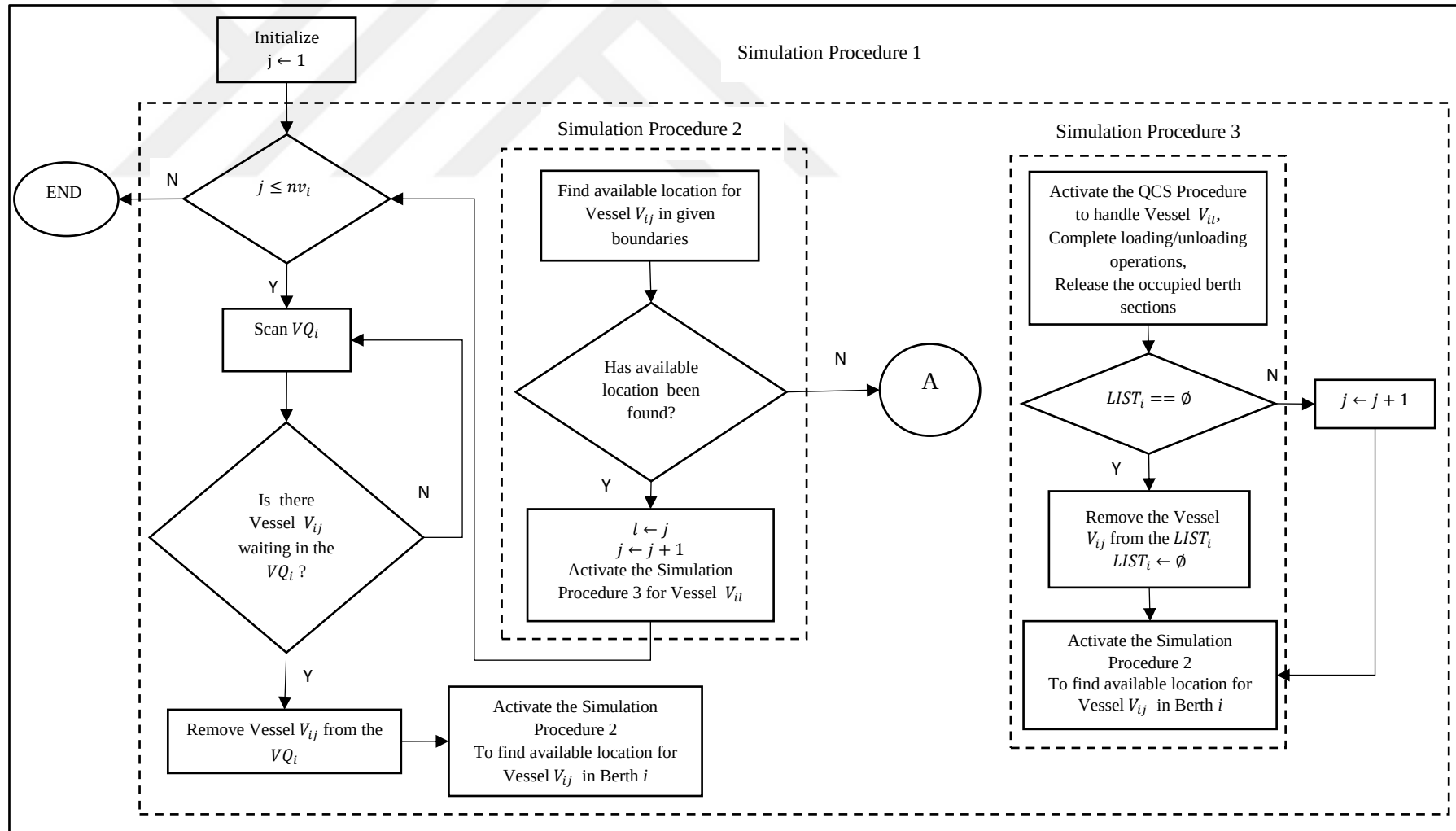


Figure 3.6 General simulation procedure

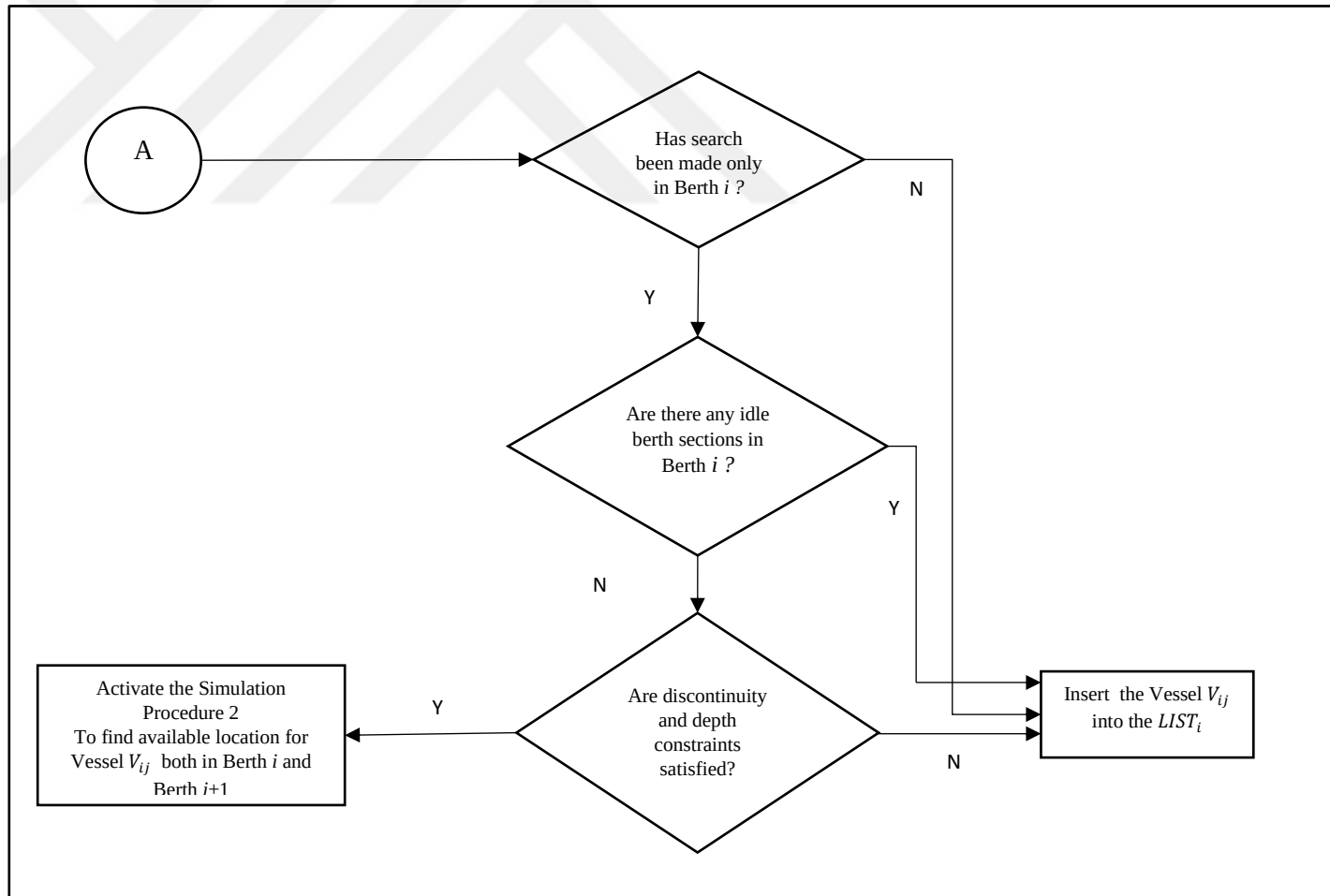


Figure 3.7 Part of simulation procedure 1

```

1. Initialize  $j \leftarrow 1$ 
2. While  $j \leq nv_i$ 
3.    $k \leftarrow S_i$ 
4.   While  $S_i \leq k + length_{V_{ij}} - 1 \leq E_i$ 
5.     Run the Sim. Procedure 2 for Vessel  $V_{ij}$ 
6.   End While
7.   If  $A == 0$ 
8.     Insert the Vessel  $V_{ij}$  into the  $LIST_i$ 
9.   Else
10.    If  $VS_{V_{ij}} == 0$ 
11.      If  $D_i == 1$ 
12.        Insert the Vessel  $V_{ij}$  into the  $LIST_i$ 
13.      Else
14.        If  $draft_{V_{ij}} > depth_i$ 
15.          Insert the Vessel  $V_{ij}$  into the  $LIST_i$ 
16.        Else
17.          While  $S_i \leq k + length_{V_{ij}} - 1 \leq E_{i+1}$ 
18.            Run the Sim. Procedure 2 for Vessel  $V_{ij}$ 
19.          End While
20.          If  $VS_{V_{ij}} == 0$ 
21.            Insert the Vessel  $V_{ij}$  into the  $LIST_i$ 
22.          Else
23.            Go to Line 28
24.          End If
25.        End If
26.      End If
27.    Else
28.       $j \leftarrow j + 1$ 
29.    End If
30.  End If
31. End While

```

Figure 3.8 Simulation procedure 1

```

1. Initialize  $counter \leftarrow 1, A \leftarrow 0, NA \leftarrow 0, VS_{V_{ij}} \leftarrow 0, VE_{V_{ij}} \leftarrow 0$ 
2. While  $counter \leq length_{V_{ij}}$ 
3.   If  $O_k == 0$ 
4.      $A \leftarrow A + 1$ 
5.   Else
6.      $NA \leftarrow k$ 
7.   End If
8.    $counter \leftarrow counter + 1$ 
9.    $k \leftarrow k + 1$ 
10. End While
11. If  $A == length_{V_{ij}}$ 
12.    $VS_{V_{ij}} \leftarrow k - length_{V_{ij}}$ 
13.    $VE_{V_{ij}} \leftarrow VS_{V_{ij}} + length_{V_{ij}} - 1$ 
14.   For  $m = VS_{V_{ij}}$  to  $VE_{V_{ij}}$ 
15.      $O_m \leftarrow 1$ 
16.   End For
17.    $l \leftarrow j$ 
18.    $j \leftarrow j + 1$ 
19.   Run the Sim. Procedure 3 for Vessel  $V_{il}$  and Sim. Procedure 1 from Line 2 for Vessel  $V_{ij}$ 
20. Else
21.    $k \leftarrow NA + 1$ 
22. End If

```

Figure 3.9 Simulation procedure 2

```

1. Initialize
2. Run the CFQCSA to handle the tasks of Vessel  $V_{il}$ 
3. For  $t = VS_{V_{il}}$  to  $VE_{V_{il}}$ 
4.    $O_t \leftarrow 0$ 
5. End For
6. If  $LIST_i \neq \emptyset$ 
7.   Remove Vessel  $V_{ij}$  from the  $LIST_i$ 
8.    $LIST_i \leftarrow \emptyset$ 
9.   Run the Simulation Procedure 1 from Line 2 for Vessel  $V_{ij}$ 
10. Else
11.    $j \leftarrow j + 1$ 
12.   Run the Simulation Procedure 1 from Line 2 for Vessel  $V_{ij}$ 
13. End If

```

Figure 3.10 Simulation procedure 3

Table 3.2 Auxiliary variables used in simulation procedures

Auxiliary Variables	Definition
A	Total number of available berth sections (temporary variable)
NA	The last occupied berth section (temporary variable)
O_k	The status of the berth section k , if it is occupied, $O_k = 1$, otherwise $O_k = 0$
D_i	Discontinuity variable, if there exists a discontinuity between Berth i and Berth $i+1$, $D_i = 1$, otherwise $D_i = 0$
VQ_i	The queue for the vessels waiting for Berth i
$LIST_i$	List of vessels which could not be located at Berth i .

3.4 Development of Simulation Optimization-based Solution Approach

Simulation is a decision-support tool which allows decision makers to model complex systems with a minimum number of assumptions. Moreover, simulation evaluates the stochastic behavior of the system and ensures flexibility for representing real life applications.

Since simulation is not a standalone optimization technique, it needs to be integrated with an optimization method in order to optimize the systems which exhibit stochastic behavior. When this is the case, simulation model takes input parameters from the optimization method, evaluates the system for this set of inputs and gives the output back to the optimization method. This procedure is called as simulation optimization. Heuristics, metaheuristics, gradient based search methods, stochastic approximation methods, sample path optimization are some of the most commonly used optimization methods in simulation optimization literature. Tekin & Sabuncuoglu (2004) presents a comprehensive survey on techniques for simulation optimization.

This dissertation proposes a simulation optimization-based solution approach for the integrated BAP and QCSP with multi-quay hybrid berth layout, dynamic arrivals of vessels, stochastic handling times and non-crossing constraints of QCs. This solution approach integrates the SA based search procedure with the simulation model of a real-world inspired container terminal. The SA algorithm was developed by Kirkpatrick, Gelatt & Vecchi (1983) for solving the NP-hard combinatorial and other optimization problems, in order to optimize the value of an objective function.

The concept of the method comes from the “annealing” process used in the metallurgical industry. The SA algorithm searches for a near global minimum by using the slow cooling structure of the physical annealing process. In the SA algorithm, the quality of the solution depends on the control parameters such as initial temperature, number of iterations and cooling rate. Figure 3.11 summarizes the procedure of SA algorithm.

```

1. Initialize
2. Obtain an initial solution  $S$  and set initial temperature  $T$  as  $T_0$ 
3. Evaluate  $G(S)$  where  $G(S)$  is the fitness value of solution  $S$ 
4. While terminating criterion is not met
5.     For  $i=1: R$  (where  $R$  is the number of iterations at each temperature)
6.         Pick a random neighbor  $S'$  of  $S$ 
7.         Set  $\Delta=G(S')-G(S)$ 
8.         If  $\Delta<0$  (downhill move)
9.             Set  $S=S'$ 
10.        Else (uphill move)
11.            Generate a random number  $x$ , from the interval (0-1)
12.            If  $x<\exp(-\Delta/T)$ 
13.                Set  $S=S'$ 
14.            End If
15.        End If
16.    End For
17. Set  $T=r*T$  (where  $r$  is the cooling rate)
18. End While

```

Figure 3.11 The procedure of SA algorithm

In the proposed solution approach, firstly, SA algorithm finds a candidate solution which gives vessel-to-berth assignments and service orders of vessels, and this solution is passed to the simulation model as input. The simulation optimization procedure is detailed in Figure 3.12. According to this figure, whenever the SA algorithm gives a candidate berth plan as input to the simulation model, the model is run for 10 replications using the candidate berth plan to estimate an average makespan. At the same time, the simulation model also forms a detailed QC schedule on the vessels using the candidate berth plan for each run. Afterwards, the average makespan is conveyed back to the simulation optimization procedure as a fitness value. When the simulation optimization procedure is terminated, a near optimum berth plan is obtained. Using the near optimum berth plan, the simulation model must be replicated to cover the variabilities in loading/unloading times. Even the simulation model is run

for the same berth plan, different solutions with different crane schedules may be obtained at each run due to the following circumstances caused by randomness in loading/unloading times: 1) QCs may be assigned to different tasks on the same vessels. 2) QCs may be assigned to tasks on different vessels. 3) Vessels may be assigned to positions exceeding the boundaries of their own berths. Therefore, a further step is required to decide which of these different solutions will be presented to the decision-maker. In this step, firstly, the obtained near optimum berth plan is used as input in the simulation model, and the model is run for R replications to get R number of solutions. The aim here is to observe the same solutions among R number of solutions which take the randomness of the loading/unloading times into account. Within these R solutions, the same solutions in which the same cranes are assigned to the same tasks on the same vessels are grouped by analyzing the simulation results in MATLAB. From among these groups, the most frequently observed solution is selected and the solution is presented to the decision-maker. At this point, it is especially important to determine the value of R since it directly affects the number of repetitions for each solution. The size of the group that contains the most frequently observed solution should be at least 30, which is sufficient to make a statistically meaningful evaluation for the performance measure (makespan). So, R is determined by trial and error method to guarantee this size. To determine the suitable R value, several experiments are conducted by replicating the simulation model for ascending R values. Hence, the minimum R value which forms the group of the frequently observed solution with a size greater than or equal to 30 is selected.

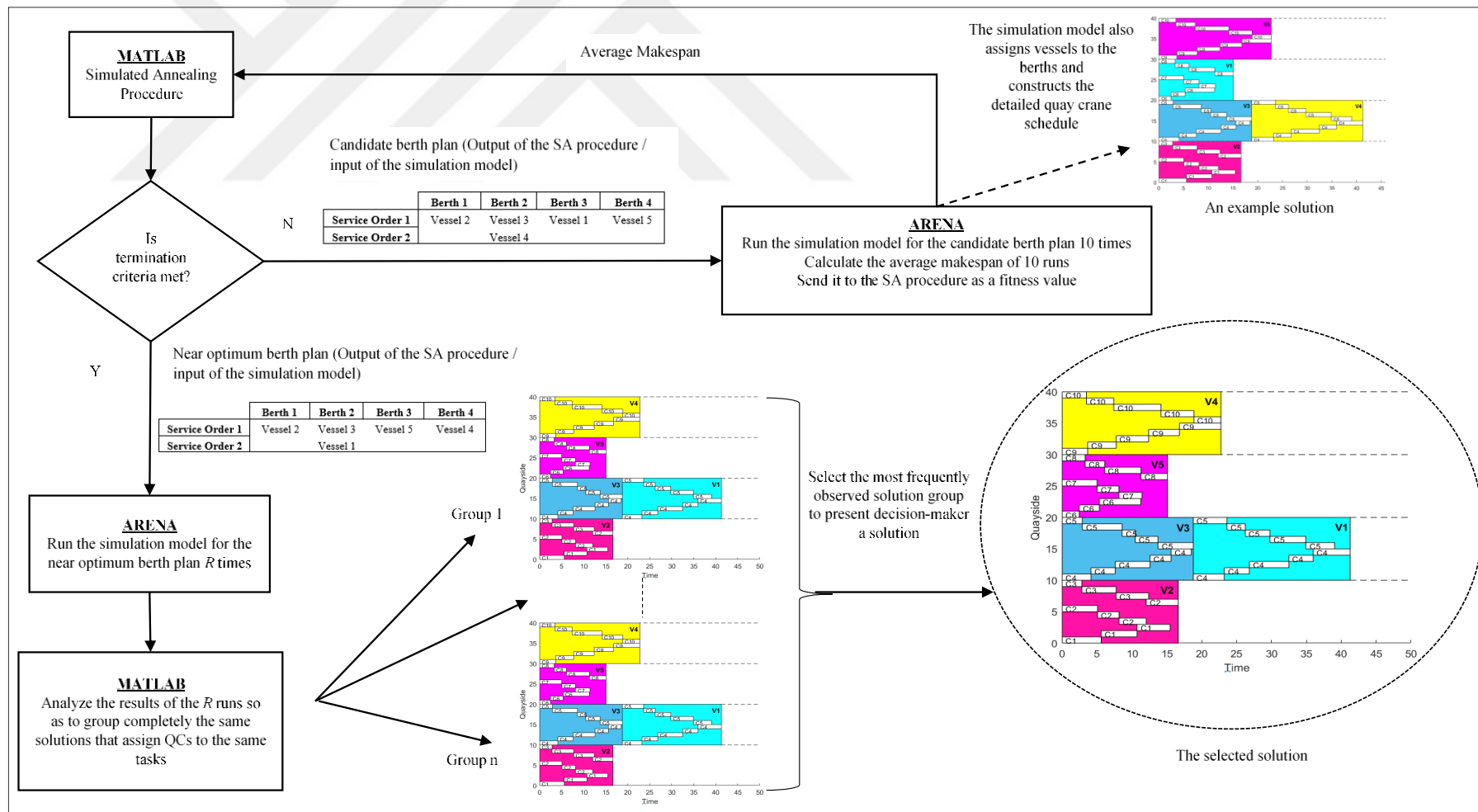


Figure 3.12 Flow chart of the proposed solution approach

In the solution representation of the proposed SA algorithm, a string structure is used in which the vessels are assigned to the berths and their service orders are shown. For example, Table 3.3 shows the solution string for the BAP including four berths and six vessels. In this representation, the first six digits belong to the first berth; the second six digits belong to the second berth and so on. The length of the string is found by multiplying the number of vessels and the number of berths. Each element in each part corresponds to a vessel and a non-zero digit indicates that the corresponding vessel is assigned to the related berth with the corresponding service order. The berth allocation plan of this representation is given in Table 3.4.

Table 3.3 The sample solution string

	Service Orders																							
Solution String	0	1	0	2	0	0	2	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6

Table 3.4 Berth allocation plan for the sample solution string

	Berth 1	Berth 2	Berth 3	Berth 4
Service Order 1	Vessel 2	Vessel 3	Vessel 6	Vessel 5
Service Order 2	Vessel 4	Vessel 1		

3.4.1 Initial Solution Generation

The initial solution is obtained by using a solution string and a dummy string which has the same length of solution string. The values of the digits are initialized to zero in solution string whereas the values of the digits in dummy string is derived from a uniform distribution with the minimum and maximum parameters of 0 and 100, respectively. The procedure for generating a feasible initial solution is as follows; the digit with the maximum value in dummy string is selected and according to the position of the selected digit, the corresponding vessel and berth numbers are identified. These numbers indicate possible vessel-to-berth assignment and this possible assignment should be checked whether it satisfies the depth and length of berth constraint. If this assignment is feasible, the remaining digits corresponding to

assigned vessel number are set to zero in dummy string, and the digit of related vessel-to-berth assignment which represents the service order of vessel in solution string is set to one if this vessel is the first vessel assigned to that berth. Each time a new vessel is assigned to a berth, the digit of related vessel-to-berth assignment is set to the service order of the most recently assigned vessel to that berth plus one. This procedure continues until all of the vessels are assigned to a berth. If there is any infeasibility, the procedure for generating a feasible initial solution is started from the beginning. The computational details of the procedure for two feasible assignments are depicted in Table 3.5 and Table 3.6. The complete list of vessel-to-berth assignments and service orders assuming that there is no infeasibility can be seen in Table 3.7.

Table 3.5 Assignment of the first vessel

	Random Numbers																							
Dummy String	95	4	58	41	78	36	25	11	3	45	7	55	69	77	12	20	36	16	8	7	41	52	6	1
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6

	Random Numbers																							
Dummy String	0	4	58	41	78	36	0	11	3	45	7	55	0	77	12	20	36	16	0	7	41	52	6	1
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6

	Service Orders																							
Solution String	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6

Table 3.6 Assignment of the second vessel

	Random Numbers																							
Dummy String	0	4	58	41	78	36	25	11	3	45	7	55	69	77	12	20	36	16	8	7	41	52	6	1
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6

	Random Numbers																							
Dummy String	0	4	58	41	0	36	0	11	3	45	0	55	0	77	12	20	0	16	0	7	41	52	0	1
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6

	Service Orders																							
Solution String	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6

Table 3.7 Final assignment of random initial solution

	Service Orders																							
Solution String	1	0	3	0	2	0	0	0	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	0
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	3	4	4	4	4	4
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6

3.4.2 Neighborhood Solution Generation

In the proposed SA based search procedure, two different neighborhood search methods which employ swap and move operators are used. The first method depends on swapping two different vessels. In other words, two different vessels are randomly selected in solution string and their positions are exchanged. As illustrated in Table 3.8, Vessel 5 which is serviced at the first berth in the second order and Vessel 6 which is serviced at the second berth in the first order are randomly selected. After swapping these two vessels, a candidate solution is obtained where Vessel 6 is serviced at the first berth in the second order and Vessel 5 is serviced at the second berth in the first order as it is depicted in Table 3.9.

Table 3.8 Random selection of vessels for swapping

	Service Orders																							
Solution String	1	0	3	0	2	0	0	0	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	0
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6

Table 3.9 Candidate solution generated by swapping

	Service Orders																							
Solution String	1	0	3	0	0	2	0	0	0	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6

The second method depends on moving a randomly selected vessel from one berth to another randomly selected berth. Also, the new service order of the moved vessel at the new berth is determined randomly. After the move operation, service orders of vessels are revised for both of the berths. As an example, in the following Table 3.10 and Table 3.11, Vessel 6 is selected and moved to Berth 1 with Service Order 2. Since

Vessel 6 leaves Berth 2, the corresponding digit of service order for this vessel in solution string is set to zero. Additionally, since Vessel 6 is moved to Berth 1 with Service Order 2, the service orders of Vessel 3 and Vessel 5 in this berth is updated as 4 and 3, respectively, in solution string.

Table 3.10 Random selection of vessel for moving

	<i>Service Orders</i>																			
Solution String	1	0	3	0	2	0	0	0	0	2	0	1	0	1	0	0	0	0	0	0
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2

Table 3.11 Candidate solution generated by moving

	<i>Service Orders</i>																			
Solution String	1	0	4	0	3	2	0	0	0	1	0	0	0	1	0	0	0	0	0	0
Berth Number	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4
Vessel Number	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2

In appendix A2, MATLAB code of this procedure is given with script functions which represent execution of the simulation model, initial solution generation and neighborhood solution generation.

3.4.3 Determination of SA Parameter Values using Taguchi Design

The parameters directly affect the quality of the solutions and improve the convergence performance of the SA algorithm. Therefore, the suitable values for the SA parameters (e.g., initial temperature, cooling rate and number of iterations at each temperature) must be tuned. Generally, better solutions are obtained with a higher initial temperature, a larger cooling rate value and a larger number of iterations at each temperature. On the other hand, higher parameter values lead to longer computational time. A set of parameters that can produce good near optimal solutions in reasonable computational time should be selected. To this aim, initially, a preliminary study is carried out comprehensively to determine three levels for each factor. A series of experiments with different combinations of the parameters are conducted to observe the diversification and intensification behavior of the SA algorithm in order to analyze

its effectiveness. For each experiment, the objective value of each candidate solution obtained at each iteration of the SA process is plotted. An example graph is given in Figure 3.13. This figure shows the run of an effective SA algorithm with suitable parameter levels. As expected, the SA algorithm accepts a worse solution not to be trapped in a local minimum as well as it uses the best solution for intensification. Accordingly, parameter levels which are shown in Table 3.12 are determined considering the effectiveness of the SA algorithm.

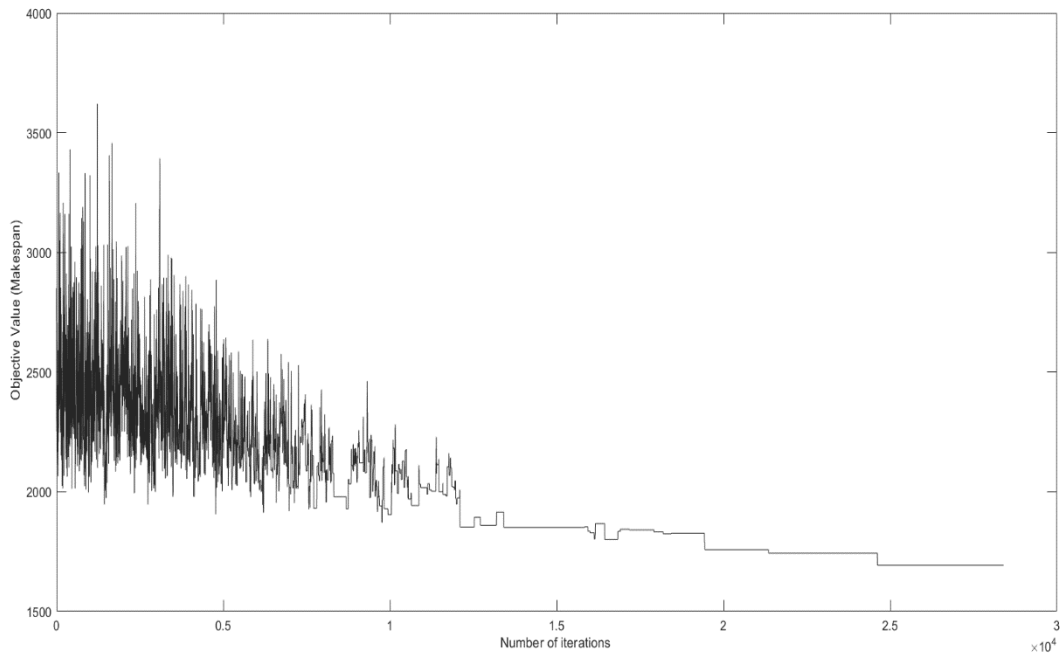


Figure 3.13 Objective value of the candidate solution at each iteration

Table 3.12 Factor levels

Factors	Factors with levels		
	Level 1	Level 2	Level 3
Initial temperature	200	250	300
Cooling rate	0.90	0.95	0.99
Number of iterations	10	30	50

After determination of the levels of the parameters, an experimental design analysis is applied to determine the suitable values of three SA parameters. According to Table 3.12, a full factorial design with 5 replications for each experiment requires 135 experiments. In order to avoid the high number of experiments required by a full factorial design, Taguchi experimental design is employed in this study. Specifically,

Taguchi's L9 orthogonal array is used since we have three factors with three levels. The experimental design scheme presented in Table 3.13 is obtained by entering the numerical values of factor levels into L9 orthogonal array.

Table 3.13 L9 orthogonal array based on experimental design employed in this study

		Experiment number								
		1	2	3	4	5	6	7	8	9
Factors	Initial temperature	200	200	200	250	250	250	300	300	300
	Cooling rate	0.90	0.95	0.99	0.90	0.95	0.99	0.90	0.95	0.99
	Number of iterations	10	30	50	30	50	10	50	10	30

The nine experiments in Table 3.13 are conducted for 5 times and results are analyzed in MINITAB 15.1 and main effect plots of the mean objective values are obtained as in Figure 3.14.

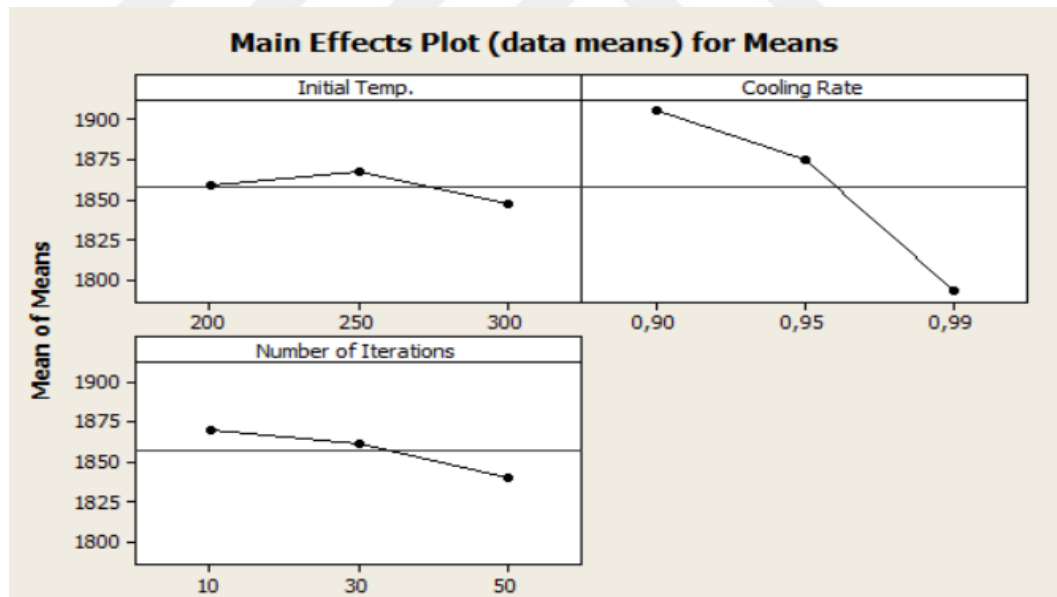


Figure 3.14 Main effect plots for mean objective values (makespan)

According to graphical representation of the mean objective values in Figure 3.14, the cooling rate impacts the objective value most. High values of the initial temperature and the number of iterations are also preferred since they give the smallest objective

value. Consequently, initial temperature, number of iterations, cooling rate are selected as 300, 50 and 0.99, respectively. The final temperature is determined as one.

3.5 Validation of the Proposed Solution Approach

To validate the proposed solution approach, the work of Lee & Wang (2010) from the related literature is used. Lee & Wang (2010) suggest a MIP-based solution approach to solve the integrated BAP and QCSP for the deterministic case to minimize the makespan. Their solution approach uses two MIP models: The first MIP model is developed for the discrete, dynamic BAP. The container handling times of vessels used in this model is obtained from the solution of the second MIP model which is proposed for the QCSP with non-crossing constraint.

Their assumptions for solving the BAP are as follows;

- Only one vessel is allowed to be handled on each berth.
- Technical and physical constraints such as vessel length, berth length, water depth and vessel draft are not considered.
- Quay crane schedule on each vessel determines the handling time of the vessel.
- Vessel arrivals occur dynamically in the planning horizon and vessels are not allowed to be handled before their arrival times

The following parameters and decision variables are used to formulate the first part of the MIP model:

Parameters:

Q : The number of berths

S : The number of vessels

q : Berth number ($1 \leq q \leq Q$)

s : Vessel number ($1 \leq s \leq S$)

a_s : The arrival time of vessel s

H_{sq} : The handling time of vessel s at berth q ($1 \leq s \leq S, 1 \leq q \leq Q$)

M : Sufficiently large positive constant number

Decision Variables:

$$X_{sq} = \begin{cases} 1, & \text{if vessel } s \text{ is assigned to berth } q \\ 0, & \text{otherwise} \end{cases}$$

y_s : The berthing time of vessel s

$$Z_{ss'} = \begin{cases} 1, & \text{if vessel } s \text{ finishes no later than vessel } s' \text{ starts handling} \\ 0, & \text{otherwise } (1 \leq s, s' \leq S, s \neq s') \end{cases}$$

C_s : The completion time of vessel s

The first part of the MIP model formulated by Lee and Wang (2010) as follows:

$$\begin{aligned} & \text{Min max } C_s & (1) \\ \text{Subject to:} & \end{aligned}$$

$$\sum_{q=1}^Q X_{sq} = 1, \forall s \in S \quad (2)$$

$$C_s = y_s + \sum_{q=1}^Q H_{sq} * X_{sq}, \forall s \in S \quad (3)$$

$$C_s - y_{s'} + Z_{ss'} * M > 0, \forall s, s' \in S, s \neq s' \quad (4)$$

$$C_s - y_{s'} - (1 - Z_{ss'}) * M \leq 0, \forall s, s' \in S, s \neq s' \quad (5)$$

$$M[(X_{sq} - 1) + (X_{s'q} - 1)] \leq Z_{ss'} + Z_{s's} - 1, \forall s, s' \in S, s \neq s', \forall q \in Q \quad (6)$$

$$y_s \geq a_s, \forall s \in S \quad (7)$$

$$X_{sq}, Z_{ss'} \in \{0, 1\}, \forall s, s' \in S, s \neq s', \forall q \in Q \quad (8)$$

Equation (3.1) represents the objective function of the first model that aims to minimize the makespan of handling all vessels while assigning vessels to the berths and sequencing them. Constraints (3.2) ensures that every single berth must be allocated by only one vessel. Constraints (3.3) define the decision variable C_s which is the completion time of vessel s . Constraints (3.4) and (3.5) represents the properties of decision variables $Z_{ss'}$. If handling operation of vessel s finishes no later than vessel s' starts ($C_s \leq y_{s'}$), $Z_{ss'}$ set to 1 otherwise 0. Constraints (3.6) ensure that two vessels

cannot be handled simultaneously if they are assigned to the same berth. Constraints (3.7) ensure that every vessel must be berthed at the time after it arrives. Constraints (3.8) are binary constraints.

The second model calculates the optimum handling times of the vessels on each berth under the following assumptions;

- The number of operating cranes on each berth is fixed. Namely, a certain number of cranes is dedicated to each berth.
- Since QCs operate on the same rail track, they cannot cross over each other.
- All of the incoming vessels are longitudinally divided into bays with respect to their lengths.
- Ship bays and QCs are indexed in increasing order.
- The handling task on each ship bay is allowed to be completed by only one crane.
- The travel time of a crane between two ship bays is ignored when compared to the handling times of ship bays.

The following parameters and decision variables are used to formulate the second part of the MIP model:

Parameters:

Q : The number of berths

S : The number of vessels

q : Berth number ($1 \leq q \leq Q$)

s : Vessel number ($1 \leq s \leq S$)

K_q : The number of quay cranes at berth q ($1 \leq q \leq Q$)

B_s : The number of ship-bays on vessel s ($1 \leq s \leq S$)

k_q, k'_q : Quay crane number at berth number q ($1 \leq k_q, k'_q \leq K_q$)

b_s, b'_s : Ship-bay number of vessel s ($1 \leq b_s, b'_s \leq B_s$)

P_{b_s} : The handling time of ship-bay b_s by a quay crane ($1 \leq b_s \leq B_s$)

Decision Variables:

$$X_{b_s k_q} = \begin{cases} 1, & \text{if ship - bay } b_s \text{ is handled by quay crane } k_q \\ 0, & \text{otherwise} \end{cases}$$

$$Y_{b_s b'_s q} = \begin{cases} 1, & \text{if ship - bay } b_s \text{ handled at berth } q \text{ finishes no later than ship - bay } b'_s \\ 0, & \text{otherwise} \end{cases}$$

$C_{b_s q}$: The completion time of ship-bay b_s handled at berth q

Lee and Wang (2010) formulate the second part of the MIP model as follows:

$$\text{Min } H_{sq} \quad (9)$$

Subject to:

$$H_{sq} \geq C_{b_s q}, \forall s \in S, \forall b_s \in B_s, \forall q \in Q \quad (10)$$

$$C_{b_s q} \geq P_{b_s}, \forall s \in S, \forall b_s \in B_s, \forall q \in Q \quad (11)$$

$$\sum_{k_q=1}^{K_q} X_{b_s k_q}, \forall s \in S, \forall b_s \in B_s, \forall q \in Q \quad (12)$$

$$C_{b_s q} - (C_{b'_s q} - P_{b'_s}) + Y_{b_s b'_s q} * M > 0 \\ \forall s \in S, \forall b_s \neq b'_s \in B_s, \forall q \in Q \quad (13)$$

$$C_{b_s q} - (C_{b'_s q} - P_{b'_s}) - (1 - Y_{b_s b'_s q}) * M \leq 0 \\ \forall s \in S, \forall b_s \neq b'_s \in B_s, \forall q \in Q \quad (14)$$

$$M * (Y_{b_s b'_s q} - Y_{b'_s b_s q}) \geq \sum_{k_q=1}^{K_q} k_q * X_{b_s k_q} - \sum_{k'_q=1}^{K_q} k'_q * X_{b'_s k'_q} + 1 \\ \forall s \in S, \forall b_s \neq b'_s \in B_s, \forall q \in Q \quad (15)$$

$$X_{b_s k_q}, Y_{b_s b'_s q} \in \{0,1\}, \forall s \in S, \forall b_s \neq b'_s \in B_s, \forall q \in Q \quad (16)$$

Equation (3.9) represents the objective function which minimizes the handling time of vessel s at berth q . Constraints (3.10) indicate that the handling time of vessel s at berth q must be equal or greater than the latest completion time of ship-bay b_s of vessel s at berth q . Constraints (3.11) ensure that completion time of ship-bay b_s must be

equal or greater than handling time of ship-bay b_s . Constraints (3.12) ensure that every ship-bay b_s can be handled by only one quay crane at berth q . In constraints (3.13) and (3.14), properties of decision variables $Y_{b_s b'_s q}$ are defined. When ship-bay b_s handled at berth q finishes no later than the handling of ship-bay b'_s starts at berth q , $Y_{b_s b'_s q}$ set to 1, otherwise 0. Constraints (3.15) are the non-crossing constraints of the quay cranes. Constraints (3.16) are binary constraints.

In order to validate our proposed solution approach, initially, since our simulation model is general and parametric, it is easily adapted to the assumptions of Lee and Wang's MIP models for the deterministic case. Afterwards, hypothetical problem instances with different parameters are generated. These instances are solved using our solution approach and Lee and Wang's models so as to compare our solutions with respect to the optimum solutions. Since the MIP models can solve only small-sized problems in reasonable times, the number of vessels and the number of ship-bays are varying between 5 and 10. All these instances have 4 berths and 3, 2, 3, 2 dedicated QCs for Berth 1, Berth 2, Berth 3 and Berth 4, respectively. For each instance, the number of vessels, their arrival times, the length of each vessel in terms of the number of ship-bays and the deterministic handling times of each ship-bay in each vessel are determined randomly. The sizes of the instances and the computational results are given in Table 3.14 and Table 3.15, respectively. The optimum solutions are obtained by using optimization software LINGO 15. Lingo formulations of the MIP models are presented in Appendix A3. After an extensive study of parameter tuning, initial temperature, number of iterations, cooling rate and final temperature are selected as 300, 50, 0.99 and 1, respectively, for the SA algorithm.

Table 3.14 The number of vessels and the lengths of vessels in terms of ship-bays

Problem Instances	The number of vessels and ship-bays									
	1	2	3	4	5	6	7	8	9	10
1	5	5	5	5	5					
9	5	6	5	6	7					
2	10	10	10	10	10					
8	8	7	6	5	8	5				
5	6	7	10	8	5	9	10			
7	5	7	8	9	8	7	10			
10	10	9	8	7	6	5	8	7	5	
3	5	5	5	5	5	5	5	5	5	5
6	5	7	8	9	10	10	10	10	10	8
4	10	10	10	10	10	10	10	10	10	10

Table 3.15 The computational results for the problem instances

Problem Instances	(Lee and Wang's MIP models)		(Proposed approach)		Gap (%)
	Optimum results	Computational time (in seconds)	Results	Computational time (in seconds)	
1	121	2	124	3120	2.48
2	211	34680	244	4347	15.64
3	200	5008	204	4432	2.00
4	355	65310	372	4353	4.78
5	150	8066	163	4255	8.67
6	279	30927	291	4382	4.30
7	186	996	201	4135	8.60
8	137	55	150	3543	9.48
9	138	7	159	3148	15.22
10	225	764	237	4498	5.33

According to Table 3.15, the percentage of deviation from the optimum solution value (Gap %) varies between 2.00% and 15.64%. Although the same deterministic problems are solved by both approaches, this deviation can be attributed to the following reasons:

The MIP model finds the solution by considering the deterministic loading/unloading times. Since the loading/unloading times are known with certainty, completion times of the tasks for each QC can be calculated. In the QC assignment phase, the MIP model takes into account these completion times and a crane schedule that prevents potential conflicts in advance can be obtained optimally. Moreover, whenever a QC schedule is made for a new incoming vessel, the MIP model ignores the final positions where the QCs have been located after completing the handling

process on the previously assigned vessel, and assigns QCs to the tasks as if the QCs are located along the quayside as in the beginning. Thus, the model only uses the completion time information of the tasks to avoid any QC conflict and ignores the initial positional constraints, which result in more favorable makespan values.

On the other hand, the CFQCS algorithm that we developed is designated for the stochastic loading/unloading times. The QC assignment decision is triggered by berthing of a new vessel along the quayside reactively. Since the simulation model can determine the completion time of the task just after finishing the handling of the containers physically as in reality, they cannot be calculated beforehand. The algorithm uses the position and status information of the QCs at that time to create a conflict-free schedule dynamically and it checks some conditions related to destination/final locations of preceding and succeeding QCs (i.e., FL_{n-1} and FL_{n+1}) dynamically (See Figure 3.5). Since the CFQCS takes into account the initial and final positions of the QCs continuously during the planning period, QC assignment decision depends on the final position where the QCs have been located after the completion of their handling tasks. This reactive and myopic scheme of our CFQCS algorithm, which is developed for real life stochastic cases, limits the decision process. Therefore, it is inevitable to get worse objective function values than those of the mathematical model even if our approach is executed with deterministic loading/unloading times. So, 7.65% average gap is reasonable to validate the proposed solution approach under these circumstances.

However, considering the computational times of the problem instances which are given in Table 3.15, as the number and length of the vessels increase, the computational time of the MIP model noticeably increases, as expected. Since the integrated BAP and QCSP is NP-hard, as the number and length of the vessels increase, the problem cannot be solved in a polynomial time. Therefore, it is necessary to apply a meta-heuristic method to solve the problem. The computational time of our proposed approach is approximately the same for each problem instance.

Although the results of the problem instances seem like the proposed solution approach performs worse than the MIP with respect to the objective value, it should be noted that the purpose of making such a comparison is to validate our proposed approach by demonstrating how it converges to the optimum solution. Besides, the proposed solution approach has many powerful features which the simulation model can handle such as hybrid berth layout, discontinuity, coverage area, whereas the MIP model cannot accommodate these features.

As an example, the input data of Problem Instance 5 are shown in Table 3.16. In Figure 3.15, it can be seen how the proposed solution approach converges to the optimum solution. Also, berth space-time diagram for the results of both of the approaches are given in Figure 3.16 and Figure 3.17.

These time-space diagrams are drawn using the results obtained from both mathematical model and proposed solution approach. MATLAB code executed for drawing a berth space-time diagram is given in Appendix A4.

Table 3.16 The input data for problem instance 5

Vessel No	Vessel Length	Arrival Time	The Handling Times of Each Ship-Bay in Each Vessel									
			1	2	3	4	5	6	7	8	9	10
1	6	5	5	26	10	18	6	19				
2	7	6	8	17	21	9	32	33	22			
3	10	20	23	26	32	37	15	20	22	18	23	30
4	8	15	20	21	44	33	27	33	41	14		
5	5	36	33	6	22	21	20					
6	9	44	19	17	9	25	40	17	27	33	42	
7	10	0	5	7	18	22	33	41	12	38	21	19

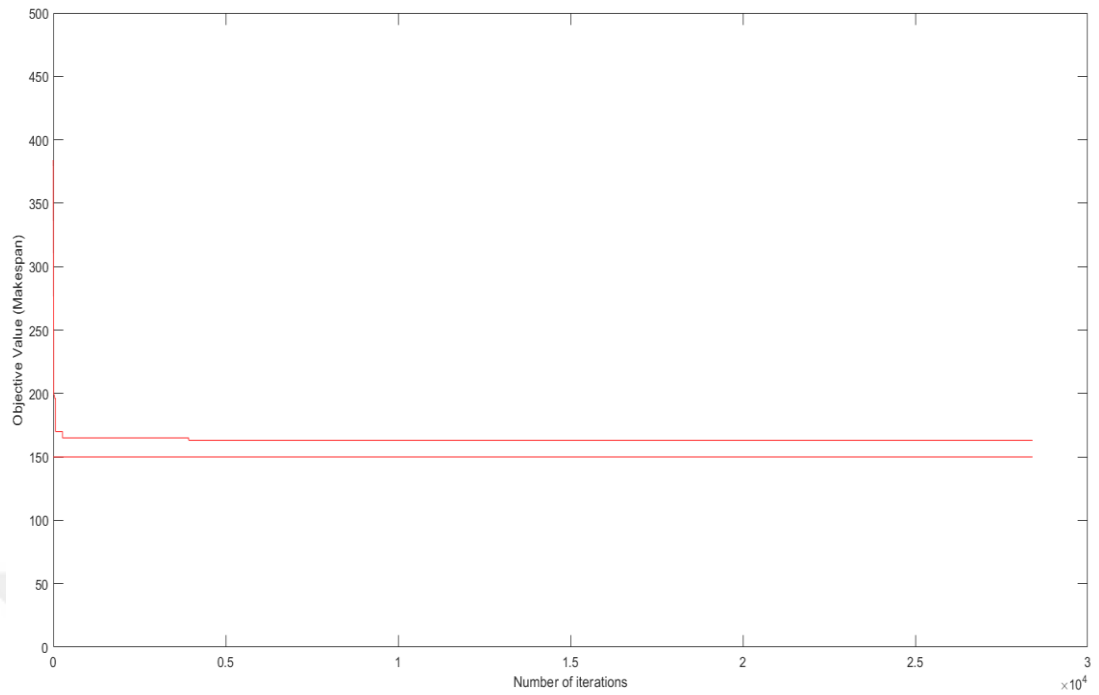


Figure 3.15 Convergence graph of the proposed solution approach

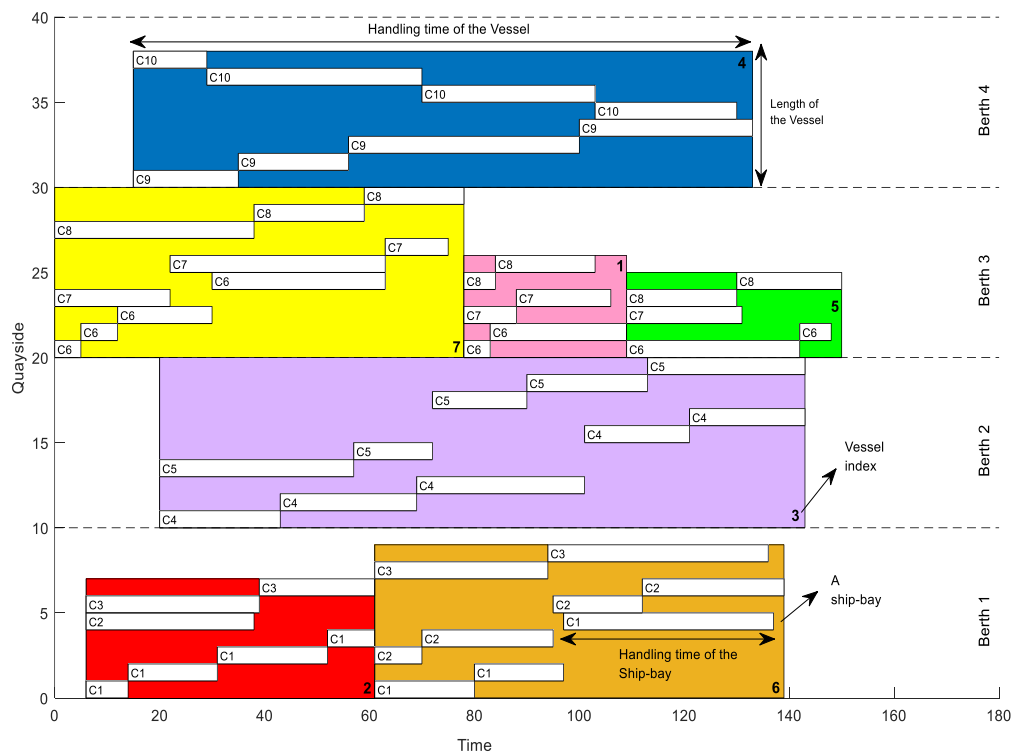


Figure 3.16 Berth space-time diagram for the optimum solution

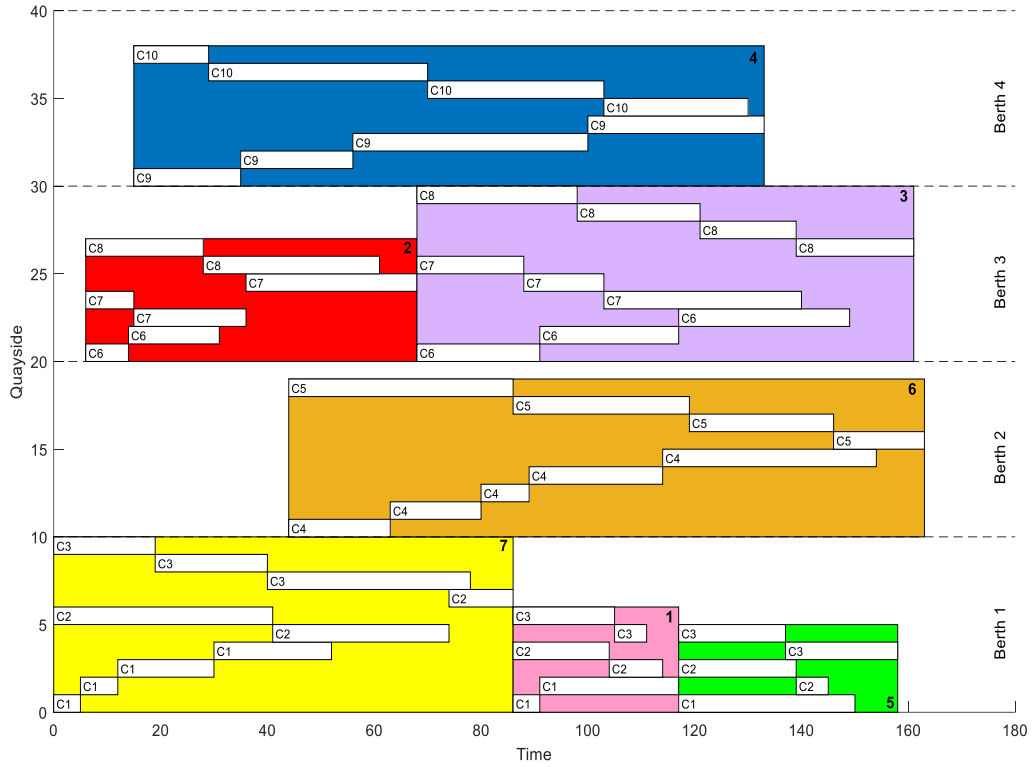


Figure 3.17 Berth space-time diagram for the solution of the proposed approach

In Figures 3.16 and 3.17, colored rectangles represent the vessels in the planning period and the same vessels are painted with the same color, also vessel numbers are indicated with bold numbers on the corresponding rectangle. Lower-left vertex of each rectangle indicates the berthing time and berthing position of a vessel. The height and width of each colored rectangle corresponds to the length and handling time of a vessel respectively. Also, QCs assigned to each ship-bay on the vessels are shown in the figure as white rectangles in colored ones. Lower-left vertex of white rectangles indicates the starting time of handling operations and corresponding berth section of the related ship-bay along the quayside. The height of each rectangle denotes a single ship-bay and the width denotes the handling time of corresponding ship-bay. In a feasible solution, these rectangles do not overlap.

In Appendix A5, for all problem instances, input data, convergence graphs, berth space-time diagrams for the optimum solutions and berth space-time diagrams for the solutions of the proposed approach are presented respectively.

CHAPTER FOUR

IMPLEMENTATION OF THE PROPOSED SOLUTION APPROACH

In order to show that the proposed solution approach can be used effectively to solve large-sized BAPs and QCSPs for the objective of minimizing makespan, three hypothetical sample problems which include most of the realistic features of a typical container terminal are randomly generated and solved using the proposed approach. The problem data for each Sample Problem is given in detail in Appendix A6. For each vessel, the number of containers that each ship-bay contains is generated from a discrete uniform distribution between 20 and 80 and given in Table A6.1, Table A6.5 and Table A6.9, respectively. The boundaries of each QC's coverage area and initial position of each QC for each Sample Problem are given in Table A6.2, Table A6.6 and Table A6.10, respectively. Berth related data which include length and depth of each berth for each Sample Problem are given in Table A6.3, Table A6.7 and Table A6.11. Vessel related data which include vessel length and draft for each Sample Problem are given in Table A6.4, Table A6.8 and Table A6.12, respectively. In order to locate the QCs homogenously along the each quay section, MATLAB code which is given in Appendix A7 is run and initial positions of the QCs are determined.

Illustrations of the sample container terminals 1, 2 and 3 which depict quay sections, berths and QCs' positions are given in Figure 4.1, Figure 4.6 and Figure 4.11, respectively. The handling time of a single container is assumed to be normally distributed with a mean of 2 minutes and a standard deviation of 0.5 minute in all problems.

These data are consistent with the maritime context and inspired of the previous studies regarding the following aspects:

- Quay discontinuities or sharp curves, as shown in Figure 3.1, naturally exist in many container terminals in reality, for examples, the Medcenter Container Terminal and the BLG Italia Terminal located at the Port of Gioia Taur (Cordeau, Gaudioso, Laporte & Moccia, 2007; Cordeau et al., 2005; Cordeau,

Laporte, Moccia & Sorrentino, 2011), the Brani Terminal at the Port of Singapore (Lee, Jin & Chen, 2012), the Port of Izmir Alsancak Container Terminal (Esmer, Yıldız & Tuna, 2013). Similarly, for example, in our Sample Problem 2, there are discontinuities between Berth 3 and Berth 4, Berth 6 and Berth 7.

- Water depth and vessel draft values are generated in accordance with the way Xu et al. (2012) expressed and generated randomly between 12 and 16. Accordingly, in our Sample Problems, these values are generated between 10 and 16.
- Lee & Wang (2010), Legato et al. (2010) and Al-Dhareri et al. (2016a) assume that vessels are longitudinally divided into bays with respect to their lengths as we do.
- Vessel lengths are generated in accordance with how Legato et al. (2010) determine. Length of the container vessels in terms of ship-bays are investigated and the small and medium sized vessels are generated. The longest vessel has 19 ship-bays in our problem.
- In our study, it is not mandatory that all ship-bays must be filled with containers. Some of the ship-bays on the vessels may be empty or not full of containers. The number of containers each ship-bay contains is generated from a discrete uniform distribution between 20 and 80, which shows similarity with Legato et al. (2010).
- Han et al. (2015) also highlight the coverage area of the QCs in their study just as we do. They express the length of the coverage areas in meters whereas we specify the boundaries of the coverage areas in terms of berth sections. In their study, they solve a real case from Ningbo Beilun Container Terminal in China which has a 1248 m length quayside. In our study, the length of a single berth section, and the length of a ship-bay are assumed to be equal. The length of a ship-bay and a single berth section is regarded as 12 meters, which is the maximum length of a container that a ship-bay can contain (40 TEU). The length of the quayside in our problem which is comprised of total 115 berth sections equals to almost 1380 meters. Considering the length of Ningbo Beilun

Container Terminal, it can be said that the length of our terminal is consistent with practice.

- Legato et al. (2010) states that handling time of a single container is 2 minutes. In this context, the handling time of a single container is assumed to be normally distributed with a mean of 2 minutes and a standard deviation of 0.5 minute

In the proposed solution approach, SA algorithm which is developed on MATLAB finds a candidate solution that gives vessel-to-berth assignments and service orders of vessels, and this solution is passed to the simulation model as input. Then, the simulation model is executed for several replications to estimate the average makespan of this berth allocation plan and passes the estimated makespan to the SA algorithm as fitness value. This process continues until termination criterion is met. In this way, the simulation optimization procedure determines a near optimum berth allocation plan and its related quay crane schedule, which minimize the average makespan.

It must be noted that the berth allocation plan obtained from MATLAB only determines vessel-to-berth assignments and service orders of vessels. However, due to the stochastic handling times and hybrid berth layout, the first berth section number occupied by a vessel, the quay crane assignments to a vessel, the starting time of handling tasks of a vessel and the completion time of all handling tasks of a vessel may vary from one simulation replication to another for the same berth allocation plan. This average makespan value may have been obtained from the makespan values of different schedules. To be able to say that two schedules are completely the same, in both schedules, vessels should be assigned to the same berthing locations and the same cranes should be assigned to the same tasks in the same order. So, in order to estimate berthing/unberthing times, berthing locations and quay crane schedules for each vessel in the near optimum berth plan, the simulation model should be run for R number of replications to find the most probable schedule scenario for the near optimum berth allocation plan.

In the simulation optimization procedure, 10 replications have been made to find out a near optimum berth allocation plan. Then, this berth allocation plan has been used in the simulation model for R replications which have been determined by the way explained in detail in Section 3.4. The same schedules obtained from R replications have been gathered, different schedule groups have been formed and the most frequently observed schedule (*i.e.*, 30 times) has been selected.

When the proposed solution approach is applied to a problem, the obtained solution which gives the estimated berthing/unberthing times with a given range, berthing locations and quay crane schedules for each vessel for the most probable scenario may be used in order to inform incoming vessels before their arrivals. It must be noted that this solution provides the decision makers with useful insights in implementation of the resulting berth and quay crane schedules in practice. Since the proposed solution approach takes into account uncertainty in loading/unloading operations, it gives more reliable results considering the variabilities with a given range in berthing/unberthing times. Moreover, the proposed approach gives solutions in a reasonable time thus, it can be applied periodically with respect to container terminal's planning routine as follows: While the incoming vessels are approaching the container terminal, they periodically report their ETA (estimated time of arrival). In addition, the information of loading/unloading tasks on the vessels is obtained from the shipping companies or the cargo agents. Based on this information, using the proposed solution approach, the container terminal generates the plan of seaside operations for the vessels in the planning period before they have arrived at the container terminal.

In the following, in order to demonstrate the effectiveness of the proposed solution approach, three different Sample Problems are solved and solution of each Sample Problem is given in detail. Moreover, the Sample Problems are solved again by proposed solution approach under the assumption that each berth section of the quayside contains one QC. In this case, the problems are no more crane-constrained and turn into Berth Allocation Problem. Since it is known that objective value of the original problem must be larger than the problem which is not crane-constrained, the result of these problems gives a lower bound for each original problem. Also, for each

Sample Problem, convergence graph, berth space-time diagram and berth plan with vessel numbers when the problems are not crane-constrained are depicted after the result of each original Sample Problem.

4.1 Representation and Solution of Sample Problem 1

The generated Sample Problem 1 has 20 incoming vessels, 4 berths spread over 2 quays and 10 QCs. The whole quayside has 40 berth sections and there is only one discontinuity between Berth 2 and Berth 3 (See Figure 4.1).

When the SA based simulation optimization procedure is applied to the problem, R is determined as 50. Eight different schedule groups have been formed and the most frequently observed schedule (*i.e.*, 35 times) has been selected.

The resulting berth space-time diagram of the selected schedule which represents the near-optimum solution is depicted in Figure 4.2. As it is depicted in Figure 4.2, each vessel is assigned to a single berth in accordance with discrete berth layout. Also, convergence graph of the selected solution is given in Figure 4.3.

In addition, for each vessel, berthing location, estimated berthing/unberthing times and QC-to-task assignments are given in Table 4.1. The information in Table 4.1 can be interpreted as follows: for example, Vessel 1 is assigned to Berth 4 starting from the 30th Berth Section of the quayside. Vessel 1 has 5 loading/unloading tasks on itself and Crane 9 is assigned to the first three of the tasks and Crane 10 is assigned to the rest of the tasks, respectively. The estimated berthing time of Vessel 1 is calculated as 676.68 by considering the minimum (661.17) and maximum (691.80) values of berthing times. In the same way, the estimated unberthing time of Vessel 1 is calculated as 937.68 by considering the minimum (917.32) and maximum (962.67) values of unberthing times. These given range in berthing/unberthing times provides the decision maker more reliable berth and quay crane schedules under uncertainty in loading/unloading operations.

Moreover, the problem is solved again assuming that the each berth section consists of one QC. With this assumption, the problem turns into BAP and the solution of this

problem provides a lower bound for the original problem. In Figure 4.4, berth space-time diagram of the problem with detailed QC schedule is given. In order to demonstrate the vessel numbers apparently, Figure 4.5 is also given. In these figures, a remarkable decrease in makespan of all vessels can be observed as expected.



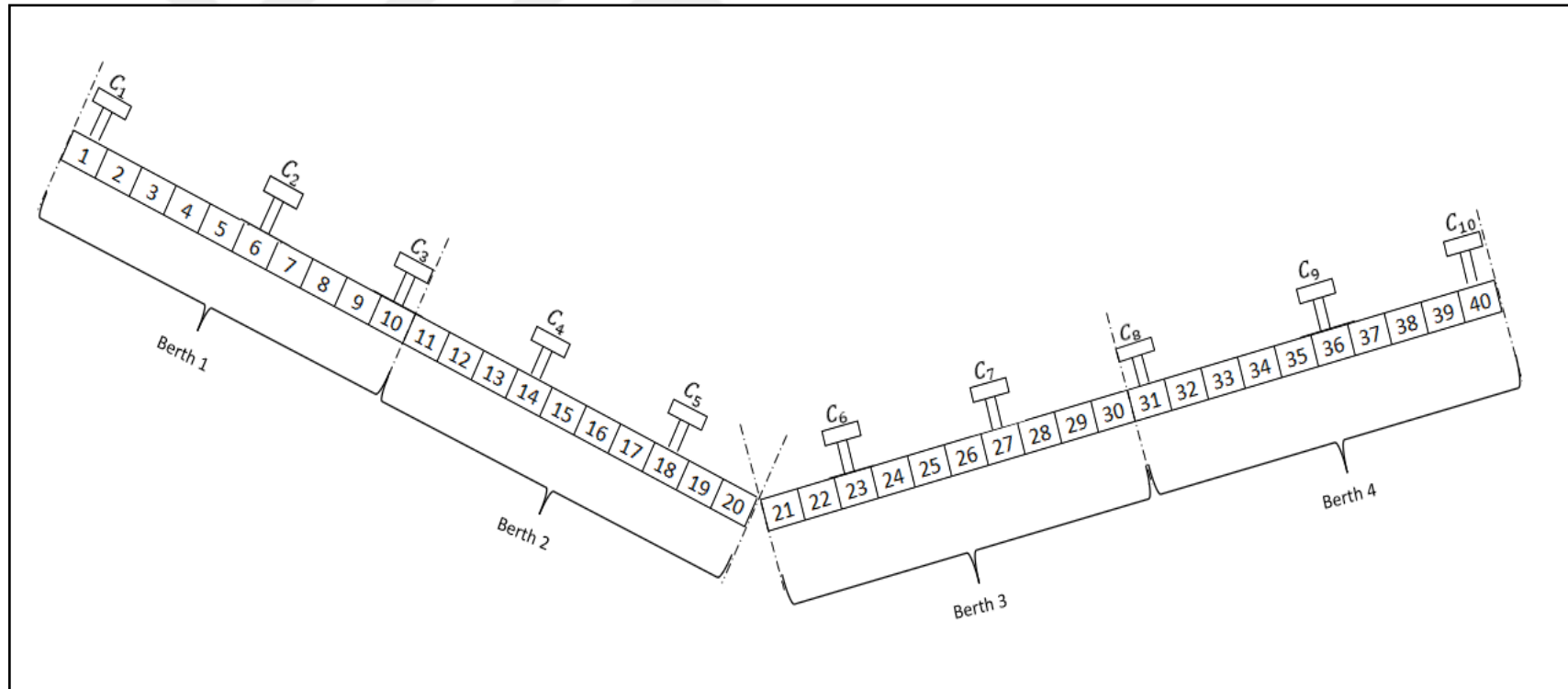


Figure 4.1 Representation of the quayside for the sample problem 1

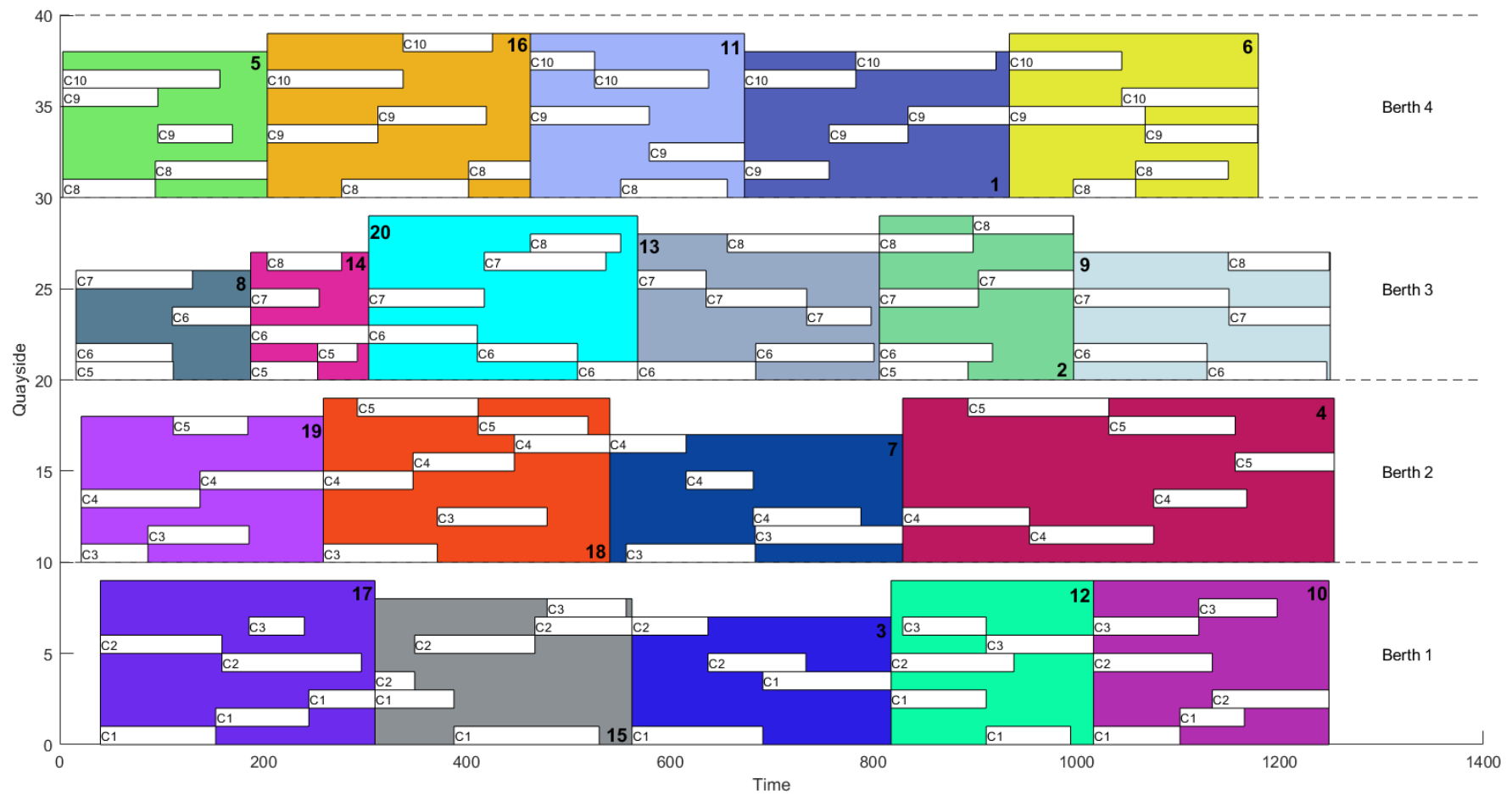


Figure 4.2 Berth space-time diagram for the solution of the proposed methodology for sample problem 1

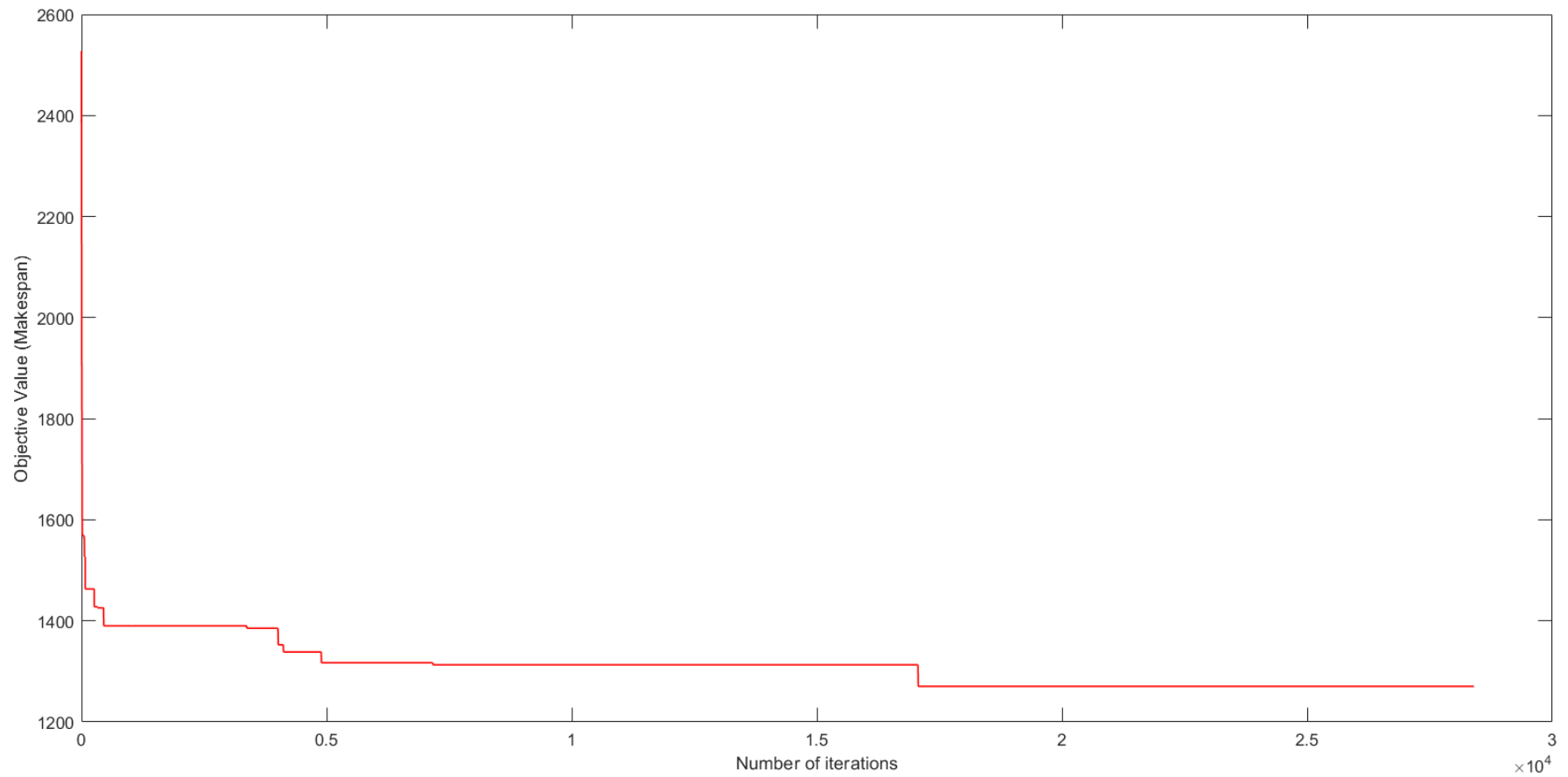


Figure 4.3 Convergence graph of the sample problem 1

Table 4.1 Berthing locations, average berthing times, departure times and crane assignment information of the vessels (sample problem 1)

Vessel no.	Berth no.	Berthing location	Min. berthing time	Estimated berthing time	Max. berthing time	The number of handling tasks of the vessels													Min. departure time	Estimated unberth. time	Max. departure time
						1	2	3	4	5	6	7	8	9	10	11	12	13			
1	4	30	661.17	676.68	691.80	C9	C9	C9	C10	C10									917.32	937.68	962.67
2	3	20	791.78	806.09	821.64	C5	C6	C7	C7	C8	C8								985.75	1002.25	1019.89
3	1	0	552.18	568.04	585.10	C1	C1	C2	C2										807.90	828.51	843.96
4	2	10	819.45	838.49	860.16	C4	C4	C4	C5	C5	C5								1225.07	1246.38	1267.51
5	4	30	3.00	3.00	3.00	C8	C8	C9	C9	C10									193.84	203.38	215.06
6	4	30	917.32	937.68	962.67	C8	C8	C9	C9	C10	C10								1167.89	1190.56	1229.84
7	2	10	539.15	550.32	560.83	C3	C3	C4	C4	C4									819.45	838.49	860.16
8	3	20	16.00	16.00	16.00	C5	C6	C6	C7										183.49	192.56	201.78
9	3	20	985.75	1002.25	1019.89	C6	C6	C7	C7	C8									1242.81	1258.43	1280.17
10	1	0	1005.39	1027.46	1048.58	C1	C1	C2	C2	C3	C3								1229.28	1255.60	1280.13
11	4	30	449.03	465.74	478.69	C8	C9	C9	C10	C10									661.17	676.68	691.80
12	1	0	807.90	828.51	843.96	C1	C1	C2	C3	C3									1005.39	1027.46	1048.58
13	3	20	549.40	570.20	585.27	C6	C6	C7	C7	C7	C8								791.78	806.09	821.64
14	3	20	183.49	192.56	201.78	C5	C5	C6	C7	C8									296.95	308.74	323.90
15	1	0	294.50	303.48	313.32	C1	C1	C2	C2	C2	C3								552.18	568.04	585.10
16	4	30	193.84	203.38	215.06	C8	C8	C9	C9	C10	C10								449.03	465.74	478.69
17	1	0	40.00	40.00	40.00	C1	C1	C1	C2	C2	C3								294.50	303.48	313.32
18	2	10	257.41	267.04	279.83	C3	C3	C4	C4	C4	C5	C5							539.15	550.32	560.83
19	2	10	21.00	21.00	21.00	C3	C3	C4	C4	C5									257.41	267.04	279.83
20	3	20	296.95	308.74	323.90	C6	C6	C6	C7	C7	C8								549.40	570.20	585.27



Figure 4.4 Berth space-time diagram for the solution of the sample problem 1 when it is not crane-constrained

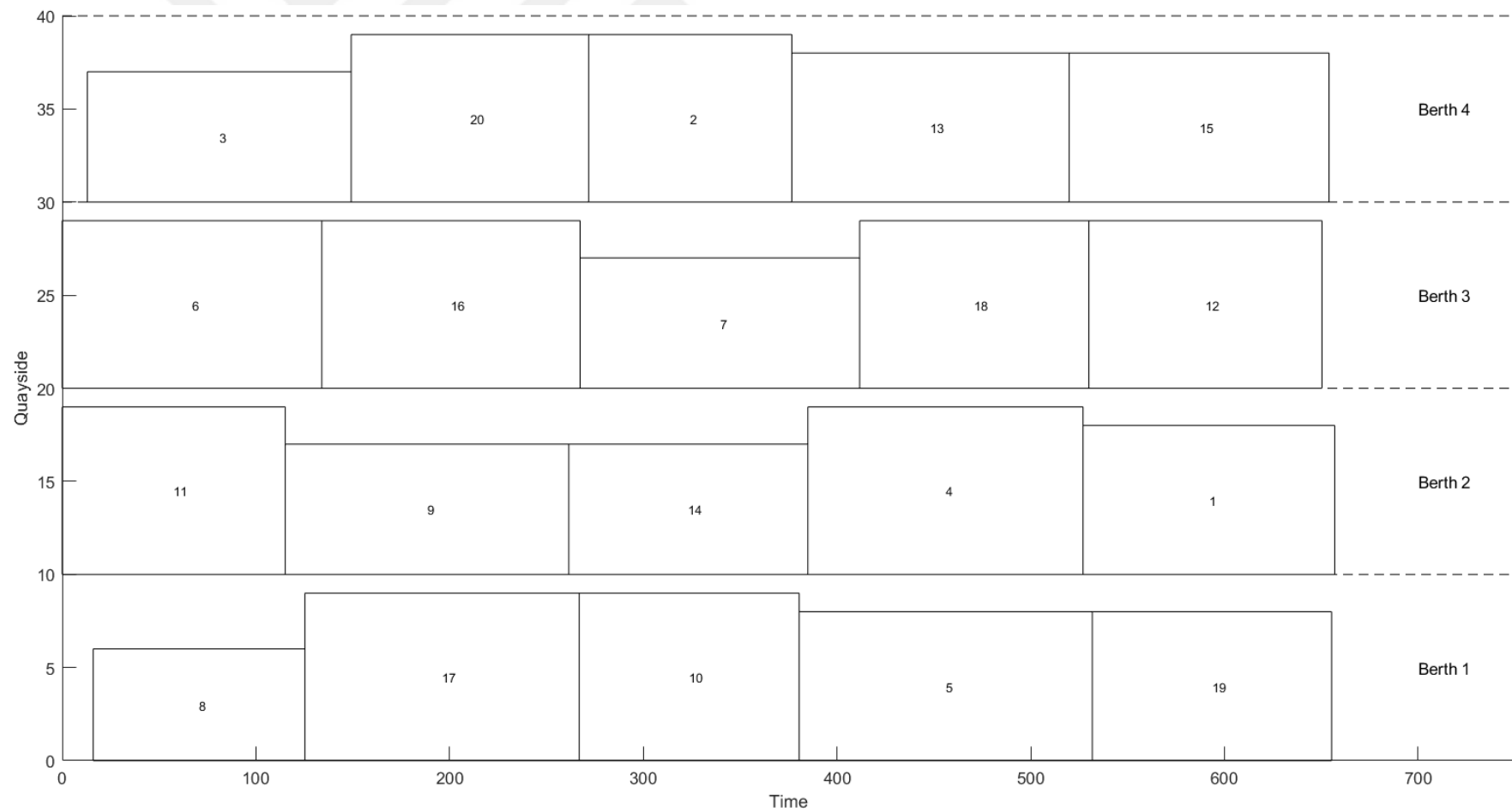


Figure 4.5 Berth space-time diagram for the solution of the sample problem 1 when it is not crane-constrained with only vessel numbers

4.2 Representation and Solution of Sample Problem 2

The generated Sample Problem 2 has 40 incoming vessels, 7 berths spread over 3 quays and 18 QCs. The whole quayside has 115 berth sections and there are two discontinuities between Berth 3 and Berth 4, Berth 6 and Berth 7, respectively (See Figure 4.6).

When the SA based simulation optimization procedure is applied to the problem, R is determined as 250. Fifty nine different schedule groups have been formed and the most frequently observed schedule (*i.e.*, 30 times) has been selected.

The resulting berth space-time diagram of the selected schedule which represents the near-optimum solution is depicted in Figure 4.7. Also, convergence graph of the selected solution is given in Figure 4.8. As it is depicted in Figure 4.7, for example vessels 22, 17, 21, 14, 31, 4 and 29 have been assigned to Berth 2 in this given service order. Since the hybrid layout is assumed, Berth 2 has been occupied by vessels 4 and 29 simultaneously. Additionally, since there is no discontinuity between Berth 2 and Berth 3. Vessel 31 has been assigned to both berths at the same time under the condition that the draft of Vessel 31 is smaller than the depth of Berth 3. It is obvious that the hybrid berth layout assumption increases the berth utilization as well as decreases the makespan.

In addition, for each vessel, berthing location, estimated berthing/unberthing times and QC-to-task assignments are given in Table 4.2. The information in Table 4.2 can be interpreted as follows: for example, Vessel 1 is assigned to Berth 4 starting from the 50th Berth Section of the quayside. Vessel 1 has 4 loading/unloading tasks on itself and Crane 9 is assigned to the first two of the tasks and Crane 10 is assigned to the rest of the tasks, respectively. The estimated berthing time of Vessel 1 is calculated as 1018.50 by considering the minimum (979.79) and maximum (1049.62) values of berthing times. In the same way, the estimated unberthing time of Vessel 1 is calculated as 1240.37 by considering the minimum (1192.82) and maximum (1274.68) values of unberthing times. These given range in berthing/unberthing times provides

the decision maker more reliable berth and quay crane schedules under uncertainty in loading/unloading operations.

Moreover, the problem is solved again assuming that the each berth section consists of one QC. With this assumption, the problem turns into BAP and the solution of this problem provides a lower bound for the original problem. In Figure 4.9, berth space-time diagram of the problem with detailed QC schedule is given. In order to demonstrate the vessel numbers apparently, Figure 4.10 is also given. In these figures, a remarkable decrease in makespan of all vessels can be observed as expected.



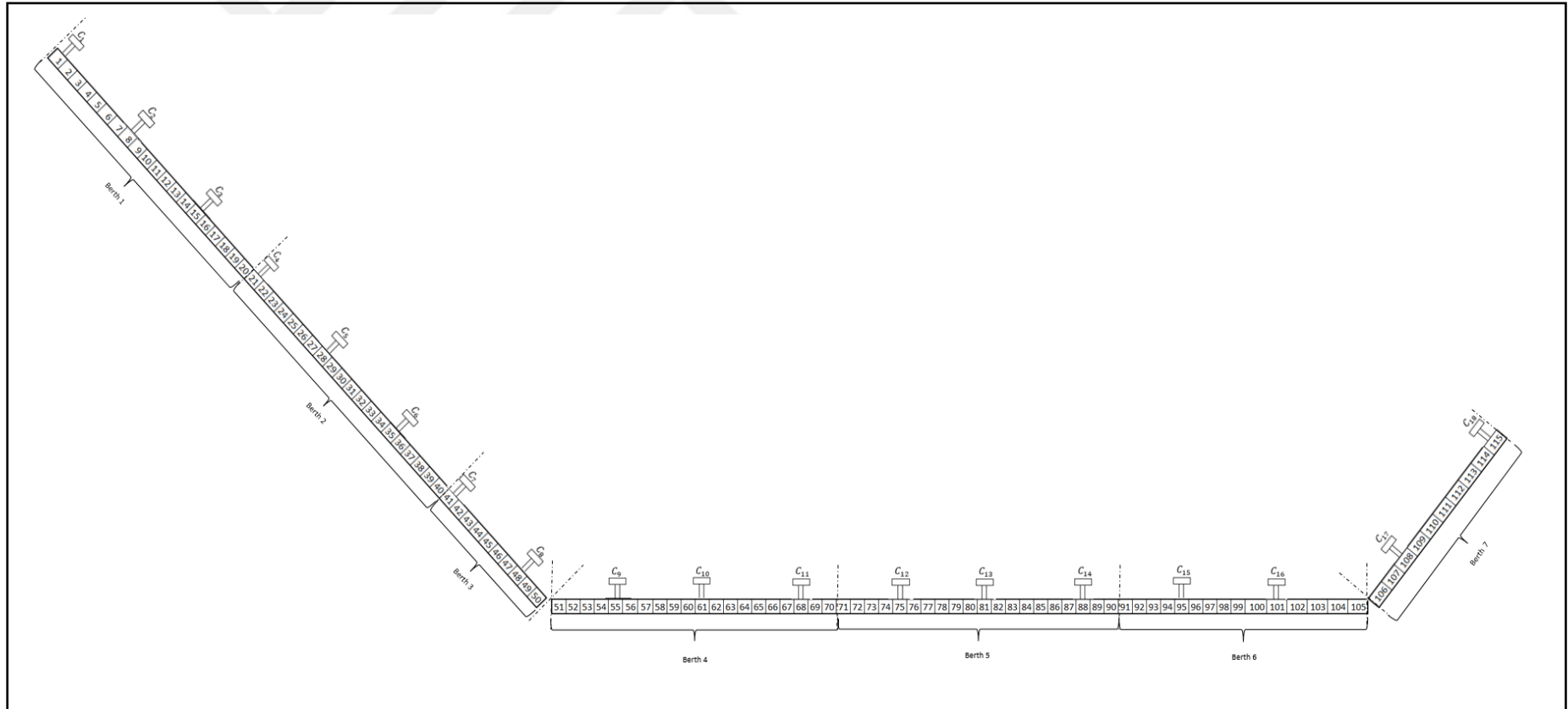


Figure 4.6 Representation of the quayside for the sample problem 2

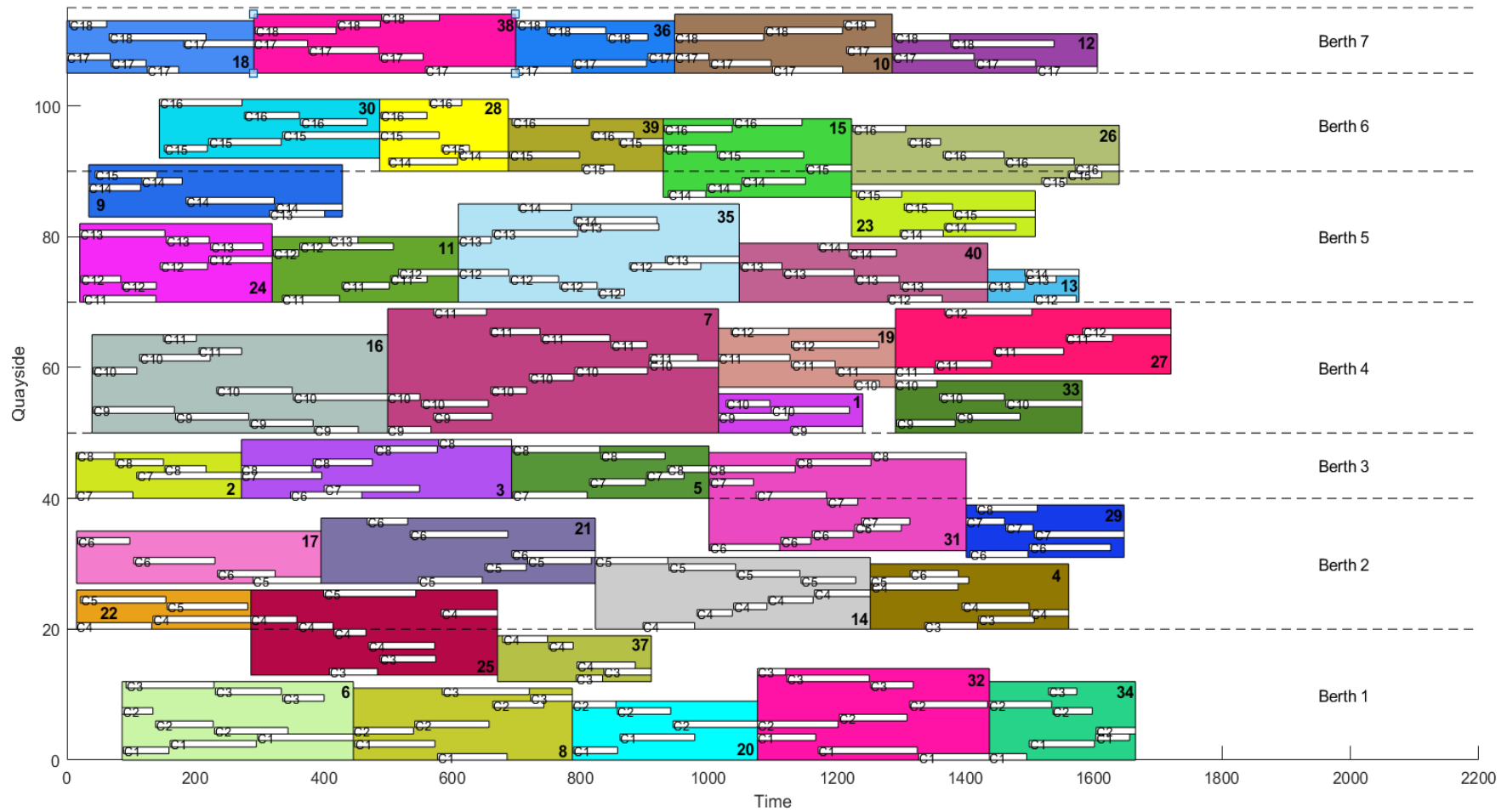


Figure 4.7 Berth space-time diagram for the solution of the proposed methodology for sample problem 2

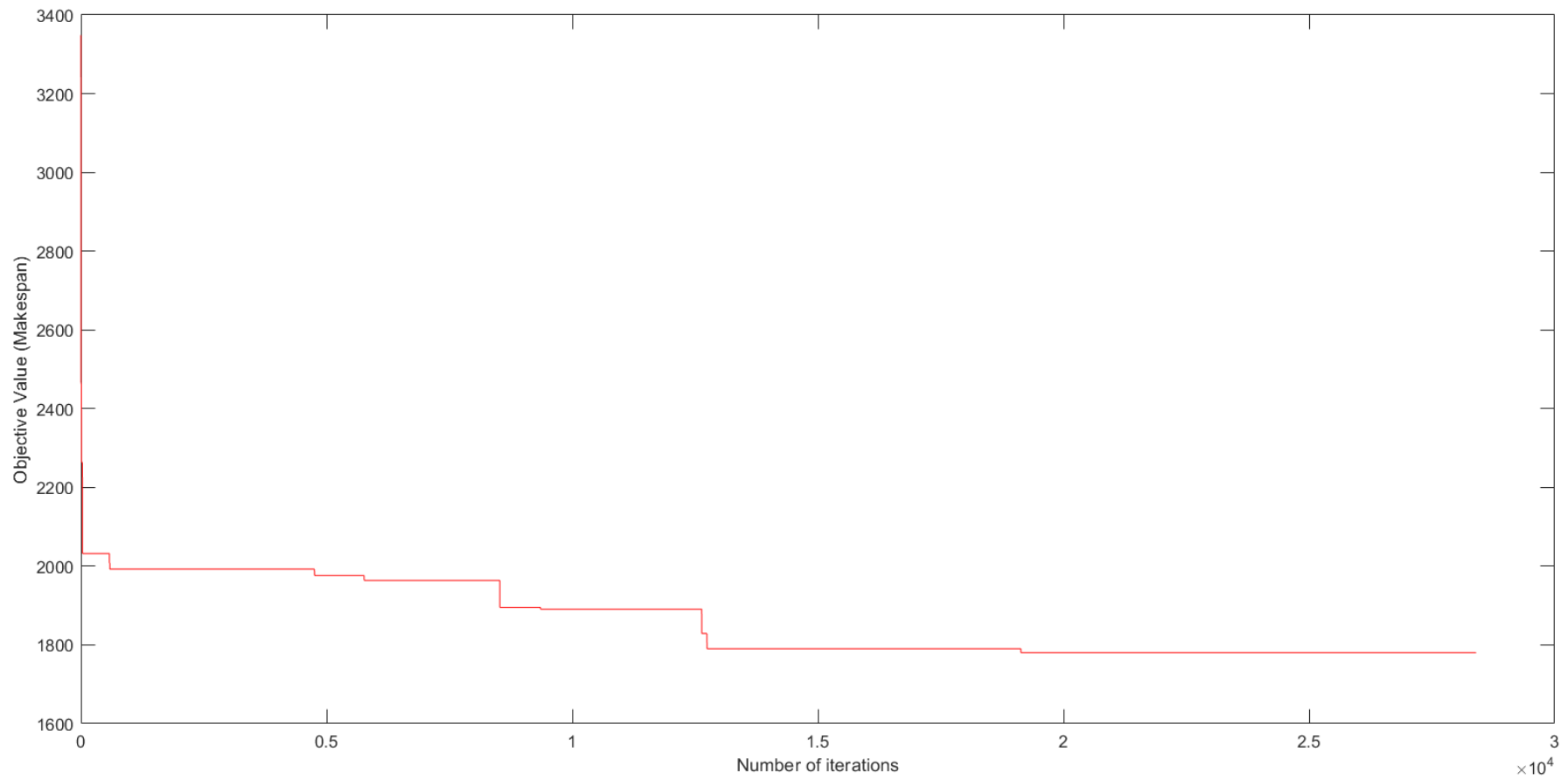


Figure 4.8 Convergence graph of the sample problem 2

Table 4.2 Berthing locations, average berthing times, departure times and crane assignment information of the vessels (sample problem 2)

Vessel no.	Berth no.	Berthing location	Min. berthing time	Estimated berthing time	Max. berthing time	The number of handling tasks of the vessels													Min. departure time	Estimated unberth. time	Max. departure time
						1	2	3	4	5	6	7	8	9	10	11	12	13			
1	4	50	979.79	1018.50	1049.62	C9	C9	C10	C10										1192.82	1240.37	1274.68
2	3	40	14.00	14.00	14.00	C7	C7	C8	C8	C8									242.33	259.96	279.36
3	3	40	242.33	259.96	279.36	C6	C7	C7	C8	C8	C8	C8							631.64	659.92	692.85
4	2	20	1231.18	1262.31	1292.11	C3	C3	C4	C4	C4	C5	C6							1531.30	1571.18	1602.33
5	3	40	631.64	659.92	692.85	C7	C7	C7	C8	C8	C8								935.14	969.70	1003.64
6	1	0	86.00	86.00	86.00	C1	C1	C1	C2	C2	C2	C3	C3	C3					428.13	449.02	470.00
7	4	50	474.90	498.85	520.11	C9	C9	C10	C10	C10	C10	C10	C10	C11	C11	C11	C11	C11	979.79	1018.50	1049.62
8	1	0	428.13	449.02	470.00	C1	C1	C2	C2	C2	C3	C3							769.13	793.39	818.31
9	5	83	34.00	34.00	34.00	C13	C14	C14	C14	C14	C15								404.18	424.04	445.68
10	7	105	902.65	935.61	966.92	C17	C17	C17	C17	C18	C18	C18							1229.50	1267.62	1309.39
11	5	70	299.30	313.61	331.83	C11	C11	C11	C12	C12	C12	C13							575.84	599.82	623.12
12	7	105	1229.50	1267.62	1309.39	C17	C17	C17	C18	C18									1542.03	1585.47	1632.62
13	5	70	1401.75	1436.23	1471.52	C12	C13	C13	C14										1550.94	1584.10	1620.53
14	2	20	807.99	832.15	859.69	C4	C4	C4	C4	C4	C5	C5	C5	C5					1231.18	1262.31	1292.11
15	5	86	889.51	922.47	962.25	C14	C14	C14	C15	C15	C15	C16	C16						1174.75	1209.45	1251.73
16	4	50	39.00	39.00	39.00	C9	C9	C9	C9	C10	C10	C10	C10	C11	C11				474.90	498.85	520.11
17	2	27	15.00	15.00	15.00	C5	C6	C6	C6										378.41	398.08	420.45
18	7	105	0.00	0.00	0.00	C17	C17	C17	C17	C18	C18								267.75	283.95	307.51
19	4	57	979.79	1018.50	1049.62	C10	C11	C11	C11	C12	C12								1253.44	1288.50	1321.51
20	1	0	769.13	793.39	818.31	C1	C1	C2	C2	C2									1043.28	1082.91	1114.71

Table 4.2 continues

Vessel no.	Berth no.	Berthing location	Min. berthing time	Estimated berthing time	Max. berthing time	The number of handling tasks of the vessels													Min. departure time	Estimated unberth. time	Max. departure time
						1	2	3	4	5	6	7	8	9	10	11	12	13			
21	2	27	378.41	398.08	420.45	C5	C5	C5	C6	C6	C6								807.99	832.15	859.69
22	2	20	15.00	15.00	15.00	C4	C4	C5	C5										268.47	283.86	298.32
23	5	80	1174.75	1209.45	1251.73	C14	C14	C15	C15	C15									1460.60	1497.72	1541.73
24	5	70	20.00	20.00	20.00	C11	C12	C12	C12	C12	C13	C13	C13						299.30	313.61	331.83
25	1	13	268.47	283.86	298.32	C3	C3	C4	C4	C4	C4	C4	C5						642.37	674.06	698.02
26	5	88	1174.75	1209.45	1251.73	C15	C15	C16	C16	C16	C16	C16							1584.19	1627.40	1666.90
27	4	59	1253.44	1288.50	1321.51	C11	C11	C11	C11	C12	C12								1679.67	1726.27	1769.29
28	6	90	470.84	488.60	513.07	C14	C14	C15	C15	C16	C16								659.38	684.55	715.32
29	2	31	1326.68	1361.96	1404.94	C6	C6	C7	C7	C7	C8								1578.15	1616.06	1662.02
30	6	92	144.00	144.00	144.00	C15	C15	C15	C16	C16	C16								470.84	488.60	513.07
31	2	32	935.14	969.70	1003.64	C6	C6	C6	C6	C7	C7	C7	C7	C8	C8	C8			1326.68	1361.96	1404.94
32	1	0	1043.28	1082.91	1114.71	C1	C1	C1	C2	C2	C2	C3	C3	C3					1412.53	1448.15	1488.63
33	4	50	1253.44	1288.50	1321.51	C9	C9	C10	C10	C10									1541.37	1586.51	1628.25
34	1	0	1412.53	1448.15	1488.63	C1	C1	C1	C2	C2	C2	C3							1640.25	1676.44	1719.09
35	5	70	575.84	599.82	623.12	C12	C12	C12	C12	C12	C13	C13	C13	C13	C14	C14			1022.16	1056.26	1090.07
36	7	105	661.42	693.78	724.58	C17	C17	C17	C18	C18	C18								902.65	935.61	966.92
37	1	12	642.37	674.06	698.02	C3	C3	C4	C4	C4									892.93	921.48	950.03
38	7	105	267.75	283.95	307.51	C17	C17	C17	C17	C18	C18	C18							661.42	693.78	724.58
39	6	90	659.38	684.55	715.32	C15	C15	C15	C16	C16									889.51	922.47	962.25
40	5	70	1022.16	1056.26	1090.07	C12	C13	C13	C13	C13	C14	C14							1401.75	1436.23	1471.52

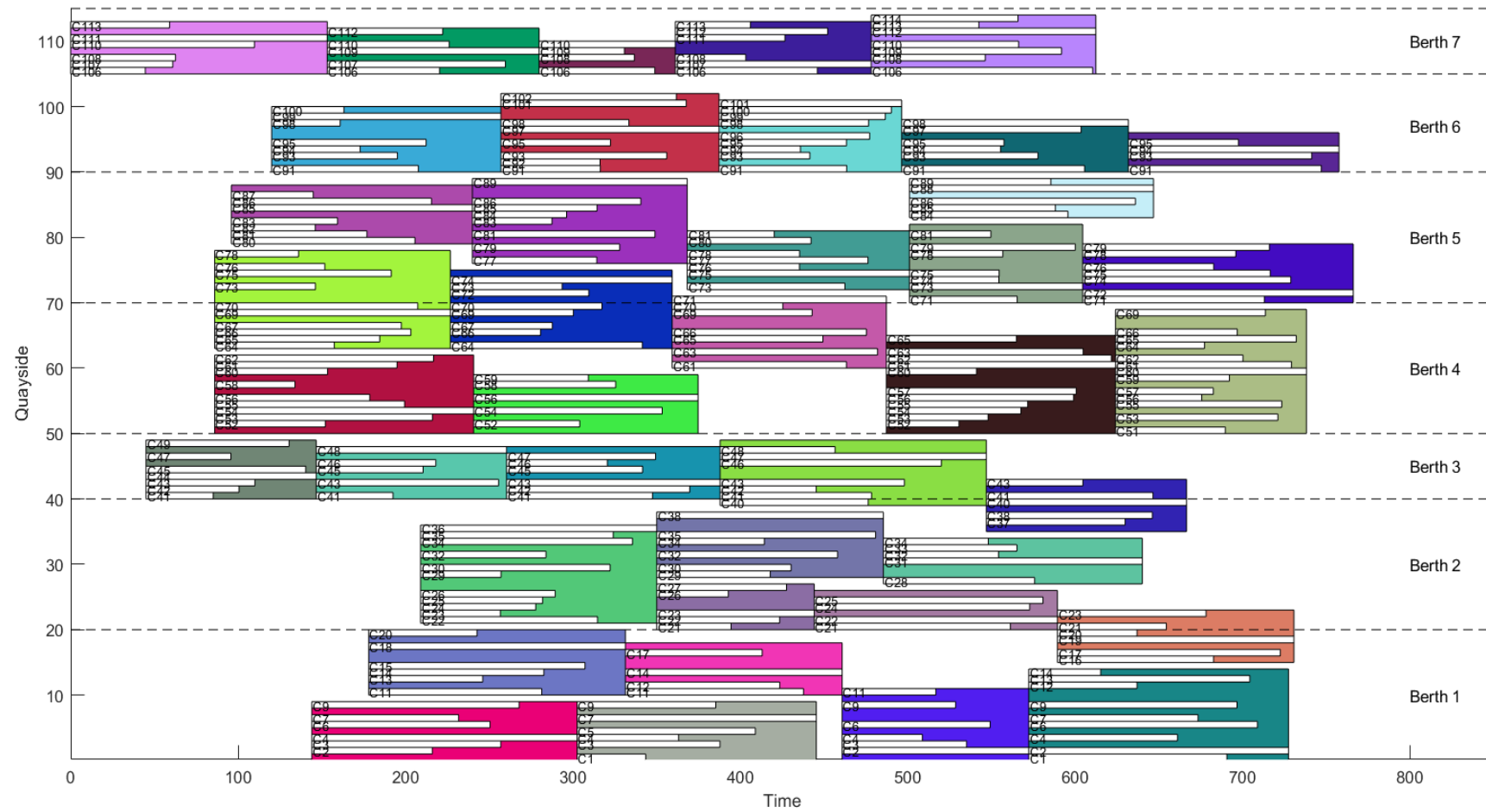


Figure 4.9 Berth space-time diagram for the solution of the sample problem 2 when it is not crane-constrained

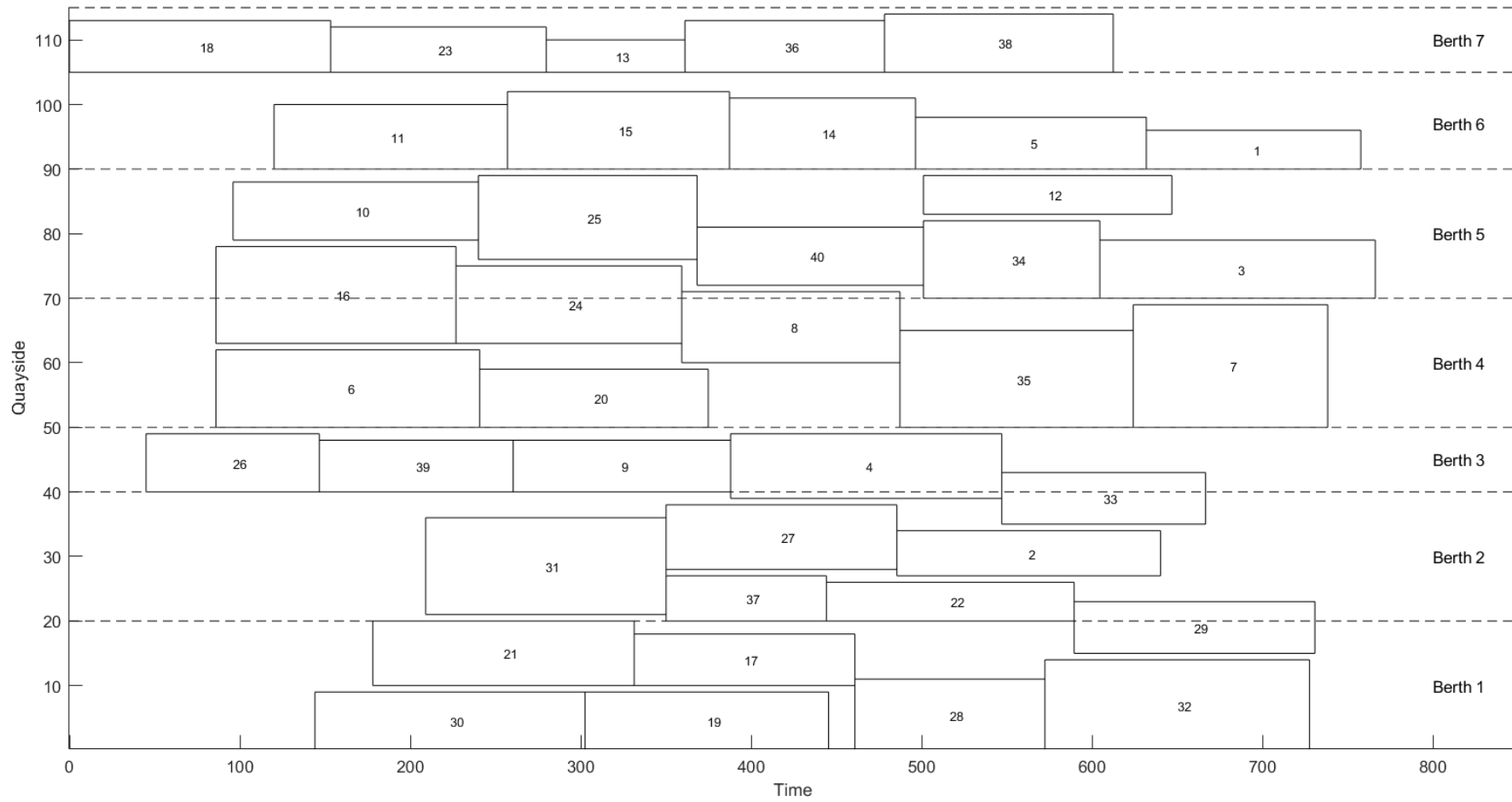


Figure 4.10 Berth space-time diagram for sample problem 2 when it is not crane-constrained with only vessel numbers

4.3 Representation and Solution of Sample Problem 3

The generated Sample Problem 3 has 60 incoming vessels, 10 berths spread over 5 quays and 25 QCs. The whole quayside has 160 berth sections and there are four discontinuities between Berth 3 and Berth 4, Berth 5 and Berth 6, Berth 6 and Berth 7, Berth 7 and Berth 8, respectively (See Figure 4.11).

When the SA based simulation optimization procedure is applied to the problem, R is determined as 3500. In this Sample Problem, 1614 different schedule groups have been formed and the most frequently observed schedule (*i.e.*, 36 times) has been selected.

The resulting berth space-time diagram of the selected schedule which represents the near-optimum solution is depicted in Figure 4.12 and Figure 4.13 are given to demonstrate vessel numbers apparently. Also, convergence graph of the selected solution is given in Figure 4.14. As it is depicted in Figure 4.12, for example vessels 57, 35, 55, 54, 24, 21, 52, 31, 41 and 48 have been assigned to Berth 1 in this given service order. Since the hybrid layout is assumed, Berth 1 has been occupied by vessels 54 and 24 simultaneously. Additionally, since there is no discontinuity between Berth 1 and Berth 2. Vessel 55 has been assigned to both berths at the same time under the condition that the draft of Vessel 55 is smaller than the depth of Berth 2. It is obvious that the hybrid berth layout assumption increases the berth utilization as well as decreases the makespan.

In addition, for each vessel, berthing location, estimated berthing/unberthing times and QC-to-task assignments are given in Table 4.3. The information in Table 4.3 can be interpreted as follows: for example, Vessel 3 is assigned to Berth 4 starting from the 50th Berth Section of the quayside. Vessel 3 has 7 loading/unloading tasks on itself and Crane 9 is assigned to the first five of the tasks and Crane 10 is assigned to the rest of the tasks, respectively. The estimated berthing time of Vessel 3 is calculated as 280.81 by considering the minimum (271.65) and maximum (291.80) values of berthing times. In the same way, the estimated unberthing time of Vessel 1 is

calculated as 905.46 by considering the minimum (891.41) and maximum (930.03) values of unberthing times. These given range in berthing/unberthing times provides the decision maker more reliable berth and quay crane schedules under uncertainty in loading/unloading operations.

Moreover, the problem is solved again assuming that each berth section consists of one QC. With this assumption, the problem turns into BAP and the solution of this problem provides a lower bound for the original problem. In Figure 4.15, berth space-time diagram of the problem with detailed QC schedule is given. In order to demonstrate the vessel numbers apparently, Figure 4.16 is also given. In these figures, a remarkable decrease in makespan of all vessels can be observed as expected.

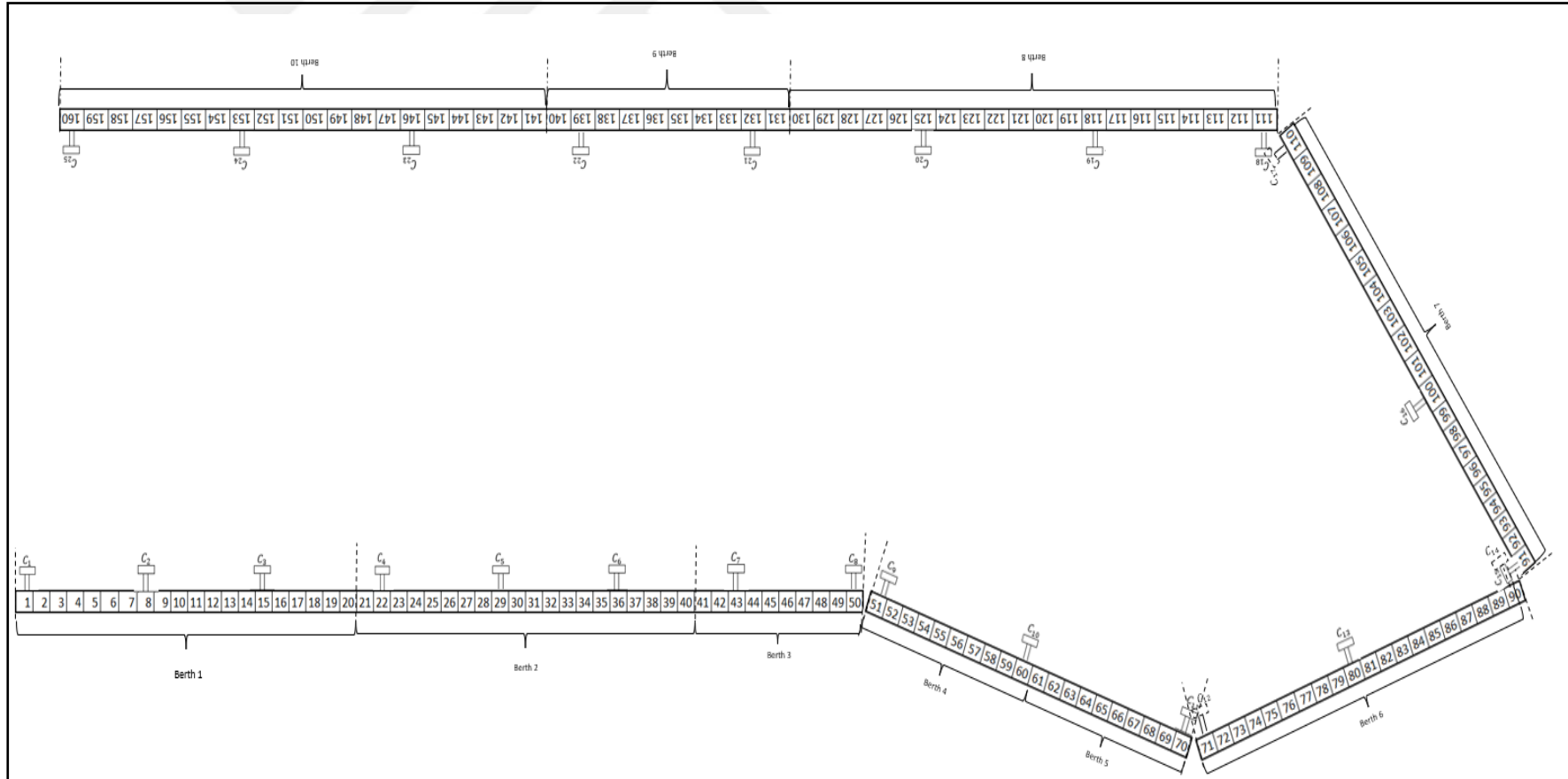


Figure 4.11 Representation of the quayside for the sample problem 3

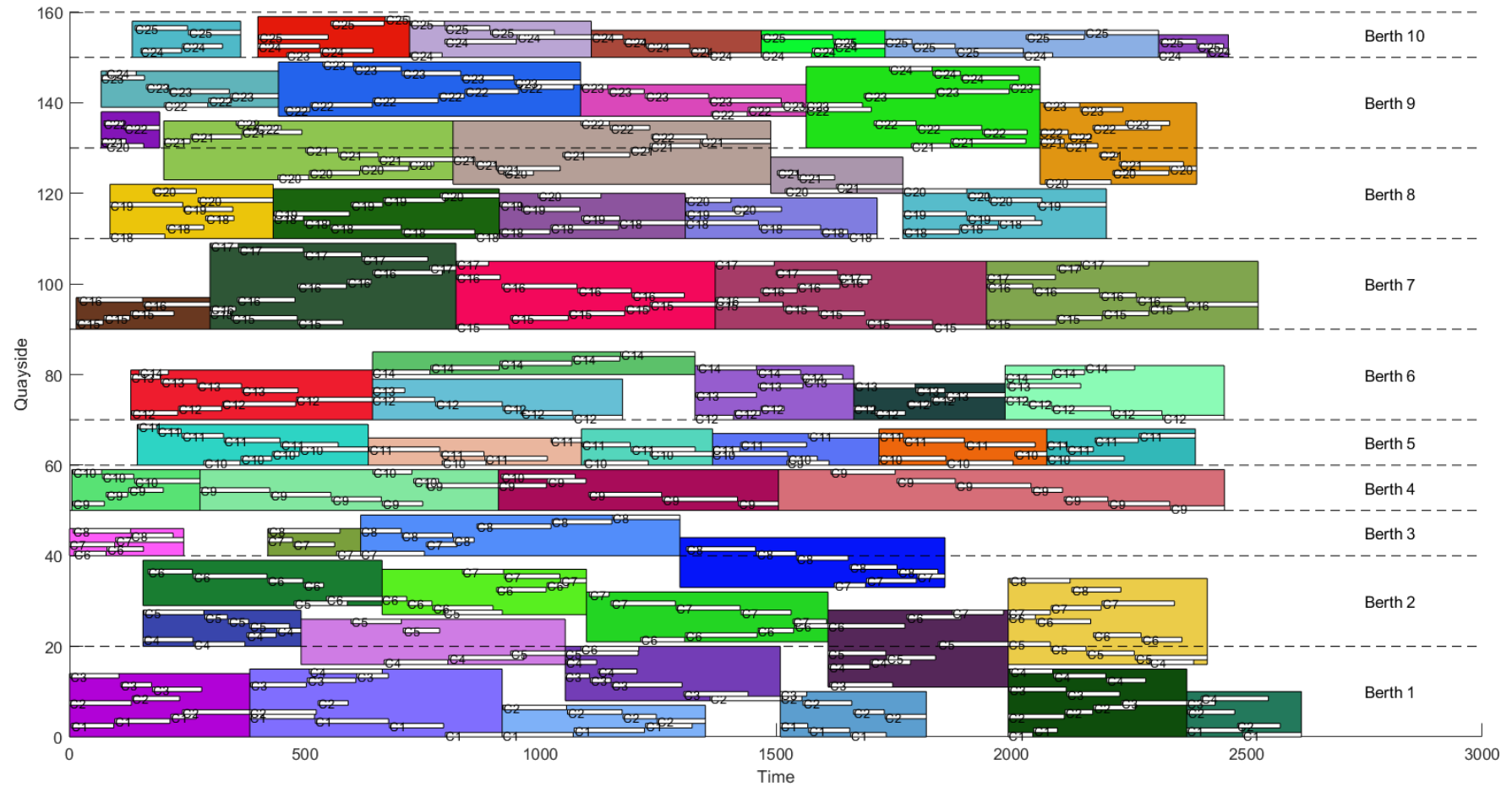


Figure 4.12 Berth space-time diagram for the solution of the proposed methodology for sample problem 3

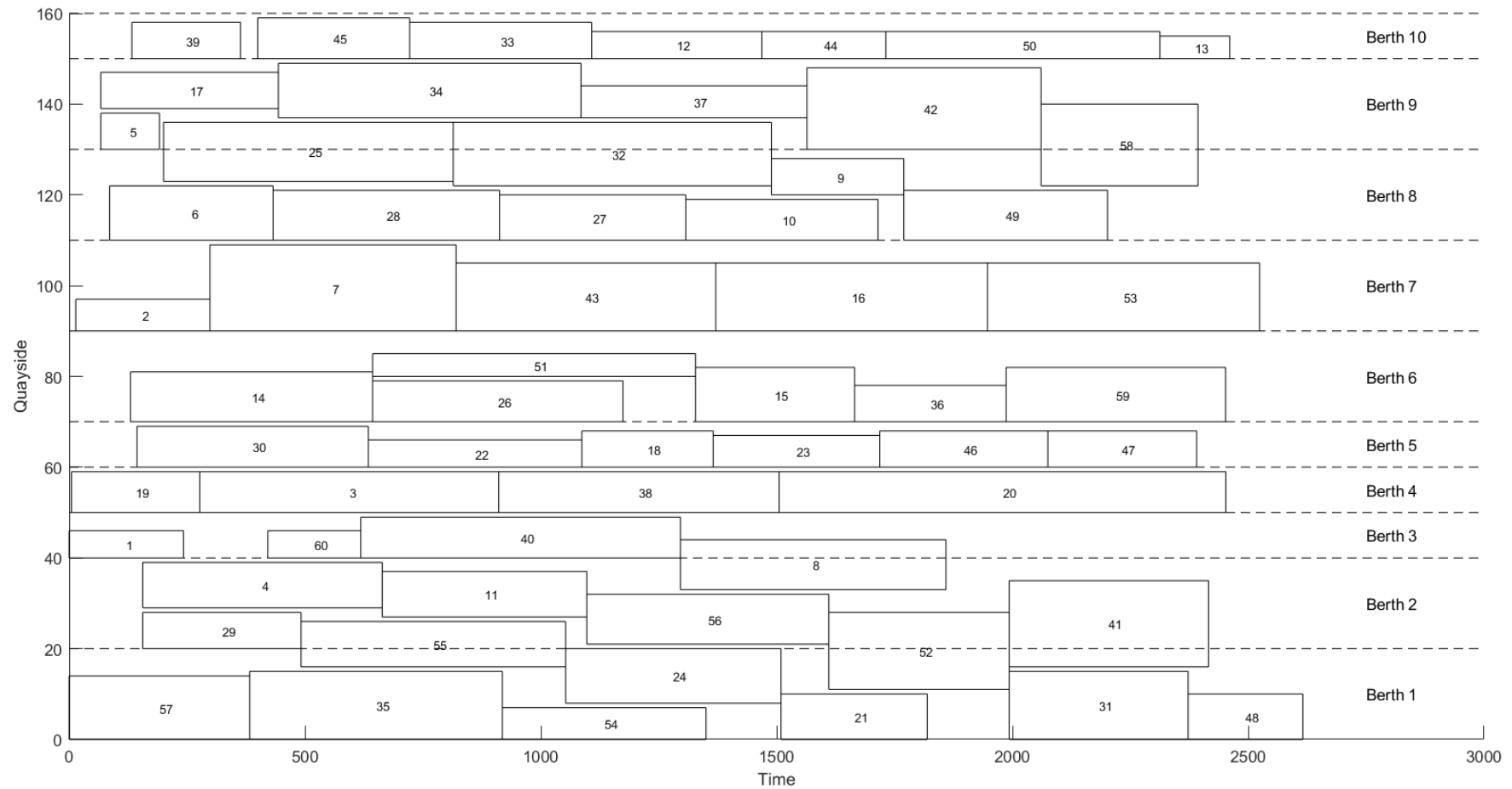


Figure 4.13 Berth space-time diagram for the solution of the proposed methodology sample problem 3 with only vessel numbers

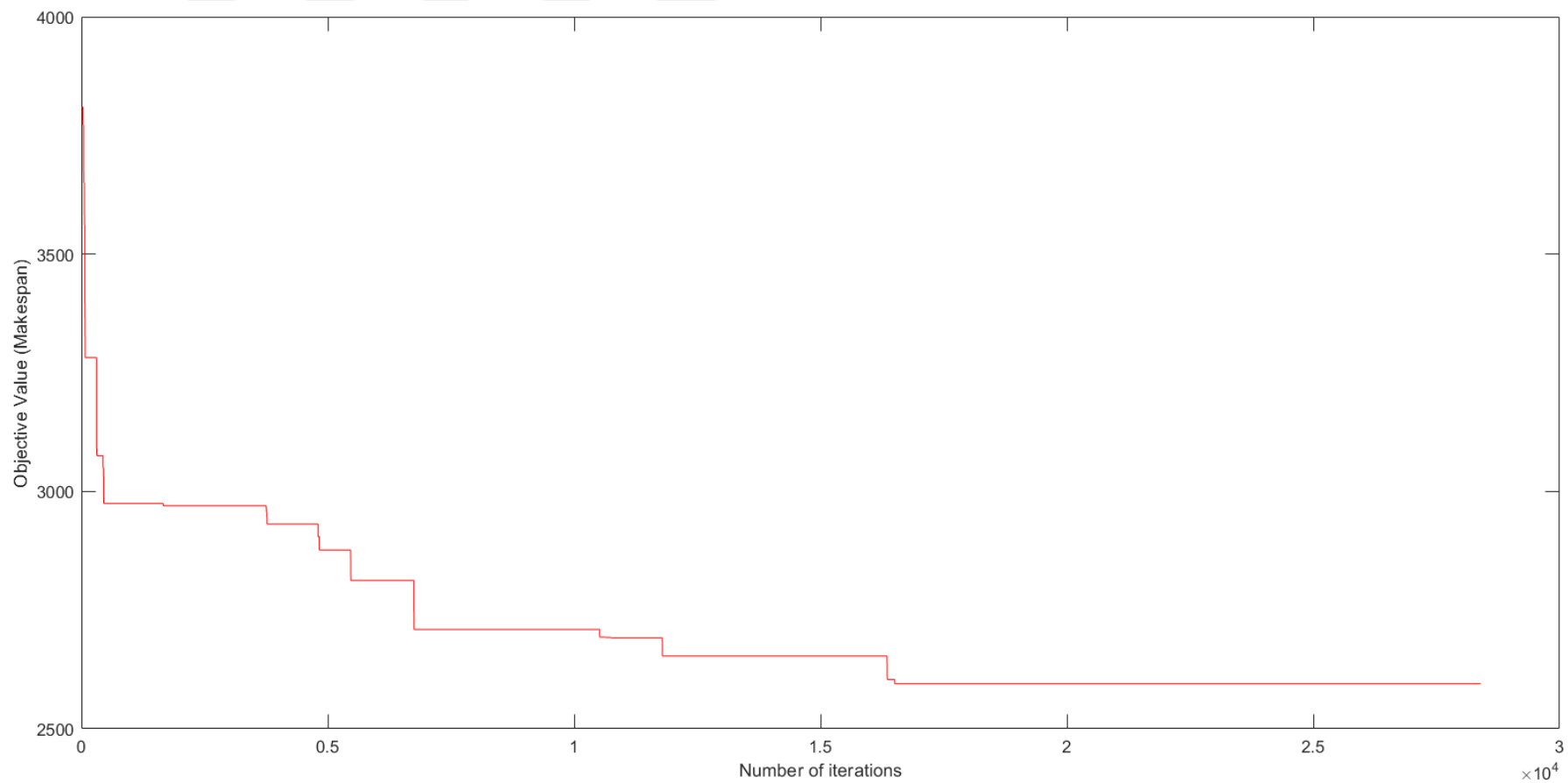


Figure 4.14 Convergence graph of the sample problem 3

Table 4.3 Berthing locations, average berthing times, departure times and crane assignment information of the vessels (sample problem 3)

Vessel no.	Berth no.	Berthing location	Min. berthing time	Estimated berthing time	Max. berthing time	The number of handling tasks of the vessels													Min. departure time	Estimated unberth. time	Max. departure time
						1	2	3	4	5	6	7	8	9	10	11	12	13			
1	3	40	0.00	0.00	0.00	C6	C6	C7	C7	C8	C8								228.96	237.82	248.66
2	7	90	14.00	14.00	14.00	C15	C15	C15	C16	C16									265.70	279.84	290.28
3	4	50	271.65	280.81	291.80	C9	C9	C9	C9	C9	C10	C10							891.41	905.46	930.03
4	2	29	156.00	156.00	156.00	C5	C6	C6	C6	C6	C6								634.98	657.62	671.59
5	9	130	67.00	67.00	67.00	C20	C21	C22	C22										197.61	204.48	214.36
6	8	110	86.00	86.00	86.00	C18	C18	C18	C19	C19	C20								415.59	430.86	444.07
7	7	90	265.70	279.84	290.28	C15	C15	C15	C16	C16	C16	C16	C16	C17	C17	C17	C17	C17	761.54	793.66	813.72
8	2	33	1285.23	1306.24	1325.10	C7	C7	C7	C8	C8	C8	C8	C8						1839.11	1861.81	1877.74
9	8	120	1295.56	1320.78	1336.57	C20	C21	C21	C21										1616.76	1643.13	1678.17
10	8	110	1284.97	1303.31	1331.78	C18	C18	C18	C18	C19	C20	C20							1690.27	1709.83	1734.99
11	2	27	634.98	657.62	671.59	C5	C6	C6	C6	C6	C6	C7	C7	C7					1063.50	1084.55	1100.68
12	10	150	1438.84	1463.50	1492.28	C24	C24	C24	C24	C24									1674.34	1693.27	1710.56
13	10	150	2367.49	2395.29	2416.01	C24	C24	C25	C25										2487.06	2514.45	2536.94
14	6	70	130.00	130.00	130.00	C12	C12	C12	C12	C13	C13	C13	C13	C14					612.24	641.43	657.82
15	6	70	1387.53	1407.90	1428.78	C12	C12	C12	C13	C13	C13	C14							1711.81	1738.88	1757.48
16	7	90	1315.35	1344.26	1376.63	C15	C15	C15	C15	C15	C16	C16	C16	C16	C17	C17	C17		1899.00	1922.94	1954.70
17	9	139	67.00	67.00	67.00	C22	C22	C23	C23	C23	C23	C24							362.87	371.92	385.19
18	5	60	1064.50	1082.68	1102.34	C10	C10	C11	C11										1342.44	1356.76	1378.64
19	4	50	5.00	5.00	5.00	C9	C9	C9	C10	C10	C10								271.65	280.81	291.80
20	4	50	1473.34	1496.15	1519.27	C9	C9	C9	C9	C9	C9	C9							2417.16	2452.25	2488.45

Table 4.3 continues

Vessel no.	Berth no.	Berthing location	Min. berthing time	Estimated berthing time	Max. berthing time	The number of handling tasks of the vessels													Min. departure time	Estimated unberth. time	Max. departure time
						1	2	3	4	5	6	7	8	9	10	11	12	13			
21	1	0	1481.63	1515.56	1546.23	C1	C1	C1	C2	C2	C2	C2	C3						1802.11	1834.16	1853.74
22	5	60	610.30	627.69	644.38	C10	C11	C11	C11	C11									1064.50	1082.68	1102.34
23	5	60	1342.44	1356.76	1378.64	C9	C10	C10	C11	C11	C11								1682.22	1703.48	1723.12
24	1	8	1037.69	1056.75	1077.03	C2	C3	C3	C3	C3	C4	C4	C5	C6					1481.63	1515.56	1546.23
25	8	113	200.00	204.66	214.36	C20	C20	C20	C20	C21	C21	C21	C21	C21	C21	C22	C22		873.13	893.26	907.01
26	6	70	612.24	641.43	657.82	C12	C12	C12	C12	C12	C13	C14							1020.21	1048.87	1066.35
27	8	110	898.12	911.91	934.89	C18	C18	C18	C19	C19	C19	C20							1284.97	1303.31	1331.78
28	8	110	415.59	430.86	444.07	C18	C18	C18	C18	C18	C19	C19	C19	C20					898.12	911.91	934.89
29	2	20	156.00	156.00	156.00	C4	C4	C4	C4	C5	C5	C5	C5						488.24	499.58	510.93
30	5	60	144.00	144.00	144.00	C10	C10	C10	C10	C11	C11	C11	C11	C11					610.30	627.69	644.38
31	1	0	1984.60	200.05	2026.19	C1	C1	C3	C3	C3	C4	C4	C4						2268.97	2293.57	2318.01
32	8	122	873.13	893.26	907.01	C20	C21	C21	C21	C21	C21	C21	C21	C22	C22	C22			1295.56	1320.78	1336.57
33	10	150	1043.47	1066.31	1091.32	C24	C24	C24	C25	C25	C25								1438.84	1463.50	1492.28
34	9	127	362.87	371.92	385.19	C22	C22	C22	C22	C22	C22	C23	C23	C23	C23	C2			908.15	927.96	953.11
35	1	0	377.22	389.13	401.90	C1	C1	C1	C1	C2	C2	C3	C3	C3	C4				906.71	924.87	945.46
36	6	70	1711.81	1738.88	1757.48	C12	C12	C12	C12	C13	C13	C13							2036.75	2059.12	2082.71
37	9	127	908.15	927.96	953.11	C22	C22	C23	C23	C23	C23	C23							1492.26	1518.02	1534.81
38	4	50	891.41	905.46	930.03	C9	C9	C9	C9	C9	C10								1473.34	1496.15	1519.27
39	10	150	133.00	133.00	133.00	C24	C24	C25	C25										349.45	360.93	371.32
40	3	40	607.58	617.06	627.30	C7	C7	C8	C8	C8	C8	C8	C8						1285.23	1306.24	1325.10

Table 4.3 continues

Vessel no.	Berth no.	Berthing location	Min. berthing time	Estimated berthing time	Max. berthing time	The number of handling tasks of the vessels														Min. departure time	Estimated unberth. time	Max. departure time
						1	2	3	4	5	6	7	8	9	10	11	12	13	14			
41	1	16	1984.60	2002.05	2026.19	C4	C5	C5	C5	C5	C6	C6	C6	C6	C7	C7	C7	C8	C8	2402.45	2429.62	2460.36
42	9	130	1492.26	1518.02	1534.81	C21	C21	C22	C22	C22	C22	C23	C23	C23	C23	C24	C24	C24		2137.44	2167.32	2186.56
43	7	90	761.54	793.66	813.72	C15	C15	C15	C15	C15	C16	C16	C16	C16	C17					1315.35	1344.26	1376.63
44	10	150	1674.34	1693.27	1710.56	C24	C24	C24	C25	C25										1938.50	1960.34	1978.04
45	10	150	400.00	400.00	400.00	C23	C24	C24	C25	C25	C25	C25								1043.47	1066.31	1091.32
46	5	60	1682.22	1703.48	1723.12	C10	C10	C10	C11	C11	C11									2032.66	2058.52	2076.60
47	5	60	2032.66	2058.52	2076.60	C10	C11	C11	C11	C11										2348.09	2377.21	2398.56
48	1	0	2268.97	2293.57	2318.01	C1	C1	C2	C2	C3	C4									2540.02	2562.02	2590.21
49	8	110	1690.27	1709.83	1734.99	C18	C18	C18	C19	C19	C19	C20	C20	C20						2114.30	2141.15	2166.30
50	10	150	1938.50	1960.34	1978.04	C24	C25	C25	C25	C25	C25									2367.49	2395.29	2416.01
51	6	80	612.24	641.43	657.82	C14	C14	C14	C14	C14										1387.53	1407.90	1428.78
52	1	11	1595.82	1618.28	1642.52	C3	C4	C4	C5	C5	C5	C6	C6	C7						1984.60	2002.05	2026.19
53	7	90	1899.00	1922.94	1954.70	C15	C15	C15	C15	C16	C16	C16	C16	C17	C17	C17				2466.22	2495.42	2531.12
54	1	0	906.71	924.87	945.46	C1	C1	C1	C2	C2	C2	C2								1321.75	1352.34	1381.92
55	1	16	488.24	499.58	510.93	C4	C4	C5	C5	C5										1037.69	1056.75	1077.03
56	2	21	1063.50	1084.55	1100.68	C6	C6	C6	C6	C7	C7	C7	C7	C7						1595.82	1618.28	1642.52
57	1	0	0.00	0.00	0.00	C1	C1	C1	C2	C2	C2	C3	C3	C3						377.22	389.13	401.90
58	8	122	2114.30	2141.15	2166.30	C20	C20	C20	C21	C21	C21	C21	C22	C22	C22	C23	C23	C23		2504.82	2538.98	257.87
59	6	70	2036.75	2059.12	2082.71	C12	C12	C12	C12	C12	C13	C14	C14	C14						2492.23	2522.27	2548.28
60	3	40	421.00	421.00	421.00	C7	C7	C7	C8											607.58	617.06	627.30

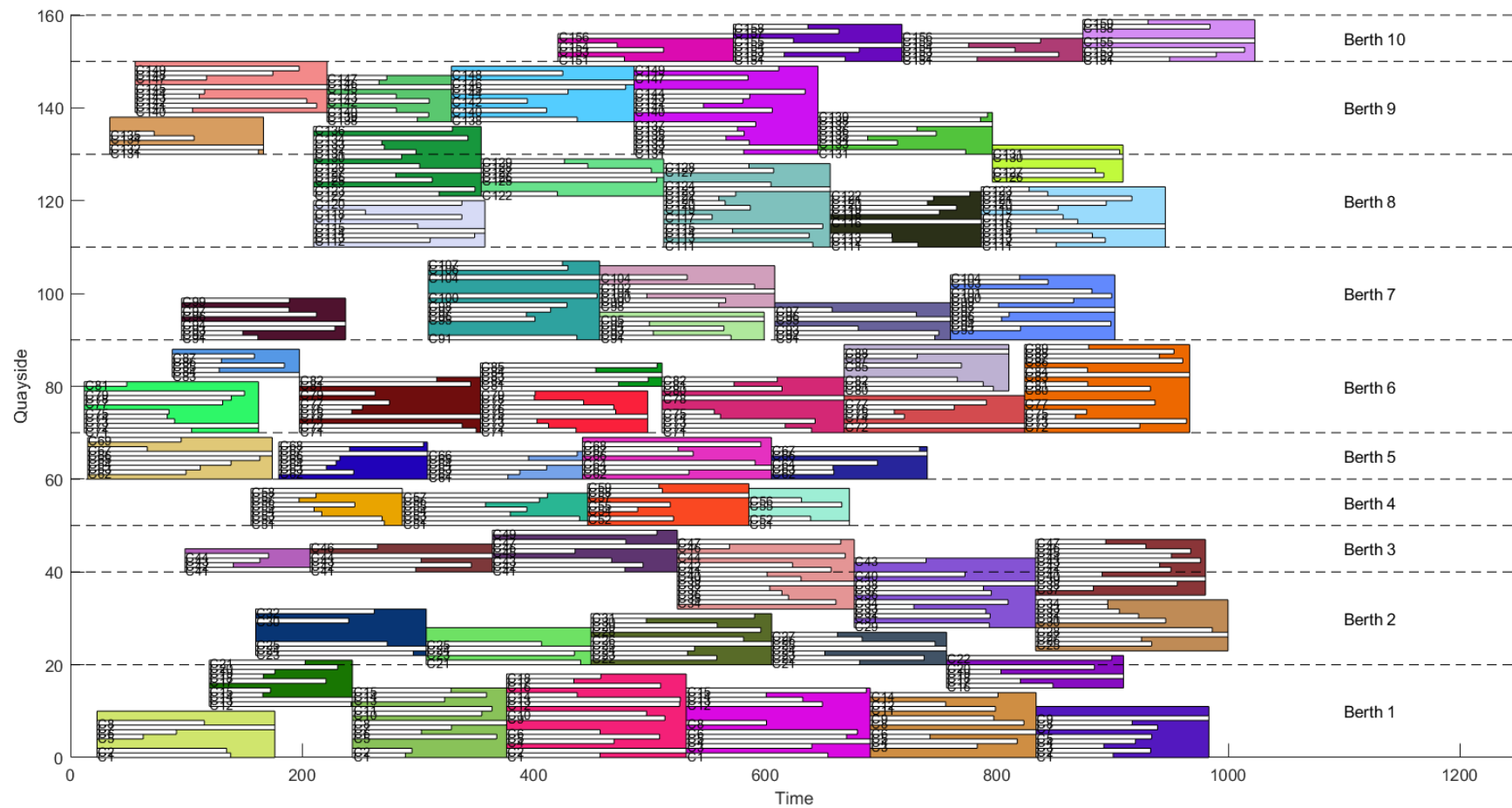


Figure 4.15 Berth space-time diagram for the solution of the sample problem 3 when it is not crane-constrained

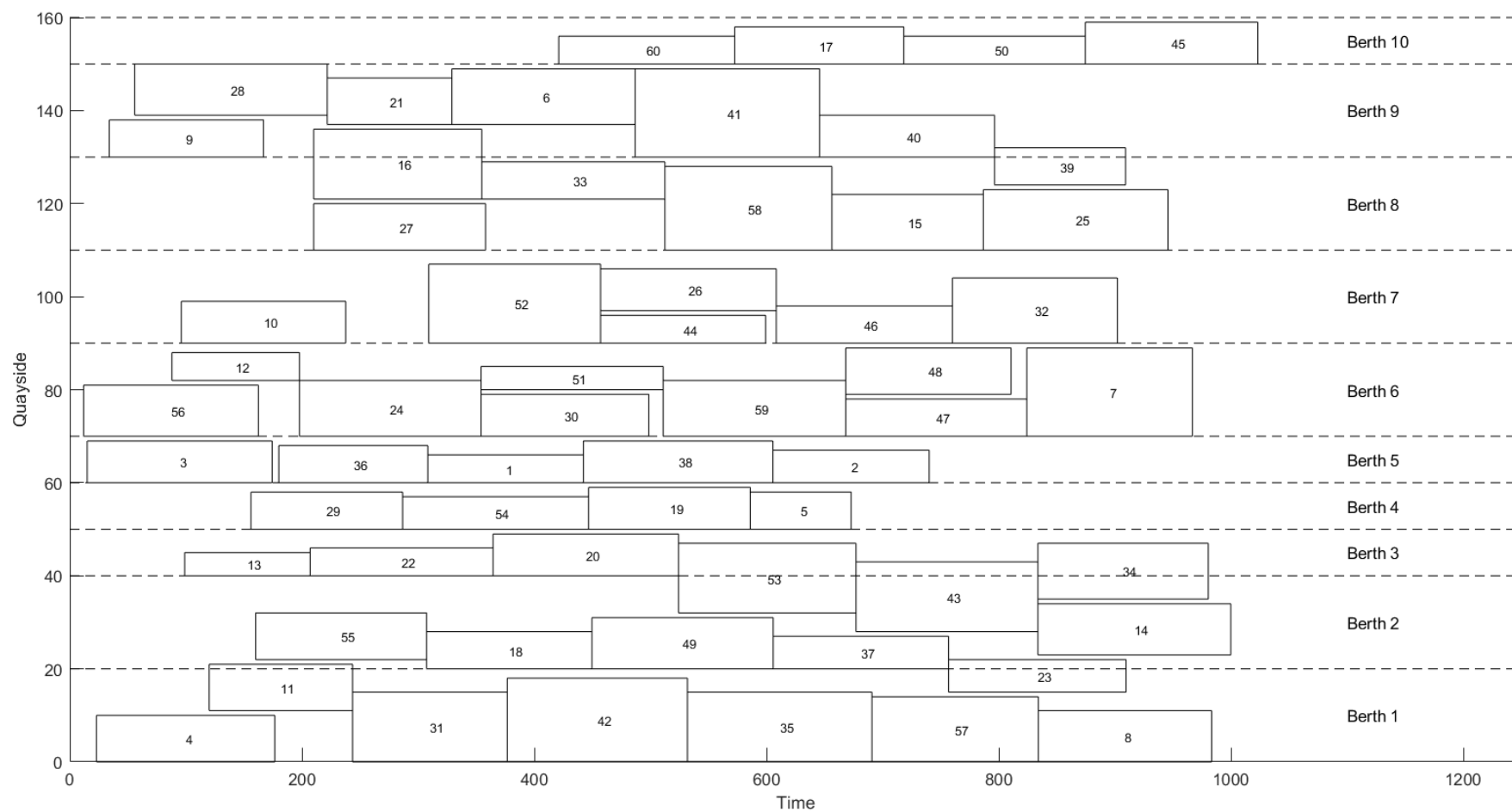


Figure 4.16. Berth space-time diagram for sample problem 3 when it is not crane-constrained with only vessel numbers

CHAPTER FIVE

CONCLUSIONS

Today, together with the increasing globalization, maritime transportation has become very important and the competition between container terminals have considerably increased. In order to gain superiority in this competitive environment, container terminals should improve their operational efficiency. To this aim, they primarily focus on the seaside operations which include berth allocation problem (BAP), quay crane assignment problem (QCAP) and quay crane scheduling problem (QCSP). Since the number and operation sequence of quay cranes assigned to a vessel determine the handling time of the vessel, the seaside problems are directly related to each other. Thus, solving these problems simultaneously give more realistic results.

In this dissertation, a simulation optimization-based solution approach is proposed for solving integrated Dynamic Multi-Quay BAP (DMQBAP) and QCSP with stochastic handling times and non-crossing constraints of QCs. The proposed solution approach for this problem has the following three contributions; 1) It integrates deeply BAP and QCSP with multi-quay hybrid berth layout, dynamic arrivals of vessels, coverage area and non-crossing constraints of QCs, and stochastic handling times, for the first time. 2) It provides a general parametric simulation model that can be adapted easily to different container terminal configurations. 3) It integrates the SA based search procedure with the developed simulation model to suggest the decision-maker the estimated berthing/unberthing times, berthing locations and quay crane schedules for each vessel for the most probable scenario.

This solution approach consists of four phases; In the first phase, an algorithm is proposed in order to obtain a conflict-free quay crane schedule assuming that the container handling times are stochastic. In the second phase, the general parametric simulation model is developed by using ARENA Simulation Software. This model handles the DMQBAP and the conflict-free QCSP simultaneously based on the proposed algorithm developed in the first phase. In the third phase, SA based simulation optimization procedure is developed in MATLAB and it is integrated with

the simulation model which is developed in the second phase. In this procedure, SA algorithm finds a candidate solution which gives vessel-to-berth assignments with service orders and this solution is passed to the simulation model as input. Then, the simulation model is executed for estimating the makespan of this berth allocation plan and passes the estimated makespan to the SA algorithm as fitness value. In the fourth phase, the proposed simulation optimization procedure is validated for deterministic handling times with respect to an existing Mixed-Integer Programming (MIP) model in literature, which solves the integrated BAP and QCSP. Finally, three different real-sized Sample Problems which have most of the realistic features of a port are generated and solved by the procedures proposed in the preceding phases. The generated Sample Problem 1 has 20 incoming vessels, 4 berths spread over 2 quays and 10 QCs. The whole quayside has 40 berth sections and there is only one discontinuity between Berth 2 and Berth 3. The generated Sample Problem 2 has 40 incoming vessels, 7 berths spread over 3 quays and 18 QCs. The whole quayside has 115 berth sections and there are two discontinuities between Berth 3 and Berth 4, Berth 6 and Berth 7, respectively. The generated Sample Problem 3 has 60 incoming vessels, 10 berths spread over 5 quays and 25 QCs. The whole quayside has 160 berth sections and there are four discontinuities between Berth 3 and Berth 4, Berth 5 and Berth 6, Berth 6 and Berth 7, Berth 7 and Berth 8, respectively. In Sample Problem 1, each vessel is assigned to a single berth in accordance with discrete berth layout. In Sample Problem 2 and Sample Problem 3, as the number of vessels in the planning period increases, some vessels occupy more than one berth and one berth contains more than one vessel at the same time in accordance with the hybrid berth layout. The results have also revealed that the proposed solution approach is effective in minimizing the makespan. Hybrid berth layout provides decrease in makespan by increasing the berth utilization.

Considering the solution of the problems, the managerial implications of the proposed solution approach are evaluated and summarized as follows: This solution approach provides the management of container terminal with useful insights in implementation of the resulting berth and quay crane schedules in practice. When the proposed solution approach is applied to a problem, the obtained solution which gives the estimated berthing/unberthing times with a given range, berthing locations and

quay crane schedules for each vessel for the most probable scenario may be used in order to inform incoming vessels before their arrivals. Since the proposed solution approach takes into account uncertainty in loading/unloading operations, it gives more reliable results considering the variabilities with a given range in berthing/unberthing times. Moreover, the proposed approach gives solutions in a reasonable time thus, it can be applied periodically in accordance with the container terminal's planning routine as follows: While the incoming vessels are approaching the container terminal, they periodically report their ETA (estimated time of arrival). In addition, the information of loading/unloading tasks on the vessels is obtained from the shipping companies or the cargo agents. Based on this information, using the proposed solution approach, the container terminal management can generate the plan of seaside operations for the vessels in the planning period before they have arrived at the container terminal. Additionally, since the proposed simulation model is general and parametric, the container terminal managers can easily adapt their real-life constraints and terminal configuration to the simulation model. By this way, they can solve different seaside operations problems. The simulation model enables the container terminal managers to make what-if analysis to evaluate any given different container terminal conditions and problems such as capacity planning and design of a container terminal, calculation of the berth and crane utilizations, effect of the changing weather conditions on the loading/unloading process, reliability of the loading/unloading equipment.

The following subjects can be considered as future research; In this dissertation, loading and unloading operations are unified as just a single task and precedence relations between them are ignored. In addition to this, it is assumed that two QCs can operate on consecutive ship-bays simultaneously. These constraints of the seaside operations can also be included in our parametric simulation model as a further study.

In this dissertation, in order to take the advantage of its ease of coding and implementation, the SA algorithm is preferred to be integrated with the simulation model to obtain berth plan and QC schedules. However, different metaheuristics can be used for the proposed simulation optimization procedure as future research.

Furthermore, the proposed simulation optimization procedure can be extended to include multiple objectives, such as minimizing average flow time of vessels and maximizing utilization of berths and quay cranes.

As it is mentioned before, the proposed solution approach provides the decision-maker with the estimated berthing/unberthing times, berthing locations and QC schedules for each vessel for the most probable scenario. The proposed simulation optimization procedure and finding the most probable scenario process can be automated and embedded in a decision support system with user interface. By this way, it can be easily used by the port management.

As another future research work, the developed simulation model and the simulation optimization procedure can be used for better understanding of the seaside operations. For example, it can be possible to design a system that automatically extracts knowledge about the management of these operations from the big data generated by these developed tools. Subsequently, this extracted knowledge can be converted to an expert system for more efficient management of seaside operations.

REFERENCES

- Agra, A., & Oliveira, M. (2018). MIP approaches for the integrated berth allocation and quay crane assignment and scheduling problem. *European Journal of Operational Research*, 264, 138-148.
- Al-Dhaheri, N., & Diabat, A. (2015). The Quay Crane Scheduling Problem. *Journal of Manufacturing Systems*, 36, 87-94.
- Al-Dhaheri, N., Jebali, A., & Diabat, A. (2016a). A simulation-based Genetic Algorithm approach for quay crane scheduling under uncertainty. *Simulation Modelling Practice and Theory*, 66, 122-138.
- Al-Dhaheri, N., Jebali, A., & Diabat, A. (2016b). The quay crane scheduling problem with nonzero crane repositioning time and vessel stability constraints. *Computers & Industrial Engineering*, 94, 230-244.
- Al-Dhaheri, N., & Diabat, A. (2017). A Lagrangian relaxation-based heuristic for the multi-ship quay crane scheduling problem with ship stability constraint. *Annals of Operations Research*, 248, 1-24.
- Arango, C., Cortes, P., Munuzuri, J., & Onieva, L. (2011). Berth allocation planning in Seville inland port by simulation and optimisation. *Advanced Engineering Informatics*, 25, 452–461.
- Bierwirth, C., & Meisel, C. (2009). A fast heuristic for quay crane scheduling with interference Constraints. *Journal of Scheduling*, 12, 345–360.
- Bierwirth, C., & Meisel, C. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202, 615–627.

- Bierwirth, C., & Meisel, F. (2015). A follow-up survey berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 244, 675-689.
- Boysen, N., Briskorn, D., & Meisel, F. (2017). A generalized classification scheme for crane scheduling with interference. *European Journal of Operational Research*, 258 (1), 343-357.
- Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., & Lusby, R. (2011). Models for the discrete berth allocation problem: A computational comparison. *Transportation Research Part E*, 47, 461–473.
- Canonaco, P., Legato, P., Mazza, R.M., & Musmanno, R. (2008). A queuing network model for the management of berth crane operations. *Computers & Operations Research*, 35, 2432 – 2446.
- Cheong, C.Y., Tan, K.C., Liu, D.K., & Lin, D.J. (2010). Multi-objective and prioritized berth allocation in container ports. *Annals of Operations Research*, 180, 63–103.
- Chang, D., Jiang, Z., Yan, W., & He, J. (2010). Integrating berth allocation and quay crane assignments. *Transportation Research Part E*, 46, 975-990.
- Chung, S.H., & Choy, K.L. (2012). A modified genetic algorithm for quay crane scheduling operations. *Expert Systems with Applications*, 39, 4213–4221.
- Cordeau, J., Laporte, G., Legato, P. & Moccia, L. (2005). Models and Tabu Search Heuristics for the Berth Allocation Problem. *Transportation Science*, 39 (4), 526-538.

- Cordeau, J.F., Gaudioso, M., Laporte, G., & Moccia, L. (2007). The service allocation problem at the Gioia Tauro Maritime Terminal. *European Journal of Operational Research*, 176 (2), 1167–1184.
- Cordeau, J.F., Laporte, G., Moccia, L., & Sorrentino, G. (2011). Optimizing yard assignment in an automotive transshipment terminal. *European Journal of Operational Research*, 215 (1), 149–160.
- Correcher, C. F., & Alvarez-Valdes, R. (2017). A biased random-key genetic algorithm for the time-invariant berth allocation and quay crane assignment problem. *Expert Systems with Applications*, 89, 112-128.
- Du, Y., Chen, Q., Quan, X., Long, L., & Funk, R.Y.K. (2011). Berth allocation considering fuel consumption and vessel emissions. *Transportation Research Part E*, 47, 1021–1037.
- Esmer, S., Yildiz, G., & Tuna, O. (2013). A new simulation modelling approach to continuous berth allocation. *International Journal of Logistics: Research and Applications*, 16 (5), 398–409.
- Giallombardo, G., Moccia, L., Salani, M., & Vacca, I. (2010). Modeling and solving the Tactical Berth Allocation Problem. *Transportation Research Part B*, 44, 232-245.
- Günther, H.O., & Kim, K. H. (2006). Container terminals and terminal operations. *OR Spectrum*, 28, 437–445.
- Han, X., Lu, Z., & Xi, L. (2010). A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. *European Journal of Operational Research*, 207, 1327-1340.

- Han, X., Gong, X., & Jo, J. (2015). A new continuous berth allocation and quay crane assignment model in container terminal. *Computers & Industrial Engineering*, 89, 15-22.
- Hansen, P., Oguz, C., & Mladenovic, N. (2008). Variable neighborhood search for minimum cost berth allocation problem. *European Journal of Operational Research*, 191, 636–649.
- Hu, Z. H. (2015a). Multi-objective genetic algorithm for berth allocation problem considering daytime preference. *Computers & Industrial Engineering*, 89, 2-14.
- Hu, Z.H. (2015b). Heuristics for solving continuous berth allocation problem considering periodic balancing utilization of cranes. *Computers & Industrial Engineering*, 85, 216-226.
- Imai, A., Nishimura, E., & Papadimitriou, S. (2001). The dynamic berth allocation problem for container port. *Transportation Research Part B*, 35 (4), 401-417.
- Imai, A., Nishimura, E., & Papadimitriou, S. (2003). Berth allocation with service priority. *Transportation Research Part B*, 37, 437-457.
- Imai, A., Sun, X., Nishimura, E., & Papadimitriou, S. (2005). Berth allocation in a container port: using a continuous location space approach. *Transportation Research Part B*, 39, 199-221.
- Imai, A., Nishimura, E., Hattori, M., & Papadimitriou, S. (2007). Berth allocation at indented berths for mega-containerships. *European Journal of Operational Research*, 179, 579–593.
- Imai, A., Nishimura, E., Hattori, M., & Papadimitriou, S. (2008). Berthing ships at a multi-user container terminal with a limited quay capacity. *Transportation Research Part E*, 44, 136–151.

- Imai, A., Chen, H.C., Nishimura, E., & Papadimitriou, S. (2008). The simultaneous berth and quay crane allocation problem. *Transportation Research Part E*, 44, 900–920.
- Karafa, J., Golias, M.M., Ivey, S., Saharidis, G.K.D. & Leonardos, N. (2013). The berth allocation problem with stochastic vessel handling times. *International Journal of Advanced Manufacturing Technology*, 65, 473–484.
- Karam, A., & Eltawil, A. B. (2016). Functional integration approach for the berth allocation, quay crane assignment and specific quay crane assignment problems. *Computers & Industrial Engineering*, 102, 458-466.
- Kasm, O.A., & Diabat, A. (2019). The quay crane scheduling problem with non-crossing and safety clearance constraints: An exact solution approach. *Computers & Operations Research*, 107, 189-199.
- Kavesghar, N., Huynh, N., & Rahimian, S.K. (2012). An efficient genetic algorithm for solving the quay crane scheduling problem. *Expert Systems with Applications*, 39, 13108–13117.
- Kim, K. H. & Moon, K.C., (2003). Berth scheduling by simulated annealing. *Transportation Research Part B*, 37, 541-560.
- Kim, K.H. & Park, Y.M. (2004). A crane scheduling method for port container terminals. *European Journal of Operational Research*, 156, 752-768.
- Kirkpatrick, S., Gelatt, C.D., & Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.
- Lalla-Ruiz, E., Izquierdo, C. E., Batista, B. M., & Moreno-Vega, J. M. (2013). A metaheuristic approach for the seaside operations in maritime container terminals. *In Lecture Notes in Computer Science* (including subseries Lecture Notes in

Artificial Intelligence and Lecture Notes in Bioinformatics) 7903 (28–35). Berlin, Heidelberg: Springer.

Lalla-Ruiz, E., González-Velarde, J. L., Melián-Batista, B., & Moreno-Vega, J. M. (2014). Biased random key genetic algorithm for the Tactical Berth Allocation Problem. *Applied Soft Computing Journal*, 22, 60–76.

Lalla-Ruiz, E., Expósito-Izquierdo C., Melián-Batista, B., & Moreno-Vega, J. M. (2016). A Set-partitioning-based model for the Berth Allocation Problem under Time-Dependent Limitations. *European Journal of Operational Research*, 250, 1001-1012.

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

Lee, Y., & Chen, C. (2009). An optimization heuristic for the berth allocation problem. *European Journal of Operational Research*, 196, 500–508.

Lee, D.H., & Wang, H.Q. (2010). Integrated discrete berth allocation and quay crane scheduling in port container terminals. *Engineering Optimization*, 42 (8), 747–761.

Lee, D., Chen J. H., & Cao, J. X. (2010). The continuous Berth Allocation Problem: A Greedy Randomized Adaptive Search Solution. *Transportation Research Part E*, 46, 1017–1029.

Lee, D.H., Jin, J. G., & Chen, J. H. (2012). Terminal and yard allocation problem for a container transshipment hub with multiple terminals. *Transportation Research Part E*, 48 (2), 516–528.

- Legato, P., Mazza, R.M., & Trunfio, R. (2010). Simulation-based optimization for discharge/loading operations at a maritime container terminal. *OR Spectrum*, 32, 543–567.
- Legato, P., Trunfio, R., & Meisel, F. (2012) Modeling and solving rich quay crane scheduling problems. *Computers & Operations Research*, 39, 2063–2078.
- Legato, P., Mazza, R.M., & Gulli, D. (2014). Integrating tactical and operational berth allocation decisions via Simulation-Optimization. *Computers & Industrial Engineering*, 78, 84-94.
- Liang, C., Huang, Y., & Yang, Y. (2009). A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Computers & Industrial Engineering*, 56, 1021–1028.
- Lin, S., & Ting, C. (2014). Solving the dynamic berth allocation problem by simulated annealing. *Engineering Optimization*, 46 (3), 308-327.
- Lin, S., Ting, C., & Wu, K. (2018). Simulated annealing with different vessel assignment strategies for the continuous berth allocation problem. *Flexible Services and Manufacturing Journal*, 30 (4), 740-763.
- Liu, J., Wan, Y., & Wang, L. (2006). Quay Crane Scheduling at Container Terminals to Minimize the Maximum Relative Tardiness of Vessel Departures. *Naval Research Logistics*, 53, 60-74.
- Lu, Z., Han, X., Xi, L., & Erera A.L. (2012). A heuristic for the quay crane scheduling problem based on contiguous bay crane operations. *Computers & Operations Research*, 39, 2915–2928.

- Mauri, G. R., Ribeiro, G. M., Lorena, L. A. N., & Laporte, G. (2016). An adaptive large neighborhood search for discrete and continuous berth allocation problem. *Computers & Operations Research*, 70, 140-154.
- Meisel, F., & Bierwirth, C. (2009). Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part E*, 45, 196–209.
- Meisel, F., & Bierwirth, C. (2011). A unified approach for the evaluation of quay crane scheduling models and algorithms. *Computers & Operations Research*, 38, 683–693.
- Meisel, F., & Bierwirth, C. (2012). A Framework for Integrated Berth Allocation and Crane Operations Planning in Seaport Container Terminals. *Transportation Science*, 47 (2), 131–147.
- Moghaddam, R.T., Makui, A., Salahi, S., Bazzazi, M., & Taheri, F. (2009) An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports. *Computers & Industrial Engineering*, 56, 241–248.
- Monaco, M.F., & Sammarra, M. (2007). The berth allocation problem: a strong formulation solved by a Lagrangean approach. *Transportation Science*, 41 (2), 265–280
- Msakni, M.K., Diabat, A., Rabadi, G., Al-Salem, M., & Kotachi, M. (2018). Exact methods for the quay crane scheduling problem when tasks are modeled at the single container level. *Computers & Operations Research*, 99, 218-233.
- Nguyen, S., Zhang, M., Johnston, M., & Tan, K.C. (2013). Hybrid evolutionary computation methods for quay crane scheduling problems. *Computers & Operations Research*, 40, 2083–2093.

- Park, Y.M., & Kim, K.H. (2005). A scheduling method for berths and quay cranes. *OR Spectrum*, 25 (1), 1–23
- Peng-Fei, Z., & Kang, H.G. (2008). Study on Berth and Quay-crane Allocation under Stochastic Environments in Container Terminal. *Systems Engineering Theory & Practice* (28). [https://doi.org/10.1016/S1874-8651\(09\)60001-6](https://doi.org/10.1016/S1874-8651(09)60001-6)
- Raa, B., Dullaert, W., & Schaeren, R.V. (2011). An enriched model for the integrated berth allocation and quay crane assignment problem. *Expert Systems with Applications*, 38, 14136–14147.
- Samarra, M., Cordeau, J.F., Laporte, G., & Monaco, M.F. (2007). A tabu search heuristic for the quay crane scheduling. *Journal of Scheduling*, 10, 327-336.
- Shang, X. T, Cao, J. X., & Ren, J. (2016). A robust optimization approach to the integrated berth allocation and quay crane assignment problem. *Transportation Research Part E*, 94, 44-65.
- Song, L., Cherrett, T., & Guan, W. (2012). Study on berth planning problem in a container seaport: Using an integrated programming approach. *Computers & Industrial Engineering*, 62, 119-128.
- Song, Y., Zhang, J., Liu, M., & Chu, C. (2019). The berth allocation optimisation with the consideration of time-varying water depths. *International Journal of Production Research*, 57 (2), 488-516.
- Stahlbock, R., & VoB, S. (2008). Operations research at container terminals: a literature update. *OR Spectrum*, 30, 1–52.
- Steenken, D., VoB, S., & Stahlbock, R. (2005). Container terminal operation and operations research – a classification and literature review. *OR Spectrum*, 26, 3–49.

- Tekin, E., & Sabuncuoglu, I. (2004). Simulation optimization: A comprehensive review on theory and applications. *IIE Transactions*, 36 (11), 1067-1081.
- Ting, C.J., Wu, K. C., & Chou, H. (2014). Particle swarm optimization algorithm for the berth allocation problem. *Expert Systems with Applications*, 41, 1543-1550.
- Türkoğulları, Y. B., Taşkın, Z. C., Aras, N., & Altinel, İ. K. (2014). Optimal berth allocation and time-invariant quay crane assignment in container terminals. *European Journal of Operational Research*, 235, 88-101.
- Türkoğulları, Y. B., Taşkın, Z. C., Aras, N., & Altinel, İ. K. (2016). Optimal berth allocation and time-invariant quay crane assignment and scheduling with crane setups in container terminals. *European Journal of Operational Research*, 254, 985-1001.
- Umang, N., Bierlaire, M., & Erera, A.L. (2017). Real-time management of berth allocation with stochastic arrival and handling times. *Journal of Scheduling*, 20, 67-83.
- Ursavaş, E. (2014). A decision support system for quayside operations in a container terminal. *Decision Support Systems*, 59, 312-324.
- Vacca, I., Salani, M., & Bierlaire, M. (2012). An exact algorithm for the integrated planning of berth allocation and quay crane assignment, *Transportation Sciences*, 47, 148-161.
- Vis, F.A. I., & Koster, R.D. (2003). Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 47 (1), 1-16.
- Wang, F., & Lim, A. (2007). A stochastic beam search for the berth allocation problem. *Decision Support Systems*, 42, 2186-2196.

- Xiang, X., Liu, C., & Miao, L. (2017). A bi-objective robust model for berth allocation scheduling under uncertainty. *Transportation Research Part E*, 106, 294-319.
- Xu, D., Li, C., & Leung, J. (2012). Berth allocation with time-dependent physical limitations on vessels. *European Journal of Operational Research*, 216, 47–56.
- Yang, C., Wang, X., & Li, Z. (2012). An optimization approach for coupling problem of berth allocation and quay crane assignment in container terminal. *Computers & Industrial Engineering*, 63, 243-253.
- Zhang, C., Zheng, L., Zhang, Z., Shi, L., & Armstrong, A.J. (2010). The allocation of berths and quay cranes by using a sub-gradient optimization technique. *Computers & Industrial Engineering*, 58, 40–50.
- Zhang, A., Zhang, W. Chen, Y., Chen, G., & Chen, X. (2017). Approximate the scheduling of quay cranes with non-crossing constraints. *European Journal of Operational Research*, 258 (3), 820-828.
- Zhen, L. (2015). Tactical berth allocation under uncertainty. *European Journal of Operational Research*, 247, 928-944.

APPENDICES

Appendix A1: ARENA model of the developed simulation model

Mod file of the ARENA model

```
4$          CREATE,          number_of_berths,0.000001:,1:NEXT(9$);

9$          ASSIGN:          c1=c1+1:
                                assigned_berth_of=c1;
13$         BRANCH,          1:

If,task_assigned_to_berth(assigned_berth_of)>0,10$,Yes:
                                Else,14$,Yes;
10$         BRANCH,          1:
                                If,assigned_berth_of==1,5$,Yes:
                                If,assigned_berth_of==2,11$,Yes:
                                If,assigned_berth_of==3,12$,Yes:
                                If,assigned_berth_of==4,51$,Yes;

5$          SCAN:
kontrol(cizelge(assigned_berth_of,kontrolv(assigned_berth_of)))==1;
24$         ASSIGN:
vesselno=cizelge(assigned_berth_of,kontrolv(assigned_berth_of)):NEXT(197$);

197$        ASSIGN:          sayac=first_node(assigned_berth_of):
                                sayac1=1:
                                available=0:
                                not_available=0:
                                bekle=0;
183$        BRANCH,          1:

If,sayac<first_node(assigned_berth_of)+berth_length(assigned_berth_of),174$,
Yes:
                                Else,182$,Yes;
174$        WHILE:          sayac1<=vessel_length#(vesselno);
175$        IF:              allocated(sayac)==0;
176$        ASSIGN:          available=available+1;
191$        ELSE;
190$        ASSIGN:          not_available=sayac;
177$        ENDIF;
178$        ASSIGN:          sayac=sayac+1:
                                sayac1=sayac1+1;

179$        ENDWHILE;
180$        BRANCH,          1:

If,available==vessel_length#(vesselno),181$,Yes:
                                Else,182$,Yes;
181$        ASSIGN:          first_allocated(vesselno)=sayac-
vessel_length#(vesselno);
184$        IF:              bekle<>1;
185$        ASSIGN:          sayac=first_allocated(vesselno):
                                sayac1=1;
186$        WHILE:          sayac1<=vessel_length#(vesselno);
187$        ASSIGN:          allocated_by(sayac)=vesselno:
                                allocated(sayac)=1:
                                sayac=sayac+1:
                                sayac1=sayac1+1;

188$        ENDWHILE;
189$        ENDIF:NEXT(6$);

182$        ASSIGN:          bekle=1;
```

```

192$      BRANCH,      1:
                        If,assigned_berth_of==1,193$,Yes:
                        If,assigned_berth_of==2,194$,Yes:
                        If,assigned_berth_of==3,195$,Yes:
                        If,assigned_berth_of==4,196$,Yes;
193$      WAIT:        assigned_berth_of,1:NEXT(197$);
194$      WAIT:        assigned_berth_of,1:NEXT(197$);
195$      WAIT:        assigned_berth_of,1:NEXT(197$);
196$      WAIT:        assigned_berth_of,1:NEXT(197$);
6$        SEARCH,      assigned_berth_of,1,nq(assigned_berth_of):
cizelge(assigned_berth_of,kontrolv(assigned_berth_of))==no;
7$        REMOVE:      j,assigned_berth_of,8$;
16$       ASSIGN:
kontrolv(assigned_berth_of)=kontrolv(assigned_berth_of)+1;
21$       DUPLICATE:    1,25$:NEXT(26$);

26$       ASSIGN:      berthing_time(vesselno)=tnow:NEXT(52$);
52$       ASSIGN:      loaded_bay(vesselno)=0:
                        c3(vesselno)=1:
                        sayac=1;
65$       WHILE:      c3(vesselno)<=vessel_length#(vesselno);
66$       IF:         bay_task(vesselno,c3(vesselno))==1;
67$       ASSIGN:
location=first_allocated(vesselno)+c3(vesselno)-1:
allocated_nodes_by_each_vessel(vesselno,location)=1:
                        bays(location)=1:
                        pick_up_points#=pick_up_points#+1;
85$       DUPLICATE:    1,68$:NEXT(86$);

86$       QUEUE,      craneq:DETACH;
68$       ENDIF;
69$       ASSIGN:      c3(vesselno)=c3(vesselno)+1;
53$       ENDWHILE:NEXT(155$);

155$      ASSIGN:      sayac=1:
                        left_most_crane=1;
156$      WHILE:      sayac<assigned_berth_of;
157$      ASSIGN:
left_most_crane=left_most_crane+number_of_cranes_at_each_berth(sayac):
                        sayac=sayac+1;
158$      ENDWHILE;
159$      ASSIGN:      req_location=first_allocated(vesselno);
169$      ASSIGN:      last_requested(left_most_crane)=req_location;
161$      REQUEST,    1:Crane(left_most_crane),,INTX(req_location);
170$      FREE:      Crane(left_most_crane);
171$      ASSIGN:      left_most_crane=left_most_crane+1:
                        sayac=1:
                        remaining_length=vessel_length#(vesselno)-2:

remaining_cranes=number_of_cranes_at_each_berth(assigned_berth_of)-2;
160$      WHILE:
sayac<=number_of_cranes_at_each_berth(assigned_berth_of)-1;
162$      IF:
sayac==number_of_cranes_at_each_berth(assigned_berth_of)-1;
163$      ASSIGN:
req_location=first_allocated(vesselno)+vessel_length#(vesselno)-1:
                        last_requested(left_most_crane)=req_location;
164$      ELSE;

```

```

172$          ASSIGN:
req_location=first_allocated(vesselno)+AINT((remaining_length/(remaining_cra
nes+1))*sayac)+1:
                                last_requested(left_most_crane)=req_location;
165$          ENDIF;
173$          REQUEST,          1:Crane(left_most_crane),,INTX(req_location);
166$          FREE:            Crane(left_most_crane);
168$          ASSIGN:          left_most_crane=left_most_crane+1:
                                sayac=sayac+1;
167$          ENDWHILE:NEXT(92$);

92$          ASSIGN:          sayac=1;
93$          WHILE:            sayac<=number_of_cranes;
94$          SIGNAL:          sayac;
95$          ASSIGN:          sayac=sayac+1;
96$          ENDWHILE;
64$          BRANCH,          1:
                                If,vesselno==1,62$,Yes:
                                If,vesselno==2,61$,Yes:
                                If,vesselno==3,59$,Yes:
                                If,vesselno==4,58$,Yes:
                                If,vesselno==5,57$,Yes:
                                If,vesselno==6,54$,Yes:
                                If,vesselno==7,55$,Yes:
                                If,vesselno==8,56$,Yes:
                                If,vesselno==9,60$,Yes:
                                If,vesselno==10,63$,Yes:
                                If,vesselno==11,78$,Yes:
                                If,vesselno==12,77$,Yes:
                                If,vesselno==13,75$,Yes:
                                If,vesselno==14,74$,Yes:
                                If,vesselno==15,73$,Yes:
                                If,vesselno==16,70$,Yes:
                                If,vesselno==17,71$,Yes:
                                If,vesselno==18,72$,Yes:
                                If,vesselno==19,76$,Yes:
                                If,vesselno==20,79$,Yes:
                                If,vesselno==21,84$,Yes:
                                If,vesselno==22,83$,Yes:
                                If,vesselno==23,82$,Yes:
                                If,vesselno==24,81$,Yes:
                                If,vesselno==25,80$,Yes;

62$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

61$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

59$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

58$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

57$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

54$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

55$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

56$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

```

```

60$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

63$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

78$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

77$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

75$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

74$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

73$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

70$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

71$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

72$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

76$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

79$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

84$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

83$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

82$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

81$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

80$          SCAN:
loaded_bay(vesselno)==pick_up_points#:NEXT(206$);

87$          CREATE,          number_of_cranes:,1:NEXT(88$);

88$          ASSIGN:          transporter_number=transporter_number+1:
                                transporter#=transporter_number;
waiting      BRANCH,          1:
                                If,transporter#==1,104$,Yes:
                                If,transporter#==2,102$,Yes:
                                If,transporter#==3,91$,Yes:
                                If,transporter#==4,97$,Yes:
                                If,transporter#==5,98$,Yes:
                                If,transporter#==6,99$,Yes:
                                If,transporter#==7,100$,Yes:
                                If,transporter#==8,101$,Yes:

```

```

If,transporter#==9,103$,Yes:
If,transporter#==10,105$,Yes;
104$      WAIT:      transporter#;
127$      BRANCH,    1:
                                If,transporter#==1,117$,Yes:
                                If,transporter#==2,118$,Yes:
                                If,transporter#==3,119$,Yes:
                                If,transporter#==4,120$,Yes:
                                If,transporter#==5,121$,Yes:
                                If,transporter#==6,122$,Yes:
                                If,transporter#==7,123$,Yes:
                                If,transporter#==8,124$,Yes:
                                If,transporter#==9,125$,Yes:
                                If,transporter#==10,126$,Yes;
117$      SCAN:      nq(craneq)<>0;
89$      BRANCH,    1:
                                If,transporter#==1,128$,Yes:
                                If,transporter#==number_of_cranes,137$,Yes:
                                Else,146$,Yes;
128$      ASSIGN:    rank=1:
                                min_distance=100000:
                                min_rank=0:
                                min_number=1000:
                                sayac=1;
129$      WHILE:    rank<=nq(craneq);
130$      IF:
                                (AQUE(craneq,rank,300)>0)&&(AQUE(craneq,rank,300)<last_requested(transporter
# +1))&&((lower_crane(transporter#)<=AQUE(craneq,rank,300))&&(upper_crane(tran
sporter#)>=AQUE(craneq,rank,300)));
131$      IF:      abs(AQUE(craneq,rank,300)-
last_requested(transporter#))<min_distance;
132$      ASSIGN:    min_distance=abs(AQUE(craneq,rank,300)-
last_requested(transporter#)):
                                min_rank=rank:
                                min_number=AQUE(craneq,rank,300);
133$      ENDIF;
134$      ENDIF;
135$      ASSIGN:    rank=rank+1;
136$      ENDWHILE:NEXT(90$);

90$      BRANCH,    1:
                                If,min_rank<>0,113$,Yes:
                                Else,waiting,Yes;
113$      ASSIGN:    task_location=AQUE(craneq,min_rank,300):
                                vesselno=AQUE(craneq,min_rank,400);
106$      REMOVE:    min_rank,craneq,107$;
114$      ASSIGN:    allocated_nodes_by_each_crane(vesselno,task_location)=transporter#:
                                picture=p2:
                                last_requested(transporter#)=task_location;
108$      REQUEST,  1:Crane(transporter#),,INTX(task_location);
115$      ASSIGN:    crane_start_time_on_each_node(vesselno,task_location)=tnow;
112$      ASSIGN:    baynumber=task_location:

baysequence=(AMOD(task_location,10)==0)*10+(AMOD(task_location,10)<>0)*AMOD(
task_location,10);
109$      DELAY:    sure(vesselno,baysequence),,Other:NEXT(111$);

111$      ASSIGN:    bays(baynumber)=0;
110$      FREE:      Crane(transporter#);
116$      ASSIGN:    loaded_bay(vesselno)=loaded_bay(vesselno)+1:

crane_finish_time_on_each_node(vesselno,task_location)=tnow:NEXT(127$);

```

```

107$          DISPOSE:          No;

137$          ASSIGN:           rank=1:
                                min_distance=100000:
                                min_rank=0:
                                min_number=10000;

138$          WHILE:           rank<=nq(craneq);
139$          IF:

(AQUE(craneq,rank,300)>last_requested(transporter#-
1))&&(AQUE(craneq,rank,300)<=number_of_intersections)&&((lower_crane(transpo
rter#)<=AQUE(craneq,rank,300))&&(upper_crane(transporter#)>=AQUE(craneq,rank
,300)));
140$          IF:               abs(last_requested(transporter#)-
AQUE(craneq,rank,300))<min_distance;
141$          ASSIGN:          min_distance=abs(AQUE(craneq,rank,300)-
last_requested(transporter#)):
                                min_rank=rank:
                                min_number=AQUE(craneq,rank,300);

142$          ENDIF;
143$          ENDIF;
144$          ASSIGN:          rank=rank+1;
145$          ENDWHILE:NEXT(90$);

146$          ASSIGN:          rank=1:
                                min_distance=100000:
                                min_rank=0:
                                min_number=1000;

147$          WHILE:           rank<=nq(craneq);
148$          IF:

(AQUE(craneq,rank,300)>last_requested(transporter#-
1))&&(AQUE(craneq,rank,300)<last_requested(transporter#+1))&&((lower_crane(t
ransporter#)<=AQUE(craneq,rank,300))&&(upper_crane(transporter#)>=AQUE(crane
q,rank,300)));
149$          IF:               abs(AQUE(craneq,rank,300)-
last_requested(transporter#))<min_distance;
150$          ASSIGN:          min_distance=abs(AQUE(craneq,rank,300)-
last_requested(transporter#)):
                                min_rank=rank:
                                min_number=AQUE(craneq,rank,300);

151$          ENDIF;
152$          ENDIF;
153$          ASSIGN:          rank=rank+1;
154$          ENDWHILE:NEXT(90$);

118$          SCAN:            nq(craneq)<>0:NEXT(89$);

119$          SCAN:            nq(craneq)<>0:NEXT(89$);

120$          SCAN:            nq(craneq)<>0:NEXT(89$);

121$          SCAN:            nq(craneq)<>0:NEXT(89$);

122$          SCAN:            nq(craneq)<>0:NEXT(89$);

123$          SCAN:            nq(craneq)<>0:NEXT(89$);

124$          SCAN:            nq(craneq)<>0:NEXT(89$);

125$          SCAN:            nq(craneq)<>0:NEXT(89$);

126$          SCAN:            nq(craneq)<>0:NEXT(89$);

102$          WAIT:            transporter#:NEXT(127$);

```

```

91$          WAIT:          transporter#:NEXT(127$);
97$          WAIT:          transporter#:NEXT(127$);
98$          WAIT:          transporter#:NEXT(127$);
99$          WAIT:          transporter#:NEXT(127$);
100$         WAIT:          transporter#:NEXT(127$);
101$         WAIT:          transporter#:NEXT(127$);
103$         WAIT:          transporter#:NEXT(127$);
105$         WAIT:          transporter#:NEXT(127$);
206$         ASSIGN:        sayac=1:
right_most_crane=number_of_cranes_at_each_berth(1);
207$         WHILE:         sayac<assigned_berth_of;
208$         ASSIGN:        right_most_crane=right_most_crane+number_of_cranes_at_each_berth(sayac+1):
                                sayac=sayac+1;
209$         ENDWHILE;
210$         ASSIGN:        sayac=1:
req_location=berth_length(assigned_berth_of)*assigned_berth_of:
                                gap=2;
211$         WHILE:
sayac<=number_of_cranes_at_each_berth(assigned_berth_of);
212$         REQUEST,       1:Crane(right_most_crane),,INTX(req_location);
213$         ASSIGN:        sayac=sayac+1:
                                last_requested(right_most_crane)=req_location;
214$         ASSIGN:        req_location=req_location-1;
215$         FREE:         Crane(right_most_crane);
217$         ASSIGN:        right_most_crane=right_most_crane-1;
216$         ENDWHILE:NEXT(17$);

17$          ASSIGN:        sayac=1:
                                sayac1=first_allocated(vesselno):
                                finish_time(vesselno)=tnow;
18$          WHILE:        sayac<=vessel_length#(vesselno);
19$          ASSIGN:        busy_time_each_node(sayac1)=
busy_time_each_node(sayac1)+finish_time(vesselno)-berthing_time(vesselno):
                                allocated(sayac1)=0:
                                sayac=sayac+1:
                                sayac1=sayac1+1;
20$          ENDWHILE;
29$          BRANCH,       1:
                                If,assigned_berth_of==1,23$,Yes:
                                If,assigned_berth_of==2,27$,Yes:
                                If,assigned_berth_of==3,28$,Yes:
                                If,assigned_berth_of==4,50$,Yes:
23$          SIGNAL:       assigned_berth_of;
15$          ASSIGN:       number_of_entity=number_of_entity+1:
                                time_in_system(vesselno)=finish_time(vesselno)-
ar(vesselno):
                                waiting_time(vesselno)=berthing_time(vesselno)-
ar(vesselno):NEXT(218$);

218$         ASSIGN:       sayac=vesselno:
                                sayac1=1;
226$         ASSIGN:       type=2:
                                rectangleno=rectangleno+1;

```

```

227$          WRITE,          file2,"%10f %10f %10f %10f %10f %10f %10f\n":
                                rectangleno,
                                type,
                                vesselno,
                                first_allocated(vesselno)-1,
                                berthing_time(vesselno),
                                vessel_length#(vesselno),
                                finish_time(vesselno)-berthing_time(vesselno);
219$          WHILE:          sayac1<=number_of_intersections;
220$          IF:
allocated_nodes_by_each_vessel(vesselno,sayac1)==1;
221$          ASSIGN:          rectangleno=rectangleno+1;
                                type=1;
222$          WRITE,          file2,"%10f %10f %10f %10f %10f %10f %10f\n":
                                rectangleno,
                                type,
                                allocated_nodes_by_each_crane(vesselno,sayac1),
                                sayac1-1,
                                crane_start_time_on_each_node(vesselno,sayac1),
                                1,
                                crane_finish_time_on_each_node(vesselno,sayac1)-
                                crane_start_time_on_each_node(vesselno,sayac1);
223$          ENDIF;
224$          ASSIGN:          sayac1=sayac1+1;
225$          ENDWHILE:NEXT(31$);

31$          IF:          number_of_entity==number_of_vessels;
32$          ASSIGN:          makespan=tnow;
34$          WRITE,          1,"%g ":
                                makespan;
35$          WRITE,          File3,"%d":
                                rectangleno:NEXT(198$);

198$          ASSIGN:          sayac=1;
199$          WHILE:          sayac<=number_of_berths;
200$          ASSIGN:          sayac1=first_node(sayac):
                                sayac2=1;
201$          WHILE:          sayac2<=berth_length(sayac);
202$          ASSIGN:          total_busy_time_of_berths(sayac)=total_busy_time_of_berths(sayac)+busy_time_
                                each_node(sayac1):
                                sayac1=sayac1+1;
                                sayac2=sayac2+1;
203$          ENDWHILE;
204$          ASSIGN:          utilization_of_berths(sayac)=total_busy_time_of_berths(sayac)/(berth_length(
                                sayac)*makespan):
                                sayac=sayac+1;
205$          ENDWHILE:NEXT(33$);

33$          ENDIF;
30$          DISPOSE:          No;

27$          SIGNAL:          assigned_berth_of:NEXT(15$);

28$          SIGNAL:          assigned_berth_of:NEXT(15$);

50$          SIGNAL:          assigned_berth_of:NEXT(15$);

25$          BRANCH,          1:

If,kontrolv(assigned_berth_of)<=task_assigned_to_berth(assigned_berth_of),22
$,Yes:
                                Else,8$,Yes;

```

```

22$          BRANCH,          1:
                                If,NQ(assigned_berth_of)>0,24$,Yes:
                                Else,10$,Yes;
8$           DISPOSE:         No;

11$          SCAN:
kontrol(cizelge(assigned_berth_of,kontrolv(assigned_berth_of)))==1:NEXT(24$)
;

12$          SCAN:
kontrol(cizelge(assigned_berth_of,kontrolv(assigned_berth_of)))==1:NEXT(24$)
;

51$          SCAN:
kontrol(cizelge(assigned_berth_of,kontrolv(assigned_berth_of)))==1:NEXT(24$)
;

14$          DISPOSE:         No;

36$          CREATE,          5,0.00001:,1:NEXT(0$);

0$           ASSIGN:          no=k:
                                k=k+1:
                                assigned_berth_of=assigned_berth(no):
                                picture=p1;
1$           DELAY:           ar(no),,Other:NEXT(3$);

3$           ASSIGN:          kontrol(no)=1;
2$           QUEUE,           assigned_berth_of:DETACH;

37$          CREATE,          1:,1:NEXT(38$);

38$          ASSIGN:          sayac=1:
                                sayacl=1:
                                picture=p2;
39$          WHILE:           sayac<=number_of_berths;
47$          ASSIGN:          sayacl=1;
40$          WHILE:           sayacl<=number_of_vessels;
45$          IF:              cizelge(sayac,sayacl)<>0;
41$          ASSIGN:          task_assigned_to_berth(sayac)=task_assigned_to_berth(sayac)+1:
                                assigned_berth(cizelge(sayac,sayacl))=sayac;
46$          ENDIF;
49$          ASSIGN:          sayacl=sayacl+1;
42$          ENDWHILE;
43$          ASSIGN:          sayac=sayac+1;
44$          ENDWHILE;
48$          DISPOSE:         No;

```

Exp. File of the ARENA model

```

PROJECT,      "Unnamed Project","Rockwell
Automation",,,No,Yes,Yes,Yes,No,No,No,No,No,No;
ATTRIBUTES:   1,list(120),DATATYPE(Real):
               121,numara,DATATYPE(Real):
               123,no,DATATYPE(Real):
               300,location,DATATYPE(Real):
               400,vesselno,DATATYPE(Real):
               available,DATATYPE(Real):
               right_idle_crane,DATATYPE(Real):
               task_location,DATATYPE(Real):

```



```

53,23,34,25,27,36,44:
kontrol_panel(40),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
busy_time_each_node(40),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
kontrol(10),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
number_of_entity,CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
vessel_length#,5,5,5,5,5:
crane_finish_time_on_each_node(10,40),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
number_of_intersections,CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real),40:
rectangleno,CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
utilization_of_berths(4),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
number,CLEAR(System),CATEGORY("None-None"),DATATYPE(Real):
crane_start_time_on_each_node(10,40),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
allocated_nodes_by_each_vessel(10,40),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
crane_panel(10),CLEAR(System),CATEGORY("NNone"),DATATYPE(Real):
number_of_cranes_at_each_berth(4),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real),3,2,3,2:
task_assigned_to_berth(4),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
sayac2,CLEAR(System),CATEGORY("None-None"),DATATYPE(Real):
requested(10),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
allocated_by(40),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real):
upper_crane(10),CLEAR(System),CATEGORY("None-
None"),DATATYPE(Real),8,9,10,19,20,28,29,30,39,40:gokce,CLEAR(System),CATEGO
RY("None-None"),DATATYPE(Real),1;
INTERSECTIONS: 1,Int1,1,FCFS(),1.0:
2,Int2,1,FCFS(),1.0:
3,Int3,1,FCFS(),1.0:
4,Int4,1,FCFS(),1.0:
5,Int5,1,FCFS(),1.0:
6,Int6,1,FCFS(),1.0:
7,Int7,1,FCFS(),1.0:
8,Int8,1,FCFS(),1.0:
9,Int9,1,FCFS(),1.0:
10,Int10,1,FCFS(),1.0:
11,Int11,1,FCFS(),1.0:
12,Int12,1,FCFS(),1.0:
13,Int13,1,FCFS(),1.0:
14,Int14,1,FCFS(),1.0:
15,Int15,1,FCFS(),1.0:
16,Int16,1,FCFS(),1.0:
17,Int17,1,FCFS(),1.0:
18,Int18,1,FCFS(),1.0:
19,Int19,1,FCFS(),1.0:
20,Int20,1,FCFS(),1.0:
21,Int21,1,FCFS(),1.0:
22,Int22,1,FCFS(),1.0:
23,Int23,1,FCFS(),1.0:
24,Int24,1,FCFS(),1.0:
25,Int25,1,FCFS(),1.0:
26,Int26,1,FCFS(),1.0:
27,Int27,1,FCFS(),1.0:
28,Int28,1,FCFS(),1.0:

```

29,Int29,1,FCFS(),1.0:
30,Int30,1,FCFS(),1.0:
31,Int31,1,FCFS(),1.0:
32,Int32,1,FCFS(),1.0:
33,Int33,1,FCFS(),1.0:
34,Int34,1,FCFS(),1.0:
35,Int35,1,FCFS(),1.0:
36,Int36,1,FCFS(),1.0:
37,Int37,1,FCFS(),1.0:
38,Int38,1,FCFS(),1.0:
39,Int39,1,FCFS(),1.0:
40,Int40,1,FCFS(),1.0:
41,Int41,1,FCFS(),1.0:
42,Int42,1,FCFS(),1.0:
43,Int43,1,FCFS(),1.0:
44,Int44,1,FCFS(),1.0:
45,Int45,1,FCFS(),1.0:
46,Int46,1,FCFS(),1.0:
47,Int47,1,FCFS(),1.0:
48,Int48,1,FCFS(),1.0:
49,Int49,1,FCFS(),1.0:
50,Int50,1,FCFS(),1.0:
51,Int51,1,FCFS(),1.0:
52,Int52,1,FCFS(),1.0:
53,Int53,1,FCFS(),1.0:
54,Int54,1,FCFS(),1.0:
55,Int55,1,FCFS(),1.0:
56,Int56,1,FCFS(),1.0:
57,Int57,1,FCFS(),1.0:
58,Int58,1,FCFS(),1.0:
59,Int59,1,FCFS(),1.0:
60,Int60,1,FCFS(),1.0:
61,Int61,1,FCFS(),1.0:
62,Int62,1,FCFS(),1.0:
63,Int63,1,FCFS(),1.0:
64,Int64,1,FCFS(),1.0:
65,Int65,1,FCFS(),1.0:
66,Int66,1,FCFS(),1.0:
67,Int67,1,FCFS(),1.0:
68,Int68,1,FCFS(),1.0:
69,Int69,1,FCFS(),1.0:
70,Int70,1,FCFS(),1.0:
71,Int71,1,FCFS(),1.0:
72,Int72,1,FCFS(),1.0:
73,Int73,1,FCFS(),1.0:
74,Int74,1,FCFS(),1.0:
75,Int75,1,FCFS(),1.0:
76,Int76,1,FCFS(),1.0:
77,Int77,1,FCFS(),1.0:
78,Int78,1,FCFS(),1.0:
79,Int79,1,FCFS(),1.0:
80,Int80,1,FCFS(),1.0:
81,Int81,1,FCFS(),1.0:
82,Int82,1,FCFS(),1.0:
83,Int83,1,FCFS(),1.0:
84,Int84,1,FCFS(),1.0:
85,Int85,1,FCFS(),1.0:
86,Int86,1,FCFS(),1.0:
87,Int87,1,FCFS(),1.0:
88,Int88,1,FCFS(),1.0:
89,Int89,1,FCFS(),1.0:
90,Int90,1,FCFS(),1.0:
91,Int91,1,FCFS(),1.0:
92,Int92,1,FCFS(),1.0:
93,Int93,1,FCFS(),1.0:

```

94,Int94,1,FCFS(),1.0:
95,Int95,1,FCFS(),1.0:
96,Int96,1,FCFS(),1.0:
97,Int97,1,FCFS(),1.0:
98,Int98,1,FCFS(),1.0:
99,Int99,1,FCFS(),1.0:
100,Int100,1,FCFS(),1.0:
101,Int101,1,FCFS(),1.0:
102,Int102,1,FCFS(),1.0:
103,Int103,1,FCFS(),1.0:
104,Int104,1,FCFS(),1.0:
105,Int105,1,FCFS(),1.0:
106,Int106,1,FCFS(),1.0:
107,Int107,1,FCFS(),1.0:
108,Int108,1,FCFS(),1.0:
109,Int109,1,FCFS(),1.0:
110,Int110,1,FCFS(),1.0:
111,Int111,1,FCFS(),1.0:
112,Int112,1,FCFS(),1.0:
113,Int113,1,FCFS(),1.0:
114,Int114,1,FCFS(),1.0:
115,Int115,1,FCFS(),1.0:
116,Int116,1,FCFS(),1.0:
117,Int117,1,FCFS(),1.0:
118,Int118,1,FCFS(),1.0:
119,Int119,1,FCFS(),1.0:
120,Int120,1,FCFS(),1.0;
QUEUES: 1,q1,FirstInFirstOut,,AUTOSTATS(Yes,,):
2,q2,FirstInFirstOut,,AUTOSTATS(Yes,,):
3,q3,FirstInFirstOut,,AUTOSTATS(Yes,,):
4,q4,FirstInFirstOut,,AUTOSTATS(Yes,,):
5,shuffleg,LowValueFirst(random),,AUTOSTATS(Yes,,):
reqq,FirstInFirstOut,,AUTOSTATS(Yes,,):
craneq,FirstInFirstOut,,AUTOSTATS(Yes,,);
PICTURES: 1,p1:
2,p2;
LINKS: 1,Link1,Int1-0,Int2-,1,1,Bidirectional,1.0:
2,Link2,Int2-0,Int3-,1,1,Bidirectional,1.0:
3,Link3,Int3-0,Int4-,1,1,Bidirectional,1.0:
4,Link4,Int4-0,Int5-,1,1,Bidirectional,1.0:
5,Link5,Int5-0,Int6-,1,1,Bidirectional,1.0:
6,Link6,Int6-0,Int7-,1,1,Bidirectional,1.0:
7,Link7,Int7-0,Int8-,1,1,Bidirectional,1.0:
8,Link8,Int8-0,Int9-,1,1,Bidirectional,1.0:
9,Link9,Int9-0,Int10-,1,1,Bidirectional,1.0:
10,Link10,Int10-0,Int11-,1,1,Bidirectional,1.0:
11,Link11,Int11-0,Int12-,1,1,Bidirectional,1.0:
12,Link12,Int12-0,Int13-,1,1,Bidirectional,1.0:
13,Link13,Int13-0,Int14-,1,1,Bidirectional,1.0:
14,Link14,Int14-0,Int15-,1,1,Bidirectional,1.0:
15,Link15,Int15-0,Int16-,1,1,Bidirectional,1.0:
16,Link16,Int16-0,Int17-,1,1,Bidirectional,1.0:
17,Link17,Int17-0,Int18-,1,1,Bidirectional,1.0:
18,Link18,Int18-0,Int19-,1,1,Bidirectional,1.0:
19,Link19,Int19-0,Int20-,1,1,Bidirectional,1.0:
20,Link20,Int20-0,Int21-,1,1,Bidirectional,1.0:
21,Link21,Int21-0,Int22-,1,1,Bidirectional,1.0:
22,Link22,Int22-0,Int23-,1,1,Bidirectional,1.0:
23,Link23,Int23-0,Int24-,1,1,Bidirectional,1.0:
24,Link24,Int24-0,Int25-,1,1,Bidirectional,1.0:
25,Link25,Int25-0,Int26-,1,1,Bidirectional,1.0:
26,Link26,Int26-0,Int27-,1,1,Bidirectional,1.0:
27,Link27,Int27-0,Int28-,1,1,Bidirectional,1.0:
28,Link28,Int28-0,Int29-,1,1,Bidirectional,1.0:
29,Link29,Int29-0,Int30-,1,1,Bidirectional,1.0:

```

30,Link30,Int30-0,Int31-,1,1,Bidirectional,1.0:
31,Link31,Int31-0,Int32-,1,1,Bidirectional,1.0:
32,Link32,Int32-0,Int33-,1,1,Bidirectional,1.0:
33,Link33,Int33-0,Int34-,1,1,Bidirectional,1.0:
34,Link34,Int34-0,Int35-,1,1,Bidirectional,1.0:
35,Link35,Int35-0,Int36-,1,1,Bidirectional,1.0:
36,Link36,Int36-0,Int37-,1,1,Bidirectional,1.0:
37,Link37,Int37-0,Int38-,1,1,Bidirectional,1.0:
38,Link38,Int38-0,Int39-,1,1,Bidirectional,1.0:
39,Link39,Int39-0,Int40-,1,1,Bidirectional,1.0:
40,Link40,Int40-0,Int41-,1,1,Bidirectional,1.0:
41,Link41,Int41-0,Int42-,1,1,Bidirectional,1.0:
42,Link42,Int42-0,Int43-,1,1,Bidirectional,1.0:
43,Link43,Int43-0,Int44-,1,1,Bidirectional,1.0:
44,Link44,Int44-0,Int45-,1,1,Bidirectional,1.0:
45,Link45,Int45-0,Int46-,1,1,Bidirectional,1.0:
46,Link46,Int46-0,Int47-,1,1,Bidirectional,1.0:
47,Link47,Int47-0,Int48-,1,1,Bidirectional,1.0:
48,Link48,Int48-0,Int49-,1,1,Bidirectional,1.0:
49,Link49,Int49-0,Int50-,1,1,Bidirectional,1.0:
50,Link50,Int50-0,Int51-,1,1,Bidirectional,1.0:
51,Link51,Int51-0,Int52-,1,1,Bidirectional,1.0:
52,Link52,Int52-0,Int53-,1,1,Bidirectional,1.0:
53,Link53,Int53-0,Int54-,1,1,Bidirectional,1.0:
54,Link54,Int54-0,Int55-,1,1,Bidirectional,1.0:
55,Link55,Int55-0,Int56-,1,1,Bidirectional,1.0:
56,Link56,Int56-0,Int57-,1,1,Bidirectional,1.0:
57,Link57,Int57-0,Int58-,1,1,Bidirectional,1.0:
58,Link58,Int58-0,Int59-,1,1,Bidirectional,1.0:
59,Link59,Int59-0,Int60-,1,1,Bidirectional,1.0:
60,Link60,Int60-0,Int61-,1,1,Bidirectional,1.0:
61,Link61,Int61-0,Int62-,1,1,Bidirectional,1.0:
62,Link62,Int62-0,Int63-,1,1,Bidirectional,1.0:
63,Link63,Int63-0,Int64-,1,1,Bidirectional,1.0:
64,Link64,Int64-0,Int65-,1,1,Bidirectional,1.0:
65,Link65,Int65-0,Int66-,1,1,Bidirectional,1.0:
66,Link66,Int66-0,Int67-,1,1,Bidirectional,1.0:
67,Link67,Int67-0,Int68-,1,1,Bidirectional,1.0:
68,Link68,Int68-0,Int69-,1,1,Bidirectional,1.0:
69,Link69,Int69-0,Int70-,1,1,Bidirectional,1.0:
70,Link70,Int70-0,Int71-,1,1,Bidirectional,1.0:
71,Link71,Int71-0,Int72-,1,1,Bidirectional,1.0:
72,Link72,Int72-0,Int73-,1,1,Bidirectional,1.0:
73,Link73,Int73-0,Int74-,1,1,Bidirectional,1.0:
74,Link74,Int74-0,Int75-,1,1,Bidirectional,1.0:
75,Link75,Int75-0,Int76-,1,1,Bidirectional,1.0:
76,Link76,Int76-0,Int77-,1,1,Bidirectional,1.0:
77,Link77,Int77-0,Int78-,1,1,Bidirectional,1.0:
78,Link78,Int78-0,Int79-,1,1,Bidirectional,1.0:
79,Link79,Int79-0,Int80-,1,1,Bidirectional,1.0:
80,Link80,Int80-0,Int81-,1,1,Bidirectional,1.0:
81,Link81,Int81-0,Int82-,1,1,Bidirectional,1.0:
82,Link82,Int82-0,Int83-,1,1,Bidirectional,1.0:
83,Link83,Int83-0,Int84-,1,1,Bidirectional,1.0:
84,Link84,Int84-0,Int85-,1,1,Bidirectional,1.0:
85,Link85,Int85-0,Int86-,1,1,Bidirectional,1.0:
86,Link86,Int86-0,Int87-,1,1,Bidirectional,1.0:
87,Link87,Int87-0,Int88-,1,1,Bidirectional,1.0:
88,Link88,Int88-0,Int89-,1,1,Bidirectional,1.0:
89,Link89,Int89-0,Int90-,1,1,Bidirectional,1.0:
90,Link90,Int90-0,Int91-,1,1,Bidirectional,1.0:
91,Link91,Int91-0,Int92-,1,1,Bidirectional,1.0:
92,Link92,Int92-0,Int93-,1,1,Bidirectional,1.0:
93,Link93,Int93-0,Int94-,1,1,Bidirectional,1.0:
94,Link94,Int94-0,Int95-,1,1,Bidirectional,1.0:

Appendix A2: MATLAB Code of the SA based Simulation Optimization Procedure

```
tic;
t=cputime;
global Solution;
global task_assigned_berth;
global assigned_berth;
global objective_value;
number_of_replications=1;
number_of_iterations=100;
number_of_vessels=10;
number_of_berths=4;
objective_value=0;
objective_value_k=0;
objective_value_best=0;
objective_value_Sc=0;
best_of_all_values=0;
X=number_of_vessels;
S=zeros(1,number_of_vessels*number_of_berths);
Sc=zeros(1,number_of_vessels*number_of_berths);
Sbest=zeros(1,number_of_vessels*number_of_berths);
Dummy=zeros(1,number_of_vessels*number_of_berths);
starttime=zeros(1,number_of_berths);
number_of_vessels_allocated_each_berth=zeros(1,number_of_vessels);
wharf_of_vessel=[10 14 12 10 10 10 14 12 10 10];
depth_of_berth=[17 17 17 17];
length_of_vessel=[9 9 9 9 9 9 9 9 9 9];
length_of_berth=[10 10 10 10];
cooling_rate=0.9;
tasksq=[1 2 3 4 5 6 7 8 9];
tasksq=tasksq(randperm(length(tasksq)));

for counter1=1:number_of_replications
    total_number_of_iterations=0;
    improvement=0;
    K=300;
    B=K;
    initial solution;
    Sk=S;
    Sbest=Sk;
    Q=zeros(number_of_berths,number_of_vessels);
    assigned_berth=zeros(1,number_of_vessels);
    task_assigned_berth=zeros(1,number_of_berths);
    for i=1:number_of_berths
        for j=(i-1)*number_of_vessels+1:i*number_of_vessels
            if Sk(j)~=0
                m=mod(j,number_of_vessels);
                if m==0
                    Q(i,Sk(j))=number_of_vessels;
                else
                    Q(i,Sk(j))=m;
                end
            end
        end
    end
end
for i=1:number_of_berths
```

```

        for j=1:number_of_vessels
            if Q(i,j)~=0
                assigned_berth(1,Q(i,j))=i;
            end
        end
    end
    for i=1:number_of_berths
        for j=1:number_of_vessels
            if Q(i,j)~=0
                task_assigned_berth(i)=task_assigned_berth(i)+1;
            end
        end
    end
    end
    Solution=(Q(:))';
    RunSim(Solution,task_assigned_berth,assigned_berth);
    objective_value_best=objective_value;
    objective_value_k=objective_value;
    while 1<K
        for counter2=1:number_of_iterations
            if counter2==100
                gokce=1;
            end
        end
        total_number_of_iterations=total_number_of_iterations+1;
        S=Sk;
        u=rand(1,1);
        if u<0.5
            transferring of vessel;
        else
            swapping;
        end
        Sc=S;
        Q=zeros(number_of_berths,number_of_vessels);
        assigned_berth=zeros(1,number_of_vessels);
        task_assigned_berth=zeros(1,number_of_berths);
        for i=1:number_of_berths
            for j=(i-1)*number_of_vessels+1:i*number_of_vessels
                if Sc(j)~=0
                    m=mod(j,number_of_vessels);
                    if m==0
                        Q(i,Sc(j))=number_of_vessels;
                    else
                        Q(i,Sc(j))=m;
                    end
                end
            end
        end
    end
    for i=1:number_of_berths
        for j=1:number_of_vessels
            if Q(i,j)~=0
                assigned_berth(1,Q(i,j))=i;
            end
        end
    end
    end
    for i=1:number_of_berths
        for j=1:number_of_vessels
            if Q(i,j)~=0
                task_assigned_berth(i)=task_assigned_berth(i)+1;
            end
        end
    end
end

```

```

end
Solution=(Q(:))';
RunSim(Solution,task_assigned_berth,assigned_berth);
objective_value_Sc=objective_value;
if
(objective_value_best<=objective_value_Sc)&&(objective_value_Sc<=obj
ective_value_k)
Sk=Sc;
objective_value_k=objective_value_Sc;
end
if objective_value_Sc<objective_value_best
Sbest=Sc;
openedFile1=fopen('C:\tez6\schedule.txt','r'); %open to read
scannedFile1=fscanf(openedFile1,'%c');
fclose(openedFile1);
openedbestschedulefile=fopen('C:\tez6\bestschedule.txt','w');
fprintf(openedbestschedulefile,'%c',scannedFile1);
fclose(openedbestschedulefile);
openedFile2=fopen('C:\tez6\rectanglenunder.txt','r'); %open to read
scannedFile2=fscanf(openedFile1,'%c');
fclose(openedFile2);
openedbestrectno=fopen('C:\tez6\bestrectno.txt','w');
fprintf(openedbestrectno,'%c',scannedFile2);
fclose(openedbestrectno);
objective_value_best=objective_value_Sc;
Sk=Sc;
improvement=improvement+1;
last_improvement=total_number_of_iterations;
end
if objective_value_k<=objective_value_Sc
u=rand(1,1);
if u<=exp(-(objective_value_k-objective_value_Sc)/B)
Sk=Sc;
end
end
total_iter(total_number_of_iterations)=total_number_of_iterations;
obj_values(total_number_of_iterations)=objective_value_best;
obj_values_optimum(total_number_of_iterations)=279;
end
K=K*cooling_rate;
B=K;
end
if objective_value_best<=best_of_all_values
best_of_all=Sbest;
end
iterationtotal(counter1)=total_number_of_iterations;
bestvalues(counter1)=objective_value_best;
bestsolutions(counter1,:)=Sbest;
end
figure
plot(total_iter,obj_values,total_iter,obj_values_optimum,'LineWidth'
,1,'Color','b');
ylim([0,5000]);
Qbest=zeros(number_of_berths,number_of_vessels);
assigned_berth=zeros(1,number_of_vessels);
task_assigned_berth=zeros(1,number_of_berths);
for i=1:number_of_berths
for j=(i-1)*number_of_vessels+1:i*number_of_vessels
if Sbest(j)~=0

```

```

        m=mod(j,number_of_vessels);
        if m==0
            Qbest(i,Sbest(j))=number_of_vessels;
        else
            Qbest(i,Sbest(j))=m;
        end

    end

end

for i=1:number_of_berths
    for j=1:number_of_vessels
        if Qbest(i,j)~=0
            assigned_berth(1,Qbest(i,j))=i;
        end
    end
end

Qbest=zeros(number_of_berths,number_of_vessels);
assigned_berth=zeros(1,number_of_vessels);
task_assigned_berth=zeros(1,number_of_berths);
for i=1:number_of_berths
    for j=(i-1)*number_of_vessels+1:i*number_of_vessels
        if Sbest(j)~=0
            m=mod(j,number_of_vessels);
            if m==0
                Qbest(i,Sbest(j))=number_of_vessels;
            else
                Qbest(i,Sbest(j))=m;
            end
        end
    end
end

for i=1:number_of_berths
    for j=1:number_of_vessels
        if Qbest(i,j)~=0
            assigned_berth(1,Qbest(i,j))=i;
        end
    end
end

fileID10=fopen('C:\tez6\bestschedule.txt','r');
fileID11=fopen('C:\tez6\bestrectno.txt','r');
formatSpec='%f';
parameters=7;
number_of_rectangles=60;
number_of_iterations=1;
sizonuc=[parameters number_of_rectangles*number_of_iterations];
length_of_vessel=[10 10 10 10 10 10 10 10 10 10];
sonuc=(fscanf(fileID10,formatSpec,sizonuc));
figure
set(gca,'YLim',[0 40])
set(gca,'XLim',[0 280])
xlabel('Time');
ylabel('Quayside');
cizelgeno=1;

```

```

for i=(cizelgeno-
1)*number_of_rectangles+1:number_of_rectangles*(cizelgeno-
1)+number_of_rectangles
    if sonuc(i,2)==1
        cranelabel=strcat('C',num2str(sonuc(i,3)));

rectangle('position',[sonuc(i,5),sonuc(i,4),sonuc(i,7),sonuc(i,6)], '
FaceColor',[1,1,1])

text(sonuc(i,5)+0.5,sonuc(i,4)+(sonuc(i,6)/2),cranelabel,'FontSize',
8);
    else

rectangle('position',[sonuc(i,5),sonuc(i,4),sonuc(i,7),sonuc(i,6)], '
FaceColor',rand(1,3))
    end

end
y=[10 20 30 40];
for i=1:length(y)
hline = reffline([0 y(i)]);
hold on
end
text(210,5, 'Berth 1');
text(210,15, 'Berth 2');
text(210,25, 'Berth 3');
text(210,35, 'Berth 4');

```

RunSim:

```

function RunSim(Solution1,task_assigned_berth1,assigned_berth1)
global objective_value
global collectData
%% Data
s1=sprintf('%d,',Solution1);
string1=s1(1:length(s1)-1); %first string to be changed
s2=sprintf('%d,',task_assigned_berth1); %second string to be changed
string2=s2(1:length(s2)-1);
s3=sprintf('%d,',assigned_berth1);
string3=s3(1:length(s3)-1);

%% Open and Read
openedFile=fopen('temp_expfile.txt','r'); %open to read
scannedFile=fscanf(openedFile,'%c');
fclose(openedFile);
%% Change values and Write .exp file
% Change values
changedText1=strrep(scannedFile,'changel',string1); %make the first
change
openedRealExp=fopen('C:\tez6\MQBAQCSP.exp','w'); %open to write data
fprintf(openedRealExp,'%c',changedText1);
fclose(openedRealExp);
%% Run the Simulation
dos('C:\tez6\calistirg.bat');
%% Post Processing
fileID = fopen('C:\tez6\result.txt');
copyResults = textscan(fileID,'%f');
results=copyResults{1};

```

```

objective_value=mean(results);
collectData=[collectData; results];
fclose(fileID);
end

```

Initial Solution

```

S=zeros(1,number_of_vessels*number_of_berths);
number_of_vessels_allocated_each_berth=zeros(1,number_of_berths);

for i=1:number_of_vessels
    ok=0;
    while ok==0
        a=randi([1,number_of_berths],1);
        if
            wharf_of_vessel(i)<=depth_of_berth(a)&&(length_of_vessel(i)+1)<=length_of_berth(a)

            number_of_vessels_allocated_each_berth(a)=number_of_vessels_allocated_each_berth(a)+1;
            S((a-1)*number_of_vessels+i)=number_of_vessels_allocated_each_berth(a);
            ok=1;
        end
    end
end
end

```

Swapping

```

S=Sk;
changed=0;
done=0;
while done==0
    Perm=randperm(number_of_vessels);
    a=Perm(1);
    b=Perm(2);
    v1=a;
    v2=b;
    for i=1:number_of_berths
        if S(a)~=0
            k1=S(a);
            a1=a;
            berth1=i;
        end
        a=a+number_of_vessels;
    end
    for i=1:number_of_berths
        if S(b)~=0
            k2=S(b);
            b1=b;
            berth2=i;
        end
        b=b+number_of_vessels;
    end
    if
        wharf_of_vessel(v1)<=depth_of_berth(berth2)&&(length_of_vessel(v1)+1)<=length_of_berth(berth2)

```

```

        if
wharf_of_vessel(v2)<=depth_of_berth(berth1)&&(length_of_vessel(v2)+1
)<=length_of_berth(berth1)
            done=done+1;
        end
    end
end
S(a1)=0;
S(b1)=0;
        S(v2+(number_of_vessels)*(berth1-1))=k1;
        S(v1+(number_of_vessels)*(berth2-1))=k2;

```

Transferring of Vessel

```

X=number_of_vessels;
    changed=0;
while changed==0
    X=number_of_vessels;
    vessels_for_each_berth=zeros(1,number_of_berths);
    a=randi([1,number_of_vessels],1,1);
    for t=1:number_of_berths
        if S(number_of_vessels*(t-1)+a)~=0
            b=t;
        end
    end
    c=b;
    while b==c
        c=randi([1,number_of_berths],1,1);
    end
    if
wharf_of_vessel(a)<=depth_of_berth(c)&&(length_of_vessel(a)+1)<=length_of_berth(c)
        for t=1:number_of_berths
            for p=1:number_of_vessels
                if S(number_of_vessels*(t-1)+p)~=0
                    vessels_for_each_berth(t)=vessels_for_each_berth(t)+1;
                end
            end
        end
        sequence=randi([1,vessels_for_each_berth(c)+1],1,1);
        for i=1:number_of_vessels
            if S(number_of_vessels*(b-1)+i)>S(number_of_vessels*(b-1)+a)
                S(number_of_vessels*(b-1)+i)=S(number_of_vessels*(b-1)+i)-1;
            end
        end
        for i=1:number_of_vessels
            if S(number_of_vessels*(c-1)+i)>=sequence
                S(number_of_vessels*(c-1)+i)=S(number_of_vessels*(c-1)+i)+1;
            end
        end
        S(number_of_vessels*(b-1)+a)=0;
        S(number_of_vessels*(c-1)+a)=sequence;
        changed=1;
    end
end
end

```

Appendix A3: Lingo Model of the MIP

Lingo Model of the first part of the MIP

```
SETS:
Q/1..2/;
S/1..2/:A,C,Y;
sq(S,Q):H,X;
ss(S,S):Z;

ENDSETS

DATA:

H=5 6
   3 5;

A=10 12;

M=1000000;

ENDDATA

MIN=zz;
@FOR(S(s1):C(s1)<=zz);
@FOR(S(s1):@SUM(Q(q1):X(s1,q1))=1);
@FOR(S(s1):C(s1)=Y(s1)+@SUM(Q(q1):H(s1,q1)*X(s1,q1)));
@FOR(S(s1):@FOR(S(s2)|s1#NE#s2:C(s1)-Y(s2)+Z(s1,s2)*M>0));
@FOR(S(s1):@FOR(S(s2)|s1#NE#s2:C(s1)-Y(s2)-(1-Z(s1,s2))*M<=0));
@FOR(Q(q1):@FOR(S(s1):@FOR(S(s2)|s1#NE#s2:M*((X(s1,q1)-1)+(X(s2,q1)-1))<=Z(s1,s2)+Z(s2,s1)-1)));
@FOR(S(s1):Y(s1)>=A(s1));
@FOR(sq(s1,q1):@BIN(X));
@FOR(ss(s1,s2):@BIN(Z));
@FOR(S(s1):@GIN(Y));

END
```

Lingo Model of the second part of the MIP

```
SETS:
Q/1..1/:K;
S/1..1/:B;
bays/1..12/;
cranes/1..3/;
sbqk(S,bays,Q,cranes):X;
sbbq(S,bays,bays,Q):Y;
sbq(S,bays,Q):C;
sq(S,Q):H;
sb(S,bays):P;

ENDSETS

DATA:
M=1000;
K=3;
B=12;
P=3 4 5 6 3 3 4 5 6 3 6 5;

ENDDATA
```

```

MIN=zz;
@FOR(S(s1):@FOR(Q(q1):H(s1,q1)<=zz));
@FOR(Q(q1):@FOR(sb(s1,b1)|b1#LE#B(s1):H(s1,q1)>=C(s1,b1,q1)));
@FOR(Q(q1):@FOR(sb(s1,b1)|b1#LE#B(s1):C(s1,b1,q1)>=P(s1,b1)));
@FOR(S(s1):@FOR(Q(q1):@FOR(bays(b1)|b1#LE#B(s1):@SUM(cranes(c1)|c1#L
E#K(q1):X(s1,b1,q1,c1)=1)));
@FOR(S(s1):@FOR(Q(q1):@FOR(bays(b1)|b1#LE#B(s1):@FOR(bays(b2)|b2#LE
#B(s1))#AND#(b2#NE#b1):C(s1,b1,q1)-(C(s1,b2,q1)-
P(s1,b2))+Y(s1,b1,b2,q1)*M>0)));
@FOR(S(s1):@FOR(Q(q1):@FOR(bays(b1)|b1#LE#B(s1):@FOR(bays(b2)|b2#LE
#B(s1))#AND#(b2#NE#b1):C(s1,b1,q1)-(C(s1,b2,q1)-P(s1,b2))-(1-
Y(s1,b1,b2,q1))*M<=0)));
@FOR(S(s1):@FOR(Q(q1):@FOR(bays(b1)|b1#LE#B(s1):@FOR(bays(b2)|b2#LE
#B(s1))#AND#(b2#GT#b1):M*(Y(s1,b1,b2,q1)+Y(s1,b2,b1,q1))>=@SUM(crane
s(c1)|c1#LE#K(q1):c1*X(s1,b1,q1,c1))-
@SUM(cranes(c1)|c1#LE#K(q1):(c1)*X(s1,b2,q1,c1)+1))));
@FOR(sbbq(s1,b1,q1,c1):@BIN(X));
@FOR(S(s1):@FOR(Q(q1):@FOR(bays(b1)|b1#LE#B(s1):@FOR(cranes(c1)|c1#L
E#K(q1):@BIN(X(s1,b1,q1,c1)))));
@FOR(sbbq(s1,b1,b2,q1):@BIN(Y));
END

```

Appendix A4: MATLAB code used for drawing berth space-time diagram

```
fileID10=fopen('C:\tez6\schedule.txt','r');
fileID11=fopen('C:\tez6\bestrectno.txt','r');
formatSpec='%f';
parameters=7;
number_of_rectangles=97;
number_of_iterations=1;
sizenuc=[parameters number_of_rectangles*number_of_iterations];
length_of_vessel=[10 10 10 10 10 10 10 10 10 10];
sonuc=(fscanf(fileID10,formatSpec,sizenuc));
figure
set(gca,'YLim',[0 40])
set(gca,'XLim',[0 450])
xlabel('Time');
ylabel('Quayside');
cizelgeno=1;
for i=(cizelgeno-
1)*number_of_rectangles+1:number_of_rectangles*(cizelgeno-
1)+number_of_rectangles
    if sonuc(i,2)==1
        cranelabel=strcat('C',num2str(sonuc(i,3)));

        rectangle('position',[sonuc(i,5),sonuc(i,4),sonuc(i,7),sonuc(i,6)],'
        FaceColor',[1,1,1])

        text(sonuc(i,5)+0.5,sonuc(i,4)+(sonuc(i,6)/2),cranelabel,'FontSize',
        10);
    else

        rectangle('position',[sonuc(i,5),sonuc(i,4),sonuc(i,7),sonuc(i,6)],'
        FaceColor',rand(1,3))
    end

end

y=[10 20 30 40];
for i=1:length(y)
    hline = reffline([0 y(i)]);
    hold on
end
text(420,5,'Berth 1');
text(420,15,'Berth 2');
text(420,25,'Berth 3');
text(420,35,'Berth 4');
```

Appendix A5: Input Data and Validation Results of 10 randomly generated Problem Instances

Table A.5.1 The input data for Problem Instance 1

Vessel No	Vessel Length	Arrival Time	The Handling Times of Each Ship-Bay in Each Vessel									
			1	2	3	4	5	6	7	8	9	10
1	5	0	23	24	25	22	34					
2	5	24	24	24	36	37	25					
3	5	35	50	45	44	37	27					
4	5	22	43	33	23	53	36					
5	5	57	25	34	35	23	44					

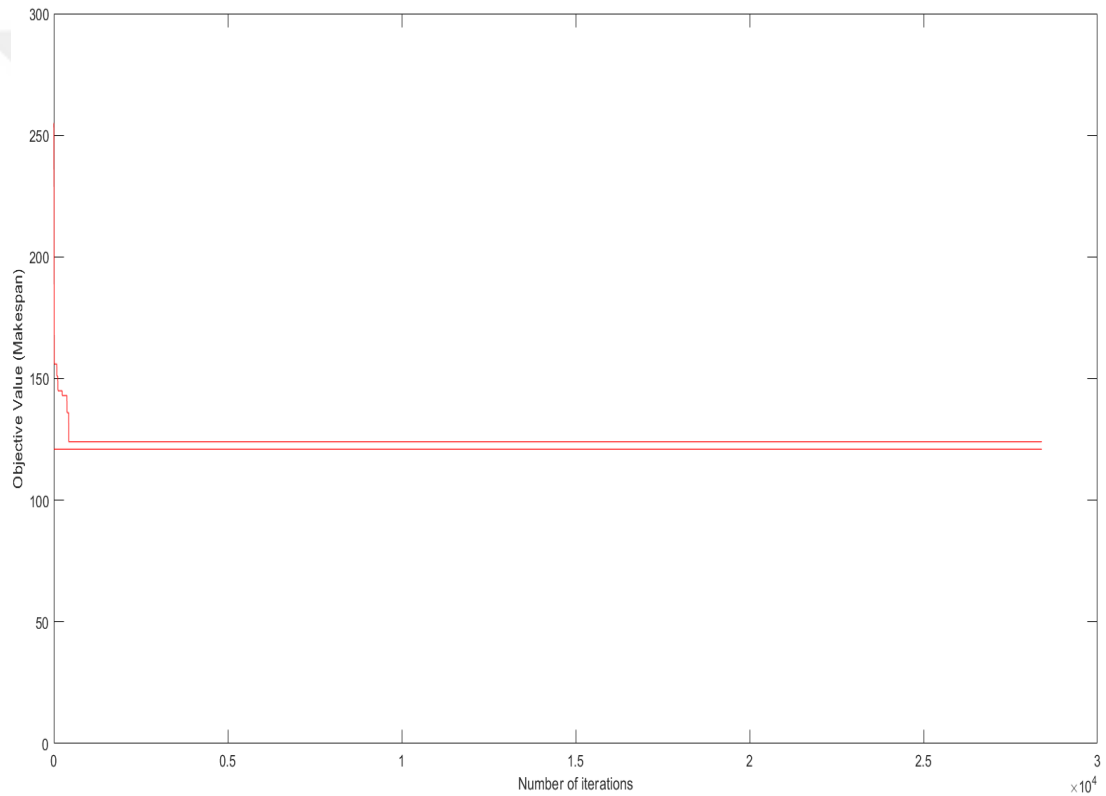


Figure A5.1. Convergence graph of the proposed approach for Problem Instance 1

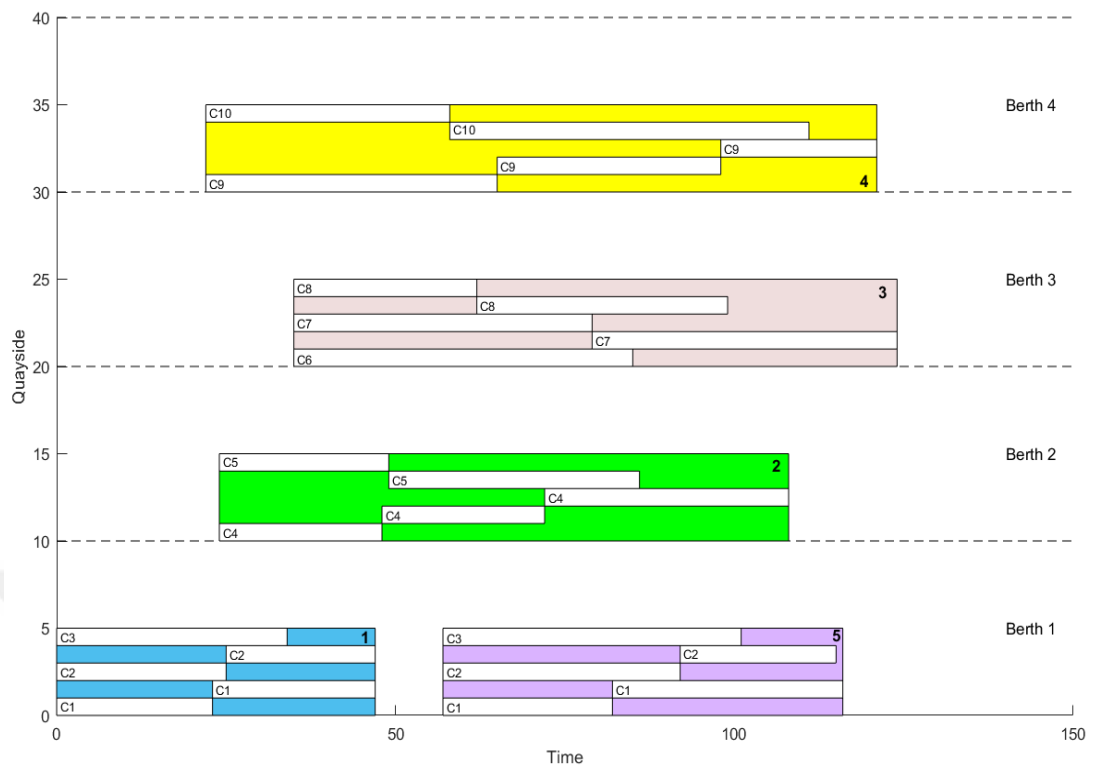


Figure A5.2 Berth space-time diagram for the optimum solution of Problem Instance 1

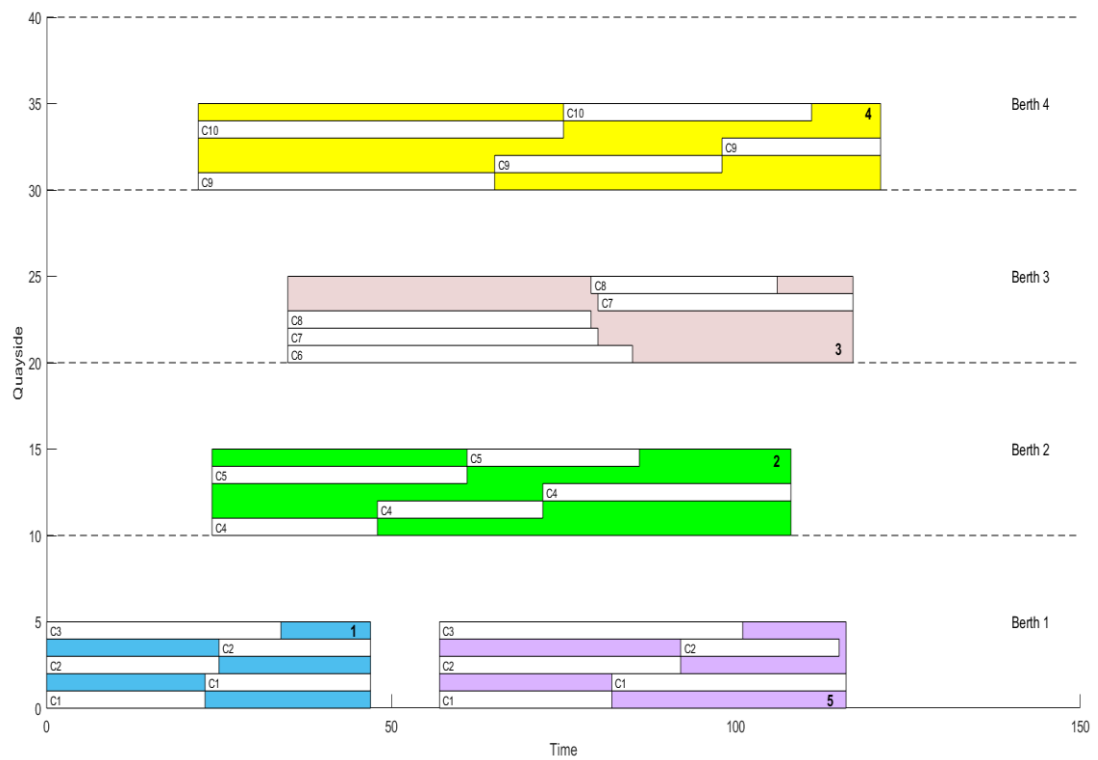


Figure A5.3 Berth space-time diagram for the proposed approach of Problem Instance 1

Table A.5.2 The input data for Problem Instance 2

Vessel No	Vessel Length	Arrival Time	The Handling Times of Each Ship-Bay in Each Vessel									
			1	2	3	4	5	6	7	8	9	10
1	10	0	23	24	25	22	34	31	29	40	35	50
2	10	24	24	24	36	37	25	28	32	25	27	38
3	10	35	50	45	44	37	27	25	28	25	27	26
4	10	22	43	33	23	53	36	24	27	29	31	32
5	10	57	25	34	35	23	44	34	26	28	42	32

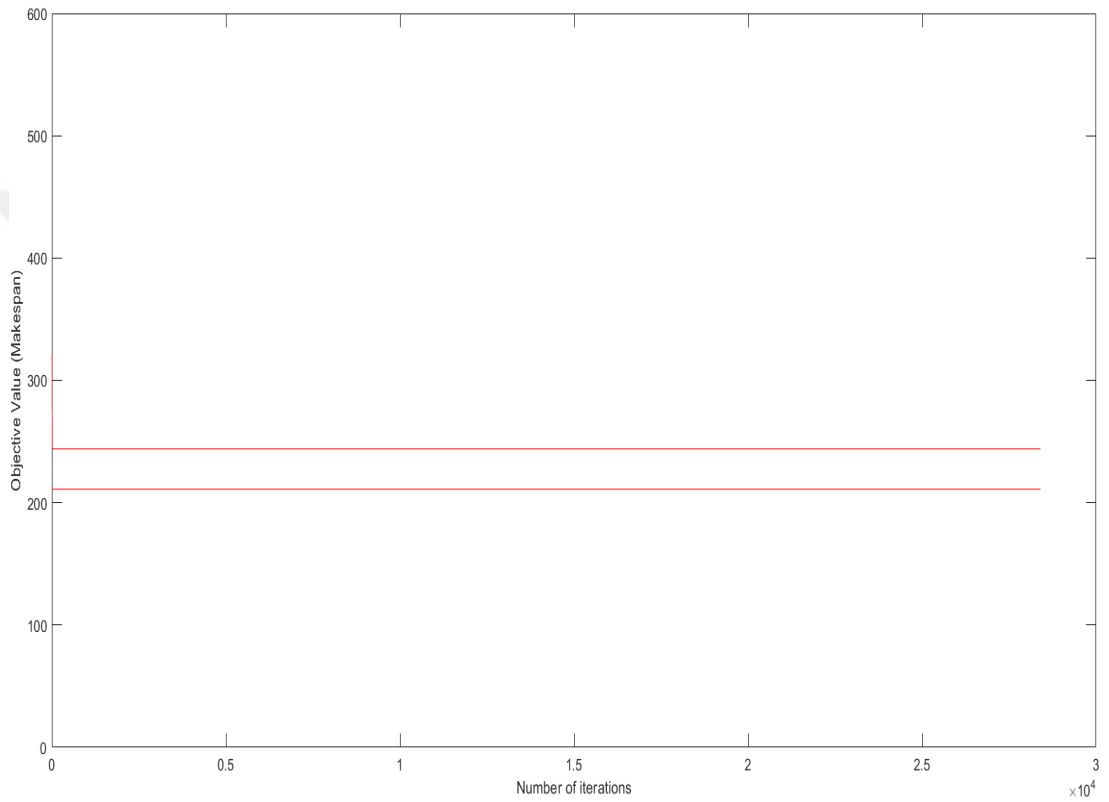


Figure A5.4. Convergence graph of the proposed approach for Problem Instance 2

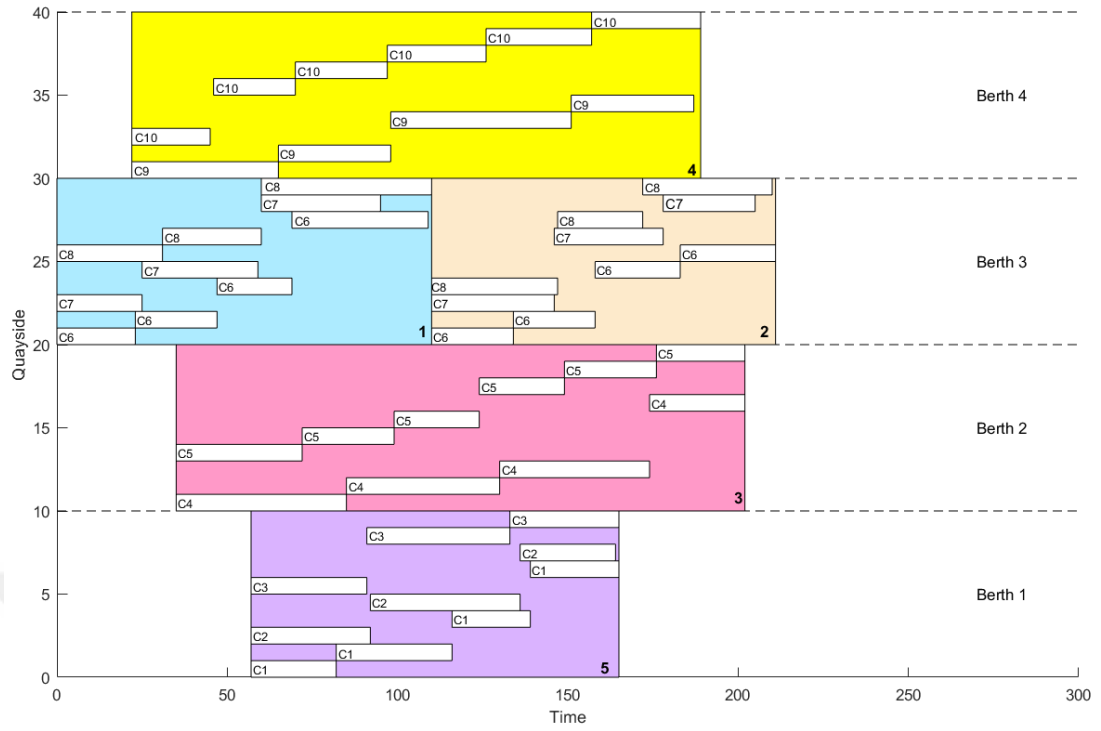


Figure A5.5 Berth space-time diagram for the optimum solution of Problem Instance 2

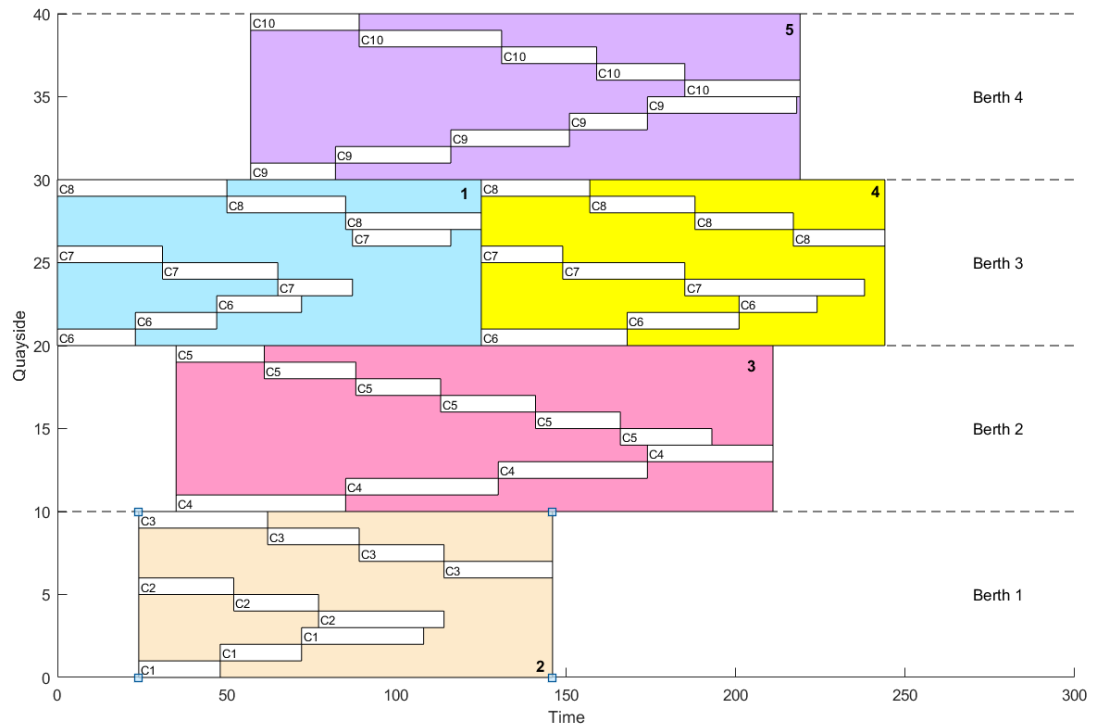


Figure A5.6 Berth space-time diagram for the proposed approach of Problem Instance 2

Table A.5.3 The input data for Problem Instance 3

Vessel No	Vessel Length	Arrival Time	The Handling Times of Each Ship-Bay in Each Vessel									
			1	2	3	4	5	6	7	8	9	10
1	5	0	23	24	25	22	34					
2	5	24	24	24	36	37	25					
3	5	35	50	45	44	37	27					
4	5	22	43	33	23	53	36					
5	5	57	25	34	35	23	44					
6	5	26	24	27	28	35	26					
7	5	70	26	27	27	29	32					
8	5	3	25	36	28	22	19					
9	5	0	52	42	57	24	28					
10	5	17	55	36	37	28	53					

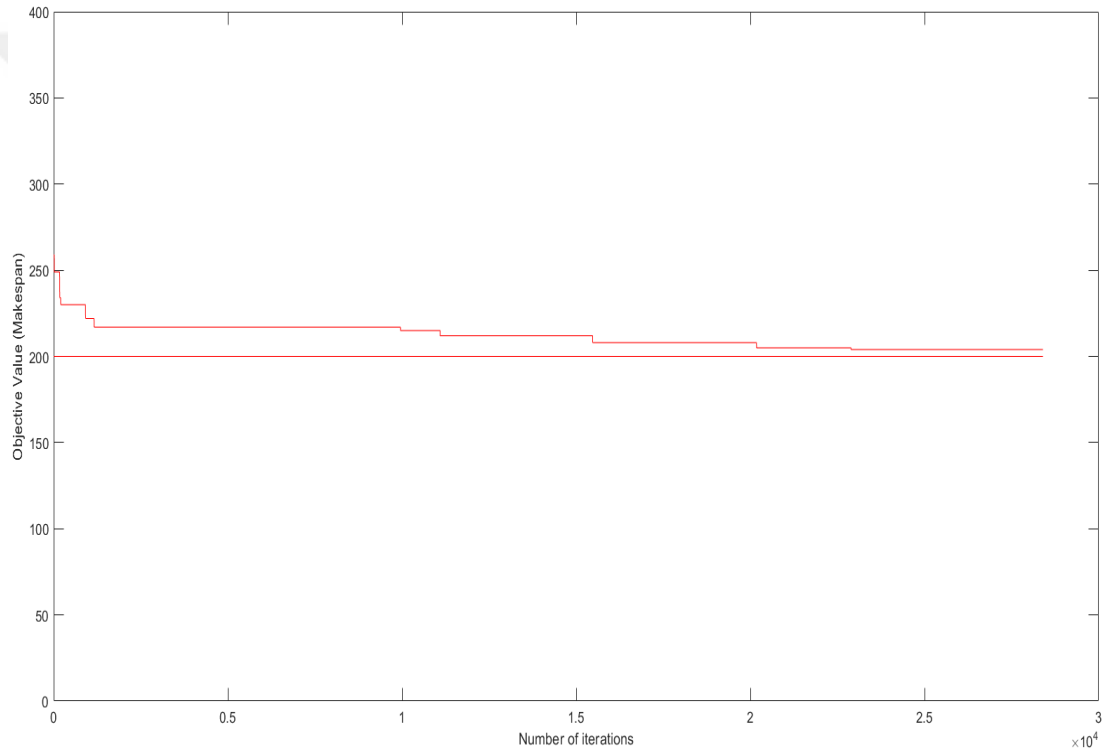


Figure A5.7. Convergence graph of the proposed approach for Problem Instance 3

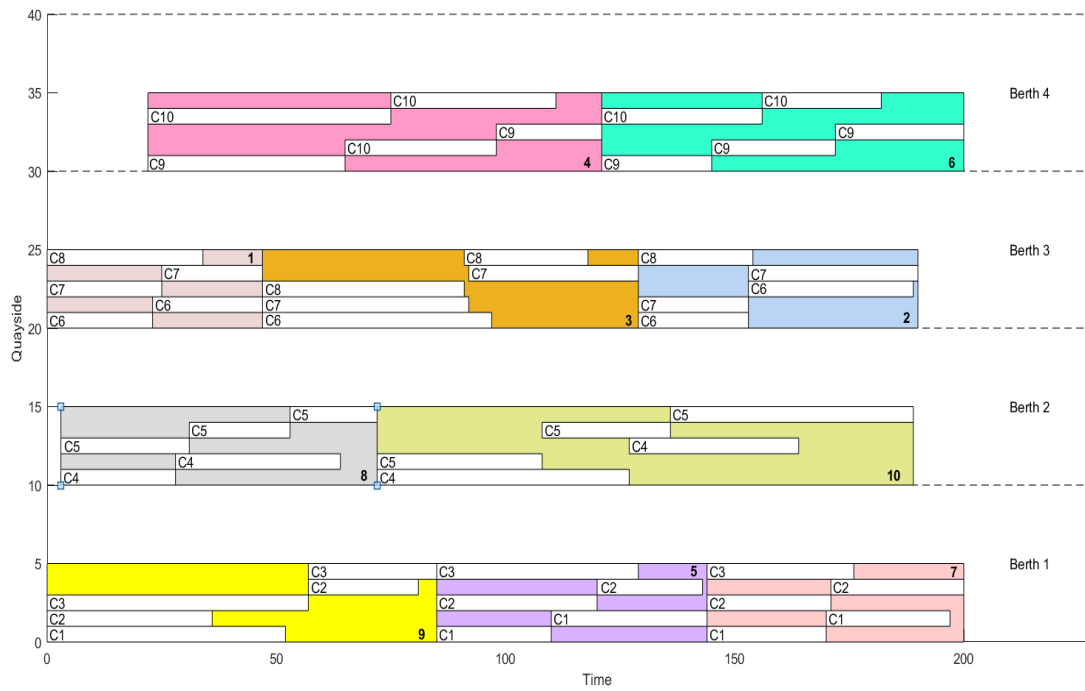


Figure A5.8 Berth space-time diagram for the optimum solution of Problem Instance 3

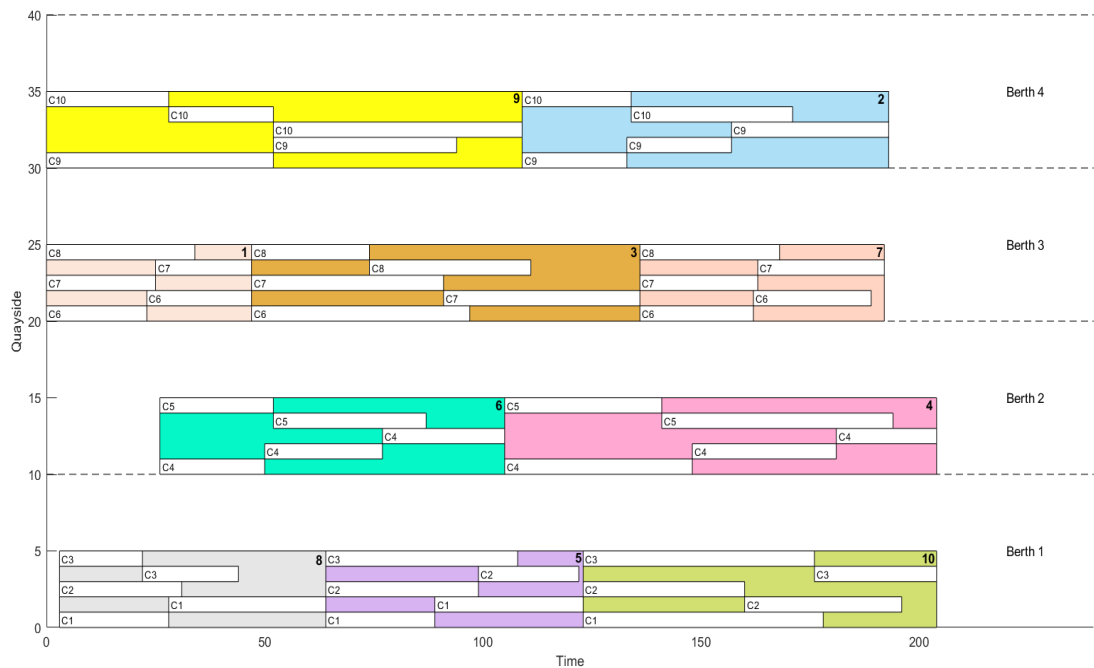


Figure A5.9 Berth space-time diagram for the proposed approach of Problem Instance 3

Table A.5.4 The input data for Problem Instance 4

Vessel No	Vessel Length	Arrival Time	The Handling Times of Each Ship-Bay in Each Vessel									
			1	2	3	4	5	6	7	8	9	10
1	10	0	23	24	25	22	34	31	29	40	35	50
2	10	24	24	24	36	37	25	28	32	25	27	38
3	10	35	50	45	44	37	27	25	28	25	27	26
4	10	22	43	33	23	53	36	24	27	29	31	32
5	10	57	25	34	35	23	44	34	26	28	42	32
6	10	26	24	27	28	35	26	40	51	53	52	25
7	10	70	26	27	27	29	32	34	33	38	42	41
8	10	3	25	36	28	22	19	36	24	32	21	17
9	10	0	52	42	57	24	28	31	32	48	52	19
10	10	17	55	36	37	28	53	26	25	28	31	32

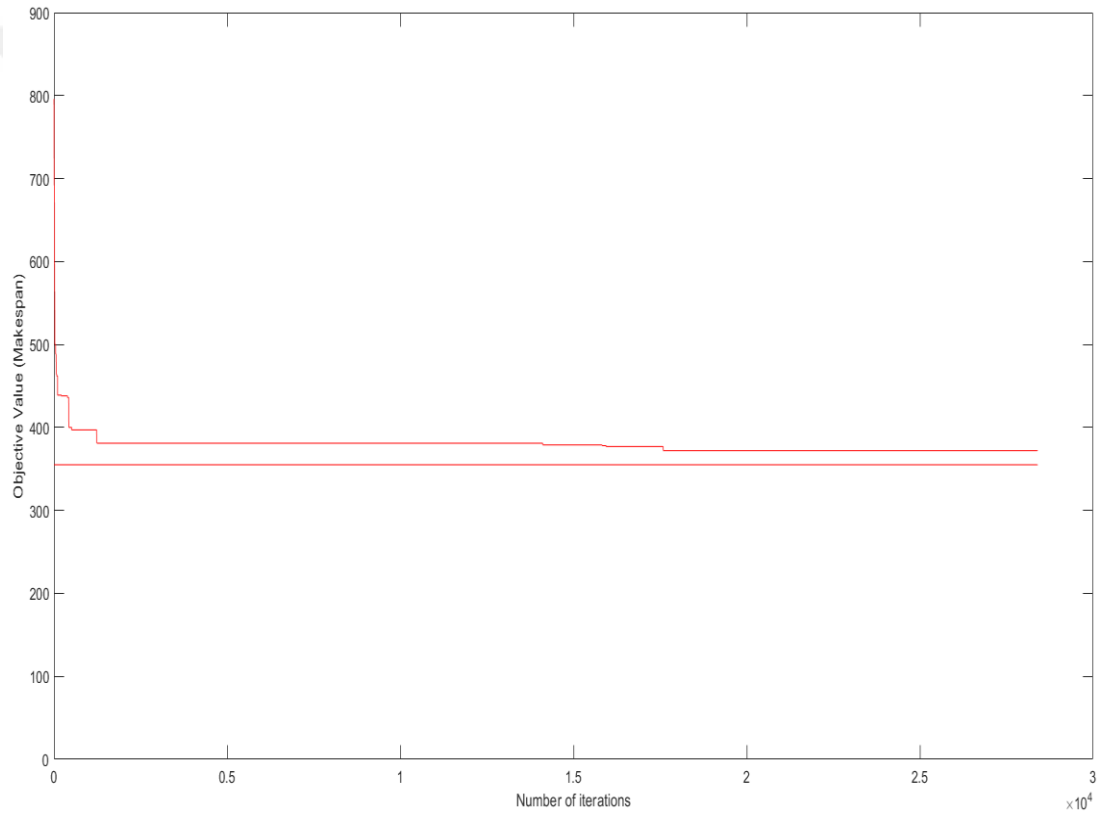


Figure A5.10 Convergence graph of the proposed approach for Problem Instance 4

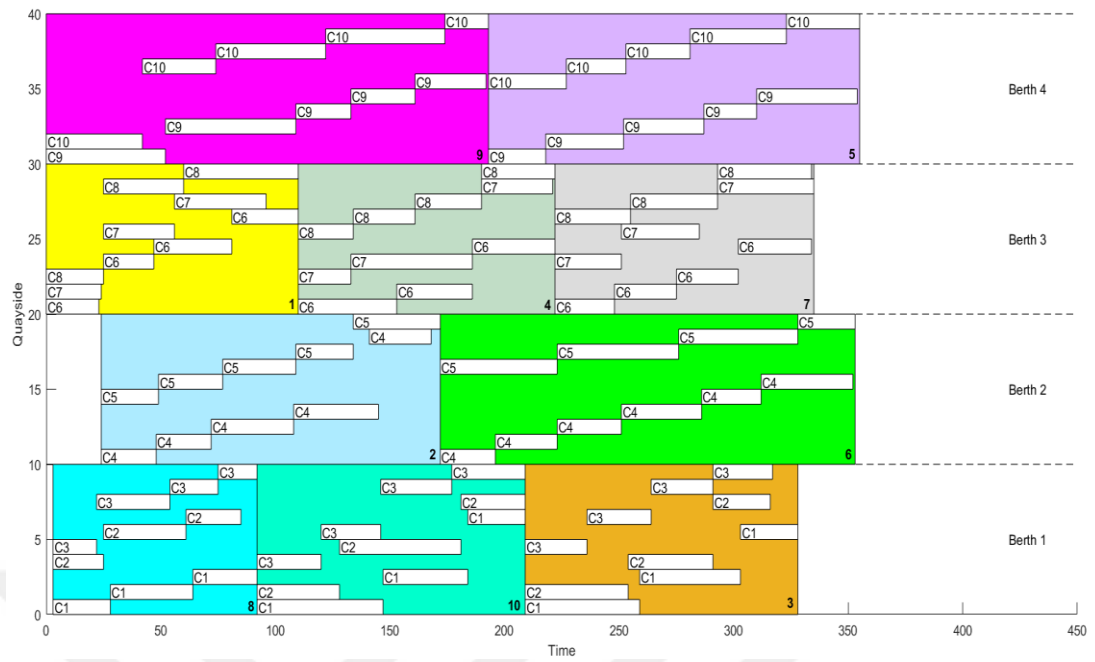


Figure A5.11 Berth space-time diagram for the optimum solution of Problem Instance 4

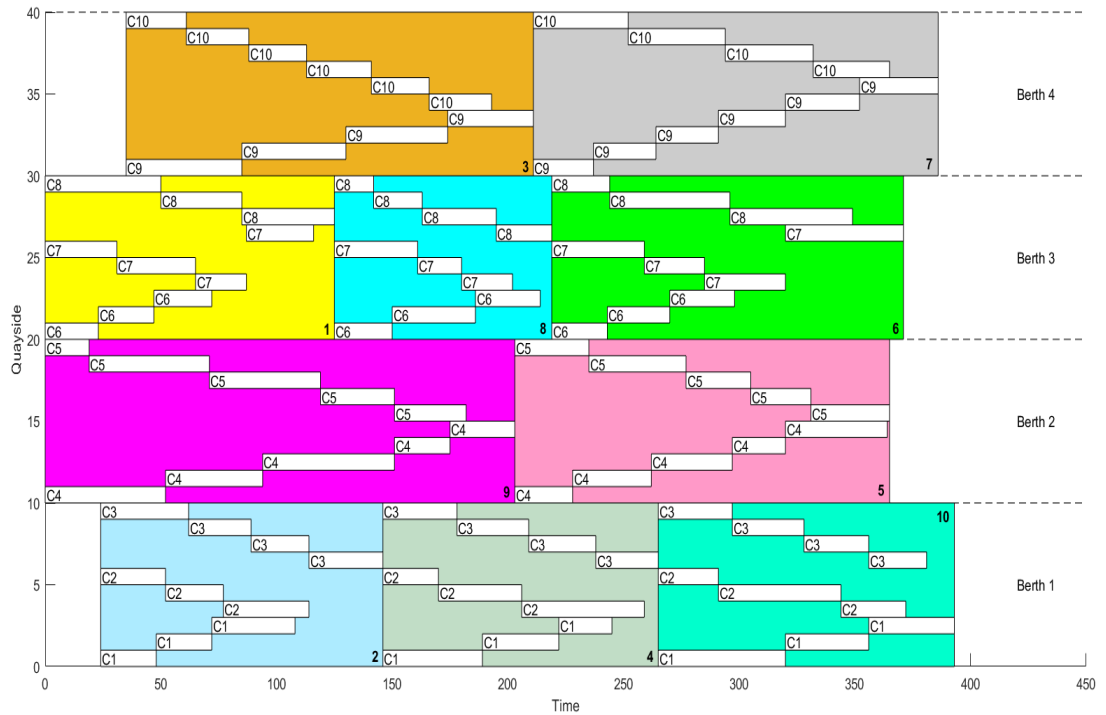


Figure A5.12 Berth space-time diagram for the proposed approach of Problem Instance 4

Table A.5.5 The input data for Problem Instance 5

Vessel No	Vessel Length	Arrival Time	The Handling Times of Each Ship-Bay in Each Vessel									
			1	2	3	4	5	6	7	8	9	10
1	6	5	5	26	10	18	6	19				
2	7	6	8	17	21	9	32	33	22			
3	10	20	23	26	32	37	15	20	22	18	23	30
4	8	15	20	21	44	33	27	33	41	14		
5	5	36	33	6	22	21	20					
6	9	44	19	17	9	25	40	17	27	33	42	
7	10	0	5	7	18	22	33	41	12	38	21	19

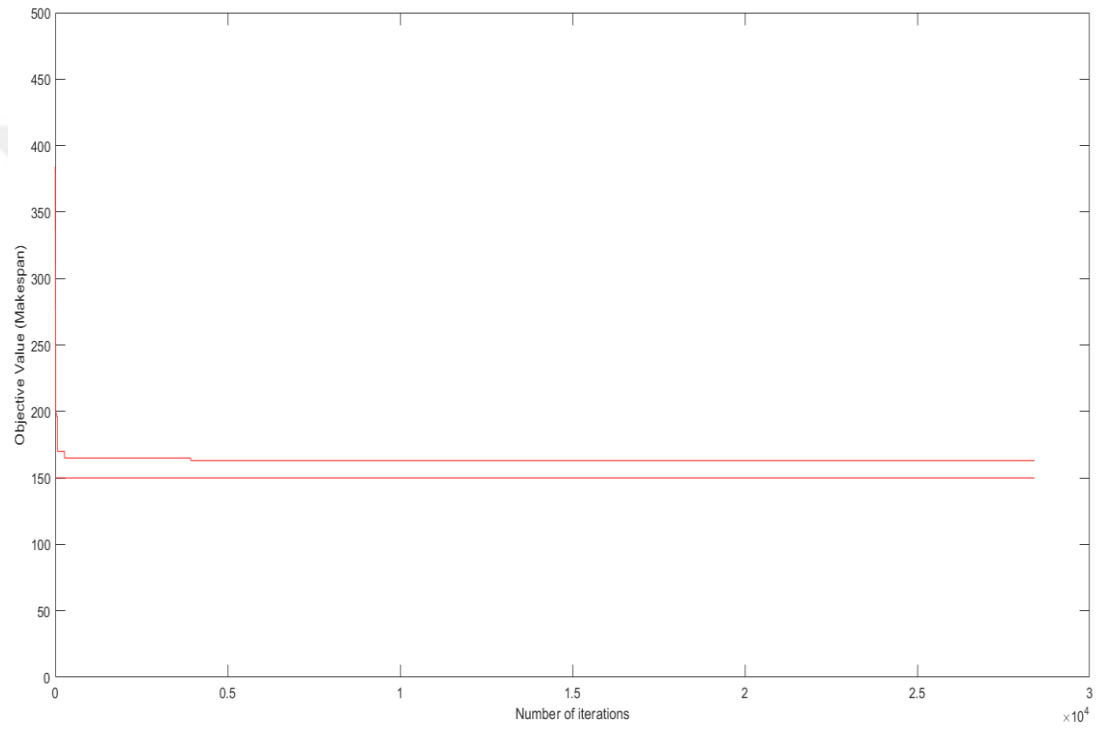


Figure A5.13 Convergence graph of the proposed approach for Problem Instance 5

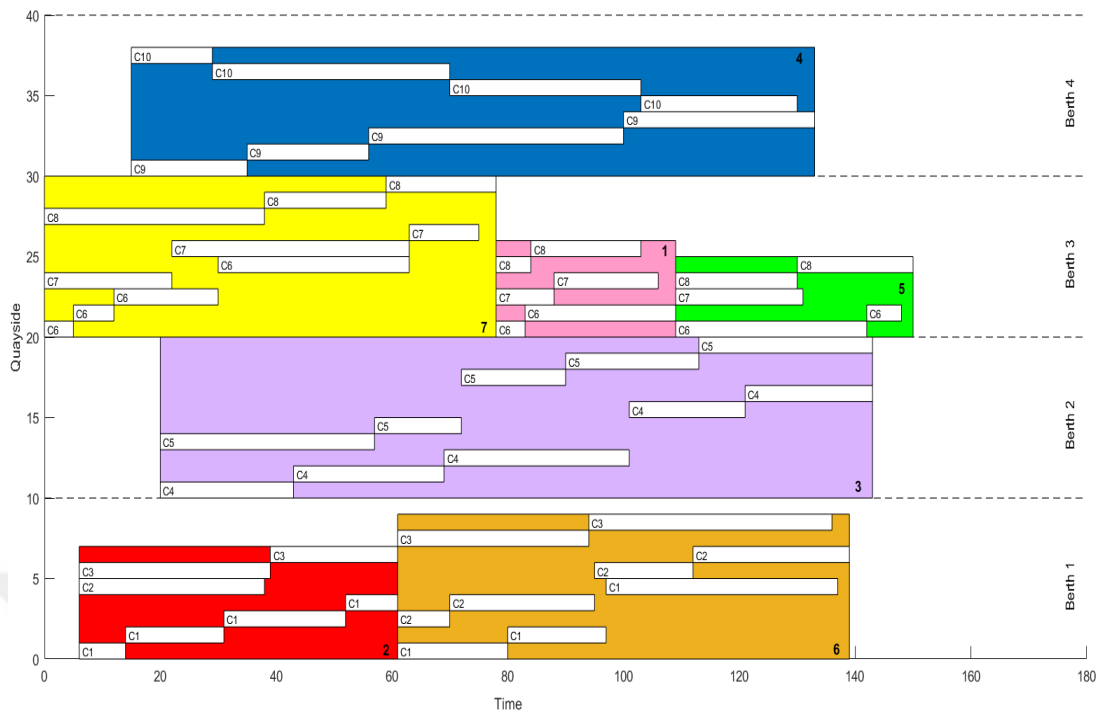


Figure A5.14 Berth space-time diagram for the optimum solution of Problem Instance 5

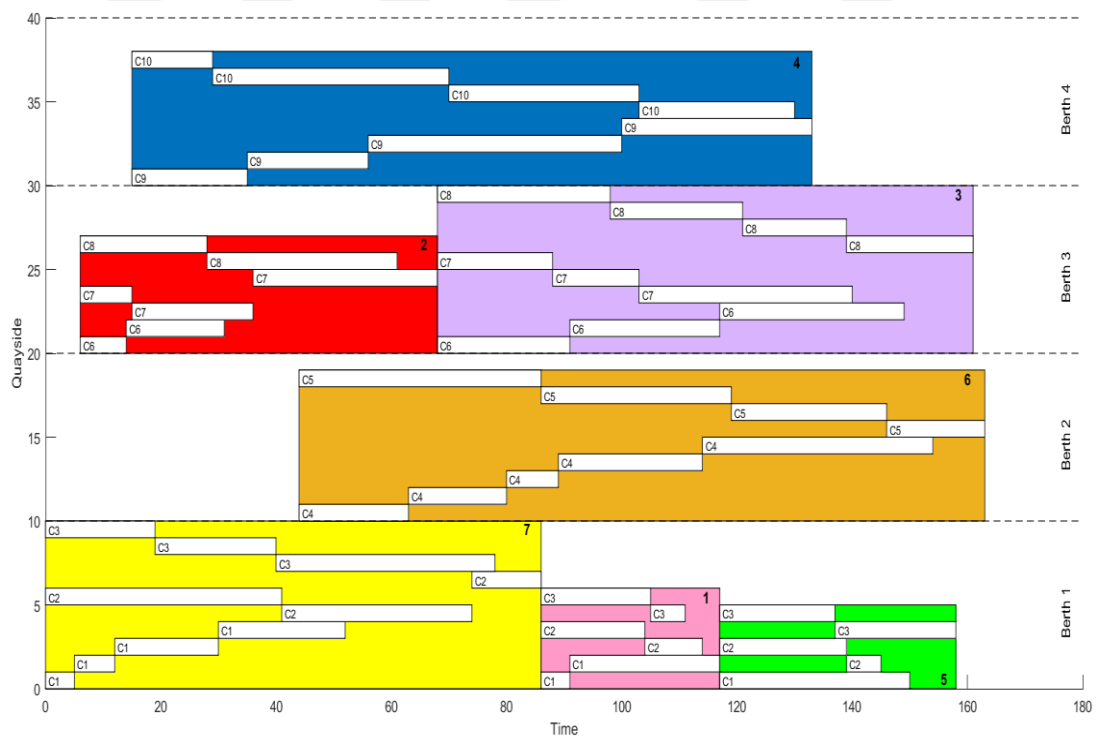


Figure A5.15 Berth space-time diagram for the proposed approach of Problem Instance 5

Table A.5.6 The input data for Problem Instance 6

Vessel No	Vessel Length	Arrival Time	The Handling Times of Each Ship-Bay in Each Vessel									
			1	2	3	4	5	6	7	8	9	10
1	5	2	46	30	40	48	30					
2	7	37	20	5	50	5	34	17	8			
3	8	24	6	24	20	17	10	29	19	43		
4	9	43	21	41	30	11	9	43	25	11	40	
5	10	18	44	34	17	25	11	40	13	41	21	17
6	10	2	18	19	36	29	32	14	47	28	27	49
7	10	46	33	41	20	11	8	42	49	9	17	39
8	10	9	36	45	28	35	15	24	30	23	43	43
9	10	20	48	35	47	36	6	42	46	14	47	9
10	8	46	40	24	14	40	12	50	42	38		

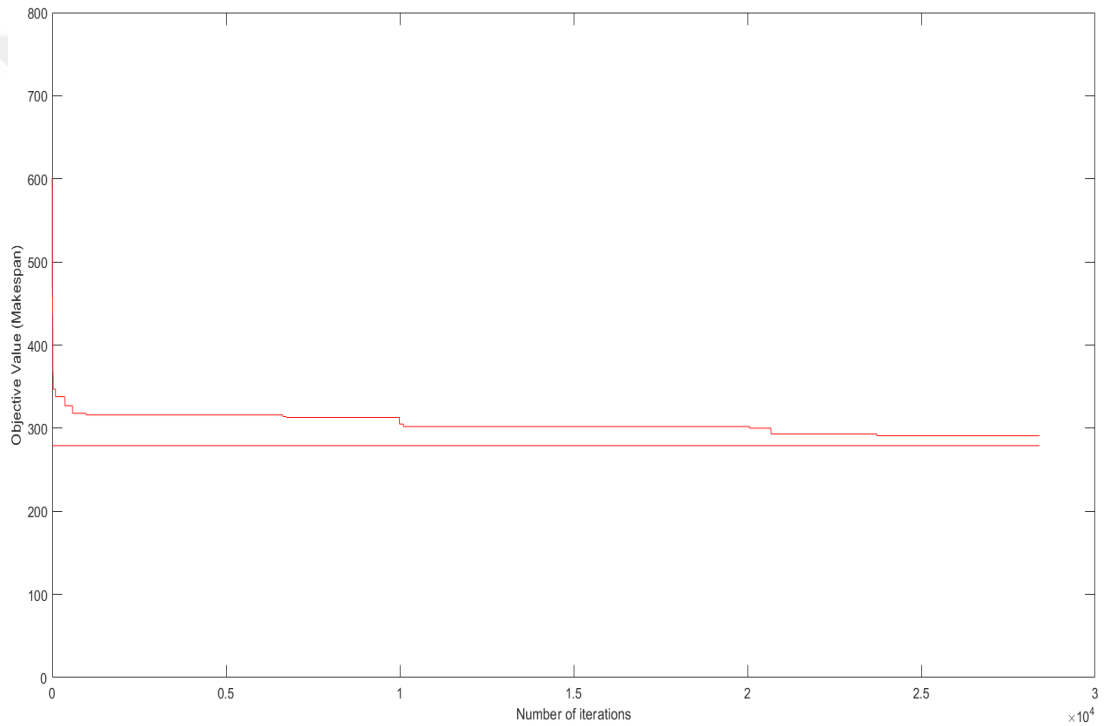


Figure A5.16 Convergence graph of the proposed approach for Problem Instance 6

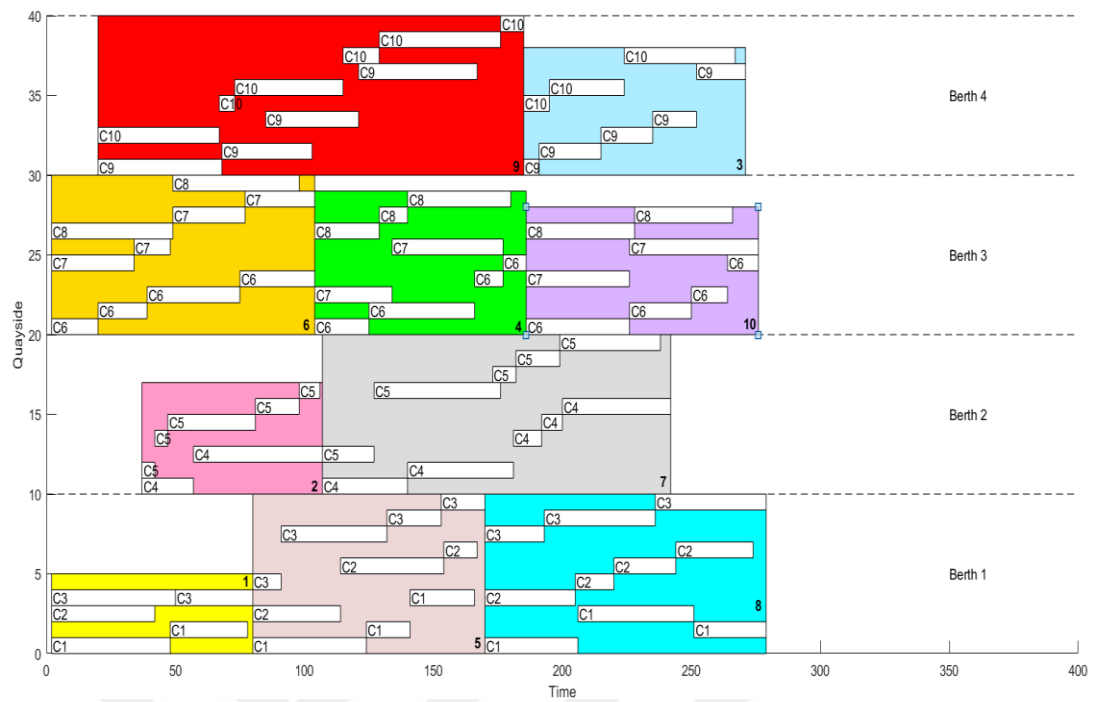


Figure A5.17 Berth space-time diagram for the optimum solution of Problem Instance 6

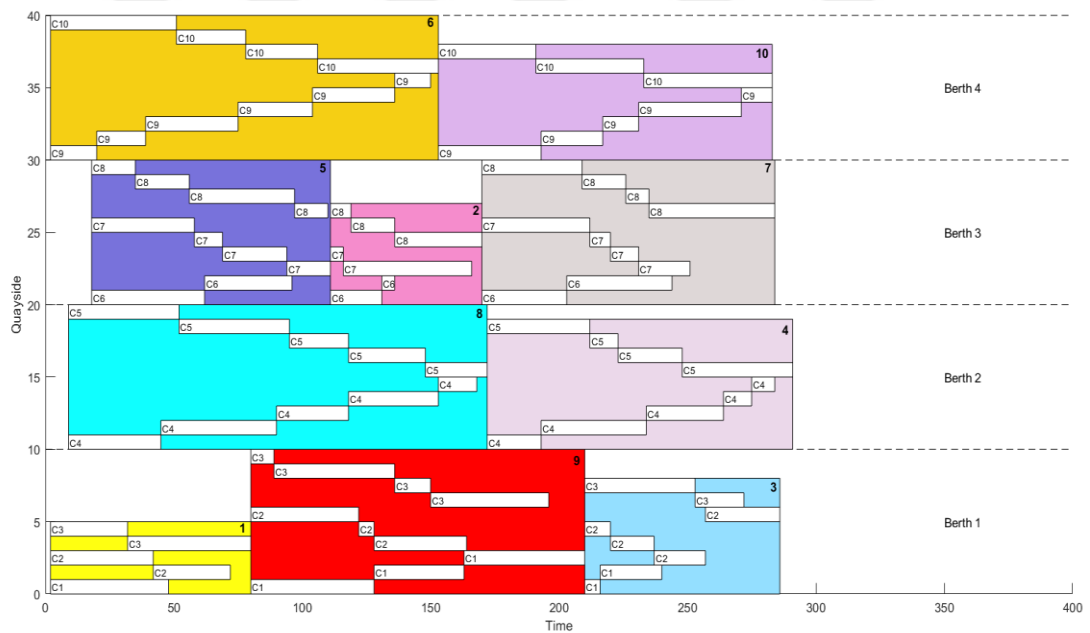


Figure A5.18 Berth space-time diagram for the proposed approach of Problem Instance 6

Table A.5.7 The input data for Problem Instance 7

Vessel No	Vessel Length	Arrival Time	The Handling Times of Each Ship-Bay in Each Vessel									
			1	2	3	4	5	6	7	8	9	10
1	5	25	16	25	47	14	5					
2	7	22	12	26	39	36	24	45	29			
3	8	26	10	22	7	29	27	20	50	50		
4	9	8	46	49	41	24	33	39	29	21	9	
5	8	31	48	8	15	10	39	29	21	9		
6	7	35	7	10	49	44	32	35	11			
7	10	43	14	47	11	3	34	40	20	28	37	32

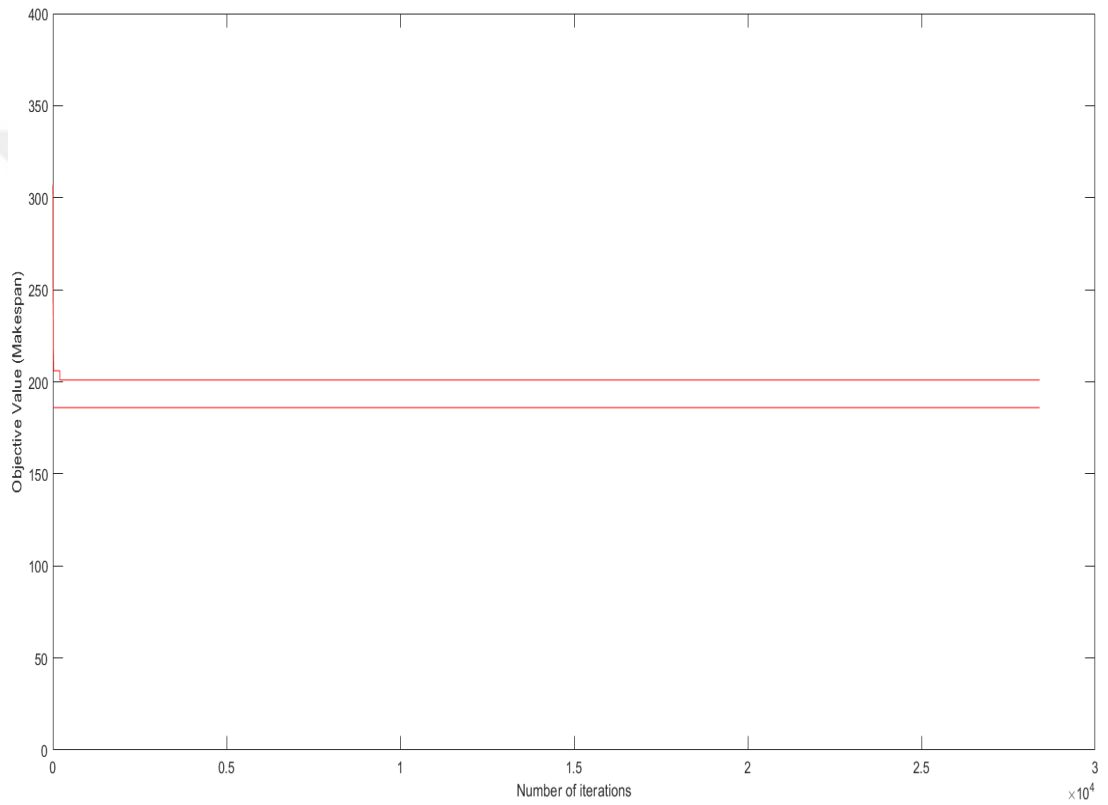


Figure A5.19 Convergence graph of the proposed approach for Problem Instance 7

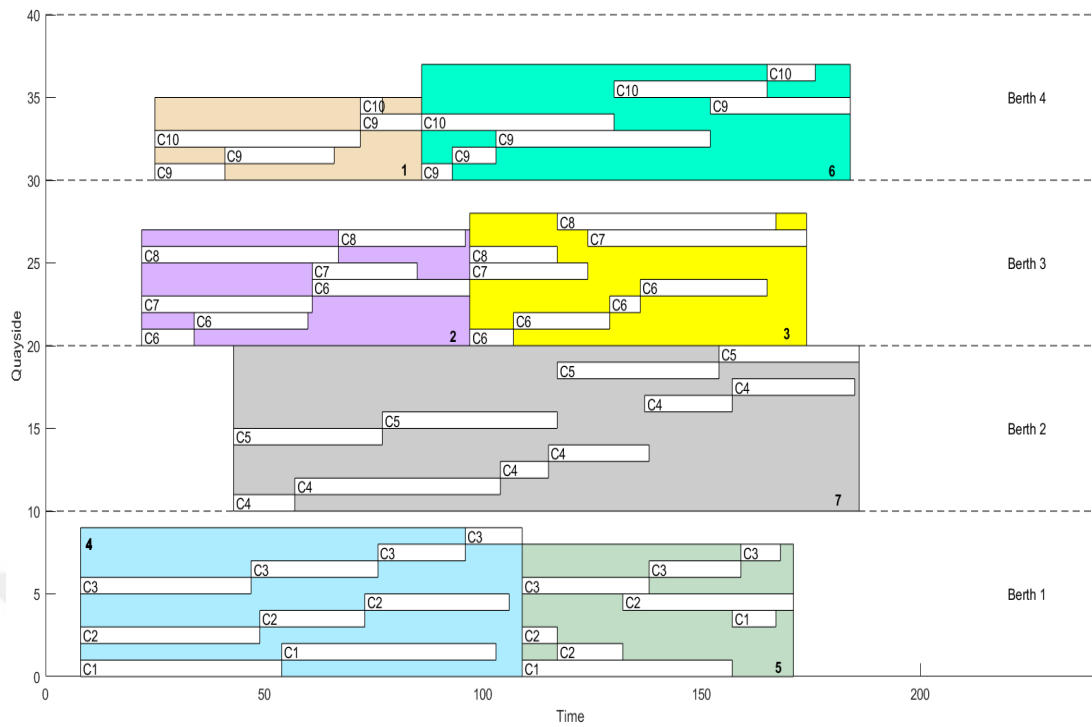


Figure A5.20 Berth space-time diagram for the optimum solution of Problem Instance 7

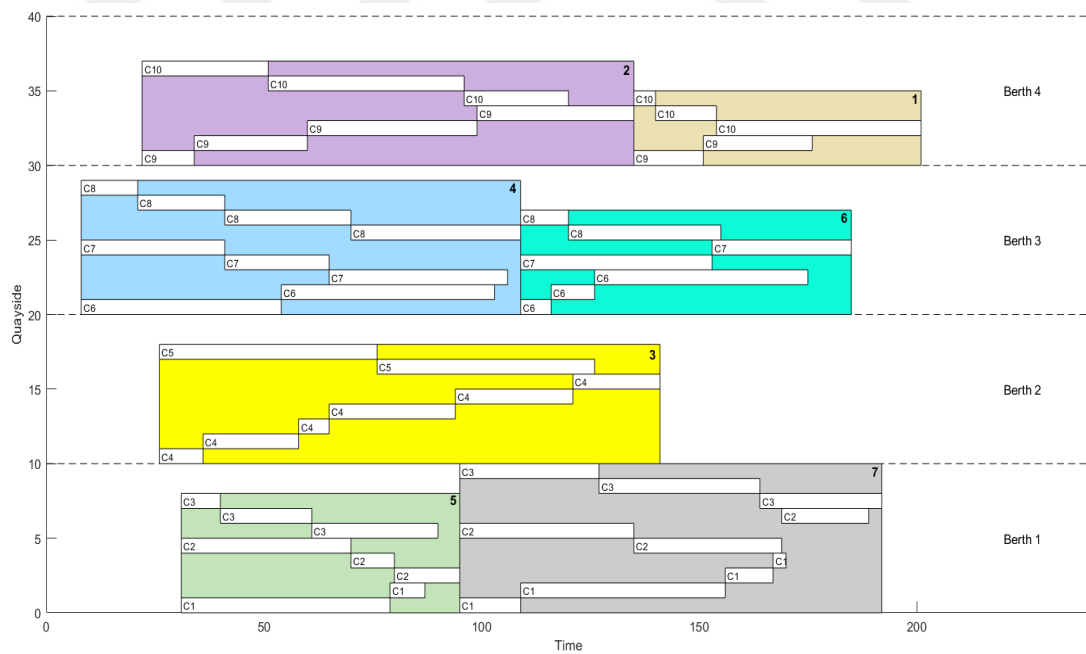


Figure A5.21 Berth space-time diagram for the proposed approach of Problem Instance 7

Table A.5.8 The input data for Problem Instance 8

Vessel No	Vessel Length	Arrival Time	The Handling Times of Each Ship-Bay in Each Vessel									
			1	2	3	4	5	6	7	8	9	10
1	8	8	40	35	28	36	29	15	24	33		
2	7	15	50	28	24	19	20	14	21			
3	6	22	44	33	22	17	43	39				
4	5	44	38	37	18	22	33					
5	8	3	41	38	27	25	19	16	47	9		
6	5	0	50	43	36	29	15					

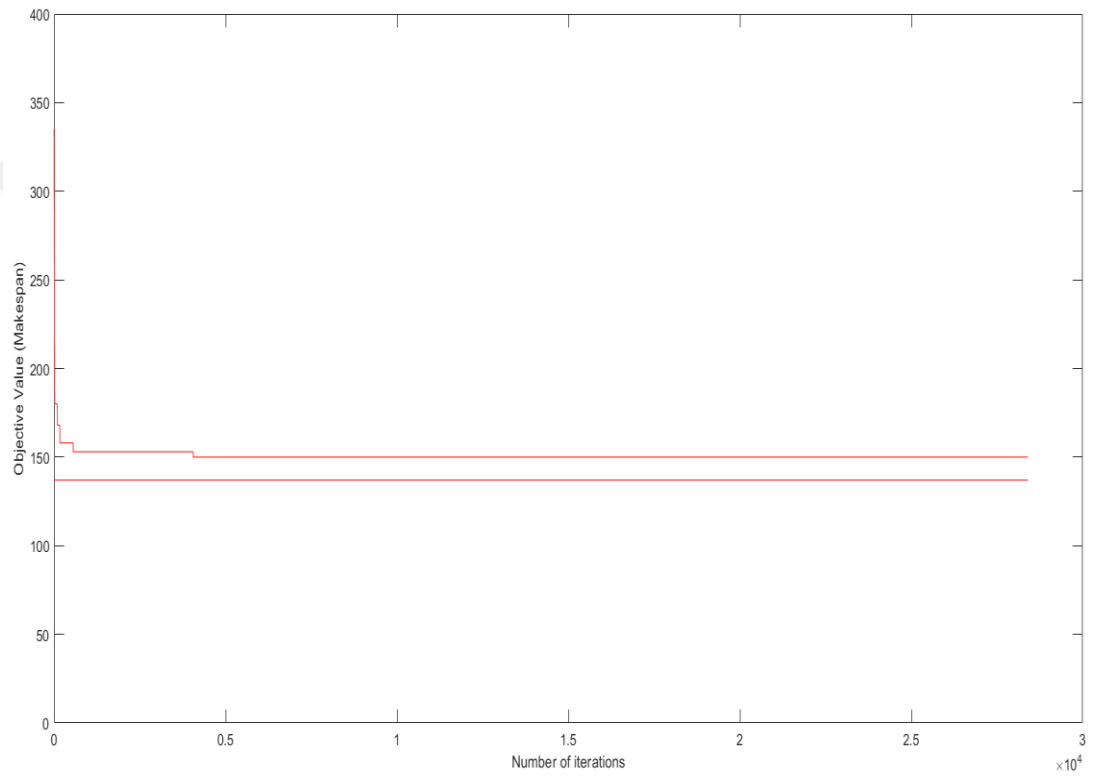


Figure A5.22 Convergence graph of the proposed approach for Problem Instance 8

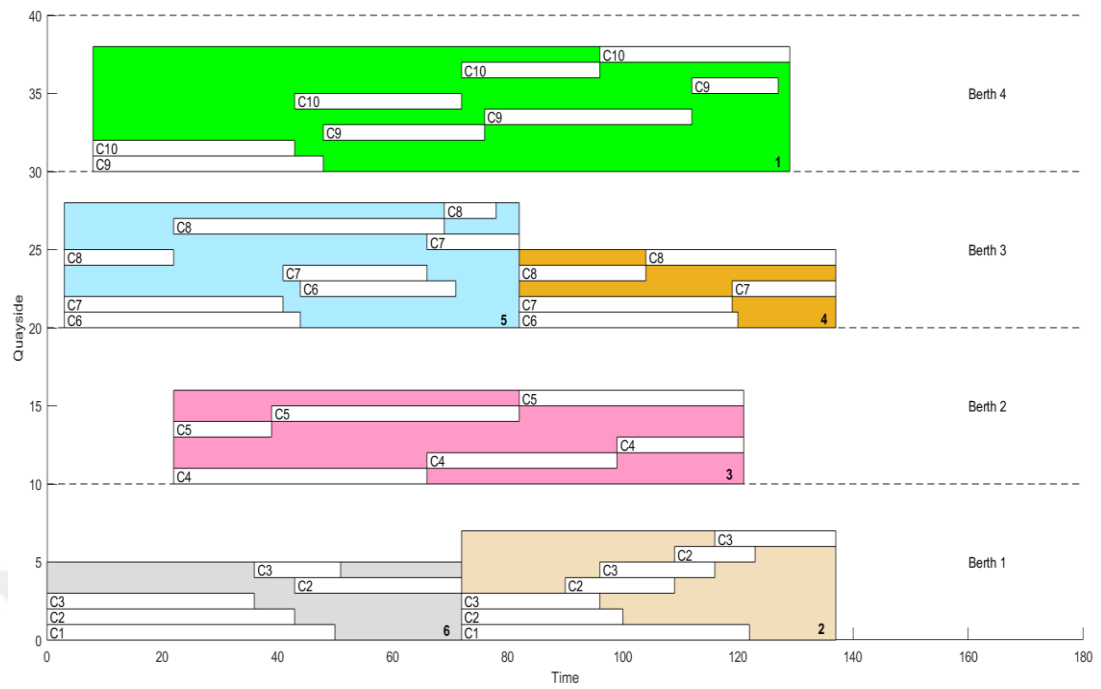


Figure A5.23 Berth space-time diagram for the optimum solution of Problem Instance 8

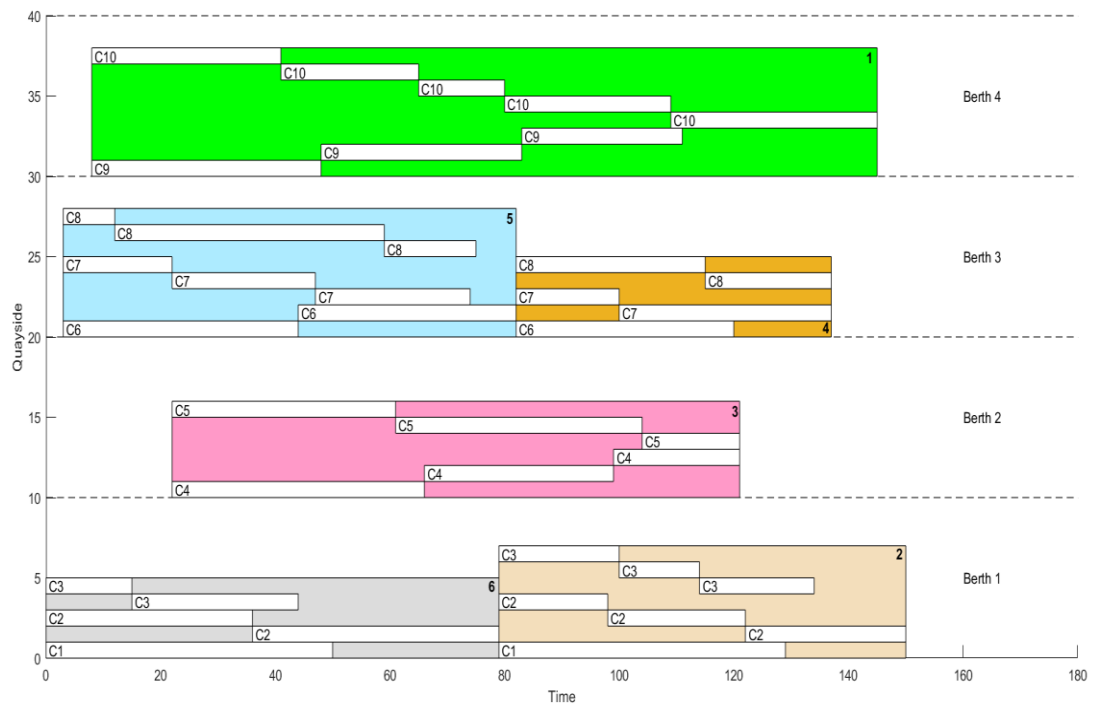


Figure A5.24 Berth space-time diagram for the proposed approach of Problem Instance 8

Table A.5.9 The input data for Problem Instance 9

Vessel No	Vessel Length	Arrival Time	The Handling Times of Each Ship-Bay in Each Vessel									
			1	2	3	4	5	6	7	8	9	10
1	5	0	19	26	35	44	50					
2	6	25	28	39	44	33	50	46				
3	5	46	39	42	38	29	25					
4	6	50	50	44	17	37	21	29				
5	7	18	40	22	15	26	35	39	44			

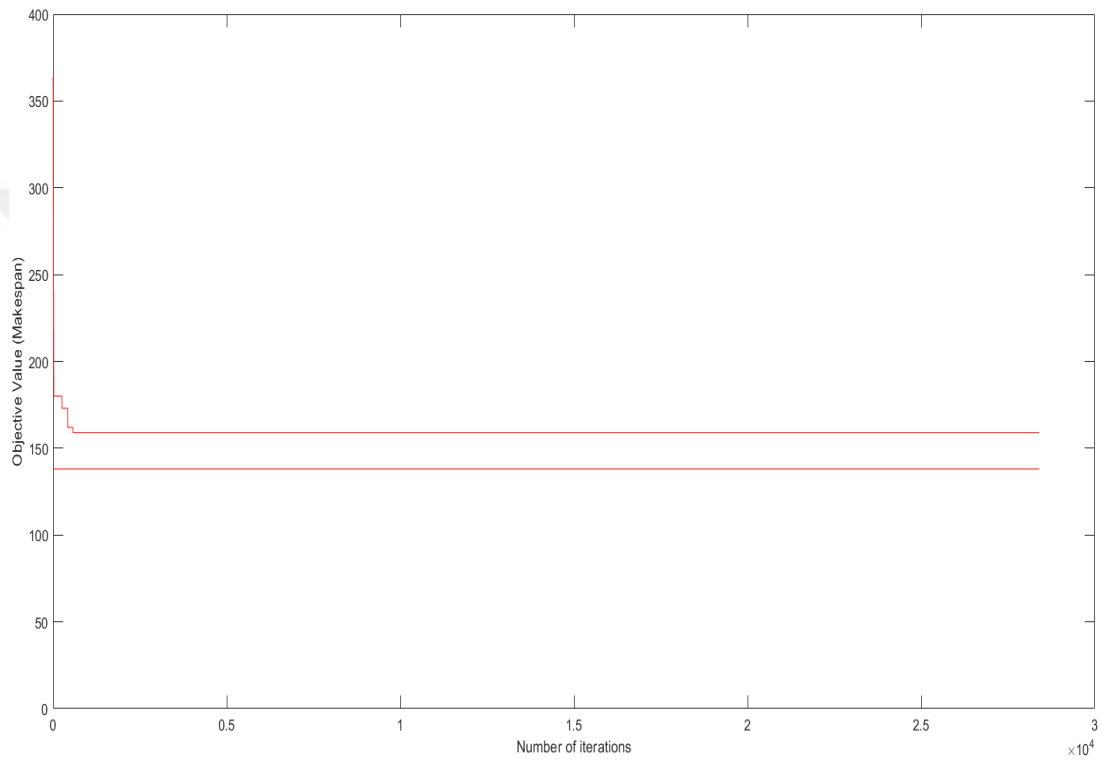


Figure A5.25 Convergence graph of the proposed approach for Problem Instance 9

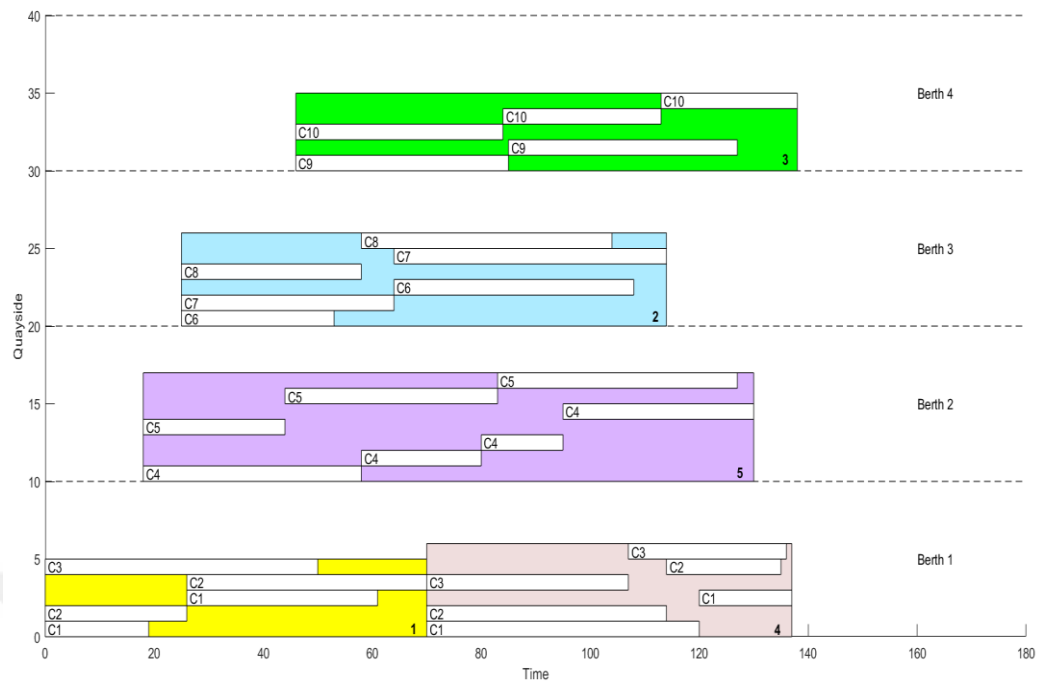


Figure A5.26 Berth space-time diagram for the optimum solution of Problem Instance 9

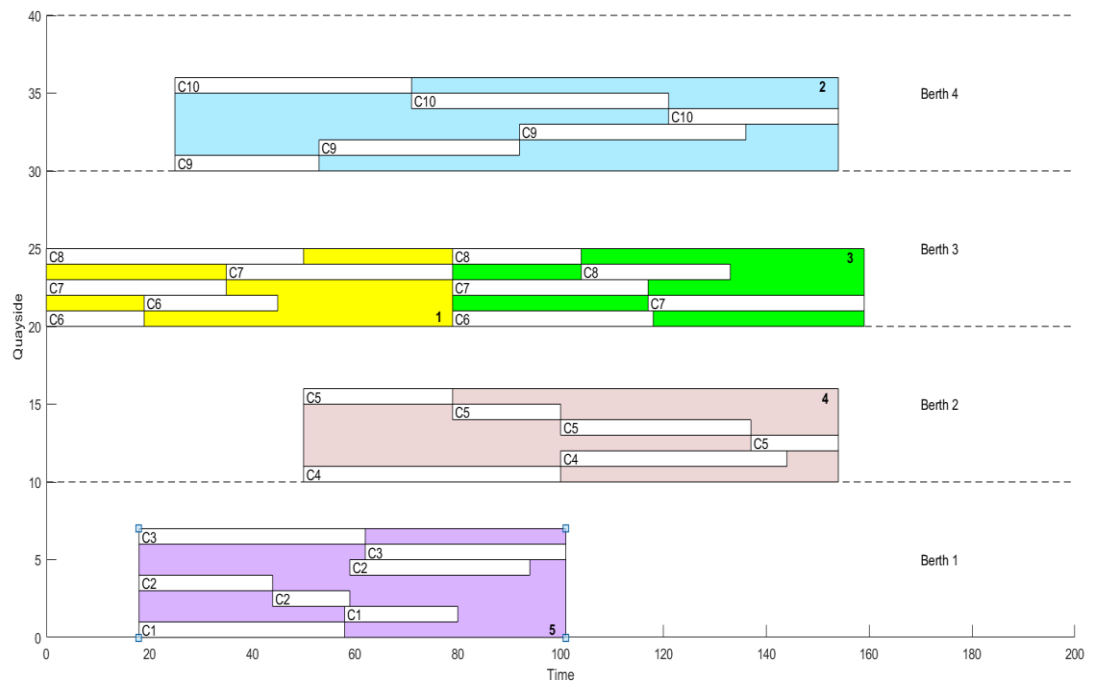


Figure A5.27 Berth space-time diagram for the proposed approach of Problem Instance 9

Table A.5.10 The input data for Problem Instance 10

Vessel No	Vessel Length	Arrival Time	The Handling Times of Each Ship-Bay in Each Vessel									
			1	2	3	4	5	6	7	8	9	10
1	10	0	35	25	33	46	18	21	39	40	18	21
2	9	20	24	32	18	15	40	27	14	12	20	
3	8	18	35	27	29	46	33	15	21	20		
4	7	35	11	25	39	21	17	50	48			
5	6	5	21	32	19	35	42	48				
6	5	40	27	13	36	41	37					
7	8	27	13	23	35	27	41	32	22	45		
8	7	3	22	34	17	26	43	32	16			
9	5	10	50	48	27	36	11					

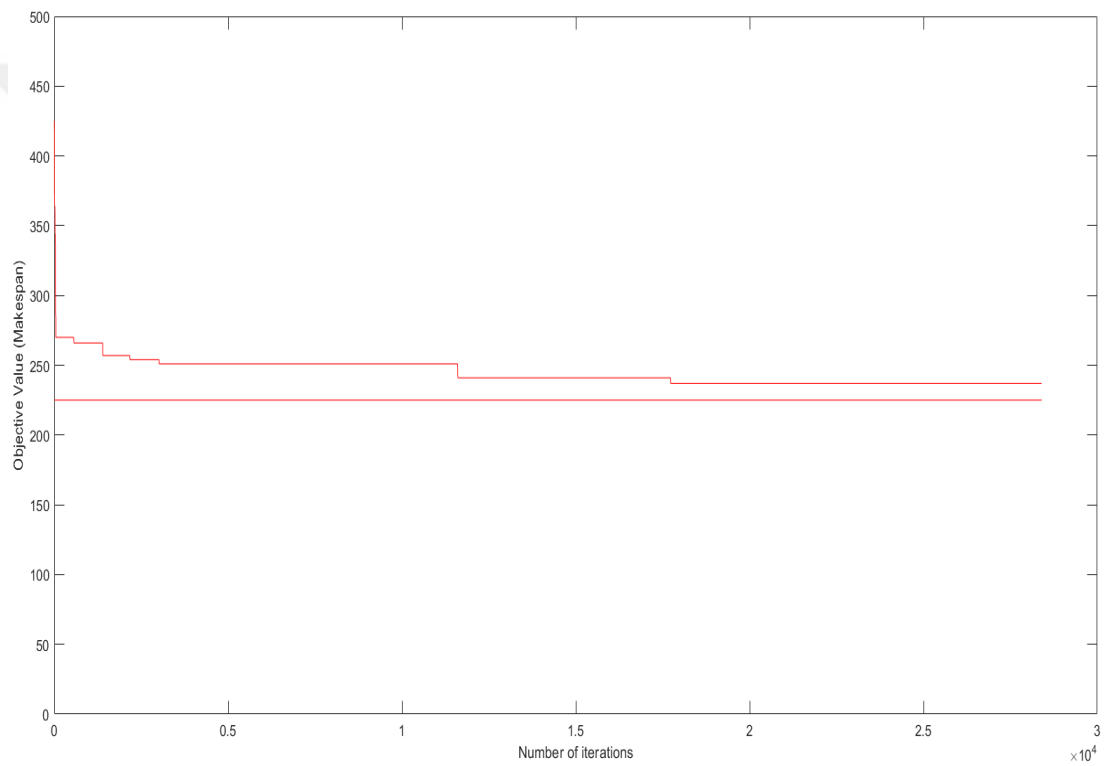


Figure A5.28 Convergence graph of the proposed approach for Problem Instance 10

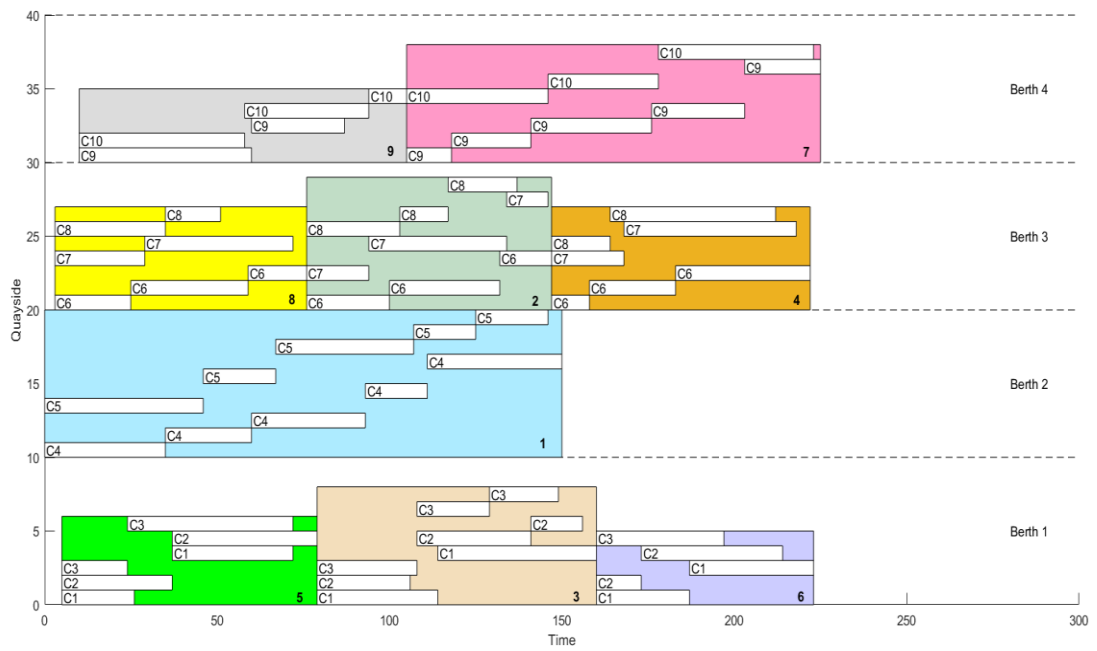


Figure A5.29 Berth space-time diagram for the optimum solution of Problem Instance 10

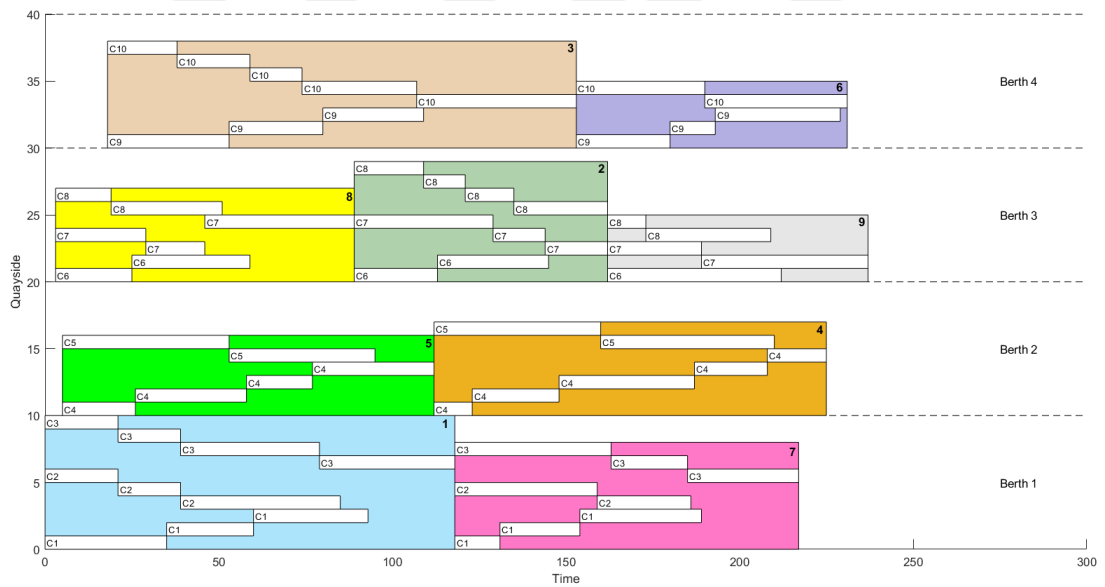


Figure A5.30 Berth space-time diagram for the proposed approach of Problem Instance 10

Appendix A6: Input data of the Sample Problems

Sample Problem 1

Table A6.1 Number of containers at each ship-bay of vessels

		Ship-Bay Number									
		1	2	3	4	5	6	7	8	9	10
Vessel Number	1	0	44	0	38	48	0	56	69	0	0
	2	44	54	0	0	49	48	0	48	48	0
	3	67	0	0	64	48	0	36	0	0	0
	4	0	60	61	49	0	48	0	36	0	0
	5	45	55	0	39	0	48	75	0	0	0
	6	31	45	0	55	66	68	0	58	0	0
	7	63	73	53	0	35	0	38	0	0	0
	8	48	50	0	39	0	57	0	0	0	0
	9	60	55	0	49	75	0	52	0	0	0
	10	41	32	58	0	56	0	51	40	0	0
	11	53	0	46	0	59	0	57	33	0	0
	12	41	0	46	0	60	50	44	0	0	0
	13	55	59	0	32	49	34	0	71	0	0
	14	33	20	58	0	35	0	38	0	0	0
	15	70	0	38	20	0	58	47	39	0	0
	16	61	32	0	56	55	0	68	0	48	0
	17	55	44	32	0	69	59	28	0	0	0
	18	56	0	55	0	48	48	46	57	59	0
	19	34	48	0	59	64	0	0	35	0	0
	20	31	49	51	0	57	0	62	43	0	0

Table A6.2 Quay crane related data

QC number	Initial Position	Lower Bound of the CA	Upper Bound of the CA
1	1	1	31
2	6	2	32
3	10	3	33
4	14	4	34
5	18	5	35
6	23	6	36
7	27	7	37
8	31	8	38
9	36	9	39
10	40	10	40

Table A6.3 Berth related data

	Berth Depth	Berth Length
Berth 1	13	10
Berth 2	13	10
Berth 3	15	10
Berth 4	14	10

Table A6.4 Vessel related data

Vessel Number	Vessel Length	Vessel Draft	Arrival Time
1	8	10	8
2	9	14	70
3	7	12	13
4	9	10	44
5	8	12	3
6	9	10	0
7	7	12	55
8	6	12	16
9	7	10	40
10	9	12	89
11	9	13	0
12	9	12	39
13	8	12	88
14	7	13	90
15	8	11	30
16	9	14	90
17	9	13	40
18	9	12	76
19	8	13	21
20	9	13	0

Sample Problem 2:

Table A6.5 Number of containers at each ship-bay of vessels

Table A6.5 Number of containers at each ship-bay of vessels

[illegible]

Table A6.6 Quay crane related data

QC number	Initial Position	Lower Bound of the CA	Upper Bound of the CA
1	1	1	43
2	8	2	44
3	15	3	45
4	21	4	46
5	28	5	47
6	35	6	48
7	41	7	49
8	48	8	50
9	55	51	98
10	61	52	99
11	68	53	100
12	75	54	101
13	81	55	102
14	88	56	103
15	95	57	104
16	101	58	105
17	108	105	114
18	115	106	115

Table A6.7 Berth related data

	Berth Depth	Berth Length
Berth 1	14	20
Berth 2	15	20
Berth 3	16	10
Berth 4	15	20
Berth 5	14	20
Berth 6	15	15
Berth 7	14	10

Table A6.8 Vessel related data

Vessel Number	Vessel Length	Vessel Draft	Arrival Time
1	6	14	0
2	7	15	14
3	9	13	15
4	10	14	23
5	8	12	67
6	12	13	86
7	19	15	89
8	11	14	100
9	8	13	34
10	9	13	96
11	10	13	120
12	6	13	88
13	5	14	99
14	11	15	130
15	12	13	155
16	15	13	39
17	8	12	0
18	8	12	0
19	9	14	5
20	9	14	8
21	10	13	178
22	6	15	15
23	7	13	130
24	12	14	20
25	13	13	200
26	9	12	45
27	10	12	210
28	11	13	56
29	8	14	156
30	9	12	144
31	15	14	209
32	14	13	217
33	8	13	255
34	12	13	288
35	15	14	300
36	8	14	180
37	7	13	190
38	9	14	122
39	8	13	133
40	9	13	145

Sample Problem 3:

Table A6.9 Number of containers at each ship-bay of vessels

		Ship-Bay Number																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Vessel Number	1	36	39	50	69	43	64	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	28	27	49	0	66	63	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	42	50	62	72	78	26	0	42	0	0	0	0	0	0	0	0	0	0
	4	57	57	0	0	20	35	76	46	0	0	0	0	0	0	0	0	0	0	0
	5	42	25	0	0	41	22	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	57	0	41	0	32	0	51	76	77	0	46	0	0	0	0	0	0	0	0
	7	0	48	71	21	26	0	59	0	0	53	24	0	72	28	0	70	59	66	28
	8	34	54	29	42	49	0	54	44	74	0	0	0	0	0	0	0	0	0	0
	9	65	70	0	39	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	33	28	67	73	0	58	50	0	48	0	0	0	0	0	0	0	0	0	0
	11	61	62	24	25	0	52	21	27	57	43	0	0	0	0	0	0	0	0	0
	12	54	21	45	23	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	13	53	21	31	37	0	23	61	21	45	73	70	33	71	70	0	55	69	40	0
	14	0	51	46	76	80	0	58	45	38	31	29	0	0	0	0	0	0	0	0
	15	38	27	25	0	0	62	0	47	53	42	44	60	0	0	0	0	0	0	0
	16	57	67	0	49	37	73	47	0	38	43	29	32	65	0	59	0	0	0	0
	17	46	24	51	69	25	0	46	31	0	0	0	0	0	0	0	0	0	0	0
	18	67	0	65	70	52	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	19	0	36	0	21	36	0	71	32	32	0	0	0	0	0	0	0	0	0	0
	20	55	64	49	0	35	79	61	0	69	0	0	0	0	0	0	0	0	0	0
	21	38	43	29	0	44	33	0	57	22	27	0	0	0	0	0	0	0	0	0
	22	47	70	46	76	0	29	0	0	0	0	0	0	0	0	0	0	0	0	0
	23	46	32	77	27	66	0	77	0	0	0	0	0	0	0	0	0	0	0	0
	24	76	71	0	76	22	26	41	0	33	0	77	59	0	0	0	0	0	0	0
	25	31	52	50	24	80	40	35	0	31	55	66	27	20	0	0	0	0	0	0

Table A6.9 continues

[illegible]

Table A6.9 continues

		Ship-Bay Number																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Vessel Number	51	59	73	78	49	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	52	66	0	0	0	43	42	53	61	0	73	0	0	0	80	0	60	58	0	0
	53	0	73	47	46	41	76	53	38	69	49	0	71	0	23	70	0	0	0	0
	54	76	75	48	51	36	59	64	0	0	0	0	0	0	0	0	0	0	0	0
	55	66	80	58	0	0	0	0	39	0	54	0	0	0	0	0	0	0	0	0
	56	47	79	40	36	37	0	58	65	70	0	20	0	0	0	0	0	0	0	0
	57	0	0	46	62	25	72	0	65	53	0	53	31	0	53	0	0	0	0	0
	58	68	0	61	27	71	0	20	0	36	27	24	31	76	48	0	0	45	38	0
	59	64	55	63	24	23	0	0	78	0	51	32	52	0	0	0	0	0	0	0
	60	27	0	44	25	0	76	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A6.10 Quay crane related data

QC Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Initial Position	1	8	15	22	29	36	43	50	51	60	70	71	80	90	91	100	110	111	118	125	132	139	146	153	160
Lower Bound of the CA	1	2	3	4	5	6	7	8	51	52	53	71	72	73	91	92	93	111	112	113	114	115	116	117	118
Upper Bound of the CA	43	44	45	46	47	48	49	50	68	69	70	88	89	90	118	119	110	153	154	155	156	157	158	159	160

Table A6.11 Vessel related data

Vessel Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Vessel Length	6	7	9	10	8	12	19	11	8	9	10	6	5	11	12	15	8	8	9	9	10	6	7	12	13	9	10	11	8	9
Vessel Draft	14	14	13	14	12	13	14	14	13	13	13	13	14	14	13	13	12	12	14	14	13	14	13	14	13	12	12	13	14	12
Arrival Time	0	14	15	23	67	86	89	100	34	96	120	88	99	130	155	39	0	0	5	8	178	15	130	20	200	45	210	56	156	144

Table A6.11 continues

Vessel Number	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Vessel Length	15	14	8	12	15	8	7	9	8	9	19	18	15	6	9	8	8	10	11	6	5	17	15	7	10	11	14	18	12	6
Vessel Draft	14	13	13	13	14	14	13	14	13	13	13	13	14	14	14	14	14	14	13	14	14	14	13	14	14	14	14	13	14	13
Arrival Time	209	217	255	288	300	180	190	122	133	145	231	330	15	201	400	167	47	231	122	67	0	309	217	25	160	12	0	156	240	421

Table A6.12 Berth Related Data

	Berth Depth	Berth Length
Berth 1	14	20
Berth 2	15	20
Berth 3	16	10
Berth 4	15	10
Berth 5	14	10
Berth 6	15	20
Berth 7	14	20
Berth 8	13	20
Berth 9	13	20
Berth 10	14	10

Appendix A7: MATLAB Code used for determining initial positions of QCs

```
length_of_quayside=20;
number_of_cranes=3;
assignment=zeros(1,length_of_quayside);
if number_of_cranes==2
    assignment(1)=1;
    assignment(length_of_quayside)=number_of_cranes;
else
    assignment(1)=1;
    assignment(length_of_quayside)=number_of_cranes;
    remaining_cranes=number_of_cranes-2;
    remaining_length=length_of_quayside-2;

    for i=1:remaining_cranes
        ratio=ceil((remaining_length/(remaining_cranes+1))*i);
        assignment(1+ratio)=i+1;
    end
end
```

Appendix A8: MATLAB Code used for grouping the same schedules and calculating estimated berthing/departure times

```
tic
fileID1=fopen('C:\tez6\schedule.txt','r');
fileID2=fopen('C:\tez6\result.txt','r');
formatSpec='%f';
parameters=7;
number_of_replications=10000;
number_of_rectangles=129;
number_of_vessels=20;
sizesonuc=[parameters number_of_replications*number_of_rectangles];
result=(fscanf(fileID2,formatSpec));
sonuc=(fscanf(fileID1,formatSpec,sizesonuc));
fclose(fileID1);
toc
similar=zeros(number_of_replications,number_of_replications);
b=0;
group=zeros(1,number_of_replications);
berthing_times=zeros(1,number_of_vessels);
departure_times=zeros(1,number_of_vessels);
average_berthing_times=zeros(1,number_of_vessels);
average_departure_times=zeros(1,number_of_vessels);

for i=1:(number_of_replications-1)
    for j=i+1:number_of_replications
        zey=sonuc((i-1)*number_of_rectangles+1:i*number_of_rectangles,2:4);
        nep=sonuc((j-1)*number_of_rectangles+1:number_of_rectangles*j,2:4);
        if zey-nep==0
            similar(i,j)=1;
        end
    end
end

toc
unique=zeros(1,number_of_replications);
for i=1:number_of_replications
    if similar(i,:)==0
        unique(i)=i;
    end
    for j=i+1:number_of_replications
        if similar(i,j)~=0
            similar(j,:)=0;
        end
    end
end
group=zeros(number_of_replications,number_of_replications);
a=1;
written=zeros(1,number_of_replications);
for i=1:number_of_replications
    b=1;
    if written(1,i)==0
        if unique(i)==i
            group(a,b)=i;
            written(1,i)=1;
        end
    end
end
```

```

else
    group(a,b)=i;
    written(1,i)=1;
    b=b+1;
for j=i+1:number_of_replications
    if similar(i,j)==1
        group(a,b)=j;
        written(1,j)=1;
        b=b+1;
    end
end
end
a=a+1;
end
end
c=a-1;
for a=1:c
    b=1;
    total_makespan=0;
    avg_makespan=0;
    for b=1:number_of_replications
if group(a,b)~=0
        total_makespan=total_makespan+result(group(a,b),1);
        avg_makespan=total_makespan/b;
end
    end
    average_makespan(a,1)=avg_makespan;
end
for i=1:number_of_replications
    nonzero_say(i)=nnz(group(i,:));
end
max_group=max(nonzero_say);
cizelgeler=group(2,1:2191);
for k=1:number_of_vessels
    for l=1:2191
        cizelgeno=cizelgeler(1,l);
for i=(cizelgeno-
1)*number_of_rectangles+1:number_of_rectangles*(cizelgeno-
1)+number_of_rectangles
    if sonuc(i,2)==2 && sonuc(i,3)==k
        berthing_times(k)=berthing_times(k)+sonuc(i,5);
        departure_times(k)=departure_times(k)+sonuc(i,5)+sonuc(i,7);
    end
end
end
end
    average_berthing_times(k)=berthing_times(k)/2191;
    average_departure_times(k)=departure_times(k)/2191;
end

```