-

THE DISCRETE AND CONTINUOUS BERTH ALLOCATION PROBLEM: MODELS AND ALGORITHMS

by

MICHAIL D. GKOLIAS

A Dissertation submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Civil and Environmental Engineering

written under the direction of

Dr. Maria Boile

and approved by

____

____

____

____

New Brunswick, New Jersey

[October, 2007]

ABSTRACT OF THE DISSERTATION

THE DISCRETE AND CONTINUOUS BERTH ALLOCATION PROBLEM: MODELS AND ALGORITHMS

By MICHAIL D. GKOLIAS

Dissertation Director:

Dr. Maria Boile

Fierce terminal competition and the need to maximize recourses utilization have led marine terminal operators to the development and application of a rich variety of Berth Scheduling Policies (BSPs). Container terminal operators seek for efficient BSPs that will reduce vessels turnaround time, increase port throughput, lead to higher revenues and increased competitiveness of the port, while at the same time keep customer satisfaction at desired levels. Several issues arise when defining the best BSPs for each port operator and the final decision depends on several factors that include the type and function of the port (dedicated or multi-user terminal, transshipment hub etc), the size and location of the port, nearby competition, type of contractual agreement with the vessel carriers etc. Some of these BSPs and issues have to a certain extend been captured by academic research, but still several attributes need to be investigated and included for these models to represent the state of the practice of container terminal operations.

In this dissertation we present new models and solution algorithms that portray different BSPs and attempt to capture the operational environment of a container terminal, while at the same time including attributes of the system that current models lack. The formulations and solutions of mathematical models presented herein, seek to optimally schedule vessels and/or quay cranes to berths in multi-user type of container terminals, without losing its applicability to the private type container terminals. The objective is to present models and algorithms that capture as much as possible of current container terminal operator practices, while minimizing the assumptions made about real world conditions that container terminals operate in.

Acknowledgement and Dedication

I would like to extend my deepest gratitude to my advisors, Dr. Maria Boilé and Dr. Sotirios Theofanis, for their guidance and support throughout the completion of this thesis. This work would not have been possible without their ability to stimulate imagination and creativity. I have the deepest respect for Dr. Maria Boilé and Dr. Sotirios Theofanis both as scientists, educators, and as my friends. It has been an honor and privilege to work under their supervision.

I would like to thank Dr. Athanasios Ziliaskopoulos, who has been an inspiration during the completion of this thesis. His influence on this dissertation has been more then he will ever acknowledge himself.

I am grateful to the committee members Dr. David Coit, Dr. Ali Maher, and Dr. Trefor Williams, for their critical review and constructive suggestions during the course of this work.

In addition, I would like to extend my sincere appreciation to Dr. David Madigan, and Dr. Demosthenis Angelides, well-known practitioners, researchers, and educators. Their influence and stimulus has been tremendous. I consider myself a very lucky person to have been educated by such professors.

Last and certainly not least I would like to express my love to my family; my sisters Panagiota and Alexandra, and my parents Ioannis Baltzidis and Maria Anninou who have always been there for me with love, support, home cooking, and especially sense of humor. I owe them everything and I hope they know it.

This thesis is dedicated to the memory of Mihail P. Anninos, Aristidis Valtzidis, and Panagiotis M. Anninos.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

1. INTRODUCTION

There are more than 2,000 ports around the world, ranging from single-berth locations handling a few hundred tons of cargo per year, to multipurpose facilities handling up to 300 million tons of cargo per year. In 2004 the world port traffic was made up of 36% liquid bulk products (mainly oil, petroleum products, and chemicals), 24% dry bulk goods (coal, iron ore, grain, bauxite, and phosphate), and 40% general cargo (UNCTAD, 2004). The later type of cargo is mainly being transferred using containers. Containers are large metal boxes made in standard dimensions and measured in multiples of 20 feet called "twenty-foot equivalent units" or in short TEU. Port terminals that handle containers are called container terminals and have different and more complex operations than passenger and dry or liquid bulk ports.

The container terminal industry has received increasing attention in the last 20 years with containerized transportation coming to the forefront of the international shipping scene. The last decade has seen an increase in container traffic that has surpassed any expectation (fig. 1.1). The world container terminal throughput for 2004 reached 348 million TEUs, an increase of 38 million TEUs from the 2003 level of 310 million TEUs[1]. To respond to the projected volumes and exploit economies of scale liner-shipping companies are investing in larger container vessels. On the other hand double-digit volume growth numbers is the norm for most of the busiest ports around the world, while some other ports, particularly the ones located in East Asia, report increase up to 140%, in the numbers of TEUs handled (Yantai: 141%; Suzhou: 57%; Nantong: 40%). According to CI-Online Asian ports are expected to burst through the 100% utilization

[1] In 2005 a decrease of 12.5 million TEUs was observed (***Source***: CI-Online)

level of their designed capacity, while US and European ports will be operating at over 80% utilization.

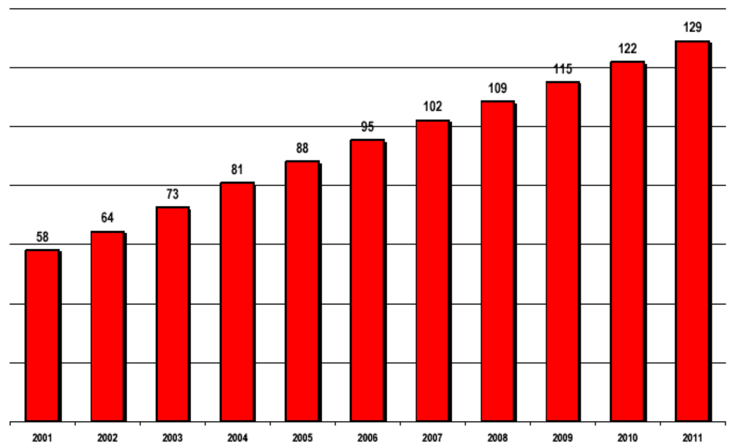


Figure 1-1 World Containerized Trade in million TEUs, 2001-2011-Forecast (Source: UNCTAD Transport Letter, No 32, Second Quarter 2006)

In maritime container transportation the hub-and-spoke arrangement is widely adopted. Deep-sea vessels, also called mother vessels, operate between a limited number of transshipment terminals (hubs). Smaller vessels (feeders) link the hubs with the other ports (spokes). This network topology results in the consolidation of capacity along the routes linking the transshipment ports and in the growth of their importance. In recent years, mother vessels have strongly increased in size, attaining up to 10,000 TEUs, while larger sizes are planned[2]. Transshipment ports are large intermodal platforms. A limited number of them handle an important share of the world traffic.

[2] The words largest container vessel currently on route (September, 2006) has a capacity of 11,000 TEU (MAERSK). In total 46 vessels exceed 8,000 TEU capacity, while the first 10,000 TEU vessels are expected for delivery in March 2008 (***Source:*** CI-Online, February 8, 2006). The biggest to date vessel in use is the EMMA a container ship owned by the A.P. Moller-Maersk Group with an estimated capacity of over 13,000 TEUs.

According to CI-Online, in 2000 the top 20 container terminals in TEUs handled 44% of the total TEU traffic, while in 2005 this number increased to 55%[3] (fig. 1-2).

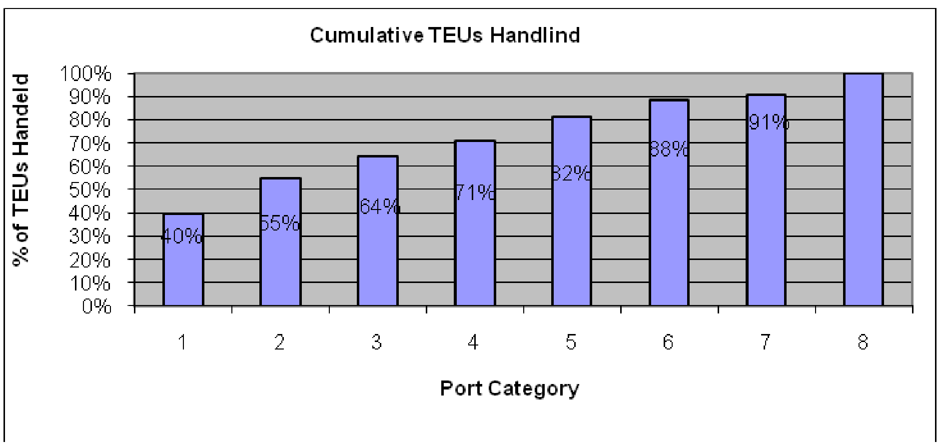
Based on their customer base, container terminals can be distinguished into: a) Privately leased and operated terminals by shipping lines (referred as dedicated terminals-DT), and b) Owned by an operator who provides berthing space to each shipping company based on the terms of a contract (PT), or on other service agreements (e.g. **F**irst **C**ome **F**irst **S**erved policy). In major ports (in countries like Japan and the US) shipping lines lease the container terminals in order to be directly involved in the processing and handling of the containers as they aim to achieve higher productivity and economies of scale.

The increasing number of container shipments combined with the growing size of container vessels causes higher demands on container terminals, container logistics, and management, as well as on technical equipment resulting in increased competition between ports, especially between geographically close ones. The ports mainly compete for ocean carrier patronage and related container volumes, as well as for the land-based truck and railroad services[4]. The competitiveness of a container seaport is marked by different factors, including the time in port for vessels (turnaround time) combined with low rates for loading and discharging (Hulten 1997; Muller 1995).

The strong competition among PT ports leads to the necessity of using the highly expensive terminal resources such as berths, storage yards, quay cranes, straddle carriers, automated guided

[3] The first 40 ports in TEUs-handled, handle over 70% of the total world TEUs traffic
[4] It is the state of the practice for the large vessel operators to arrange for feeder vessels services on their own, thus port operators need to compete only for the large vessels

vehicles, stacking cranes etc. as efficiently as possible. A key factor of success is the optimization of these logistic processes (Daganzo, 1990; De Castilho and Daganzo, 1993; Taleb-Ibrahimi et al., 1993). It is clear that rapid turnover of containers, which corresponds to a reduction of the container vessel turnaround time, and of the costs of the process, is a crucial competitive advantage for a port. With respect to the terminal operations this can translate to the minimization of the time a vessel is at the berth as an overall objective (Steenken et al., 2004). Customer satisfaction focuses on waiting times rather than charges since inequitable waiting time for vessels may not make the port attractive for carriers even if the port charges are low[5] (Imai et al. 2003).



Port Categories: 1: First 10 Ports, 2: First 20 Ports, 3: First 30 Ports, 4: First 40 Ports, 5: First 50 Ports, 6: First 60 Ports, 7: First 80 Ports, 8: Total

Figure 1-2 Cumulative TEUs Handled by Port Groups: Port Category/Number of Ports per Category (Data Source: CI-Online)

[5] Although according to Fung et al. (2006) high terminal handling charges (THCs) lowered Hong Kongs container terminal throughput

Research Scope

Global container terminal operating companies have faced strong criticism from their ocean carrier customers for most of this year as a combination of marine and landside congestion has led to deteriorating levels of service (CI-Online, September 2007). Fierce terminal competition and the need to maximize recourses utilization have led marine terminal operators to the development and application of a rich variety of Berth Scheduling Policies (BSPs) to deal with the marine side of the congestion. Container terminal operators seek for the efficient BSPs that may reduce vessels turnaround time, increase port throughput, leading to higher revenues and increased competitiveness of the port, while at the same time keeping the customers' satisfaction at a desired level (usually set by contractual agreements).

Several issues arise when defining the best berth-scheduling policies (BSP) for each port operator and depend on several factors including the type and function of the port (DT or PT, transshipment hub etc), the size of the port, the location, nearby competition, type of contractual agreement with the vessel carriers and other. Some of these BSPs and issues have to a certain extend been captured by academic research, but still several attributes need to be investigated and included for these models to represent the state of the practice of container terminal operations. For example the majority of the models do not explicitly deal with the relationship of customer satisfaction and the port operator costs/benefits. Furthermore most of these models ignore the multi-objective and stochastic environment that port managers have to operate in and make future decisions.

In this dissertation we present new models portraying BSPs that attempt to capture the operational environment of a container terminal and include some of the attributes of the system that current models lack. The formulation and solution of mathematical models presented herein, will seek to optimally schedule vessels to berths in PT type of container terminals, without losing its applicability to the DT container terminals. The objective is to present models that reflect BSPs that capture as much as possible of the current container terminal operator practices, while minimizing assumptions made about real world conditions. Insight will be provided of how these BSP may be used by the port operator to negotiate future contracts with shipping companies.

This dissertation may be broken down into four major sections. The first section, composed by Chapters 5, 6, 7, and 8, deals with the discrete berth allocation problem presenting a generic formulation and four solution approaches for the problem. The next section (Chapter 9) focuses on the continuous berth allocation with simultaneous quay crane scheduling and presents a generic formulation and a solution approach. The third section, (Chapters 10 and 11) deals with the multi-objective berth allocation problem and investigates the applicability of a multi-objective optimization environment for the BAP. In these two chapters we present a general formulation of the multi-objective discrete BAP, a solution approach, and a heuristic to obtain the optimal Pareto set. The last section (Chapter 12) deals with the stochastic aspect of the berth allocation problem and investigates the applicability of stochastic modeling to the BAP, presenting a general formulation and four heuristic solution approaches.

2. SYSTEM DESCRIPTION

Container Terminals

Container terminals are open systems of material flow with two external interfaces: the quayside where containers arrive/leave via vessel and the landside where containers arrive/leave the terminal via trucks or trains. Within the terminal we can distinguish three areas: the berth area where vessels are berthed for service, the storage yard area where containers are stored waiting to be exported or imported, and the terminal gate that connects the container terminal to the hinterland. Accordingly operations in a container terminal can be broken down to three categories: seaside operations, landside operations, and yard operations, all of which interact with each other. Seaside operations consist of the vessels' berthing operations at the quay, and the loading and unloading of containers onto the vessel. The seaside operations interact with the yard operations via the internal transport equipment used to transport containers from/to the vessel and to/from the storage yard. The yard operations manage the containers during the transfer between the landside and the seaside. It includes operations such as the internal transport of the containers from/to the vessel and from/to the trucks/rail, and the storage operations in the storage yard. The landside operations deal with activities of receiving and delivering inbound and outbound containers to and from the storage yard. While each system can be viewed as an independent entity, and its' operations are usually studied as such, interactions between the systems are unavoidable and play a crucial role in the efficient management and operation of a container terminal. A schematic description of a container terminal operations and interactions between the different systems is portrayed in figure 2-1.

Although all container terminals serve the same purpose, they are mainly differentiated on the handling/internal transport/stacking equipment that they use. The most common systems are: a) the tractor-trailer (all chassis) system, b) the straddle carrier direct system, c) the straddle carrier relay system, d) the yard gantry crane system, and e) the front end loader (reach stacker) system (direct or relay). Combinations of these systems can also be found at several container ports. Determining the type of equipment that should be used is usually viewed in terms of a cost VS productivity relationship, but is also influenced by historical, social and cultural aspects (i.e. US ports use mainly conventional systems even though the cost benefit ratios are probably in favor of automated systems, and chassis only systems, which is found nowhere else in the world).

Terminal System and Equipment Overview

Seaside Operations

Vessel Berthing

Vessel operation consumes a large portion of the turnaround time of containerships in ports. Different types of vessels are serviced in a container terminal ranging from deep-sea vessels with a loading capacity up to 13,000 container units (TEU) to feeder vessels with a capacity up to 4,000 TEU. When the vessel arrives at a port it has to moor at the quay. For this reason a number of berths are available by the port operator to service the vessels. A typical berth can accommodate a number of vessels depending on the length of the quay. Before the vessels' arrival, information on the type of the vessel, the number of containers to be (un)loaded, and a proposed arrival and departure time is sent to the terminal operators. When planning for berth allocation and usage, the berthing time and the exact position of each vessel at the quay, as well as various quayside resources are determined. Several variables are considered, including the length-overall and (expected) arrival time of each vessel, the number of containers for discharging and loading on each vessel, and the storage location of inbound/outbound containers to be loaded onto/discharged from the corresponding vessel.

Quay Cranes

After a vessel is berthed a number of quay cranes (QC), a subset of the assigned cranes to the specific berth, are used in order to load and unload the vessel based on the vessels' stowage plan (fig 2-2). Depending on the vessel's size commonly two to five cranes operate on deep-sea vessels, and one to three cranes on feeder vessels. Two commonly used types of quay cranes at medium and large container terminals are the single-trolley cranes and dual-trolley cranes. The trolleys travel along the arm of a crane and are equipped with spreaders, which are specific devices to pick up containers. Modern spreaders allow the move of two 20 ft containers simultaneously (twin-lift mode). Conventionally single-trolley cranes are engaged at container terminals. Dual-trolley cranes represent a new development only applied at very few terminals. The maximum performance of quay cranes depends on the crane type. The technical performance of cranes is in the range of 50–60 boxes/hour, while in operation the performance is in the range of 20–30 boxes/hour. The time required for loading/unloading operations depends on the cycle time of the quay cranes and transfer cranes, and on the relative position of the berth place to the designated container stocking blocks for the vessel. Also, the cycle time of a quay crane depends on the loading sequence of slots and is affected by the loading sequence of containers in the yard.

At small container terminals mobile cranes may also be used. These cranes may also be used in medium to large size container terminals as *backup* quay cranes for special occasions (i.e. increase productivity on a specific vessel without interrupting service of other vessels).

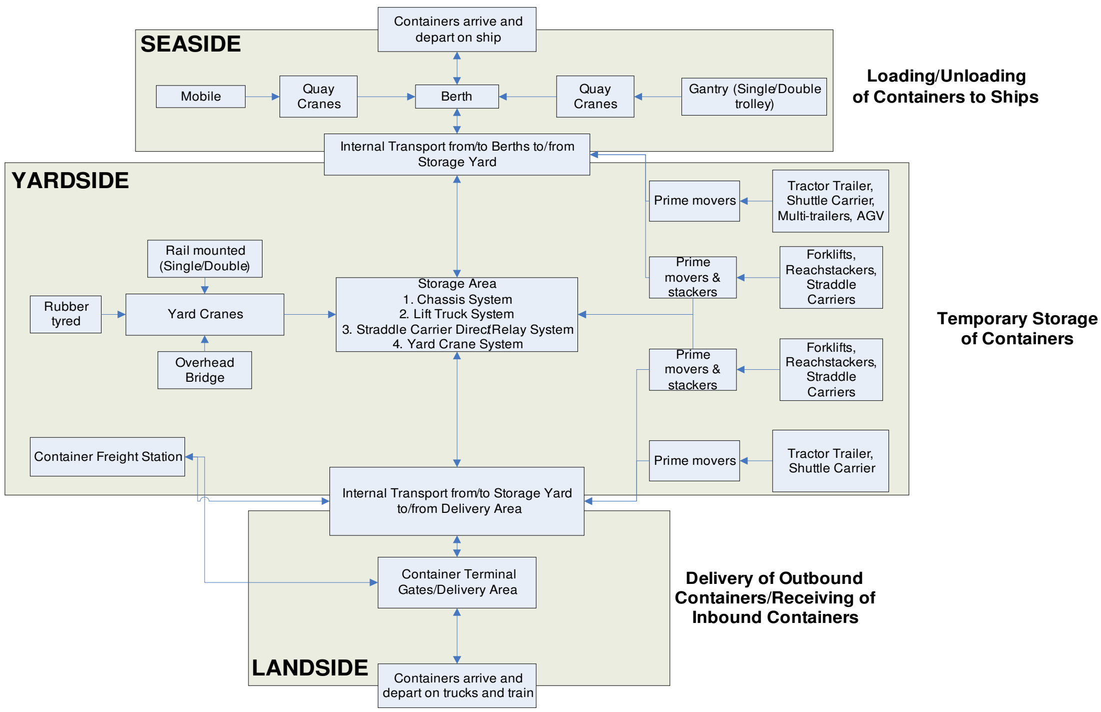


Figure 2-1 Schematic Representation of Container Terminal Activities and Operations

Stowage Plan

Loading of export containers onto a vessel is based on a stowage plan. Stowage planning involves finding the optimal plans of positioning containers into a container vessel, with respect to a set of structural and operational restrictions, and is performed by the terminal operators based on a plan provided by the shipping company. In contrast with the unloading process, there is hardly flexibility in the loading process. Containers are placed on the vessel in a last-in-first-out manner and therefore temporary unloading and reloading in subsequent ports along the route (shifting) is common and results in high costs (a shift move is regarded by many terminal operators as a regular container move regardless that the container is unloaded and loaded on the same vessel).

Storage Yard

Containers that arrive at the terminal (inbound or outbound) usually have to be stored for a certain period of time, usually less than a week. For this reason a designated area in the container yard is reserved. For the time period that containers remain in the terminal they are stored at designated areas, within the terminal, known as the container or storage yards. Container yards are divided into two categories based on the storage system that they use: a) storage on chassis, and b) storage on the ground. Though the former way of storing containers provides flexibility and high performance (with a chassis system each container is individually accessible) it requires amplitude of land, something that today's terminals do not have. In most container terminals space is limited and so stacking containers on the ground is the most common approach. Although stacking containers saves space it creates a significant increase of operational time.

The container storage area is usually separated into different stacks (or blocks), which are differentiated into rows, bays and tiers. A typical block has seven rows (or lanes) of spaces, six of which are used for storing containers in stacks or columns, and the seventh reserved for truck passing to pickup or deliver a container when yard cranes are used for the stacking of the container. Each row typically consists of over twenty TEU container stacks stored lengthwise end to end. For storing a 40 ft. container stack, two TEU stack spaces are used. Some stack areas are reserved for special containers like reefers, which need electrical connection, dangerous goods, or overheight/overwidth containers, which do not allow for normal stacking. A usual policy is to separate stacks into areas for export, import, special, and empty containers. For the later category of containers, due to space limitations, storage areas also exist outside the marine terminal, and usually the majority of these containers are stored there.

Containers are distinguished into inbound (import), outbound (export), and transshipment. The former are containers that arrive via vessel, unloaded and then usually placed in a designated area in the container terminal, until they are picked up by truck or rail to be moved to their inland destination. Outbound containers arrive on trucks or rail at the terminal, usually few days before the arrival of the vessel to be loaded on (ranging from one to nine depending on the ports policy with few extremes over 10 days), and are also placed in the storage yard. Transshipment containers on the other hand arrive on a vessel, are unloaded in the port and then loaded onto another vessel having as a final destination another port.

Inbound and outbound containers have different arrival patterns. Inbound containers arrive at the port predictably and at large batches and depart one by one in a more unpredictable manner (Sideris et al., 2002). This demands flexible handling and explains why straddle carries or rubbertyred cranes are generally chosen for import operations. On the other hand, outbound containers arrive at the port in a random way and are stored on land, and depart the port in batches. Figure 2-2 presents an illustrative dwell time distribution of import and export containers with the assumption that the ports' policy is: a) not to receive export containers earlier than 10 days from the vessels' arrival, and b) the dwell time of import containers is limited to 7 days.

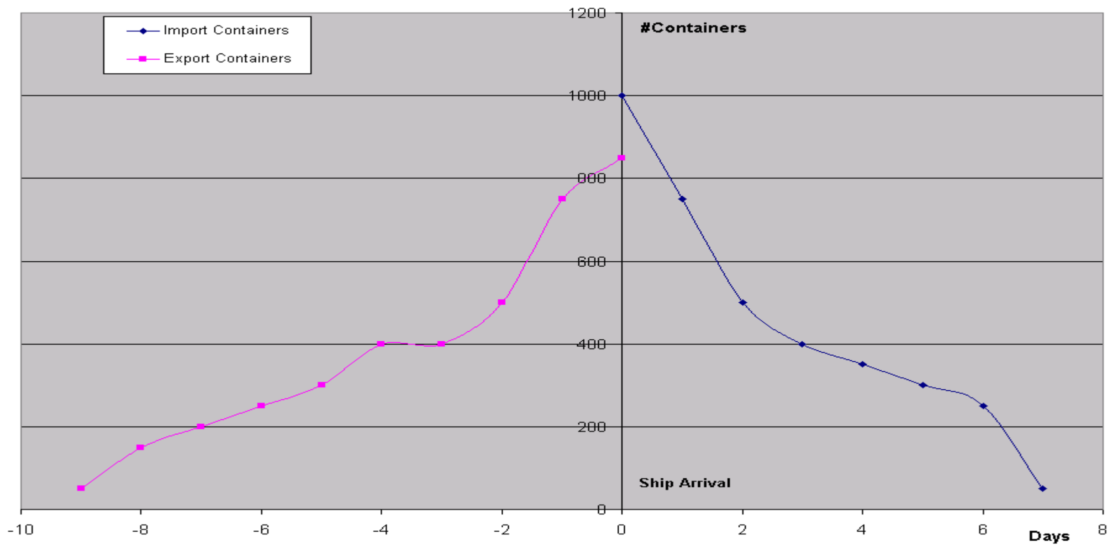


Figure 2-2 Theoretical Distribution of Dwell Time for Import and Export Containers

Storage Yard Equipment

One of the main decisions to be made when designing the storage system (planning level) of a container terminal is the type of stacking equipment to be used. These include: forklift trucks, reachstackers, yard cranes and straddle carriers. Forklifts and reachstackers are usually used to move and stack light containers (like empty ones). The main equipment used for stacking containers on stacks though are yard cranes and straddle carries. The latter can also (and are in many cases) used for inter-terminal transport of containers (e.g. from/to the vessel to/from the storage yard). There are three types of yard cranes: rail mounted gantry cranes (RMG), rubber tired gantries (RTG), and overhead bridge cranes (OBC). Rubber tired gantries are more flexible in operation while rail mounted gantries are more stable (more productive when working in one block) and overhead bridge cranes are mounted on concrete or steel pillars. Commonly gantry cranes span up 6–12 rows and allow for stacking containers 4–10 high. To avoid operational interruption in case of technical failures and to increase productivity and reliability, two RMGs are often employed at one stack area. A new development in yard cranes is, the so-called, double-RMG systems. They consist of two RMGs of different height and width able to pass each other thus avoiding a handshake area. This results in a slightly higher productivity of the system. The technical performance of gantry cranes is approximately 25 moves/hour. Similarly, these types of cranes can be used for the loading and unloading of trains.

Internal Transport

As mentioned previously containers need to be transported from/to the quay and from/to the gate to the storage yard. A variety of vehicles are employed for the horizontal transport of containers within a terminal, both for the vessel-to-shore transportation and the landside operation. The transport vehicles can be classified into two different types: vehicles that are not able to lift containers by themselves and vehicles that have the capability of lifting containers. For the first class loading and unloading of containers is done by cranes (quay cranes at the seaside and gantry cranes at the landside). Trucks with trailers, multi-trailers and automatic guided vehicles belong to this class of transport vehicles. It should be noted that for the internal transport between storage yard and delivery area multi-trailers and AGVs are not used.

Transport vehicles of the second class are straddle carriers, forklifts, and reachstackers. The former ones are the most important since they can also be used for the stacking operation in the stack yard. Therefore, they can be regarded, as 'cranes' not locally bound, with free access to containers independent of their position in the yard. The straddle carriers' (SC) spreader allows the transport of either 20 or 40 ft. containers (recently twin mode SC able to transport/stack two 20' containers simultaneously is becoming available, but as of today are only used in a handful of terminals around the world). Straddle carrier systems are very flexible and dynamic. Straddle carriers exist in numerous variants. Usually straddle carriers are man-driven and able to stack 1 over 2 or 1over 3. During the last years progress was made to develop automatic straddle carriers (Stenken, 2004). An automatic straddle carrier system has gone into production at Patrick Terminal/Brisbane, Australia.

Terminal Gates/ Delivery Area

The receipt/delivery operations at the terminal gates is the fourth and final (after unloading, transfer to the yard and storage) sequence of activities in the terminal operation for inbound containers, and the first for outbound. Containers arrive at a terminal from the landside either via truck or rail. Trucks arrive at the terminals' in-gate where the data of the containers have to be checked and then drive to transition points where the containers are loaded or unloaded by the internal equipment. (Maher container terminals at New Jersey, US are estimated to handle over 13,000 trucks per day in the next years). Landside operations for the rail part are similar, but not identical, to the quayside operations. A loading plan exists that specifies which container has to be placed on each wagon, depending on its destination, type, weight, the wagons maximum load etc. We should note that large container terminals serve some thousand trucks a day and efficient gating systems are crucial not only to the levels of congestion within the terminal but also to the surrounding street network of the port. Although delays to container handling caused by the receipt/delivery are no more serious than those arising elsewhere in the container terminal, they are the most obvious and visible to inland transport operators, and are particularly damaging to the terminals' reputation (Portworker Development Program, 1999).

Container Freight Station (CFS) Facilities

Door-to-door container transport may not be possible (container arrives at destination without being stripped) because the shipper or receiver: a) has insufficient cargo to make up a full container load, b) lacks suitable equipment or facilities for handling, c) has the premises that cannot accommodate container vehicles, d) has no road connections suitable for container vehicles, and/or e) has no rail or inland waterway connections to the premises. In these cases, and when an inland clearance depot (ICD) is not available[6], the usual alternative strategy for the cargo owner is the use of a container freight station (CFS) located at the container terminal.

The CFS is a cargo consolidation area, container packing/unpacking and cargo distribution center equivalent to an ICD. Shippers can have their cargoes transported in break-bulk form, by road, rail or inland waterway, to the CFS for consolidation and packing into containers ready for loading aboard a vessel. Similarly, receivers of goods can arrange for them to be unpacked from

[6] And for several other reasons like a shortage of suitable road vehicles, rail wagons or barges to transport the containers; container owners and lessors may place too high a price on the movement of containers inland from the ports; customs authorities may insist on examining the contents of all containers before allowing them out of the port, so reducing the benefits of door-to-door containerization etc.

containers at the CFS, separated into break-bulk consignments, and collected on their behalf by the most convenient form of inland transport. At the CFS the following broad functions are performed: receive, sort and consolidate export break-bulk cargoes from road vehicles, rail wagons and inland waterway craft; pack export cargoes into containers ready for loading aboard a vessel; unpack import containers, and sort and separate the unpacked cargoes into break-bulk consignments ready for distribution to consignees; deliver import cargoes to inland transport—road vehicles, rail wagons and inland waterway craft; store import and export cargoes temporarily, between the times of unloading and loading, while various documentary and administrative formalities are completed. The facilities and resources of the CFS are all designed to carry out these basic functions effectively and efficiently.

3.LITERATURE REVIEW

Introduction

Container terminals are open systems of material flow with two external interfaces: the quayside where containers are unloaded/loaded onboard the vessel and the landside where containers are delivered to/from the terminal via trucks or trains. Within the terminal we can distinguish three areas: the berth area where vessels are berthed for service, the storage yard area where containers are stored temporarily waiting to be exported or imported, and the terminal gate that connects the container terminal to the hinterland. Accordingly, operations in a container terminal can be broken down to three categories: seaside operations, landside operations, and yard operations, all of which interact with each other. Seaside operations consist of the vessels' berthing operations at the quay, and the loading and unloading of containers onto the vessel. The seaside operations interact with the yard operations via the internal transport equipment used to transport containers from/to the vessel and to/from the storage yard. The yard operations manage the containers during the transfer between the landside and the seaside. It includes operations such as the internal transport of the containers from/to the vessel and from/to the trucks/rail, and the storage operations in the storage yard. The landside operations deal with activities of receiving and delivering inbound and outbound containers to and from the storage yard. While each subsystem can be viewed as an independent entity, and its' operations are usually studied as such, interactions between the systems are unavoidable and play a crucial role in the efficient management and operation of a container terminal. A schematic description of a container terminal operations and interactions between the different systems is portrayed in figure 1.

Although seaside operations are interrelated with container transfer and storage yard operations, they present a special interest on their own regarding the relationship between the shipping lines and terminal operators. The tremendous increase of containerized trade during the last years, the resulting congestion in container terminals worldwide, the remarkable increase in containership capacity, the increased operating cost of container vessels and the adoption by shipping lines of yield management techniques, originally adopted by the airline industry, strain the relationships between shipping lines and container terminal operators. Shipping lines want their vessels to be served immediately upon arrival or according to a favorable priority pattern and complete their loading/unloading operations within a prearranged time window, irrespective of the problems and shortage of resources terminal operators are facing. Therefore, in many cases allocating the scarce berthing resources is considered to be a problem deserving both practical and theoretical attention.

This chapter presents a comparative and analytical, up-to-date, review of the existing research efforts relating to berth planning. Existing models have been critically reviewed based on their a) efficiency in addressing key operational and tactical questions relating to vessel service, and b) relevance and applicability to different berth planning marine terminal operator strategies and contractual service arrangements between terminal operators and shipping lines. Strengths and deficiencies of the existing models to address real world problems in a systematic and coherent manner are being discussed. The chapter concludes with a critical overview of issues to be addressed to make these models more relevant to real world applications.

The Berth Allocation Problem (BAP)

As mentioned earlier, one of the most critical operations inside a container terminal is the berth planning process, which has an immediate effect on the vessel's turnaround time. In the past, terminal operators applied First Come First Served (FCFS) service policies in allocating berth space. The increasing trade volumes, vessel size and the restructuring of the shipping lines service networks were eventually followed by customer service agreements and differentiation service policies that deviate from the FCFS rule. This in turn calls for a more sophisticated and informed resource utilization of the berthing capacity. Due to the high cost of building new berth capacity, container terminal operators and managers prefer solutions that focus on the operational aspects of berth planning and less on the strategic/tactical. In this chapter the various aspects of the Berth Allocation Problem (BAP) are reviewed.

The BAP can be simply described as the problem of allocating berth space for vessels in a container terminal. Vessels usually arrive over time and the port operator needs to assign them to berths to be serviced (unload and load containers) as soon as possible. Shipping lines and therefore vessels compete over the available berths and different factors, discussed in detail later, affect the berth and time assignment. The BAP has two planning/control levels: the strategic/tactical, and the operational. At the strategic/tactical level the number and length of berths/quays that should be available at the port are determined. This is done either at the initial development of the port or when an expansion is considered. At the operational level, the allocation of berthing space to a set of vessels scheduled to call at the port within a few days time horizon has to be decided upon. Since liner shipping vessels follow a regular schedule, in most cases the assignment of a berth to the vessel has to be decided upon on a regular and usually periodical basis. At the operational level the BAP is typically formulated as combinatorial optimization problem (i.e. machine scheduling problem, 2D packaging problem).

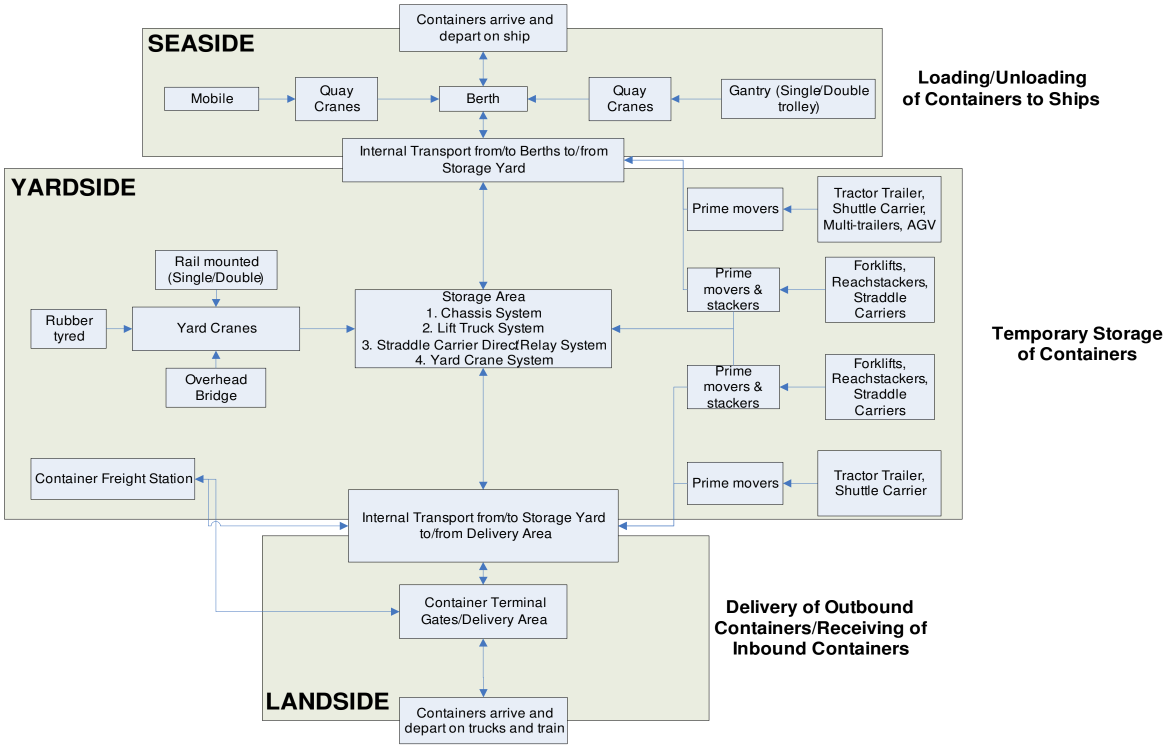


Figure 3-1 Schematic Representation of Container Terminal Activities and Operations

After the BAP has been solved, the resulting Berth Scheduling Plan (BSP) is usually presented using a space-time diagram. Figure 2 presents a simple example of space-time diagrams applied to berth planning. The x-axis represents the time and the y-axis the berth(s). Each rectangle represents a vessel. The rectangles and their size correspond to the berth space required and the service time.

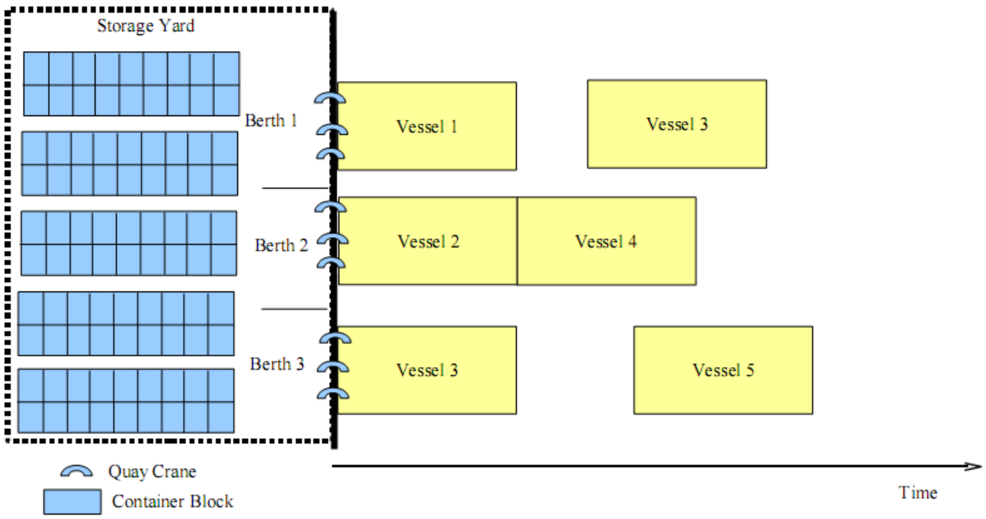


Figure 3-2 Space-Time Diagrams of Berth Planning

The BAP can be considered and formulated according to the following variations a) Discrete versus Continuous Berthing Space, b) Static VS Dynamic Vessel Arrivals, and c) Static VS Dynamic Service Time. The BAP can be modeled as a discrete problem if the quay is viewed as a finite set of berths, where each berth is described by fixed-length segments or as points. Usually though vessels are of different lengths and dividing the quay into a set of segments is difficult mainly due to the dynamic change of the length requirements for each vessel. One solution to this problem is to use longer segments, a solution resulting poor space utilization), or short segments (infeasible solutions). To overcome these drawbacks continuous models have appeared in the literature where vessels can berth anywhere along the quay. In the discrete case, the BAP can be modeled as an unrelated parallel machine-scheduling problem (Pinedo, 2002), where a vessel is treated as a job and a berth as a machine, whereas in the continuous case as a packaging or the two dimensional cutting stock problem. The BAP can also be modeled as a static problem (SBAP) if all the vessels to be serviced are already in the port at the time scheduling begins or as a dynamic problem (DBAP) if all vessels to be scheduled for berthing have not yet arrived but arrival times are know in advance. Service time at each berth depends on several factors, with the two most important being the number of cranes operating on each vessel and the distance from the preferred berth position (PBP). If the model does not take under consideration the number of cranes to be operating at each vessel then the problem can be considered as static in terms of the handling time. On the other hand if the number of cranes that will work on each vessel is decided upon from the model then the formulation can be considered as dynamic in terms of the vessels service times. Finally technical restrictions such as berthing draft, inter-vessel and end-berth clearance distance, that bring the problem formulation closer to real world conditions, are further assumptions that have been adopted by researchers.

Table 3-1 summarizes the several categories of BAP variations and the respective assumptions that have appeared in the literature. The models that have appeared in the literature usually combine two or more of these assumptions. In most cases the formulation of the problem leads to NP-hard or NP-complete problems that require heuristic and meta-heuristic to be employed for a computationally acceptable solutions time.

Table 3-1Berth Plan Model Variations

Assumption	Description
Static VS Dynamic Berthing	Static: All vessels are in the port when the berth plan is determined Dynamic: All vessels are not in the port when the berth plan is determined
Discrete VS Continuous Berth Space	Discrete: Allocation scheme based on the berth Continuous: Allocation scheme not based on the berth
Static VS Dynamic Handling Time	Static: Constant Handling Time Dynamic: Handling Time Depends on several parameters (i.e. Vessel size, Berthing location, Quay Crane assignment etc)
Technical/Operational Restrictions	Inter vessel clearance, End Berth clearance etc

Literature Review

In this section a complete and up-to-date, in our opinion, literature review on BAP is provided. The advantages and limitations of each approach are presented along with a brief description of the model formulation.

Table 3-2 summarizes the most important papers that have appeared in the literature that dealt with the BAP. The first column states the main author and the date of publication. The second column states the BSP (dynamic/static, discrete/continuous) while the third and fourth the objective and the problem formulation. Since most of the formulations lead to NP-Hard or NP-Complete problems the last column states the adopted solution approach, which as can be seen is usually some sort of (meta)heuristic.

The first paper to appear on the BAP problem was by Nikolaou (1969), followed by a paper by Sabria and Daganzo (1989). This research focused on queuing theory using a serial representation of the problem. Issues related to the applicability of this modeling approach to the BAP were discussed by Edmond and Maggs (1978). After 1990, research in the BAP has solely focused on mathematical programming and simulation, since queuing models failed to capture several of the attributes of the problem. The only exception was a study by Legato and Mazza (2001) and a study by Dragovic et al (2006). The later used queuing theory and simulation to evaluate the efficiency of models used at the Pusan East Container Terminal, while the former presented a closed queuing network model and performed simulation for estimating congestion effects at the berth.

One of the early works that appeared in the literature and did not follow a serial approach was by Lai and Shih (1992). The authors assumed that a wharf, represented as a continuous line that could be partitioned into several sections, to each of which only one vessel could be allocated at a specific time. The berth-allocation rules considered the following factors: the available sections of a wharf, the expected completion time of a vessel at each available section, and size and arrival time of each vessel. A heuristic algorithm was developed considering a first-come-first-served (FCFS) rule. Through a simulation experiment three berth-allocation rules for container vessels are compared.

Brown et al. (1994, 1997) treated the BAP in naval ports. They identified the optimal set of vessel-to-berth assignments that maximizes the sum of benefits for vessels while in port. Berth planning in naval ports has important differences from berth planning in commercial ports though. In the former, a berth shift occurs when for proper services, a newly arriving vessel must be assigned to a berth where another vessel is already moored.

Imai et al. (1997) was the first to introduce the idea that for high port throughput, optimal vessel-to-berth assignments should be found without considering the FCFS basis. However, this may result in some vessels' dissatisfaction regarding order of service. In order to deal with the two criteria to evaluate, i.e., berth performance and dissatisfaction on order of service, they developed a heuristic algorithm to find a set of non-inferior solutions while maximizing the former and minimizing the later. They introduce of a multi-objective approach, new to machine scheduling problems. A two objective non-linear integer program is formulated to identify the set of non-inferior berth allocations to minimize the dual objectives of overall staying time and dissatisfaction on order of berthing. Dissatisfaction was expressed as the sum of the number of cases in which a vessel arrives later than a particular vessel and is moored earlier. After defining the two-objective non-linear IP they reduced the problem into a single objective problem consisting of the summation of the waiting times and dissatisfaction. From numerical experiments, it was concluded that the trade-off increases if the size of the port increases. Their berthing principle, however, could not treat the dynamic allocation. This paper was the only paper that looked specifically in the customer satisfaction problem, even though it was done in an aggregate way.

Li et al. (1998) formulated the BAP as a scheduling problem with a single processor through which multiple jobs can be processed simultaneously. The problem assumed that all vessels had already arrived, and the minimization of the make-span was attempted based on that assumption. They present two cases of the problem: a) the fixed position case, and b) the non-fixed position case. The authors stated that both cases could be applicable to the berth-scheduling problem under different assumptions (infinite and negligible vessel setup time and cost for job interruption/position change after the job has started). They also consider the case where the processor is partially available. They suggested a first-fit-decreasing heuristic rule for which numerical experiments were conducted for all three cases. The results showed that the average relative errors of the heuristics are less than 20% among all the parameters tested, and the results suggest that the heuristics are effective in producing a near-optimal solution.

Similar to Li et al. (1998), Guan et al. (2002) considered the berth allocation problem as a multiprocessor task scheduling. A vessel (job) is assigned to a number of cranes (m parallel processors/decision variables). They developed a heuristic to minimize the total weighted completion time of vessel service and performed worst-case analysis. Weights were assigned to each job, dependent on the vessels size. They considered two cases of weights both as different functions of the vessels size. No priority service rules were employed, though the weighting of the jobs could be considered as a form implicit rule implementation, and no numerical examples were presented or discussed.

Lim (1998), along with Li et al. (1998), addressed the continuous BAP, with the objective of minimizing the maximum amount of quay space used at any time with the assumption that once a vessel is berthed, it will not be moved to any place else along the quay before it departs. He also assumed that every vessel is berthed as soon as it arrives at the port. This approach is very restricting since it does not solve the problem in which the berthing time is a decision variable and the handling time varies along the quay. The problem was represented as a graph with directed and undirected edges and transformed into a restricted version of the two-dimensional packing problem. A heuristic was presented that performed well under historical data.

Unlike Lim (1998), Imai et al (2001), Brown et al. (1994) and Lai and Shi (1992) Park and Kim (2002) consider the continuous DBAP and used the sub-gradient optimization technique. Their objective was to estimate the berthing time and location by minimizing the total waiting and service time and the deviation from the preferred berthing location. They are the first ones to include penalization of the deviation from the optimal berth.

Tong et al. (1999) were the only ones to follow the implementation of the Ant Colony Optimization (ACO) technique and showed how it can be applied effectively to solve the BAP. The objective was to minimize the necessary wharf length subject to several space and time constraints. Experimental results, with problems that dropped the constraints of clearance distance between vessels and fixed and forbidden positions (that do not affect the main focus of the BAP), were presented but no comparison was made to any of the other available methods and there is no indication that the algorithm may perform well in real life problems.

Imai et al. (2001) was the first one to address the DBAP. Their objective was to minimize the sum of a vessels waiting and handling time. Handling time was assumed to be dependent on the berth but was not modeled, as with most of the papers, and was considered deterministic. Computational experiments showed that the proposed heuristic works well from a practical point of view. In the same context Nishimura et al. (2001) addressed the same problem but for a public berth system. In this paper the authors extended the work done by Imai et al. (2001) to include physical restrictions (water-depth and quay length). They also dropped the assumption that each berth can handle one vessel at a time. Service priority relied on the FCFS rule, and was not dependent on the vessels cargo volume, though the authors note that this is a usual constraint. The goal was to optimize service time (including waiting time). A heuristic based on genetic algorithms (GA) is employed to obtain a good solution within a reasonable computational time. Experimental results are presented for both simultaneous and single occupancy of a berth. For small size problems the optimality gap was 10% while for larger size problems 20%.

Hansen and Oguz (2003) criticized the model formulation by Imai et al. (2001) and supported that the formulation was incorrect, presenting a new formulation. Imai et al. (2005) published a Corrigendum to clarify this issue. Several computational experiments were performed for the purposes of this paper, using both the formulations of Imai et al. (2001) and Hansen and Oguz (2003). In all the test instances the same optimal objective function value was obtained. We should note that due to the multi-solution space (fig. 3) different assignments might provide the same optimal objective value.

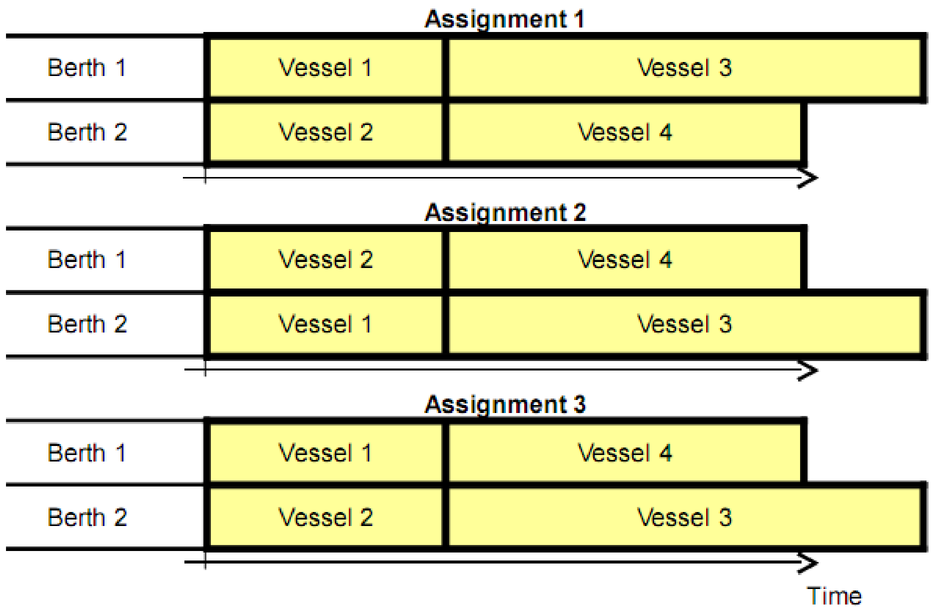


Figure 3-3 Multi-solution space example

Imai et al. (2003) modified and extended the DBAP formulation of Imai et al. (2001) and Nishimura et al. (2001) in order to include service priority constraints. The objective was to minimize the total service time while differentiating priorities to vessels by variation of their service time in the solution. They assumed that only one vessel can be moored per berth at a time, the service time was berth dependent, while no physical/technical restrictions were considered. Several numerical examples are presented using different weight priority formulas and a small discussion on the choice for the value of the weights is provided.

Kim and Moon (2003) also studied the continuous DBAP and formulated a MIP model, similar to Imai et al. (2001), but used simulated annealing instead of the Lagrangian Relaxation to find near optimal solutions. The objective was to minimize delays and handling cost by non-optimal locations of the vessels berthing, attempting to simultaneously determine the berth time and location. Unlike Imai et al. (2001) they apply a cost penalty to berthing in non-preferred berths. Priority service rule inclusion was not stated explicitly, though a penalty was included in the minimization function for the late departure of each vessel. In their paper a comparison of the simulated annealing method to classical optimization technique is presented using a numerical example (a more comprehensive comparison and more details on the model can be found in Moon, 2000). In their comparison the authors use only small instances of the problem because the continuous case of the BAP is intractable by exact methods and results show that the simulated annealing method provided near-optimal solutions within a reasonable time frame.

Park and Kim (2003) extended their previous work to combine the BAP with consideration of quay crane capacities. Their study determined the optimal start times of vessel services and associated mooring locations while at the same time determines the optimal assignment of quay cranes to vessels. They assumed that handling times vary linearly with the number of quay cranes assigned to a vessel, in order to solve the integrated problem. The handling time was considered independent from the mooring location of the vessel. The formulation of the objective function is similar to the ones by Park and Kim (2002) but more elaborate since it includes four different cost penalties (handling cost, early arrival, late arrival, and late delay).

Similar to Park and Kim (2003), Meisel and Bierwirth (2006) presented a model for the Berth and Quay Crane Assignment problem. A heuristic was also provided, based on priority rule methods, for the integrated solution of these problems and computational results based on real world data was presented. The objective was the minimization of unused Quay crane capacity. The authors stated that further research is being conducted.

Table 3-2 Different Berth Planning Assignment Problem Assumptions

Author	*BSF*	*Objective*	*Formulation*	*Solution Approach*
Li et al., 1998	SBAPCQ	Makespan	Single Processor Scheduling	Heuristic
Lim et al., 1998	SBAPC	Amount of Quay	2D Packaging	Heuristic
Guan et al., 2002	SBAPCQ	Total Weighted Completion Time	Multiprocessor Scheduling	Heuristic
Guan & Chen, 2004	SBAPC	Total Weighted Completion Time	MIP	Heuristic
Park & Kim, 2002	SBAPC	Cost From Delayed Departures and Cost of Non-Preferred Berth	MIP	Lagrangian relaxation and sub-gradient optimization
Kim and Moon, 2003	SBAPC	Cost From Delayed Departures and Cost of Non-Preferred Berth , Cost from early or late start of vessel handling against ETA (Estimated Time of Arrival)	MIP	Simulated Annealing
Park and Kim, 2003	CBAPD	Handling Cost, Penalties from Berthing prior or after ETA, Penalties from Late Departures	MIP	Lagrangian Relaxation and Subgradient optimization, Dynamic Programming
Nishimura et al., 2001	DBAPD	Total Completion Time	MIP	GA
Imai et al., 2001	DBAPD	Total Completion Time	MIP	Lagrangian relaxation and sub-gradient optimization
Imai et al., 2003	DBAPD	Weighted Total Completion Time	QAP	GA
Imai et al., 2005	DBAPC	Total Completion Time	MIP	Heuristic
Imai et al , 2007	DBAPC	Total Completion Tim	MIP	GA
Imai et al., 2006	DBAPC	External Berth Service Time	MIP	GA
Cordeau et al., 2005	DBAPC	Total Completion Time/ Weighted Total Completion Time	MIP	Tabu Search Heuristic
Briano et al., 2005	SBAPC	Cost From Delayed Departures & Cost of Non-Preferred Berth	MIP	Heuristic
Lee et al., 2006	SBAPCQ	Makespan & Quay Crane Time	MIP	GA
Wang & Lim, 2006	SBAPC	Position, Delay, & Unallocation Cost	MIP	Stochastic Beam Search
Hansen et	DBAPD	Delayed Departures, Waiting and	MIP	Variable Neighborhood

al., 2007		Handling Cost		Search
Moorthy & Teo, 2006	CBAPD	Operational Cost and Maximize Service Levels	Rectangle Packing	Simulated Annealing
Golias et al. (2007)	DBAPD	Delays from Late and Maximize Premiums from Early/Timely Departures	MIP	GA
Boile et al. (2007)	DBAPD	Multi-objective: Total Service Time and Preferred Customers Satisfaction	MIP	GA

DBAPD: Dynamic BAP Discrete, **DBAPC**: Dynamic BAP Continuous, **SBAPD**: Static BAP Discrete, **SBAPC**: Static BAP Continuous

Lee et al. (2006) following the work of Park and Kim (2003) presented a method for scheduling berth and quay cranes, which are critical resources in container terminals. Excluding simulation models, this paper along with Park and Kim (2003), Li et al (1998), Guan et al. (2002) and Meisel and Bierwirth (2006) are the only two papers that combine berth scheduling and quay crane assignment. A bi-level programming model, in which the berth allocation problem with the objective of minimizing the sum of waiting time and handing time of each vessel is dealt with as upper level problem whilst the quay crane scheduling problem with the objective of minimizing the sum of makespan of all the vessels and the completion time for all the quay cranes is dealt with as lower level problem, is formulated by considering various practical constraints such as interference between the quay cranes. To solve this model, a genetic algorithm is used to determine the near optimal solution. A computational experiment is conducted to examine the performance of the proposed bi-level programming model and algorithm. Unfortunately results are presented for small instances of the problem. According to the author further research is in progress to address the issue of larger instances.

In an unpublished paper Dai et al. (2004) study the static and dynamic berth allocation problem as a rectangle-packing problem with arrival constraints. Given a scheduling window and information on the vessels that will be arriving within the window, they structure the associated berth planning problem as one of packing rectangles in a semi-infinite strip with general spatial cost structure. The goal was to minimize the delays faced by vessels, with higher priority vessels receiving promised level of services, while at the same time address the desirability to berth the vessels on designated locations along the terminal minimizing the movement and exchange of containers within the yards and between vessels. Constraints on draft, equipment availability, etc. are not taken into consideration. They show that a pair of permutations of all the vessels can encode any BP problem and formulate a time and space cost minimization model. They provide a lower bound for the SBAP in order to evaluate the performance of their procedure. They use simulated annealing, with 5 different neighborhood schemes, in order to search the feasible solution space.

Guan and Cheung (2004) presented a berth allocation model that allows multiple vessels moor at a berth, considers vessel arrival time and optimizes the total weighted flow time. Two formulations were presented: a) the Relative Position Formulation (similar to the model of Kim and Moon, 2003; with a slightly different objective function), and b) the Position Assignment Formulation. Following the idea by Imai et al. (2003) they apply a weight coefficient to each vessel limited to vessels departing later then the requested time. They develop a tree procedure and a heuristics that combines the tree procedure with the heuristic in Guan et al. (2002). Computational experiments show that the composite heuristic is quite effective.

Imai et al. (2005) extend their previous work by solving the DBAP in a continuous berth space. They assume that handling time depends on the quay location where the vessel is handled (all the existing BAPC studies up to date assume unchanged handling times regardless of where the vessels are handled), and is a function of the berth location relative to the container storage yard and the assigned yard trailers to transport containers to/from the vessel. In their formulations thought handling time is again considered deterministic. No consideration is given to service priorities and the objective is restrained, as most of the papers, to the minimization the total service and wait time.

Cordeau et al. (2005), similar to Imai et al. (2005), considered the discrete DBAP, that works with a finite set of berthing points, and provided two formulations: a formulation similar to Imai et al. (2001) and a formulation similar to the Multi Depot Vehicle Routing Problem with Time Windows (Legato et al. 2001). They also developed a heuristic for the continuous case. In contrast to previous models their work is capable of handling a weighted sum of service times as well as windows on berthing times. In the discrete case, medium-size instances were solved exactly under some assumptions, which enabled an assessment of the quality of the heuristic. Because the continuous problem could not be solved exactly, the assessment of the heuristic developed was only be inferred from the discrete scenario. To avoid simplifications contrary to Park and Kim (2003) the authors do not solve the BAP and the Quay Crane Assignment Problem (QCAP) simultaneously. The objective is the minimization of the total (weighted) service time for all vessels, defined as the time elapsed between the arrival in the harbor and the completion of handling.

Briano et al. (2005) outlined the integration between a flexible simulator, which represents the marine-side operations of a container terminal, with a Linear Programming model for improving berth assignment and yard stacking management policies. The proposed methodology starts with a Mathematical Model for supporting Berth Planning. The optimal position of the berth for each vessel was considered the nearest docking place where the containers have to be taken or dropped. The goal of this part of the model is to minimize the penalty cost resulting from delayed departures of vessels and the additional handling cost resulting from deviation of the berthing position from the best location on the berth. The authors note that the problem is NP-hard and can only be solved in reasonable time for a maximum of seven vessels and a 72 hour plan horizon. This model is combined with a simulation model used for identifying optimal positions for containers. The paper is very brief and does not go into detail of how the integration was accomplished or problems that were or may be encountered during implementation.

Lokuge and Alahakoon (2006) present a unique approach, differentiating itself from all the previous work presented herein. They use Artificial Intelligence (AI), and more specifically the Beliefs, Desires and Intention (BDI) agent architecture. They describe the use of a hybrid BDI agent architecture for a vessel berthing application system. An extended hybrid BDI agent system with intelligent tools (neural networks and adaptive neuro-fuzzy inference system (ANFIS)) was proposed for improved performance in the terminal. The assignment of vessel to berth is based on several factors that include: minimization of the waiting time of the vessel, berth productivity, minimum distance for vessel berthing and sailing etc. Results show a reduced average waiting time of vessels while several other measures of port productivity are also presented.

Imai et al., (2007) addressed the berth allocation problem at a multi-user container terminal with indented berths for fast handling. A new integer linear programming formulation was presented, which was then extended to model the berth allocation problem at a terminal with indented berths, where both mega-containerships and feeder vessels are to be served for higher berth productivity. A genetic algorithm heuristic was used to solve the problem to optimality. From derived computational results it was concluded that while the indented terminal served the mega-vessel faster than the conventional terminal, the total service time for all vessels was longer than the one in the conventional terminal. Imai et al. (2006) addressed a variation of the berth allocation problem at multi-user terminals, where vessels normally served at the terminal with expected wait time exceeding a certain time limit, were assigned to an external terminal. The objective of the problem was to minimize the total service time of vessels at the external terminal.

Wang and Lim (2006) solved the DBAP by minimizing un-allocation, position and delay costs, using a Stochastic Beam Search Heuristic that outperformed both the current state-of-the-art metaheuristics and the traditional beam search. The authors concluded that the formulation and solution approach is fast, easy to modify and implement, and can be directly applied to solving multi-stage decision making problems.

Hansen et al. (2007) studied the DBAPD considering the minimization of total costs for waiting and handling as well as earliness or tardiness of completion, for all vessels. They presented a general formulation that can be reduced to the BSP by Imai et al (2001) and (2003). A Variable Neighborhood Search (VNS) heuristic was proposed, and compared with Multi-Start (MS), a Genetic Search algorithm (GA) and a Memetic Search algorithm (MA). VNS provided optimal solutions for small instances solved to optimality for the DBAPD with the objective to minimize the total service time. The authors claim that VNS outperforms MS, MA and GA on large instances though this statement is not based on elaborate computational experiments (i.e. very small computational time). Furthermore, instances used for the computational examples cannot be considered descriptive of a medium to heavily congested port, since the ratio of vessel to berth per day is very low (the maximum ratio was less then 1 vessel per berth per day).

Moorthy and Teo (2006), expanded on the work of Dai et al. (2004), and presented a new approach (perhaps the most interesting so far) for the DBAPC. The problem was modeled as a bicriteria optimization problem. The first objective dealt with the trade-off between the operational cost of moving containers from one vessel to the other and the delays (difference of actual arrival and start of mooring) of customers. The second objective dealt with the stochasticity in the arrival of ships and the robustness of the final schedule. The authors try to minimize the expected delays of transshipment vessels. The focus of the experimental results lies in the stochastic nature of the problem. As also stated by the authors the approach adopted is limited by the fact that the berth solution is relevant only when a substantial number of vessels arrive periodically and within the same period. They use a continuous representation but note that in the final solution overlapping of vessels is not avoided, especially when demand increases. Nevertheless this is the first paper that studied the BAP incorporating the stochastic nature of the vessel arrivals with very promising results. The authors stated that further research is underway.

Monaco and Sammara (2007) presented a compact formulation for the discrete and dynamic BAP and developed a Lagrangean heuristic to solve the problem. Imai et al. 2007 proposed a formulation for the simultaneous berth–crane allocation that minimizes the total service time and developed a genetic algorithm-based heuristic to solve to the resulting problem.

Discussion of current Berth Allocation Research

Port Operator Service Agreements and Berth Allocation Models

The SBAP and DBAP along with the discrete and continuous BAP have been widely studied in different combinations. Most of the studies tried to minimize the total service and waiting time (total completion time-TCT) and/or the deviation from the preferred berth, since it is expected that minimization of the deviation from the preferred berthing position will reduce service time and operator's cost, while very few studies incorporated the minimization of the cost endured by a vessels late departure after an agreed point in time. These objectives satisfy most of the port operators' objectives but fail to portray most of the service priority agreements (fig. 3-4). These contractual arrangements can vary from berthing (and start of cargo handling operations) upon arrival, to guaranteed service time window and/or guaranteed service productivity (UNCTAD, 1986). Earliness or lateness of a vessels start or completion time of handling operations (loading/unloading of containers) implies benefits or costs to both the port operator and the ocean carrier. If these operations are completed after a specified and agreed time, the port operator may pay a penalty to the ocean carrier, while if these operations are completed before that date the carrier may pay a premium fee to the port operator, subject to the contractual arrangements, although in practice premium may be compensated with past or future penalties assigned to the port operator due to failure to meet the terms of the contract. Although early departures are seldom reported to happen, they can help ocean carriers in managing the time factor of their service schedules, by providing time buffer to cope with time lost in other ports (Notteboom, 2006). Early premiums can be offset by reducing voyage operating cost through reducing the voyage speed and therefore the fuel consumption. In fact, recently, ocean carriers seek to reduce operating cost through voyage speed reduction, while maintaining service punctuality (Savvides, 2006 and Lloyds List, 2006). It was not until the early 2007 (Hansen et al., 2007) that researchers began recognizing the significance of premiums from the vessels' early departures. Nevertheless, service deadlines (start or finish or service) in the form of time windows, penalties and premiums from the early start of service and premiums from the start of finish of service within the deadline time window have not yet been investigated, though they represent one of the most basic BSPs.

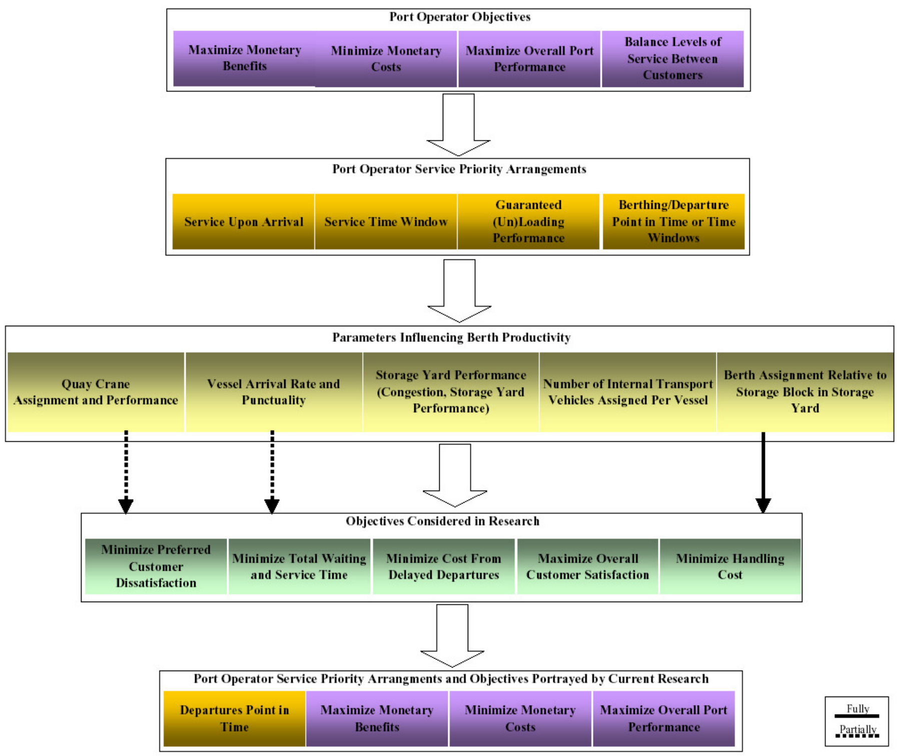


Figure 3-4 Port Operator Objectives and Service Priority Arrangements VS Parameters Influencing BAP and Objectives and Service Priority Arrangement Considered in Current Research

Service Priority Schemes based on Weights

Allocating vessels to berths by simply minimizing the total completion time can lead to problems where vessels with smaller handling volumes receiving higher priorities than vessels with larger handling volumes (Pinedo, 2002). The later, end up serviced at the end of the queues at each berth. That is to say given the situation that two vessels with different handling volumes arrive at the same time, the large vessel will wait for the smaller vessel to get serviced (if they are both serviced at the same berth). Assignment policies based on this objective have the consequence of larger waiting times for larger vessels. Some large vessels though, for a number of reasons (call at another port, time sensitive cargo etc), might need to be assigned for service and/or finish their service as soon as possible, after their arrival at the port. In practice, the problem of assigning priority status to vessels is more complex since for some vessels contractual agreements signed between ports and customers do not allow for arbitrarily assignment (Dai et al., 2004).

To illustrate this disadvantage assume the following case of a single berth with vessel arrival and handling times as given in table 3a. In this case solving the problem with the objective of minimizing the total completion time, vessel 1 will have to wait and be served last (table 3b). The question arises though how beneficial is that for the port operator to have a large customer waiting for such a long period of time. This also implies that the berth might be unutilized for certain periods of time waiting for the small vessels to arrive (while other vessels are already in port waiting for service), resulting in extra cost to port operators.

As described previously a number of studies tried to address this issue by assigning weights to the vessels, representing in this way the vessels priority. Service priority schemes based on the assignment of arbitrary weights to vessels were introduced by Imai et al. (2003) and were adopted by a number of studies that followed. The main issue with this approach lies in the determination of weights, since there is no intelligent way, excluding iterative processes, to assign these weights in order to meet specific contractual agreements. To avoid these issues Kim and Moon (2003), Park and Kim (2003), and lately Wang and Lim (2006), and Hansen et al. (2007), used monetary penalization-premiums of delayed-early/timely departures.

Table 3-3 Example arrival, handling and finish times per vessel, for a simple BAP problem

3a: Arrival and Handling Time

Vessel	**Arrival**	**Handling Time**
1	0	50
2	0	4
3	0	4
4	0	4
5	16	4
6	25	4
7	30	4
8	40	4
9	50	4

3b: Assignment Minimizing Total Service Time

Service Order	**Vessel**	**Finish Time**	**Idle Berth Time**	**Wait Time**
1	2	4	0	0
2	3	8	0	4
3	4	12	0	12
4	5	20	4	0
5	6	29	4	0
6	7	34	1	0
7	8	44	6	0
8	9	54	6	0
9	1	104	0	54

Multi-objective berth allocation

As with all engineering problems determining berthing times and positions of containerships at a port container terminal has several objectives; i.e. minimizing service and waiting time, meeting contractual agreements, minimizing port operators' operational costs, etc. All the models that have been presented in the related literature focus on the formulation of a single objective BSP model. Even though most of the authors recognize the multi-objective character of the problem, and at several cases consider it as such, they end up combining the multiple objectives into a single scalar value by using weighted aggregating functions according to the preferences set by the decision-makers and then, find a solution that satisfies these preferences. However, in many real scenarios involving multi-objective scheduling problems (such as the BAP), it is preferable to present various compromising solutions to the decision-makers, so that the most adequate schedule can be chosen. Although this can be achieved by performing the search several times using different preferences each time, another approach is to generate the set of compromise solutions in a single execution of the algorithm. The consideration of many objectives has the advantage of a wider range of alternatives for the participants in the planning and decision-making processes, (i.e. "analyst" or "modeler"– who generates alternative solutions, and "decision maker" who uses the solutions generated by the analyst to make informed decisions) while modeling the problem in a more realistic way.

Most of the BAP have a number of constraints (some hard and other soft). As correctly pointed out by Fonseca and Fleming (1995) in a research report, in many cases satisfying constraints is a difficult problem itself. When different constraints cannot be satisfied simultaneously, the problem is often deemed to admit no solution. The answer came from Coello Coello (2000), who showed that it is possible to treat constraints as objectives and consistently outperform the single objective approach without a significant sacrifice in terms of performance. This observation could be proven very valuable when formulating BSP model that deal with hard constraints that lead to infeasibility (such as the service upon arrival service arrangement) since a number of the constraints can be viewed as objectives.

The main drawback of a multi-objective formulation lies in the solution approach and optimality of the results. To solve multi-objective problems evolutionary algorithms have been used exclusively (Toboada, 2007). On the other hand though heuristic approaches have also been used exclusively to solve medium to large (and in some cases small) instances of all the single objective BSP presented to date.

Stochastic Arrivals and Handling Time

Another issue not yet fully addressed, but mentioned regularly in the berth allocation literature, is how robust are the BAP assignments when the uncertainty of the vessels arrival and handling times is not considered. Excluding the study by Moorthy and Teo (2006), the rest of the studies assumed that the arrival and handling time of each vessel was known with certainty. Usually though vessels provide the port operator with a time window in which they may arrive at the port and request service. These time windows are not known with certainty until few hour of a vessels arrival. Furthermore, and due to several factors quay crane availability and performance varies and this influences the handling time of the vessels, which should be considered stochastic. Although berth allocation models including arrival and handling time uncertainty may be more beneficial to a port operator, as of today have not yet been investigated..

Berth and Quay Crane Scheduling

One of the most crucial issues in the BAP is the consideration of the handling times as constants and only dependent on the berth assignment of the vessel. As mentioned earlier, a small number of researchers presented studies that deviated from this approach. Nevertheless only part of the service priority agreements has been modeled. Further research is required that takes under consideration the rest of the service priority agreements with the focus on the guaranteed (un)loading performance, which is one of the most important factors during negotiations of future contractual agreement between the port and vessel operators.

Conclusions

From the review of the system and the related literature, it becomes obvious that several objectives need to be met and optimized at the berth part of the terminal and in general at the seaside (table 3-4). A trade-off exists between the total staying time in the port, the dissatisfaction of vessel owners caused by the order in which vessels are berthed and finish service (expressed in the total waiting and service time) and the port operators operating and capital costs. The issue of meeting contractual agreements based on berth productivity and vessel berth assignment is very important for a ports' competitiveness.

Some researchers tried to address the issue of port competitiveness by assigning service priority rules either as the vessels' weight factor or as a time cost penalty for late departure. Unfortunately the former formulations did not present a sensitivity analysis of the weights, while the later based the BSP of vessels serviced before the scheduled time on the minimization of the PBP. On the other hand premium benefits, cost of keeping a berth idle, guidance of how to use/modify weights or cost penalties to achieve the terminal operators' service contractual agreements have never been stated explicitly. This, is somewhat expected, since these studies looked at the problem from a single objective using weighted methods, that most of the times have an inherent problem in selecting the weights (or utility functions) that characterize the decision-maker's preferences. In practice, and in most scientific fields, it can be very difficult to precisely and accurately select these weights, even for someone familiar with the problems domain. BAP formulations that could accommodate flexibility in that aspect would be more realistic. Furthermore the stochastic nature of the BAP has not been present in any model so far.

In our opinion several issues remain to be addresses in the BAP (table 3-4, 3-5, 3-6 and 3-7) including: a) dynamic formulation of the service time in respect to the assigned quay cranes, internal transport vehicles, and the preferred berthing position, b) formal methodology of weight application to service priority modeling and improvement of weight definitions, c) operators cost-service priority modeling, d) stochastic nature of BAP, and e) multi-objective formulation of the BAP.

Table 3-4 Existing BSP

Existing approaches	**Container Terminal Operator Practices**
Minimize Total Completion Time	Maximize Overall Berth Performance
Minimize Late Departures, Maximize Early Departures and Distance from Preferred Berth	Maximize Overall Berth Performance, Customer Dissatisfaction, Penalty Cost
Minimize Total Weighted Completion Time	Maximize Overall Berth Performance, Customer Dissatisfaction
Minimize Position Cost, Delay Cost, and Unallocation Cost	Minimize Customer Dissatisfaction, Indirectly Minimize Service Time
Minimize Cost From Delayed Departures, Cost of Non-Preferred Berth, and Cost from Early or Late start of vessel handling against estimated times of vessel arrival	Minimize Customer Dissatisfaction, Indirectly Minimize Service Time
Minimize the maximum amount of Quay occupied	None

Table 3-5 Future BSP

Future Research	**Container Terminal Operator Practices**
Minimize Late Berthing/Departures, Maximize Early/Timely Departures/Berthing (Departure/Berthing Time or Time Window), Minimize Handling and Wait Time Cost	Maximize Profit, Minimize Customer Dissatisfaction
Minimize cost from failing to meet Guaranteed (Un)Loading Performance/Service Time and Berthing and Departure Time Window	Maximize Profit, Minimize Customer Dissatisfaction
Multi-Objective BAP	Balance between Profit Maximization, Customer Dissatisfaction, Total Service Time Minimization, Customer Specific Service Time Minimization etc
Stochastic Arrival/Service Time BAP	Minimize Risk of Interruptions and Delays. Robust Scheduling

Table 3-6 Current Formulations and Port Operator Objectives and Service Priority Arrangements

		Port Operator Objectives				Port Operator Service Priority Arrangements			
Author	BSF	Overall Berth Performance	Minimize Operational Cost	Balance Level of Service Between Customers	Maximize Profits	Service Upon Arrival	Service Time Windows	Guaranteed (Un)Loading Performance	Berthing/Departure Point in Time/Time Windows
Li et al., 1998	Makespan	X(P)		X(P)					
Lim et al., 1998	Amount of Quay	X(P)	X(P)						
Guan et al., 2002, Guan & Chen, 2004, Imai et al., 2003, Cordeau et al., 2005	Total Weighted Completion Time	X(P)		X	X(P)				
Park & Kim, 2002, Briano et al., 2005	Cost From Delayed Departures and Cost of Non-Preferred	X(P)	X		X(P)				X(P)

	Berth								
Kim and Moon, 2003	Cost From Delayed Departures Cost of Non-Preferred Berth Cost from early or late start of vessel handling against ETA (Estimated Time of Arrival)	X(P)	X			X(I)			X(P)
Li et al., 1998	Makespan	X(P)		X(P)					
Lim et al., 1998	Amount of Quay	X(P)	X(P)						
Guan et al., 2002, Guan & Chen, 2004, Imai et al., 2003, Cordeau et al., 2005	Total Weighted Completion Time	X(P)		X	X(P)				

I: Indirectly, P: Partially

Table 3-6 Current Formulations and Port Operator Objectives and Service Priority Arrangements (Continued)

		Port Operator Objectives				Port Operator Service Priority Arrangements			
Author	BSF	Overall Berth Performance	Minimize Operational Cost	Balance Level of Service Between Customers	Maximize Profits	Service Upon Arrival	Service Time Windows	Guaranteed (Un)Loading Performance	Berthing/Departure Point in Time/Time Windows
Park & Kim,	Cost From Delayed	X(P)	X		X(P)				X(P)

2002, Briano et al., 2005	Departures and Cost of Non-Preferred Berth						
Kim and Moon, 2003	Cost From Delayed Departures Cost of Non-Preferred Berth Cost from early or late start of vessel handling against ETA (Estimated Time of Arrival)	X(P)	X		X(I)		X(P)
Park and Kim, 2003	Handling Cost Penalties from Berthing Prior or After ETA Penalties from Late Departures Total Number of Crane Setup	X(P)	X		X(I)		X(P)
Imai et al., 2001; Nishimura et al., 2001; Cordeau et al., 2005; Imai et al., 2005; Imai et al., 2007;	Total Completion Time	X					
Imai et al., 2006	External Berth Service Time	X(P)					
Cordeau et al., 2005	Housekeeping		X	X(P)			
Lee et al., 2006	Makespan & Quay Crane Time	X	X(P)				

Wang & Lim, 2006	Position, Delay, & Unallocation Cost		X	X(P)			X(P)
Hansen et al., 2007	Delayed Departures Waiting and Handling Cost		X	X(P)			X(P)
Moorthy & Teo, 2006	Operational Cost Service Levels		X	X(I)			

I: Indirectly, P: Partially

Table 3-7 Proposed Formulations and Port Operator Objectives and Service Priority Arrangements

	Port Operator Objectives				**Port Operator Service Priority Arrangements**			
Proposed BSF	*Overall Berth Performance*	*Minimize Operational Cost*	*Balance Level of Service Between Customers*	*Maximize Profits*	*Service Upon Arrival*	*Service Time Windows*	*Guaranteed (Un)Loading Performance*	*Berthing/Departure Point in Time/Time Windows*
Minimize Late Departures, Maximize Early and Timely Departures (Time Window)	X(P)		X(P)	X		X(P)		X(P)
Minimize Late Berthing, Maximize Early Berthing (Time Window)	X(P)			X	X(P)	X(P)		X(P)
Minimize Late Berthing, Maximize Early and Timely Berthing (Time Window)	X(P)			X	X(P)	X(P)		X(P)
Guaranteed (Un)Loading Performance/Service Time								
Minimize Late Departures/Berthing, Maximize Early and Timely Departures/Berthing (Time Window)	X(P)	X(P)	X(P)	X	X(P)	X(I)	X(I)	X

Minimize Late Departures/Berthing, Maximize Early and Timely Departures/Berthing (Time Window), Minimize Handling Cost, Minimize Waiting Time Cost	X(P)	X	X	X	X(I)	X(I)	X(I)	X
Multi-Objective	X(D)	X(D)	X(D)	X(D)	X(D)	X(D)	X(D)	X(D)
Stochastic Service Time	X(D)	X(D)	X(D)	X(D)	X(D)	X(D)	X(D)	X(D)
Guarantee Performance	X(P)	X(P)		X			X	

I: Indirectly, P: Partially, D: Depending on Objectives

4. BASIC CONCEPTS OF MULTI-OBJECTIVE AND STOCHASTIC OPTIMIZATION

The BAP belongs to the class of scheduling problems, which include a wide variety of problems such as machine scheduling, events scheduling, personnel scheduling and others. Many real-world scheduling problems are multi-objective and stochastic by nature (Ehrgott & Gandibleux, 2000; Pinedo 2002). This chapter will provide the basic concepts of Multi-Objective and Stochastic Optimization without going into a depth analysis. The information provided though, will be enough for the reader to follow up on the application of these methods to the BAP formulation and solution approaches presented herein.

Multi-Objective Optimization

Basic Concepts

The general N-objective optimization problem (or in general the multi-objective optimization problem - MOO) can be defined in the following way (as stated by Coello Coello, 1999): Find the vector of decision variables (also called solution) $\boldsymbol{X=[x_1, x_2, \ldots.., x_n]}$ that optimizes (minimizes or maximizes) a vector objective function: $\boldsymbol{F(X)}=[f_1(\boldsymbol{X}), f_2(\boldsymbol{X}),\ldots\ldots., f_n(\boldsymbol{X})]$, subject to m inequality constraints $G_i(\boldsymbol{X}),\ i=(1,2,3\ldots.,m)$ and k equality constraints $H_j(\boldsymbol{X}),\ j=(1,2,3,\ldots.,k)$. If the variables $\boldsymbol{x}$ are discrete, then the problem is called Multi-Objective Combinatorial Optimization (MOCO) problem.

Due to the conflicting nature of the objectives it is usually the case that there is no unique optimal solution. It is possible to improve separately at least one (but not all) objective function of a given solution but this will usually causes the declining of its remaining objective functions (or at least one of them). Thus, several different solutions could be thought of as "optimal", because no one dominates the other.

The main difficulty with the multi-objective approach lies in the comparison of the solutions. By definition one solution outperforms another if the values of all objective functions of the first solution are better than the second. In other words if $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ are two solutions then $\boldsymbol{F(X_1)}$ dominates $\boldsymbol{F(X_2)}$ if and only if $f_i(X_1) \geq fi(X_2), \forall i$, and $f_i(X_1) \succ fi(X_2)$, for at least one i. Such solutions are called "Pareto-optimal". If no solution can dominate the given solution then it can be considered to be optimal.

All Pareto-optimal solutions compose a certain boundary between the space, which contains dominated solutions and the space where no solutions exist. This boundary is called the trade-off surface or Pareto-front. It can be depicted as a surface in the N-dimensional space, where N is the number of objectives. An example of the Pareto front of a bi-objective space is presented as a curve in figure 4-1. For a more analytical description of these concepts the reader is referred to: Jaszkiewicz (2001), and Van Veldhuizen and Lamont (2000).

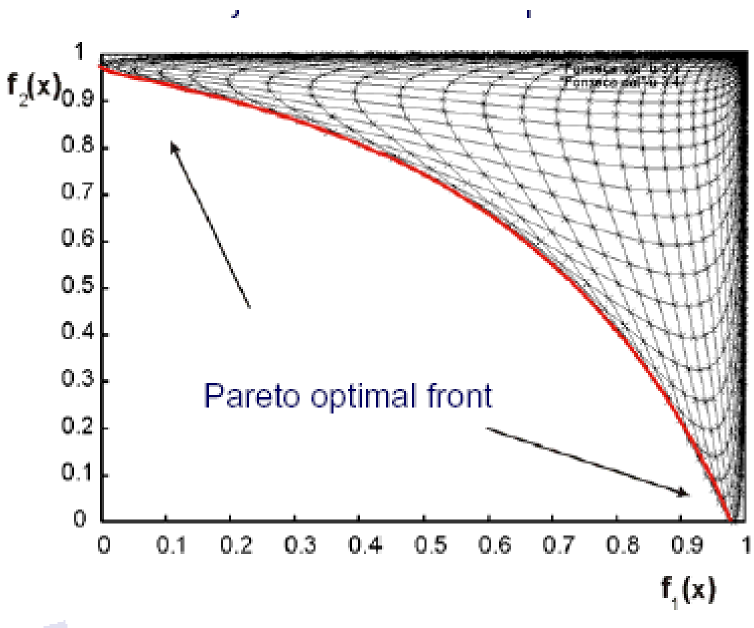


Figure 4-1 Pareto Optimal Solutions and Pareto Optimal Front (Curve)

(Source: Mehnen, 2005)

Modeling Techniques

There are two general approaches to multiple-objective optimization, in terms of the solution approach. One is to combine the individual objective functions into a single composite function or move all but one objective to the constraint set. In the former case, determination of a single objective is possible with methods such as utility theory, weighted sum method, etc., but the problem complexity and accuracy lies in the proper selection of the weights or utility functions that are used to depict the decision-maker's preferences. In practice, it can be very difficult to precisely and accurately select these weights, even for someone familiar with the problem domain (Coello Coello, 2000). In the latter case, the problem is moving objectives to the constraint set, a constraining value must be established for each of these former objectives and can be rather arbitrary. In both cases, an optimization method would return a single solution rather than a set of solutions that can be examined for trade-offs. For this reason, decision-makers often prefer a set of good solutions considering the multiple objectives, which leads to the second approach.

The second general approach is to determine an entire Pareto optimal solution set or a representative subset. While moving from one Pareto solution to another, there is always a certain amount of sacrifice in one objective(s) to achieve a certain amount of gain in the other(s). Pareto optimal solution sets are often preferred to single solutions because they can be practical when considering real-life problems since the final solution of the decision maker is always a trade-off. Pareto optimal sets can be of varied sizes, but the size of the Pareto set usually increases with the increase in the number of objectives.

MOO Algorithms

The use of exact methods to solve MOO problems is time consuming and the most common approach for solving MOO problems is the use of Multi-Objective Metaheuristics (MOM) (as stated by many researchers; see for example: Konak et al. 2006; Silva et al. 2004).

Furthermore according to Coello Coello (2003[7]) heuristics (and metaheuristics) seem particularly suitable to solve multi-objective optimization problems, because they are less subject to the shape or continuity of the Pareto front (e.g., they can easily deal with discontinuous or concave Pareto fronts), whereas this is a real concern for mathematical programming techniques. Additionally, many current heuristics (e.g., evolutionary algorithms, particle swarm optimization, etc.) are population-based, which means that we can aim to generate several elements of the Pareto optimal set in a single run.

[7] Online document <http://www.cs.cinvestav.mx/~EVOCINV/download/tutorial-sis-2003.pdf>

The most popular MOM are: evolutionary algorithms (EA), and alternative multi-objective metaheuristics as shown in table 4-1, such as: tabu search, simulated annealing, and memetic algorithms that explicitly use local search or neighborhood exploration (instead of genetic operators) to drive the search or as an important component of the process (hybrid approaches). Jones et al. (2002) reported that 90% of the approaches to multi-objective optimization aimed to approximate the true Pareto front for the underlying problem and 70% of all meta-heuristics approaches were based on evolutionary approaches.

Evolutionary Algorithms

There is no universally accepted definition of evolutionary algorithm, but in the strict sense an evolutionary algorithm handles a population of solutions, evolves this population by means of cooperation (recombination) and self-adaptation (mutation) and uses a coded representation of solutions (Hertz and Klober, 2000). EAs such as Evolution Strategies and Genetic Algorithms (GA) have become the method of choice for optimization problems that are too complex to be solved using deterministic techniques such as linear programming or gradient methods. EAs require little knowledge about the problem being solved, and they are easy to implement, robust, and inherently parallel. To solve a certain optimization problem, it is enough to require that one is able to evaluate the objective (cost) function(s) for a given set of input parameters. Because of their universality, ease of implementation, and fitness for parallel computing, EAs often take less time to find the optimal solution than gradient methods. However, most real-world problems involve simultaneous optimization of several often mutually concurrent objectives. Multi-objective EAs are able to find optimal trade-offs in order to get the Pareto optimal set (Taboada, 2007).

Table 4-1 Alternative Multi-Objective Metaheuristics

Method	Author
Simulated Annealing for Multi-objective Optimization	Serafini, 1992
Multi-objective Tabu Search (MOTS)	Hansen, 1997
Pareto Simulated Annealing (PSA)	Czyzak & Jaszkiewicz, 1998
Multi-objective Simulated Annealing (MOSA)	Ulungu, et al., 1999
Memetic Pareto Archived Evolutionary Strategy (M-PAES)	Knowles & Cornel, 2000
Genetic Local Search (GLS)	Jaszkiewicz, 2002
Simulated Annealing for Multi-objective Optimization	Suppapitnarm et al., 2000
Other Multi-objective Metaheuristics Using Local Search	

GAs have been the most popular heuristic approach to multi-objective design and optimization problems. Following the success of metaheuristics in single objective optimization many researchers proposed the use of GA based metaheuristics in MOO. Since the proposition of the Vector Evaluated Genetic Algorithm (Schaffer, 1985) a significant number of different multiple objective metaheuristics have been proposed.

According to Konak et al. (2005) several GA based multi-objective evolutionary algorithms have been developed. In their paper they present the well-known and credible algorithms that have been used in many applications along with their advantages and disadvantages. These algorithms are: Multi-objective Genetic Algorithm (MOGA) (Fonseca and Fleming, 1993), Niched Pareto Genetic Algorithm (NPGA) (Horn et al. 1994), Random Weighted Genetic Algorithm (RWGA) (Murata et al., 1996), Nondominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb, 1995), Strength Pareto Evolutionary Algorithm (SPEA) (Zitler and Thiele, 1999), Improved SPEA (SPEA2) (Zitler et al. 2001), Pareto- Archived Evolution Strategy (PAES) (Knowles and Corne, 2000), Pareto Envelope-based Selection Algorithm (PESA) (Corne et al. 2000), Region-based Selection in Evolutionary Multi-objective Optimization (PESA-II) (Corne et al. 2001), Fast Nondominated Sorting Genetic Algorithm (NSGA-II) (Deb et al. 2002), Multi-objective Evolutionary Algorithm (MEA) (Sarker et al. 2002), Micro-GA (Coello and Pulido, 2001), Rank-

Density Based Genetic Algorithm (RDGA) (Lu and Yen, 2003), and Dynamic Multi-objective Evolutionary Algorithm (DMOEA) (Yen and Lu, 2003).

Performance Measures

Of the various multi-objective EAs available, we are interested in the ones that provide the best approximation for a given problem. For this reason, comparative studies have been conducted (Zitler and Thiele, 1998; Van Veldhuizen and Lamont, 2000; Tan et al, 2001) aiming at revealing strengths and weaknesses of certain approaches and at identifying the most promising algorithms. This in turn, led to the question of how to compare the performance (quality of outcome and computational resources) of multi-objective optimizers.

It is difficult to define appropriate quality measures for approximations of the Pareto-optimal front, and as a consequence graphical plots have been used to compare the outcomes of multi-objective EAs until recently, as Van Veldhuizen, (1999) points out, but several studies can be found in the literature that address the problem of comparing approximations of the trade-off in a quantitative manner. Most popular are the unary quality measures, i.e., the measure assigns each approximation set a number that reflects a certain quality aspect, and usually a combination of them is used, e.g., (Van Veldhuizen and Lamont, 2000; Kalyanmoy et al. 2000). Other methods are based on binary quality measures, which assign numbers to pairs of approximation sets, e.g., (Zitler and Thiele, 1998; Hansen and Jaszkiewicz, 1998).

A third, and conceptually different approach, is the attainment function approach (Grunert da Fonseca et al. 2001), which consists of estimating the probability of attaining arbitrary goals in objective space from multiple approximation sets. Despite of this variety, it has remained unclear up to now how the different measures are related to each other and what their advantages and disadvantages are. Recently, a few studies have been carried out to clarify this situation.

Hansen and Jaszkiewicz (1998) studied and proposed some quality measures that induce a linear ordering on the space of possible approximations—on the basis of assumptions about the decision maker's preferences. They first introduced three different out-performance" relations for multi-objective optimizers and then investigated whether the measures under consideration are compliant with these relations. The basic question they considered was: whenever an approximation is better than another according to an "outperformance" relation, does the comparison method also evaluate the former as being better (or at least not worse) than the latter?

Knowles et al. (2000) compared the information provided by different assessment techniques on two database management applications. Later, Knowles (2002) and Knowles and Corne (2002) discussed and contrasted several commonly used quality measures in the light of Hansen and Jaszkiewicz's approach as well as according to other criteria such as, e.g., sensitivity to scaling. They showed that about one third of the investigated quality measures are not compliant with any of the "outperformance" relations introduced by Hansen and Jaszkiewicz (1998).

Zitler et al. (2002) showed that: a) there exists no unary quality measure that is able to indicate whether an approximation A is better than an approximation B, b) the above statement even holds if we consider a finite combination of unary measures, c) most existing quality measures that have been proposed to indicate that A is better than B at best allow to infer that A is not worse than B, d) unary measures being able to detect that A is better than B exist, but their use is in general restricted, and e) binary quality measures overcome the limitations of unary measures and, if properly designed, are capable of indicating whether A is better than B.

Multi-Objective Scheduling Literature

Literature on multi-objective scheduling is vast. Excellent reviews of the principles of evolutionary multi-objective optimization and recent developments are provided by Coello et al. (2002), Van Valdhuizen and Lamont (2000), and Taboada (2007). We will focus the literature review on multi-objective machine scheduling related problems, since the discrete BAP belongs to this general category of problems. The interested reader is referred to Silva et al. (2004), and the excellent online directory of multi-objective optimization by Coello[8] for a more analytical literature review and an introduction to multi-objective metaheuristics for scheduling. Murata et al. (1996) proposed a multi-objective genetic algorithm (MOGA) and applied it to flow-shop scheduling. Hyun et al. (1998) developed a new selection scheme in GA, and showed its superiority for multi-objective scheduling problems in assembly lines. Also using GAs, Chen et al. (1996) studied the radiological worker allocation problem in which multiple constraints are considered. Constraints are classified as hard and soft. Each solution must satisfy the hard constraints and performance of the solution is measured by the violation of soft constraints. The GA approach was compared with conventional optimization techniques such as goal programming and simplex method, and the GA showed superior results. Other heuristics such as simulated annealing and tabu search have also been studied.

Marett and Wright (1996) compared these two heuristics for flow shop scheduling problems with multiple objectives. The performance of the methods was compared as the number of objectives increased. Simulated annealing was found to perform better than tabu search as the number of objectives increased. They also mentioned that the complexity of combinatorial problems is strongly influenced by the type of objectives as well as their number. Yim and Lee (1996) used Petri nets and heuristic search to solve multi-objective scheduling for flexible manufacturing

[8] http://www.lania.mx/~ccoello/EMOO/

systems. Pareto optimal solutions were obtained by minimizing the weighted summation of the multiple objectives. Jaszkiewicz (1997) combined the genetic algorithm with simulated annealing to solve a nurse-scheduling problem. The maximization of multiple objectives is represented by a single scalar function, which is the summation of a scalar multiplying the difference between current and previous solutions for each objective. The scalar is greater than one if the objective is improved or less than one if no improvement is found. Cohran et al. (2003) proposed a two-stage multi-population genetic algorithm (MPGA) to solve parallel machine scheduling problems with multiple objectives. Their approach is applied in parallel machine scheduling problems with two objectives: makespan and total weighted tardiness (TWT). The MPGA was compared with a benchmark method, the multi-objective genetic algorithm (MOGA), and showed better results for all of the objectives over a wide range of problems. The MPGA was extended to scheduling problems with three objectives: makespan, TWT, and total weighted completion times (TWC), for which also performed better than MOGA. Chang et al. (2005) introduced a two-phase sub population genetic algorithm to solve the parallel machine-scheduling problem. The two-phase sub-population genetic algorithm was applied to solve the parallel machine-scheduling problems in testing of the efficiency and efficacy. Experimental results were reported and the superiority of this approach was discussed. Taboada & Coit (2006a) and Taboada et al. (2007) formulated the redundancy allocation problem (RAP) as a multi-objective problem with the system reliability to be maximized, and cost and weight of the system to be minimized. The Pareto-optimal set was initially obtained using the fast elitist nondominated sorting genetic algorithm (NSGA-II) originally proposed by Deb et al. (2002). Then, the decision-making stage was performed by applying two proposed pruning methods to reduce the size of the Pareto-optimal set and obtain a smaller representation of the multi-objective design space. For those studies, NSGA-II was effective. However, NSGA-II is a general multi objective evolutionary algorithm (MOEA) for any type of problem. This implies that the problem formulation needs to be properly adapted. Moreover, in these studies, the final Pareto front found by NSGA-II contained many repeated solutions, so in order to obtain a large number of solutions; several runs had to be performed. Thus, if a decision-maker must solve many similar RAP problems, then a custom MOEA, especially designed to solve multi-objective design allocation problems, offers great advantage. MOEA-DAP, (Taboada & Coit, 2006b) was developed to address these difficulties. MOEA-DAP is a multiple objective evolutionary algorithm specifically designed to solve system design allocation problems. Thus, this new approach has the strength of a problem-oriented technique. MOEA-DAP, mainly differs from other MOEAs in the type of crossover operation performed. In this step, several offspring are created through multi-parent recombination. Thus, the mating pool contains a great amount of diversity of solutions. This disruptive nature of the proposed type of crossover, subsystem rotation crossover (SURC), appears to encourage the exploration of the search space.

Genetic Algorithms

A GA is a programming technique that mimics biological evolution as a problem-solving strategy. The input to the GA is a set of potential solutions, encoded in some fashion (usually binary form), and a metric called a fitness function that allows each candidate to be quantitatively evaluated. These candidates may be solutions already known to work, with the aim of the GA being to improve them, but more often they are generated at random. GAs are called ``blind'' because they have no knowledge of the problem.

The members of this initial population are each evaluated for their fitness or goodness in solving the problem. If the problem is to maximize a function f(x) over some range [a,b] of real numbers and if f(x) is nonnegative over the range, then f(x) can be used as the fitness of the bit string encoding the value x.

From the initial population of chromosomes, a new population is generated using three genetic operators: reproduction, crossover, and mutation. These are modeled on their biological counterparts. With probabilities proportional to their fitness, members of the population are selected for the new population. This means that in a pool of randomly generated candidates, some of which will not work at all, are not deleted and in a random manner are kept and allowed to reproduce.

Pairs of chromosomes in the new population are chosen at random to exchange genetic material, their bits, in a mating operation called crossover. This produces two new chromosomes that replace the parents. Randomly chosen bits in the offspring are flipped, a progress called mutation. The new population generated with these operators replaces the old population. The algorithm has performed one generation and then repeats for some specified number of additional generations. The population evolves, containing more and more highly fit chromosomes. When the convergence criterion is reached, such as no significant further increase in the average fitness of the population, the best chromosome produced is decoded into the search space point it represents.

The expectation is that the average fitness of the population will increase each round, and so by repeating this process for hundreds or thousands of rounds, very good solutions to the problem can be discovered.

GAs differ substantially from more traditional search and optimization methods. The most significant differences of GAs to traditional search and optimization methods (Pohleheim, 2004) are:

a) They search a population of points in parallel, not just a single point

b) They do not require derivative information or other auxiliary knowledge; only the objective function and corresponding fitness levels influence the directions of search

c) They use probabilistic transition rules, not deterministic ones

d) They are generally more straightforward to apply, because no restrictions for the definition of the objective function exist

e) They can provide a number of potential solutions to a given prob-lem. The final choice is left to the user. (Thus, in cases where the particular problem does not have one individual solution, for example a family of pareto-optimal solutions, as in the case of multi-objective optimization and scheduling problems, then the evolutionary algorithm is potentially useful for identifying these alternative solutions simultaneously).

Genetic algorithms are used in search and optimization, such as finding the maximum of a function over some domain space. In contrast to deterministic methods like hill climbing or brute force complete enumeration, genetic algorithms use randomization. Points in the domain space of the search, usually real numbers over some range, are encoded as bit strings, called chromosomes. Each bit position in the string is called a gene. Chromosomes may also be composed over some other alphabet than {0,1}, such as integers or real numbers, particularly if the search domain is multidimensional.

Implementation Issues

Although genetic algorithms have proven to be an efficient and powerful tool certain limitations exist in their application. The most important ones according to Marczyk, (2004)[9] are:

[9] http://www.talkorigins.org/faqs/genalg/genalg.html

a) ***Representation***. According to there are two main ways of achieving this. The most common approach is to define individuals as lists of numbers (binary-valued, integer-valued, or real-valued) where each number represents some aspect of a candidate solution

b) ***Fitness function***. Defining the fitness function so that higher fitness is attainable and actually does equate to a better solution for the given problem. If the fitness function is chosen poorly or defined imprecisely, the genetic algorithm may be unable to find a solution to the problem, or may end up solving the wrong problem (e.g. Graham-Rowe, 2002)

c) ***Other Parameters.*** Defining the other parameters of a GA (the size of the population, the rate of mutation and crossover, the type and strength of selection) must be also chosen with care. If the population size is too small, the genetic algorithm may not explore enough of the solution space to consistently find good solutions. If the rate of genetic change is too high or the selection scheme is chosen poorly, beneficial schema may be disrupted and the population may enter error catastrophe, changing too fast for selection to ever bring about convergence

d) ***Premature convergence***. If an individual that is more fit than most of its competitors emerges early on in the course of the run, it may reproduce so abundantly that it drives down the population's diversity too soon, leading the algorithm to converge on the local optimum that that individual represents rather than searching the fitness landscape thoroughly enough to find the global optimum (Forrest, 1993; Mitchell, 1996). This is an especially common problem in small populations, where even chance variations in reproduction rate may cause one genotype to become dominant over others. The most common methods implemented by GA researchers to deal with this problem is ranking, scaling and tournament selection

Finally, several researchers (Holland, 1992; Forrest, 1993; Haupt and Haupt 1998) advise against using genetic algorithms on analytically solvable problems. It is not that genetic algorithms cannot find good solutions to such problems; it is merely that traditional analytic methods take much less time and computational effort than GAs and, unlike GAs, are usually mathematically guaranteed to deliver the one exact solution.

Stochastic Scheduling

In many real world problems (and in scheduling) the problem data cannot be known accurately due to insufficient information about the future or to the uncertainty in the technical parameters. Stochastic programming is an approach to model these problems by taking these uncertainties into account (Birge & Louveaux, 1997). In stochastic scheduling some of the parameters of the system are random variables (arrival time, processing time, machine up-time, etc). Models can be classified into three broad categories: a) models for scheduling a batch of stochastic jobs, b) multi-armed bandit models, and c) models for scheduling queuing systems. The BAP studied in this dissertation belongs to the category (a) and (c) with the random variable being the arrival time (also known as the release date of a job).

Regarding methods and techniques, it seems fair to say that no unified and practical approach has been developed to design and analyze (nearly) optimal policies across the range of stochastic scheduling models. Although many such models can be cast in the framework of dynamic programming, straightforward application of this technique has not proven very effective, due to the large (or infinite) size of the resulting formulations (curse of dimensionality). Most results have been instead obtained through problem-specific arguments, which often do not extend to seemingly related models.

Decision-making under uncertainty has traditionally focused on a priori methods such as two-stage stochastic optimization and Markov Decision Processes. The methods have been successful for a variety of applications. However, for large, multi-stage, and highly dynamic applications, these methods face the so-called curse of dimensionality as they search for optimal solutions in large search spaces. In conclusion there are three approaches for dealing with uncertainty of the release dates: a) Stochastic optimization, b) Online scheduling, and c) Online stochastic scheduling. All three approaches are described in the next subsections.

Stochastic Optimization

Stochastic optimization models have been formulated as multi-stage optimization models, with the majority focusing on two-stage problems. While successful solutions to these problems give us some insight to the random structure of the domains, they do not translate easily into efficient solutions to the related online problems and most of the times suffer from computational overload.

The two-stage stochastic programming approach is the simplest one, where the decision variables are partitioned into two sets. The first-stage variables are decided before the realization of the uncertain parameters, while the second stage or recourse variables are determined once the stochastic parameters are revealed. In order to decide, the policy utilizes the complete information contained in the partial schedule up to time t, as well as information about unscheduled jobs that arrive before the end of time t. For the large, multi-stage, and highly dynamic applications these methods face the so-called curse of dimensionality as they search for optimal policies in gigantic search spaces (Van Hentenryck et al., 2005; Shapiro, 2005)

Integer/Mixed-integer stochastic programs, like most formulations of the BAP (table 3-2), are particularly challenging. Even though the problem can be reduced to a specially structured IP if the probability distribution of the uncertain parameters is finite, traditional integer-programming techniques are not suited to exploit the special problem structure and deal with the problem of dimensionality.

An alternative approach, in particular for problems with large sample spaces, is to use Monte-Carlo sampling to generate i.i.d realizations and approximate the problem a with a sample average. Shapiro (2005) has shown that the Sampling Average Approximation method (Kleywegt et al. 2001) cannot be extended efficiently to multistage stochastic optimization problems since the number of required samples method must grown exponentially with the number of iterations, which is typically large or infinite in online applications. Combinatorial solutions suffer from similar exponential explosion. BAP formulations so far presented in the literature are already very computationally expensive. Thus applying this approach to the BAP problem with stochastic release dates would not be the right direction.

For a survey of stochastic scheduling problems consult Megow et al (2005) and Dean (2005). There is also an online database by Dr. Weber R.R for 343 papers related to stochastic scheduling at: http://www.statslab.cam.ac.uk/~rrw1/stoc_sched/index.html.

Online scheduling

Online algorithms have been used to deal with operational issues. In an online scheduling problem data is a sequence of requests, which are revealed over time and the algorithm must decide which request to process next. Online algorithms are inherently dynamic in nature but they typically ignore historical data or data about some of the future requests. Thus they are not suitable for the BAP with stochastic release dates. Online algorithms deal with operational decisions. For related literature consult Van Hentenryck and Bent (2006).

Online Stochastic Scheduling (OSS)

It is only until recently that OSS was recognized by researchers (Eliiyi and Kocer, 2005). Online stochastic optimization in general combines the later two approaches, discussed earlier, to decision making under uncertainty and exploits their respective strengths, by focusing on the data and uncertainty of the problem instance online and using stochastic information to make more informed decisions, while gradually reducing the uncertainty over time. The uncertainty in online stochastic combinatorial optimization concerns the requests: which requests come and when. In the next subsection we present the offline, online, and stochastic online general formulations. The former two are presented to make the concept straightforward.

Bent and Van Hentenryck (2004) consider online stochastic optimization problems where time constraints severely limit the number of offline optimizations, which can be performed at decision time and/or in between decisions. They propose a novel approach, which combines the salient features of the earlier approaches: the evaluation of every decision on all samples (expectation) and the ability to avoid distributing the samples among decisions (consensus). The key idea underlying the algorithm is to approximate the regret of a decision d. The regret algorithm is evaluated on two fundamentally different applications: online packet scheduling in networks and online multiple vehicle-routing with time windows. On both applications, it produces significant benefits over prior approaches. They assumed though that a black-box simulator could model the arrival process of the tasks. They had no knowledge of the distribution inside the simulator but could obtain samples. This has the disadvantage of not being able to weight each sampled scenario.

Chou et al. (2005) consider a model that combines the features of stochastic and online scheduling. They prove asymptotic optimality of the online weighted shortest expected processing time rule for the single machine problem assuming that the weights and the processing times can be bounded from above and below by constants.

Megow et al. (2005) consider a non-preemptive, stochastic parallel machine-scheduling model with the goal to minimize the weighted completion times of jobs. They propose a simple online scheduling policy for the first model, and prove a performance guarantee that matches the currently best-known performance guarantee for stochastic parallel machine scheduling. For the more general model with job release dates they derive an analogous result, and for distributed processing times they improve upon the previously best known performance guarantee for stochastic parallel machine scheduling.

Eliiyi and Kocer (2005) focus on a model that generalizes stochastic scheduling and online scheduling. They assume that the jobs arrive online. Once a job arrives, its expected processing time is revealed, but the actual processing time remains unknown until the job is completed. They model the problem and identify application areas. They also review some solution procedures that can be utilized for the optimal solution of this problem.

Shultz (2005) consider the stochastic identical parallel machine scheduling problem and its online extension, when the objective is to minimize the expected total weighted completion time of a set of jobs that are released over time. They give randomized as well as deterministic online and offline algorithms that have the best known performance guarantees in either setting, online or offline and deterministic or randomized. Their analysis is based on a novel linear programming relaxation for stochastic scheduling problems that can be solved online.

Van Hentenryck et al. (2005) in a tutorial, consider online stochastic combinatorial optimization problems where uncertainties, i.e., which requests come and when, are characterized by distributions that can be sampled and where time constraints severely limit the number of offline optimizations which can be performed at decision time and/or in between decisions. They propose online stochastic algorithms that combine the frameworks of online and stochastic optimization.

Wu et al (2005) examine whether Bent and Van Hentenryck (2004) combination methods can be adapted to scheduling with uncertain release dates, and to determine how effective and significant the methods are. In particular, they develop four ways of applying Bent and Van Hentenryck (2004) consensus approach to the problem. In addition, they propose a probabilistic sampling method to handle lead-time uncertainty. That is they use this knowledge to select samples, and associate with them weights corresponding to their probability. This method allows them to generate fewer samples and have a more accurate model of future scenarios.

In October 2006 Van Hentenryck and Bent published a book on Online Stochastic Combinatorial Optimization. The book presents a novel framework for online stochastic optimization, and address decision-making under uncertainty and time constraints.

Online Stochastic Optimization Modeling Issues

One of the critical issues faced by online stochastic algorithms is how to use time wisely in time constraint problems, since only a few samples can be optimized within the time constraints. In other words, the algorithm must find an effective approach to optimize the samples and extract information from their solutions in order to make more informed decisions. When time is not a factor, a traditional approach (Chang, Givan, & Chong, 2000) consists of using an expectation algorithm, which works as follow: at time *t*, generate a number of samples S_i, solve each sample once per available request *r* by serving *r* at *t*, and select the schedule with the best request overall.

Unfortunately, the expectation approach does not perform well under time constraints, since it must distribute its available optimizations across all requests. This issue was recognized and addressed in (Bent & Van Hentenryck 2004a) where a consensus approach was proposed. Its key idea was to solve as many samples as possible and to select the request, which is chosen most often in the sample solutions at time *t*. The consensus approach was shown to outperform the expectation method on online packet scheduling under time constraints. However, as decision time increases the quality of the consensus approach levels off and is eventually outperformed by the expectation method. It is also possible to hybridize the expectation and consensus approaches but the resulting method loses some of the benefits of consensus under strict time constraints (Bent & Van Hentenryck 2004b)

Algorithms

There are three major approached to solving OSS problems and they all focus on how to choose the best scenario at time t. These approaches are (Bent and Van Hentenryck, 2004b; Wu et al., 2005):

Expectation: Obtain samples of possible future arrival scenarios, compute schedules for al samples and choose schedule with the best reward. (Best results, too much time)

Consensus: Obtain samples of possible future arrival scenarios, compute schedules for all the samples and chooses the decision appearing the most in the optimal schedules of the samples. It is faster than expectation since it requires a fewer number of optimizations.

Regret: It is similar to consensus, but as well as computing the optimal schedules also computes the loss of reward for each other possible decision, and then chooses the decision that has the lowest total loss. This method approaches the former one (expectation) when there is sufficient computation time and the later one (consensus) when time is limited.

We will present the general formulation of these approaches as presented by various Researches (Chang, Givan, & Chong 2000; Bent and Van Hentenryck, 2004; Wu et al. 2005).

Expectation (E): This is the primary method proposed by (Chang, Givan, & Chong 2000) for online packet scheduling. Informally speaking, the method generates future requests by sampling and evaluates each available request against that sample. A simple implementation can be specified as follows:

```
chooseRequest-E(R, t)
1  A ← ready(R, t);
2  for r ∈ A
3     do f(r) ← 0;
4  for i ← 1 ... O/|A|
5     do S ← R ∪ getSample([t + 1, t + Δ]);
6        for r ∈ A
7           do σ ← optimalSolution(S \ {r}, t + 1);
8              f(r) ← f(r) + w(r) + w(σ);
9  return argmax(r ∈ A) f(r);
```

(Source: Bent and Van Hentenryck, 2004b)

Where: *ω(r)*: weight (usually representing gain if request *r* served), *R*: number of requests, *O*: number of offline optimizations at each step, *f(r):* reward function

Lines 2-3 initialize the evaluation function *f(j)* for each request *r*. The algorithm then generates a number of samples for future requests (line 4). For each such sample, the algorithm computes the set *R* of all available and sampled requests at time *t* (line 5). The algorithm then considers each available request *r* successively (line 6), it implicitly schedules *r* at time *t*, and applies an optimal offline algorithm (line 7) using $S \backslash \{r\}$ and the time horizon. The evaluation of request *r* is updated in line 8 by incrementing it with its weight and the score of the corresponding optimal offline solution. All scenarios are evaluated for all available requests and the algorithm then returns the request *r* with the highest evaluation.

Consensus (C): This algorithm uses stochastic information in a fundamental different way. Algorithm C was introduced in (Bent & Van Hentenryck 2004a) as an abstraction of the sampling method used in online vehicle routing (Bent & Van Hentenryck 2001). Instead of evaluating each possible request at time *t* with respect to each sample, algorithm C executes the offline algorithm on the available and sampled requests and to count the number of times a request is scheduled at time t in each resulting solution. Then the request with the highest count is selected. Algorithm C can be specified as follows:

```
chooseRequest-C(R, t)
1  for r ∈ R
2     do f(r) ← 0;
3  for i ← 1 ... O
4     do S ← R ∪ getSample([t + 1, t + Δ]);
5        σ ← optimalSolution(S, t);
6        f(σ(t)) ← f(σ(t)) + 1;
7  return argmax(r ∈ R) f(r);
```

(Source: Bent and Van Hentenryck, 2004b)

At line 5 the offline algorithm is called with all available and sampled requests and a time horizon starting at *t* and line 6 which increments the number of times request (*t*) is scheduled first. Line 7 simply returns the request with the largest count. Algorithm C has several appealing features. First, it does not partition the available samples between the requests, which is a significant advantage when the number of samples is small and/or when the number of requests is large. Second, it avoids the conceptual complexity of identifying symmetric or dominated requests.

Regret (R): The key insight in Algorithm R is the recognition that, in many applications, it is possible to estimate the local loss of a request *r* at time *t* quickly. In other words, once the optimal solution of a scenario is computed, it is easy to compute the local loss of all the requests, thus approximating E with one optimization. This intuition can be formalized using the concept of regret.

Definition 2 (Regret). *A regret is a function which, given an optimal solution* $\gamma = \mathcal{O}(\mathbf{S}_{t-1}, \mathbf{R}_t)$*, over-approximates the local loss of r, i.e.,*

$$\text{REGRET}(\mathbf{S}_{t-1}, \mathbf{R}_t, r, \gamma) \geq \text{LocalLoss}(\mathbf{S}_{t-1}, \mathbf{R}_t, r).$$

Moreover, there exists two functions f_o *and* f_r *such that*

- $\mathcal{O}(\mathbf{S}_{t-1}, \mathbf{R}_t)$ *runs in time* $O(f_o(\mathbf{S}_{t-1}, \mathbf{R}_t))$*;*
- $\text{REGRET}(\mathbf{S}_{t-1}, \mathbf{R}_t, r, \gamma)$ *runs in time* $O(f_r(\mathbf{S}_{t-1}, \mathbf{R}_t))$*;*
- $|R| f_r(\mathbf{S}_{t-1}, \mathbf{R}_t)$ *is* $O(f_o(\mathbf{S}_{t-1}, \mathbf{R}_t))$.

(Source: Van Hentenryck et al., 2005)

Intuitively, the complexity requirement states that the computation of the |R| regrets does not take more time than the optimization. Regrets typically exist in practical applications. Algorithm R works as follows:

```
CHOOSEREQUEST-R(S_{t-1}, R_t)
1   F ← FEASIBLE(S_{t-1}, R_t);
2   for r ∈ F
3     do f(r) ← 0;
4   for i ← 1 ... T
5     do A ← R_t :: SAMPLE(R_t, Δ);
6        γ ← O(S_{t-1}, A);
7        f(γ_t) ← f(γ_t) + w(γ);
8        for r ∈ F \ {γ_t}
9          do f(r) ← f(r) + (w(γ) − REGRET(S_{t-1}, A, r, γ));
10  return argmax(r ∈ F) f(r);
```

(Source: Van Hentenryck et al., 2005)

Its basic organization follows algorithm C. However, instead of assigning some credit only to the request selected at time *t* for a given scenario *s*, algorithm R (lines 8-9) uses the regrets to compute, for each available request r, an approximation of the best solution of *s* serving *r* at time *t*, i.e., $\omega(\gamma) - REGRET(St-1, A, r, \gamma)$. Hence every available request is given an evaluation for every scenario for the cost of a single offline optimization. Observe that algorithm R performs T offline optimizations at time *t*.

5. DISCRETE DYNAMIC BERTH ALLOCATION

Introduction

As discussed in the previous sections container terminal operators seek for the efficient BSPs that may reduce vessels turnaround time, increase port throughput, leading to higher revenues and increased competitiveness of the port, while at the same time keeping the customers' satisfaction at a desired level (usually set by contractual agreements). In practice vessels arrive at the port over a period of time and normally request start or finish of service within a time window. These time windows are usually determined through contractual agreements between the port operator and the carrier, in terms of time of start or finish of service after the vessel's arrival at the port. Based on these contractual agreements, serving a vessel prior or within these time windows provides certain premiums to the port operator while service past these time window results in penalties. Port operators usually assign vessels to berths with the objective to minimize/maximize these costs/ premiums. Furthermore, port operators are interested in retaining satisfactory levels of service, (usually in terms of the total service or wait time), for all the customers, since this is usually a measure for negotiating future contractual agreements with new customers. Finally one of the main port operator concerns is the minimization of the cost associated with the vessels' handling operations (Vis and de Koster, 2003; Steenken et al., 2004).

In light of the above discussion this chapter presents a generic formulation for the discrete BAP that addresses these issues. Our research deviates from BSPs presented so far in the literature by incorporating several parameters of the BAP including minimization of the total (un)weighted service time, costs from vessel waiting, cost from vessels delayed departures, premiums from early and timely departures, minimization of the handling time. Furthermore, and to our knowledge, this is the first time that the BAP is addressed in a time window setup.

This chapter is organized as follows. The next section discusses the problem formulation, while the third section presents how the general model can be reduced to BSPs found in the literature. The fourth section introduces a Genetic Algorithm (GA) based heuristic solution algorithm. The fifth section provides a number of computational examples to evaluate the performance of the heuristic and the last section concludes the chapter.

Problem Formulation

To formulate the generic discrete and dynamic BAP (GDDBAP) we define the following:

i = $(1,\ldots\ldots,I) \in B$ set of berths,

j = $(1,\ldots.,T) \in V$ set of vessels,

k = $(1,\ldots.,T) \in O$ set of service orders,

S_i = Time when berth i becomes available for the first time in the current planning horizon,

A_j = Arrival time of vessel j,

C_{ij} = Handling time of vessel j at berth i,

y_{ijk} = Idle time of berth i before vessel j is serviced as the k^{th} vessel,

X_{ijk1} = 1 if vessel j serviced at berth i as the k^{th} vessel and departs or berths before the requested time window and zero otherwise,

X_{ijk2} = 1 if vessel j serviced at berth i as the k^{th} vessel and departs or berths after the requested time window and zero otherwise,

X_{ijk3} = 1 if if vessel j serviced at berth i as the k^{th} vessel and departs or berths within the requested time window and zero otherwise,

a_{j1} = Hourly earliness departure premium for vessel j,

a_{j2} = Hourly earliness berthing premium for vessel j,

a_{j3} = Hourly cost of wait time of vessel j,

b_{j1} = Hourly lateness departure penalty for vessel j,

b_{j2} = Hourly lateness berthing penalty for vessel j,

γ_{j1} = Hourly timely departure premium for vessel j,

γ_{j2} = Hourly timely berthing premium for vessel j,

t_{j1} = Requested early departure/berthing time of vessel j,

t_{j2} = Requested late departure/berthing time of vessel j,

WST_j = 1 if vessel j sets a request for early, timely and late departure 0 otherwise,

HC_j = 1 if handling cost is considered and 0 otherwise,

CC_{ij} = Handling cost of vessel j serviced at berth i,

DT_{ijk} = Difference of early/late actual and requested finish/start time of vessel j serviced at berth i as the k^{th} vessel,

DT_{ijk}^{+} = $\max(0, DT_{ijk})$,

DT_{ijk}^{-} = $\min(0, DT_{ijk})$,

DTT_{ijk} = Difference of early timely requested and actual timely finish/start time of vessel j serviced at berth i as the k^{th} vessel.

R_{ij} = 0 if vessel j cannot be serviced at berth i due to physical or technical restrictions and 1 otherwise

The formulation of the GDDBAP is as follows:

[GDDBAP]:

$$\min - \sum_i \sum_j \sum_k \left\langle \begin{array}{l} \{a_{j1}WST_j + a_{j2}(1-WST_j)\}DT^{+}_{ijk} + \{b_{j1}WST_j + b_{j2}(1-WST_j)\}DT^{-}_{ijk} \\ -\{\gamma_{j1}WST_j + \gamma_{j2}(1-WST_j)\}DTT_{ijk} - HC_j CC_{ij}(\sum_{r=1}^{3} X_{ijkr}) \\ -a_{j3}WST_j\{(t_{j1}-A_j-C_{ij})X_{ijk1} - DT^{+}_{ijk} + (t_{j2}-A_j-C_{ij})X_{ijk2} - DT^{-}_{ijk} \\ +(t_{j1}-A_j-C_{ij})X_{ijk3} - DTT_{ijk}\} \end{array} \right\rangle, \quad \text{(Eq. 5-1a)}$$

or

$$\min - \sum_i \sum_j \sum_k \left\langle \begin{array}{l} \{(a_{j1}+a_{j3})WST_j + a_{j2}(1-WST_j)\}DT^{+}_{ijk} + \\ \{(b_{j1}+a_{j3})WST_j + b_{j2}(1-WST_j)\}DT^{-}_{ijk} + \\ \{(\gamma_{j1}-a_{j3})WST_j + \gamma_{j2}(1-WST_j)\}DTT_{ijk} - HC_j CC_{ij}(\sum_{r=1}^{3} X_{ijkr}) - \\ a_{j3}WST_j\{(t_{j1}-A_j-C_{ij})X_{ijk1} + (t_{j2}-A_j-C_{ij})X_{ijk2} \\ +(t_{j1}-A_j-C_{ij})X_{ijk3}\} \end{array} \right\rangle, \quad \text{(Eq. 5-1b)}$$

Subject to: $\sum_i \sum_j \sum_k \sum_{r=1}^{3} X_{ijkr} = 1, \forall j$, (Eq. 5-2)

$$\sum_{j \in V} \sum_{r=1}^{3} X_{ijkr} \le 1, \forall i \in B, k \in O \quad \text{(Eq. 5-3)}$$

$$\sum_{m \ne j \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk} - (A_j - S_i)\sum_{r=1}^{3} X_{ijkr} \ge 0, \forall i \in B, j \in T, k \in O \quad \text{(Eq. 5-4)}$$

$$y_{ijk} \le M \sum_{r=1}^{3} X_{ijkr}, \forall i \in B, j \in T, k \in O \quad \text{(Eq. 5-5)}$$

$$DT^{+}_{ijk} \le (t_{j1} - WST_j C_{ij} - S_i)X_{ijk1} - y_{ijk} - \sum_{j \ne m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + alpha_{ijk}, \quad \text{(Eq. 5-6)}$$
$$\forall i \in B, j \in T, k \in O$$

$$t_{j1} X_{ijk1} \ge \sum_{j \ne m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{ijkr} + y_{imh}) + y_{ijk} + (WST_j C_{ij} + S_i)X_{ijk1} - M(1 - X_{ijk1}), \quad \text{(Eq. 5-7)}$$
$$\forall i \in B, j \in T, k \in O$$

$$DT^{-}_{ijk} \leq (t_{j2} - WST_j * C_{ij} - S_i)X_{ijk2} - y_{ijk} - \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + M(1 - X_{ijk2}) \text{,(Eq. 5-8)}$$
$$\forall i \in B, j \in T, k \in O$$

$$t_{j2} X_{ijk2} \leq \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk} + (WST_j C_{ij} + S_i) X_{ijk2}, \quad \text{(Eq. 5-9)}$$
$$\forall i \in B, j \in T, k \in O$$

$$alpha_{ijk} \leq M(1 - X_{ijk1}), \forall i \in B, j \in T, k \in O \quad \text{(Eq. 5-10)}$$

$$alpha_{ijk} \leq \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk}, \forall i \in B, j \in T, k \in O, \quad \text{(Eq. 5-11)}$$

$$DT^{+}_{ijk} \leq MX_{ijk1}, \forall i \in B, j \in T, k \in O, \quad \text{(Eq. 5-12)}$$

$$t_{j1} X_{ijk3} \leq \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk} + (WST_j C_{ij} + S_i) X_{ijk3} + M(1 - X_{ijk3}), \quad \text{(Eq. 5-13)}$$
$$\forall i \in B, j \in T, k \in O$$

$$t_{j2} X_{ijk3} \geq \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk} + (WST_j C_{ij} + S_i) X_{ijk3} - M(1 - X_{ijk3}), \quad \text{(Eq. 5-14)}$$
$$\forall i \in B, j \in T, k \in O$$

$$DTT_{ijk} \leq t_{j1} X_{ijk3} - \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) , \quad \text{(Eq. 5-15)}$$
$$- y_{ijk} - (WST_j C_{ij} + S_i) X_{ijk3} + M(1 - X_{ijk3}), \forall i \in B, j \in T, k \in O$$

$$\sum_{k} \sum_{r} X_{ijkr} \leq MR_{ij}, \forall i \in B, j \in T \quad \text{(Eq. 5-16)}$$

$X_{ijk1}, X_{ijk2}, X_{ijk3} \in \{0,1\}$, $y_{ijk} \geq 0$, $DT^{-}_{ijk} \leq 0, DT^{+}_{ijk} \geq 0, DTT_{ijk} \leq 0$, $alpha_{ijk} \geq 0$, M is a large positive number

In the objective function (5-1a and 5-1b) the first three terms correspond to the total cost from delayed departures/berthing, the total premiums from early departures/berthing, and the total premium from timely departures/berthing (depending on if WST_j=1 or 0). The fourth term corresponds to the total handling cost, while the last term to the total wait time cost. We should note that waiting costs are only applicable to customers with early and late departure requests.

Further explanations for the objective function are provided later in this section. Constraints (5-2) ensure that vessels must be serviced once; constraints (5-3) ensure that each berth services one vessel at a time; and constraints (5-4) and (5-5) ensure that each vessel is serviced after its arrival. Constraints (5-6) and (5-15) enforce the declaration of the decision and auxiliary variables. In the model waiting time costs are only applied to customers requesting a departure window deadline, since waiting time costs for customers requesting a start service time window are already considered. This formulation provides the port operator to differentiate between the customers that request berthing from departure. Thus smaller wait hourly costs will be assigned to the customers requesting time window service for departures, since wait time is of no interest. Further explanations are needed for the constraints (5-6) through (5-15) that estimate the start or completion time of each vessel. Equations (5-16) enforce different physical and technical constrains that do not allow the berthing of certain vessels at certain berths (i.e. berth depth).

If $X_{ijk1}= X_{ijk2}= X_{ijk3}=0$ then from (5-6), (5-10), (5-11) and (5-12) we obtain $DT_{ijk}^{+}=0$, from (5-7) we obtain $DT_{ijk}^{-}=0$, and from (5-15) we obtain that $DTT_{ijk}=0$; while constraints (5-7), (5-9), (5-13), and (5-14) are satisfied. Furthermore the wait time term in the objective function is reduced to zero.

On the other hand if: $X_{ijk1}= X_{ijk3}=0$, then due to (2) $X_{ijk2}=1$ and:

$$DT_{ijk}^{+} \leq -\sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) - y_{ijk}) + alpha_{ijk} \qquad \text{(Eq. 5-17)}$$

$$alpha_{ijk} \leq \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk} \qquad \text{(Eq. 5-18)}$$

$$DT_{ijk}^{+} \leq 0, \forall i \in B, j \in T, k \in O \qquad \text{(Eq. 5-19)}$$

$$DT^{-}_{ijk} \leq (t_{j2} - \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) - y_{ijk} - (WST_j * C_{ij} + S_i)) \quad \text{(Eq. 5-20)}$$

$$DTT_{ijk} \leq M \quad \text{(Eq. 5-21)}$$

This means that:

$$alpha_{ijk} = \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk} \, , \quad \text{(Eq. 5-22)}$$

$$DT^{+}_{ijk} = 0 \, , \quad \text{(Eq. 5-23)}$$

$$DT^{-}_{ijk} = (t_{j2} - \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) - y_{ijk} - (WTS_j * C_{ij} + S_i)) \leq 0 \, , \quad \text{(Eq. 5-24)}$$

$$DTT_{ijk} = 0 \, , \quad \text{(Eq. 5-25)}$$

If: $X_{ijk2} = X_{ijk3} = 0$, then $X_{ijk1} = 1$. This means that:

$$DT^{-}_{ijk} \leq M \, , \forall i \in B, \, j \in T, k \in O \, , \quad \text{(Eq. 5-26)}$$

$$alpha_{ijk} \leq 0 \, , \quad \text{(Eq. 5-27)}$$

$$DT^{+}_{ijk} \leq (t_{j1} - \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) - y_{ijk} - (WST_j * C_{ij} + S_i)) + alpha_{ijk} \quad \text{(Eq. 5-28)}$$

$$DTT_{ijk} \leq M \quad \text{(Eq. 5-29)}$$

This means that:

$$alpha_{ijk} = 0 \, , \quad \text{(Eq. 5-30)}$$

$$DT^{-}_{ijk} = 0 \, , \quad \text{(Eq. 5-31)}$$

$$DT_{ijk}^{+} = (t_{j1} - \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) - y_{ijk} - (WST_j * C_{ij} + S_i)) \geq 0 \qquad \text{(Eq. 5-32)}$$

$$DTT_{ijk} = 0, \qquad \text{(Eq. 5-33)}$$

Finally if: $X_{ijk1}= X_{ijk2}=0$, then $X_{ijk3}=1$ and as shown previously: $DT_{ijk}^{+} = 0$, $DT_{ijk}^{-} = 0$. Furthermore:

$$DTT_{ijk} = t_{j1} X_{ijk3} - \sum_{j \neq m \in T} \sum_{h<k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) - y_{ijk} - (WST_j C_{ij} + S_i) X_{ijk3}, \qquad \text{(Eq. 5-34)}$$

Model Adaptation to Different Berth Allocation Policies

Depending on the port operators' goals and customer agreements, the GDDBAP formulation can produce a number of different BSPs. In this section we will show how the GDDBAP can be reformulated to portray BSPs found in the literature. The BSPs discussed herein are: a) The minimum total weighted service time (Imai et al., 2003), b) The minimum total service time (Imai et al., 2001), c) The minimum total service time at an external berth (Imai et al., 2006b), d) The minimum cost berth allocation (Hansen et al., 2007), and e) The minimum cost with time windows and service deadlines.

Minimum total weighted service time (MTWST) BSP

***Proposition 5-I*:** [GDDBAP] can be reduced to a linear formulation of the BSP of Imai et al. (2003).

Proof: Assume that the time window is reduced to a point in time: $t_{j1} = t_{j2} = A_j, \forall j$. Also assume that: $a_{j1} = a_{j2} = a_{j3} = \gamma_{j1} = \gamma_{j2} = b_{j2} = 0,\ b_{j1} = b_j > 0,\ WST_j = 1,$ and $CH_j = 0, \forall j$. The formulation is then reduced to a model that will minimize the total weighted service time.

Minimum total service time (MTST) BSP

Proposition 5-II: [GDDBAP] is the general case of the BSP of Imai et al. (2001).

Proof: Since the formulation of Imai et al. (2001) is a special case of the formulation of Imai et al. (2003), it follows that the formulation Imai et al. (2001) is a special case of GDDBAP.

Minimum Cost Berth Allocation (MCBA) BSP

***Proposition 5-III*:** [GDDBAP] can be reduced to the BSP of Hansen et al. (2007).

Proof: Assume all vessels have not arrived at the port before we start the berth allocation. Assume that the requested departure time is reduced to a point in time $t_{j1}=t_{j2}=t_j$, and that all vessels only set early and late departure requests ($WST_j=1,\ a_{j2}=b_{j2}=0,\ \gamma_{j1} = \gamma_{j2} = 0$). The GDDBAP can be reduced to:

$$\min - \sum_i \sum_j \sum_k \left\langle \begin{array}{l} (a_{j1} + a_{j3}) DT^+_{ijk} + (b_{j1} + a_{j3}) DT^-_{ijk} - HC_j CC_{ij} (\sum_{r=1}^{2} X_{ijkr}) \\ - a_{j3} \{ (t_j - A_j - C_{ij}) \sum_{r=1,2} X_{ijkr} \} \end{array} \right\rangle, \qquad \text{(Eq. 5-35)}$$

Subject to: $\sum_i \sum_j \sum_k \sum_{r=1}^{2} X_{ijkr} = 1, \forall j,$ (Eq. 5-36)

$$\sum_{j\in V}\sum_{r=1}^{2} X_{ijkr} \leq 1, \forall i \in B, k \in O \tag{Eq. 5-37}$$

$$\sum_{m\neq j\in T}\sum_{h<k\in O}(C_{im}\sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk} - (A_j - S_i)\sum_{r=1}^{3} X_{ijkr} \geq 0, \forall i \in B, j \in T, k \in O \tag{Eq. 5-38}$$

$$y_{ijk} \leq M\sum_{r=1}^{3} X_{ijkr}, \forall i \in B, j \in T, k \in O \tag{Eq. 5-39}$$

$$DT^{+}_{ijk} \leq (t_{j1} - WST_j C_{ij} - S_i)X_{ijk1} - y_{ijk} - \sum_{j\neq m\in T}\sum_{h<k\in O}(C_{im}\sum_{r=1}^{2} X_{imhr} + y_{imh}) + alpha_{ijk}, \quad \forall i \in B, j \in T, k \in O \tag{Eq. 5-40}$$

$$t_{j1}X_{ijk1} \geq \sum_{j\neq m\in T}\sum_{h<k\in O}(C_{im}\sum_{r=1}^{2} X_{ijkr} + y_{imh}) + y_{ijk} + (WST_j C_{ij} + S_i)X_{ijk1} - M(1 - X_{ijk1}), \quad \forall i \in B, j \in T, k \in O \tag{Eq. 5-41}$$

$$t_j X_{ijk2} \leq \sum_{j\neq m\in T}\sum_{h<k\in O}(C_{im}\sum_{r=1}^{2} X_{imhr} + y_{imh}) + y_{ijk} + (C_{ij} + S_i)X_{ijk2}, \forall i \in B, j \in T, k \in O \tag{Eq. 5-42}$$

$$alpha_{ijk} \leq M(1 - X_{ijk1}), \forall i \in B, j \in T, k \in O \tag{Eq. 5-43}$$

$$alpha_{ijk} \leq \sum_{j\neq m\in T}\sum_{h<k\in O}(C_{im}\sum_{r=1}^{2} X_{imhr} + y_{imh}) + y_{ijk}, \forall i \in B, j \in T, k \in O, \tag{Eq. 5-44}$$

$X_{ijk1}, X_{ijk2} \in \{0,1\}$, $y_{ijk} \geq 0$, $DT^{-}_{ijk} \leq 0, DT^{+}_{ijk} \geq 0$, $alpha_{ijk} \geq 0$, M is a large positive number

The first term of the objective function minimizes the total cost from delayed departures and maximizes the total premiums from early departures; the second term minimizes the total handling cost, and the third term the total waiting cost. This formulation will produce the same results as with the BSP formulation of Hansen et al. (2007).

Minimum external berth total service time (MEBTST) BSP

Proposition 5-IV: [GDDBAP] is a general case of the BSP of Imai et al. (2006b).

Proof: Assume all vessels have not arrived at the port before we start the berth allocation. Assume that the time window is reduced to a point in time, where $t_{j1}=0$, $t_{j2}=L_j+A_j$, where L_j is the limit of waiting time of vessel j. Also assume that $a_{j1}= a_{j2}=a_{j3}=0$, $b_{j1}= b_{j2}=0$, $WST_j=0$, $HC_j=0$, $\gamma_{j1}=0$ and $\gamma_{j2}=-C_{Qj}$ where Q is the external berth similar to Imai et al (2006b). The fourth index in the decision variable X_{ijkl} (l=1, 2, 3) can be ignored and the GDDBAP is then reduced to:

$$\min \sum_i \sum_j \sum_k C_{Qj} X_{ijk}\ , \qquad \text{(Eq. 5-45)}$$

Subject to: $$\sum_i \sum_j \sum_k X_{ijk} = 1, \forall j\ , \qquad \text{(Eq. 5-46)}$$

$$\sum_{j \in V} X_{ijk} \le 1, \forall i \in B, k \in U\ , \qquad \text{(Eq. 5-47)}$$

$$\sum_{m \ne j \in T} \sum_{h<k \in O} (C_{im} X_{imh} + y_{imh}) + y_{ijk} - (A_j - S_i) X_{ijk} \ge 0, \forall i \in B, j \in T, k \in O\ , \qquad \text{(Eq. 5-48)}$$

$$L_j X_{ijk} \ge \sum_{j \ne m \in T} \sum_{h<k \in O} (C_{im} X_{imh} + y_{imh}) + y_{ijk} + (S_i - A_j) X_{ijk} - M(1 - X_{ijk})\ , \qquad \text{(Eq. 5-49)}$$
$$\forall i (\ne Q) \in B, j \in T, k \in O$$

This formulation will produce the same results as with the BSP formulation of Imai et al. (2006b). The proof for the static case is similar and thus omitted.

Minimum Cost with Time Window Service Deadlines (MCTWSD) BSP

Proposition 5-V: [GDDBAP] is a general case of the Minimum Cost with Time Window Service Deadlines BSP.

Proof: Assume all vessels have not arrived at the port before we start the berth allocation. Assume that $a_{j2}=a_{j3}=0$, $b_{j2}=0$, $WST_j=1$, and $CH_j=0$. The GDDBAP is then reduced to the BSP with minimum cost with time window service deadlines.

Solution Procedure

A heuristic was developed for the GDDBAP, since it is not likely that an efficient exact solution procedure exists, leading to an optimal solution in polynomially bounded computation time. The procedure we employ for the heuristic is the Genetic Algorithms (GAs). GAs based heuristics are widely applied for plenty of practical problems of mathematical programming, which are difficult to solve in terms of polynomially bounded computation time. They work on the principle of evolving a population of trial solutions over a number of iterations, to adapt them to the fitness landscape expressed in the objective function (Taboada, 2007).

Representation

Although binary-coded GAs are commonly used, there is an increasing interest in alternative encoding strategies, such as integer and real-valued representations. For some problem domains, like scheduling problems, it can be argued that the binary representation is, in fact, deceptive since it obscures the nature of the search (Taboada, 2007). Thus, in this chapter we use an integer chromosomal representation in order to exploit in full the characteristics of the problem. For instance, consider the following example of 5 vessels and 2 berths (fig.5-1). For this problem each chromosome will have ten cells {chromosome length= (Number of Berths) x (Number of Customers)}.

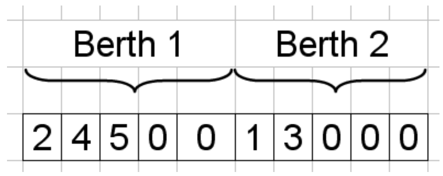


Figure 5-1 Chromosome Representation

The first 5 cells represent the 5 possible service orders in Berth1 and the last 5 cells the 5 possible service orders in Berth 2. In this assignment Vessel 2, 4, and 5 are serviced at Berth 1 as the first, second and third vessel respectively, and Vessel 1 and 3 are serviced in Berth 2 as the first an second vessel respectively. No vessel will be serviced after vessel 5 and three (zero value of cell).

Genetic Operations: Crossover & Mutation

Crossover can combine information from two parents while mutation can introduce new information. Crossover is explorative[10]; it makes a big jump to an area somewhere "in between" two parent areas (Eiben, and Smith, 2003). On the other hand mutation is exploitative[11]; it creates random diversions staying near or not in the area of the parent, depending on the mutation (insert, swap, inversion, and scramble). There is a debate on the use of crossover and mutation and which approach is the best to use. The main conclusion is that the performance of either mutation or crossover is highly affected by the problems' domain. In our problem, at each generation the crossover operation will generate a large number of infeasible children in terms of constraint set (2) (i.e., a child chromosome may not service all the vessels while other vessels are served twice). In the BAP literature simple heuristics were applied to eliminate this problem (Nishimura 2001; Imai 2003; Imai 2006a; 2006b). After running several computational examples with and without crossover results showed that problems solved with crossover returned worse solutions than

[10] ***Exploration***: Discovering promising areas in the search space, i.e. gaining information on the problem
[11] ***Exploitation***: Optimizing within a promising area, i.e. using information

problems using only mutation and were computationally more expensive. We do acknowledge that complex crossover techniques (partially mapped crossover, cycle crossover, and edge recombination) could eliminate the former insufficiency of the crossover operation. This could result though to a significant increase of the computational time and was not implemented within this chapter.

Instead of crossover we experimented with four different types of mutation: insert, swap, inversion, and scramble mutations (fig.5-2) that were applied to all the chromosomes at each generation. Each of the four types of mutations has its own characteristics in terms of preserving the order and adjacency information. Insert mutation picks two cells at random and moves the second one to follow the first, thus preserving most of the order and adjacency information. Inversion mutation picks two cells at random and then inverts the substring between them preserving most adjacency information (only breaks two links) but disrupting the order information. Swap mutation picks two cells from a chromosome and swaps their positions preserving most of the adjacency information but disrupting the order. Finally, scramble scrambles the position of a subset of cells of the chromosome.

Computational experiments showed that when all four mutations were applied the GA algorithm converged at a faster rate and there was significant improvement in the value of the objective function. Thus in our algorithm we employed all four mutation types but as the GA progressed the weight was shifted from the Inverse and Scramble mutation to the Insert and Swap mutation. In this way in the beginning of the search the heuristic performs large *jumps* and as the objective function improves the heuristic searches in an increasing smaller region.

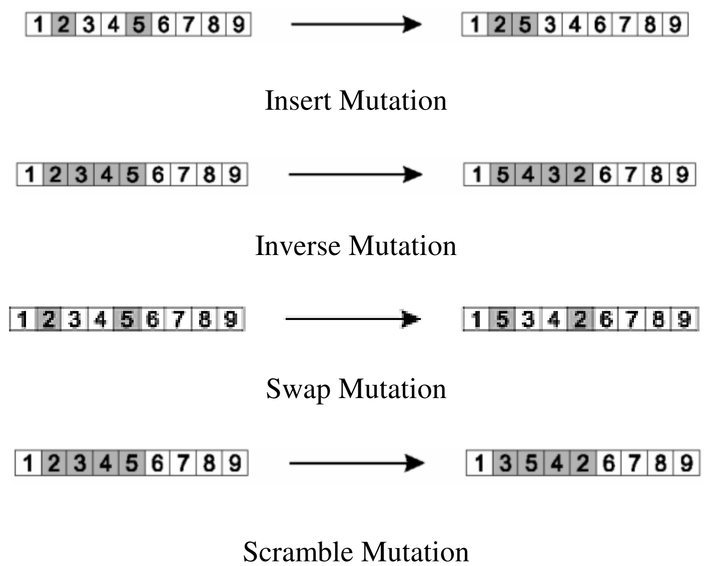


Figure 5-2 Example of the Mutation Operation

Technical/Physical Restrictions

In a mutated chromosome, all vessels cannot be serviced at the assigned berth, because of the physical or technical conditions. After the mutated individuals have been created and before the fitness function evaluation if a vessel does not satisfy constraint (Eq. 5-16), then the chromosome is assigned a very large function value.

Fitness/Selection

The GDDBAP is a minimization problem; thus the smaller the objective function value is, the higher the fitness value must be. The best solutions likely have an extremely good fitness value among solutions obtained where there is no significant difference between them in the objective function value. In order to avoid trapping the algorithm at local optimal locations of the solution space and instead of using a fitness function different then the objective function (as in Nishimura et al., 2001; and Imai et al., 2003), we use the objective function as the fitness function and select a number of medium and low fitness solutions probabilistically among the children of the next generation. As discussed previously all the chromosomes from the previous generation are candidates for the next generation. Several selection algorithms exist in the literature (Taboada, 2007). One of the most common one is the so-called roulette wheel selection (Goldberg, 1989), which is implemented in this chapter as follows:

Roulette Wheel Selection Algorithm

Step 0: Create an empty set *S* to hold the individuals that will proceed as the next generation

Step 1: Normalize the fitness values of the individual chromosomes[12]. Normalization means multiplying the fitness value of each individual by a fixed number, so that the sum of all fitness values equals 1.

Step 2: The population is sorted by descending fitness values.

Step 3: Accumulated normalized fitness values are computed (the accumulated fitness value of an individual is the sum of its own fitness value plus the fitness values of all the previous individuals).

Step 4: Choose a random value *R* between 0 and 1

Step 5: Select the first individual whose accumulated normalized value is greater than *R* and add it to *S*.

Step 6: If $|S|$ is equal to the initial population Stop else removed that individual and go to Step 1.

[12] In our heuristic the number of chromosomes that enter the wheel selection is 4 times larger then the ones that will move on as the next generation

The procedure of the GA heuristic algorithm is outlined in figure 5-3.

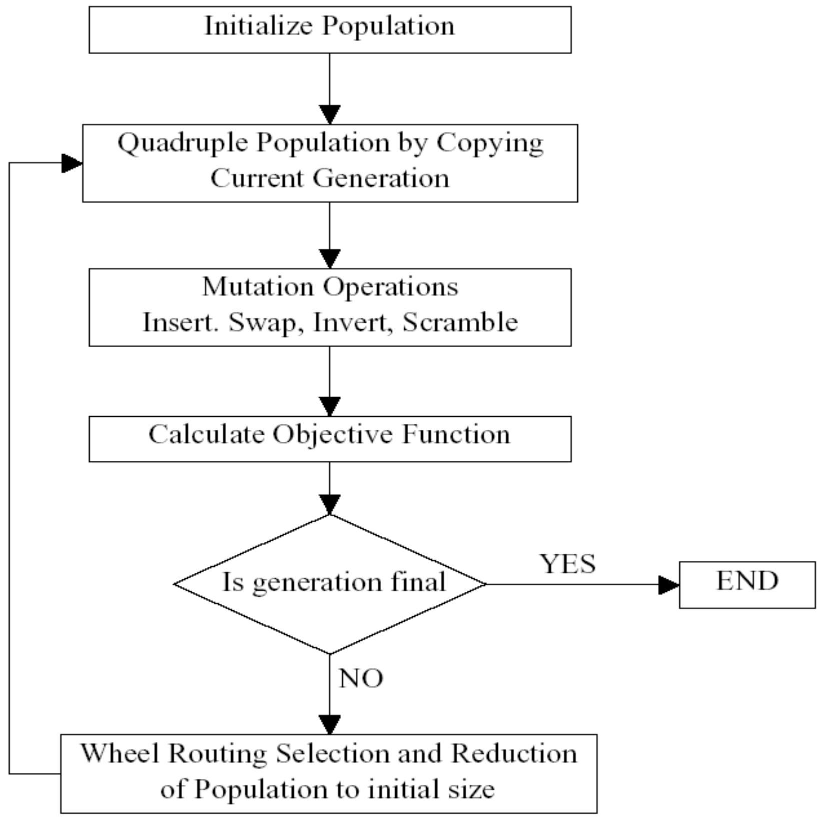


Figure 5-3 GA Heuristic

Applicability of GA Heuristic to different BSP

The proposed heuristic is applicable to any of the reduced GDDBAP BSPs, with the exception of the MEBTST BSP. This is due to the infeasibility due to constraints (Eq. 5-49). Due to the nature of the problem there is no intelligent way of using mutation or crossover operations without obtaining a large number of infeasible solutions due to constraint set (Eq. 5-49). In order to avoid this issue and when solving the MEBTST problem we propose the following small neighborhood variable search heuristic:

Heuristic 5-1: Small neighborhood variable search heuristic

Step 0: Randomly move all the vessels assigned to the external berth to the other internal berths

Step 1: Select the first internal berth

Step 2: Reassign the vessels on the selected internal and the external berth using a branch and bound algorithm

Step 3: If the previous berth was the last internal berth end else one select the next berth and move to step 1

This heuristic is applied to each chromosome after the mutation operations are finished.

Computational Examples

Dataset Description

Problems used in the experiments were generated randomly but systematically. When creating the experimental data the focus was in creating datasets that would be computationally challenging. We developed forty problem sets where vessels are served with various handling volumes at a multi-user container terminal (MUT) with five and ten berths, with two planning horizons of one and two weeks (Tables 5-1 and 5-2). The random generation process was based on data from two real world container terminals with similar terminal operating systems (one in Europe and one in the US). The range of variables and parameters considered were chosen according to the data obtained from these two container terminals. Vessel handling volumes ranged from 500 to 4,000 (TEU) based on a uniform distribution pattern. The handling time of a vessel was dependent on the berth assigned, and was a function of the number of the cranes that may be assigned. We assume that 2 to 3 cranes operate on small sized vessels (<2000 TEU), 3 to 4 cranes on medium sized vessels (<3000 TEU), and 4-6 on large mother vessels (<4000 TEU). The average crane productivity was assumed to be 25 TEU/hour. The average vessels per berth per week equivalence (VBWE) were 5 while the minimum, maximum and average handling time was 10, 51, and 32 hours. Testing instances with lower VBWE averages would not be provide representative evaluation of the heuristics performance.

The minimum handling time was calculated by dividing the handling volume by the average productivity of a crane multiplied by the number of cranes operates on the vessel. The average handling time per berth was 24 hours. Random numbers were used to generate the handling time of vessels at the other berths, always in relation to the berth with the minimum handling time. The association of the minimum time with the berth was also made randomly. Availability of berths was calculated using a uniform probability with a minimum of zero and a maximum of 10 hours. One of the most crucial issues in these experiments was the selection of the interarrival vessel distribution. We assumed that vessels arrived randomly with a minimum of 20 and a maximum of 25 vessels per week. Arrival times of vessels within the week period were randomly generated. Figure 5-4 shows the vessel arrival distribution in the form of a bar chart. Weights for the vessels were also randomly generated.

Experimental Results

The solution procedure was coded in Scilab 4.1 on a Dell Precision 670 Workstation, 2GB RAM. The number of individuals was set to 50 and the number of generations was set to 50. The solution procedure was evaluated using two basic BSPs: a) Minimum Total Service Time (MTST), and b) Minimum Total Weighted Service Time (MTWST). The focus of the experiments was on the optimality, robustness and computational efficiency of the proposed algorithm. The initial schedule for each experiment was obtained in a random fashion without using any rules since this would bias results.

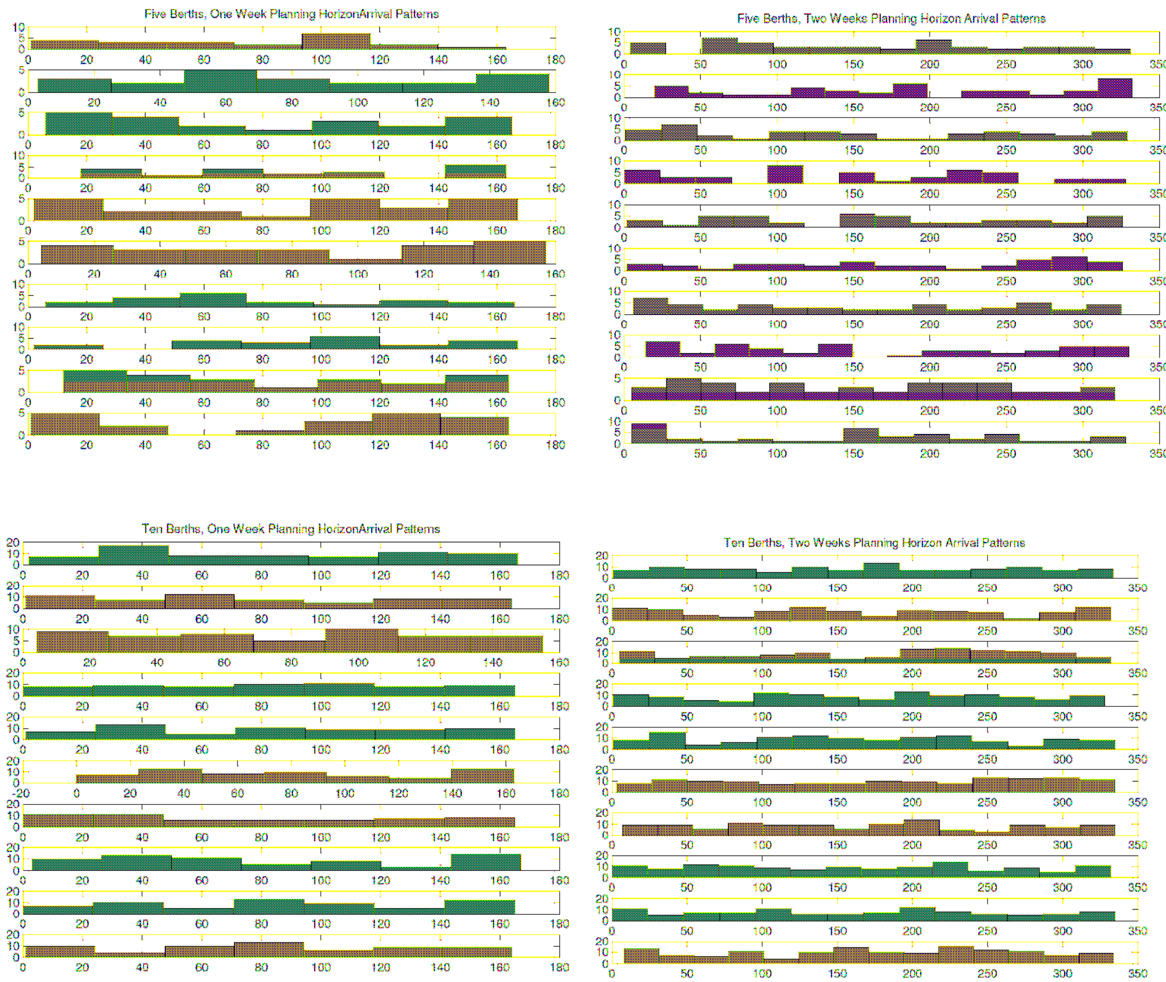


Figure 5-4 Vessel Arrivals

Table 5-3 shows the minimum and maximum values of the objective function for the MTST BSP weekly schedule for both berth capacities obtained from five trials using the GA algorithm and the optimal values obtained using CPLEX 9.0. For the rest of the problems optimal solutions were not obtained even after several hours of computations.

Table 5-1 Dataset Information One Week Time Horizon

	FIVE BERTHS		***Ten Berths***	
Instance	Vessels	Volume (TEU)	Vessels	Volume (TEU)
1	22	41,734	68	159,362
2	21	55,212	58	118,981
3	21	50,938	53	115,885
4	20	37,405	63	140,794
5	23	55,120	64	145,334
6	23	50,877	60	135,584
7	20	46,970	55	136,616
8	21	45,252	64	141,741
9	22	50,957	61	147,214
10	20	44,752	61	143,987
Average	21	47,922	61	138,550
Annual	554	1,245,964	1,578	3,602,295

For each one of the eighty problems, the ratio of the range of the objective function values for 5 trials (different starting populations) to the lowest objective value, which can be expressed by the highest objective value during the five trials divided by the lowest objective value during the five trials, was calculated. Results are reported in figure 5-4. The first ten sets of bars of each sub-graph show the range of objective values for each planning horizon (one and two weeks) and berth capacity (five and ten berths) for each one of the ten test instances. The last (eleventh bar set) shows the average range of objective values over all the test instances. The average ratio was less than 10%, and thus we can conclude that results obtained from the GA algorithm are consistent for different trials.

The computational time was also measured during the evaluation of the heuristic. Figure 5-5 shows the average computational time for the five trials for each of the 20 instances of the problem.

Table 5-2 Dataset Information Two Weeks Time Horizon

	Five Berths		*Ten Berths*	
Instance	Vessels	Volume (TEU)	Vessels	Volume (TEU)
1	47	107,490	115	263,108
2	42	96,000	106	237,840
3	44	104,113	124	274,267
4	44	100,231	118	259,327
5	44	105,915	124	300,197
6	40	84,162	137	307,169
7	47	103,250	116	281,467
8	49	95,117	131	306,686
9	44	83,792	105	241,788
10	41	93,250	138	318,769
Average	44	97,332	121	279,062
Annual	1,149	2,530,632	3,156	7,255,607

Table 5-3 Objective function values (in hours) for the MTST BSP

	Planning Horizon of One Week			
	Five Berths		Ten Berths	
Problem Instance	% Difference CLEX-GA (Min and Max)		% Difference CLEX-GA (Min and Max)	
1	0%	0%	0%	6%
2	0%	0%	0%	0%
3	0%	2%	0%	3%
4	0%	1%	0%	3%
5	0%	0%	0%	3%
6	0%	1%	0%	2%
7	0%	1%	0%	7%
8	0%	2%	0%	2%
9	0%	1%	0%	2%
10	0%	1%	0%	2%

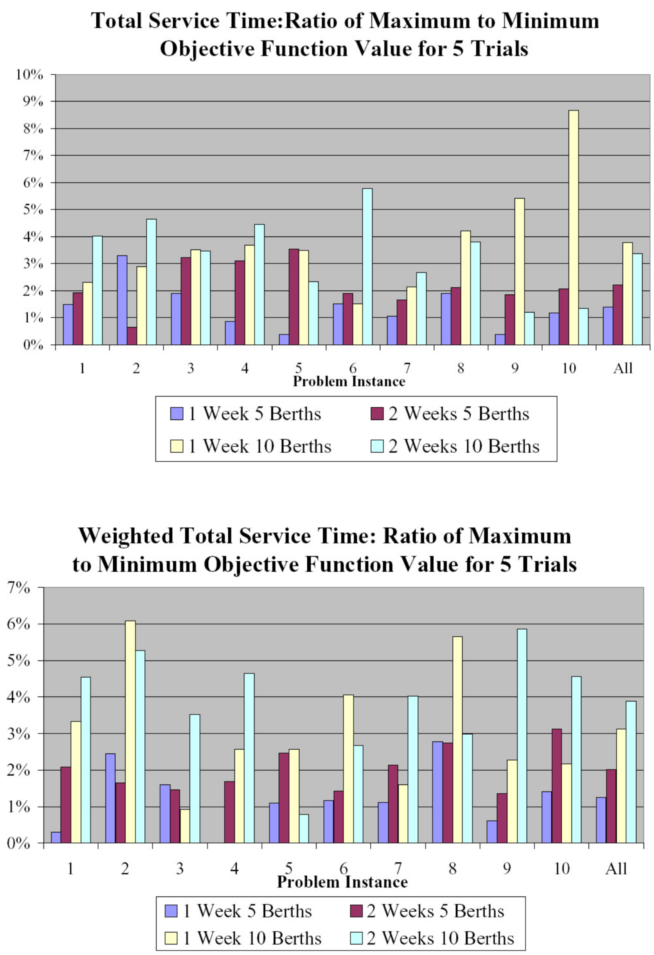
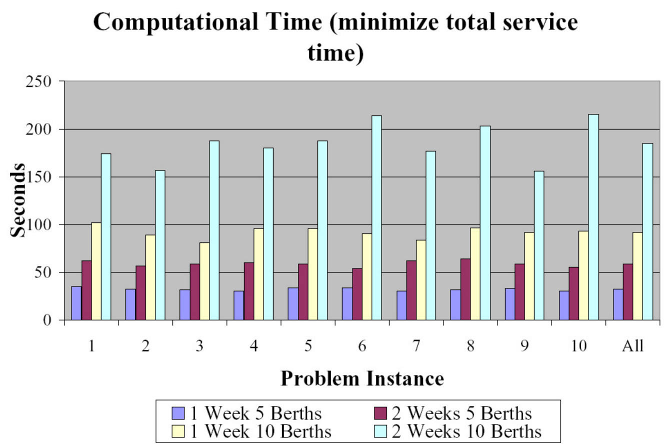


Figure 5-5 Range of the objective function values

We also evaluated the iterations that are required for the algorithm to stabilize. Figure 5-6 shows the typical pattern of the progress of the objective function value at each step for the smallest and largest instance of all the instances used. We observe the algorithm stabilizes after 200 iterations. The maximum time per iteration was less then 1 sec. As we increase the population convergence of the algorithm is established in fewer generations and the progression is smoother.



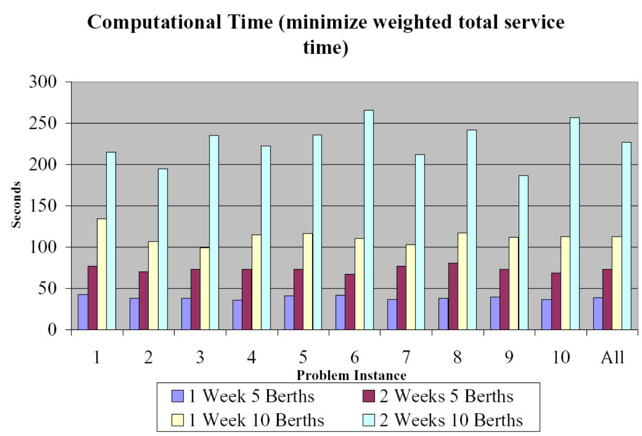


Figure 5-6 Computational Time

Conclusions

In this chapter we presented a generic linear MIP formulation for the discrete and dynamic BAP. Several proofs of how the model can be reformulated to portray a number of different BSPs were presented. A GA based heuristic solution approach that can be applied to each of the reformulations was also presented. The proposed heuristic was tested for robustness and computational efficiency using two existing BSPs with promising results. In the next three chapters we will present three additional heuristics for the BAP.

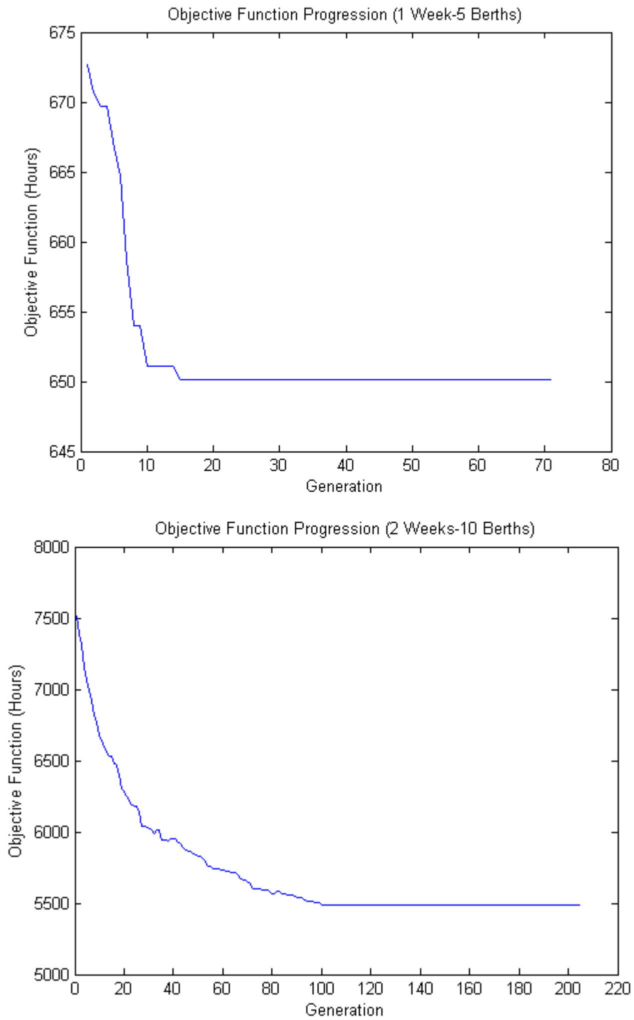


Figure 5-7 MTST BSP Progression of Objective Function Value (Smallest and Largest Problem Instance)

6 AN OPTIMIZATION BASED GENETIC ALGORITHM HEURISTIC FOR THE BERTH ALLOCATION PROBLEM

Introduction

Genetic algorithm based heuristics are very popular as a BAP solution approach. In this chapter we develop an optimization based genetic algorithm (GA) heuristic to solve medium to large instances of the BAP. The heuristic can be applied to the discrete and dynamic BAP, and is independent of the objective function. We also present a heuristic for the single BAP that we use in the GA heuristic. For the purposes of this chapter we study the discrete and dynamic berth allocation problem with the objective to minimize the total weighted service time (WBAP) and deal with calling vessels with various service priorities. We use the linear MIP model formulation presented in the previous chapter and conduct numerical experiments to evaluate the efficiency and effectiveness of the proposed method.

The remainder of this chapter is organized as follows: Section 2 presents the mathematical formulation. Section 3 describes a solution procedure. Section 4 reports the numerical results. Section 5 concludes the chapter.

Problem Formulation

The berth scheduling policy modeled in this chapter was originally proposed by Imai et al. (2003). In the same manner the BAP presented in this chapter assumes only one long wharf at a multi-user terminal. The wharf is divided into several berths and we obtain a set of assignments of vessels to those berths. We also assume that each berth can service one vessel at a time and that there are no physical and/or technical restrictions such as the relationship between vessel draft and effective quay water depth. Furthermore, as with most papers presented in the literature, the vessel handling time is assumed dependent on the berth where it is assigned, since it is related to the time of the landside transfer operations.

In formulating the WBAP we define the following variables: i=(1,……,I) $\in$ B set of berths, j=(1,….,T) $\in$ V set of vessels, k=(1,….,T) $\in$ O set of service orders, S_i=time when i berth becomes idle, A_j=arrival time, C_{ij}= handling time of vessel j at berth i, X_{ijk}=1 if vessel j is serviced at berth i with $(k\text{-}1)$ successors, y_{ijk}= idle time of berth i before vessel j is serviced as the k^{th} vessel. The problem was initially formulated by Imai et al. (2003) is shown in equations 6-1 through 6-4.

(WBAP): $$\sum_i \sum_j \sum_k \{C_{ij} + S_i - A_j + \sum_{m \neq j} \sum_{h<k} C_{im} X_{imh}\} a_j X_{ijk} + \sum_i \sum_j \sum_k (y_{ijk} + \sum_{m \neq j} \sum_{h<k} y_{imh}) a_j \,, \quad (6\text{-}1)$$

Subject to: $$\sum_{i \in B} \sum_{k \in U} X_{ijk} = 1, \forall j \in V \,, \quad (6\text{-}2)$$

$$\sum_{j \in V} X_{ijk} \leq 1, \forall i \in B, k \in O \,, \quad (6\text{-}3)$$

$$\sum_{m \neq j \in T} \sum_{h<k \in O} (C_{im} X_{imh} + y_{imh}) + y_{ijk} - (A_j - S_i) X_{ijk} \geq 0, \forall i \in B, j, \in T, k \in O \,, \quad (6\text{-}4)$$

$X_{ijk} \in \{0,1\}$, Integer, $y_{ijk} \geq 0$ Positive (decision variables), where a_j is a weight for vessel j.

The objective function seeks to minimize the weighted service time. Constraints (6-2) ensure that vessels must be serviced once; constraints (6-3) that each berth services one vessel at a time; and constraints (6-4) that each vessel is serviced after its arrival. For further explanations of the objective function and the constraints the reader is referred to Imai et al. (2003). The resulting formulation is non-linear (MINLP). MINLP problems are precisely so difficult to solve, because they combine all the difficulties of both of their subclasses: the combinatorial nature of mixed integer programs (MIP) and the difficulty in solving nonconvex (and even convex) nonlinear programs (NLP). Because subclasses MIP and NLP are among the class of theoretically difficult problems (NP-complete), so it is not surprising that solving MINLP can be very challenging. Imai et al. (2003) used a Lagrangian relaxation of the problem in order to look into the availability of the subgradient optimization. Although the subgradient method was adaptable to this problem, enormous computational effort was expected because the relaxed problem was a quadratic assignment problem which was NP-hard. Therefore, they eventually employed a GA based heuristic algorithm, an approach widely utilized for complicated combinatorial problems.

In this chapter the problem is reformulated as a linear problem. The new formulation is shown in equations 6-5 through 6-9.

(LWBAP): $\min \sum_i \sum_j \sum_k -a_j DT_{ijk}$, (6-5)

Subject to: $\sum_i \sum_j \sum_k X_{ijk} = 1, \forall j$, (6-6)

$\sum_{j \in V} X_{ijk} \leq 1, \forall i \in B, k \in O$, (6-7)

$\sum_{m \neq j \in T} \sum_{h < k \in O} (C_{im} X_{imh} + y_{imh}) + y_{ijk} - (A_j - S_i) X_{ijk} \geq 0, \forall i \in B, j, \in T, k \in O$, (6-8)

$DT_{ijk} \leq M(1 - X_{ijk}) - (C_{ij} + S_i - A_j) X_{ijk,} - y_{ijk} - \sum_{j \neq m \in T} \sum_{h < k \in O} (C_{im} X_{imh}) - \sum_{j \neq m \in T} \sum_{h < k \in O} y_{imh}$, (6-9)

$\forall i \in B, j \in T, k \in O$

$X_{ijk} \in \{0,1\}$, Integer, $y_{ijk} \geq 0$, $DT_{ijk} \leq 0$, where DT_{ijk} is an auxiliary variable.

A proof that this model is the linear version of Imai et al. (2003) was provided in Chapter 5.

Solution Procedure

A heuristic was developed for the LWBAP, since it is not likely that an efficient exact solution procedure exists, leading an optimal solution in polynomially bounded computation time. The procedure we employ for the heuristic is the Genetic Algorithms (GAs). Unlike the previous GA heuristic presented in chapter this heuristic incorporates an optimization component in the heuristics procedure that aims to improve the performance of the heuristic in terms of the final value of the objective function. Further detail are provided in the following subsections of this chapter.

Representation

The same representation as with the heuristic in Chapter 5 is applicable and was implemented.

Genetic Operations: Crossover & Mutation

The genetic operations described in Chapter 5 are applicable and were implemented.

Optimization Component

After the completion of the genetic operations (crossover and mutation) a typical genetic algorithm procedure would continue with the selection of the next generation. Instead of moving directly to that step we embed an optimization element applied at each iteration of the genetic algorithm and immediately after the genetic operations are completed but before the next generation selection. The optimization procedure is applied as follows:

Heuristic 6-1: GA Optimization Procedure

STEP 1: Select randomly a number of individuals F from the total individuals A ($|F| \leq |A|$).

STEP 2: Create an empty set B to hold the new individuals.

STEP 3: Select the next individual from F and remove it from F.

STEP 4: At each berth of this individual reassign the vessels using a branch and bound algorithm with the objective of minimizing the total weighted service time.

STEP 5: If F is empty end else go to Step 3.

Single Berth Optimization

The implementation of the optimization component becomes time consuming if the number of customers (at each berth) exceeds 5. In order to improve the computational time performance we propose the following heuristic for the single berth assignment. In the appendix we prove that this heuristic will give the optimal value if all weight are equal to 1.

Heuristic 6-2: Rolling Time Window Heurist for the Single Berth BAP

STEP 0: Sort vessels in ascending order of arrival time $S=\{S_1, S_2, \ldots\ldots, S_n, S_{n+1}, S_j\}$, where $A_n<A_{n+1}$

STEP 1: Select the first n vessels $N=\{S_1, S_2, \ldots\ldots, S_n\}$, with arrival times smaller than the service time of the largest vessel if it was assigned first. If the number of vessels selected are less than 6 then continue to add vessels in order of arrival until $|N|=6$.

STEP 2: Solve LWBAP using *N* vessels

STEP 3: Check how many vessels from *N* have finished service before the arrival of vessel S_{n+1}. Name this set *ND*.

STEP 4: If *ND*=empty then include in *N* all vessels that arrive before the finish of the earliest "job" from *N* and go to step 2 else remove from *N* vessels in *ND* and add S_{n+1} to *N*

STEP 5: Go to step 2 until *N*=*ND*= empty set

Fitness/Selection

The fitness/selection criterion applied in this chapter was presented in Chapter 5.

Computational Experiments

Dataset Description

The same dataset as with Chapter 5 was used for the computational experiments of this chapter.

Experimental Results

The solution procedure was coded in SciLab 4.1[13] on a Toshiba Dual Core Intel T2250 with 2GB of RAM. For the OBGA the population size was set to 20, the number of generations to 40 and the size of the individuals to be optimized within each generation equal to 5. In order to evaluate the effectiveness of the optimization component of the heuristic we performed the same

[13] Copyright © 1989-2005. INRIA ENPC <www.scilab.org>

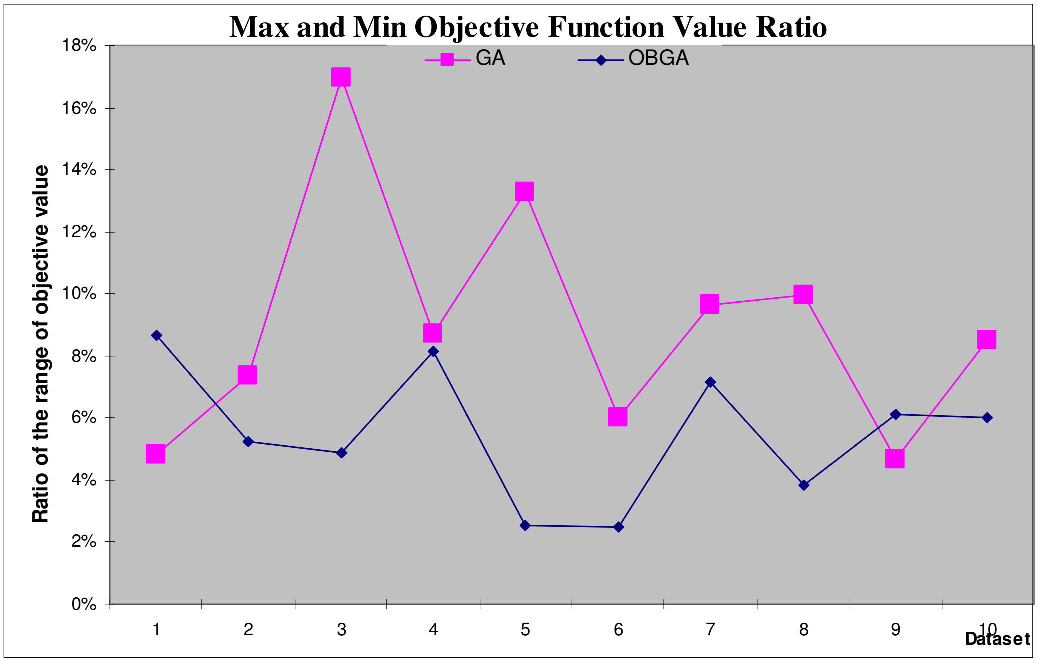
experiments excluding the optimization step (from now on we will refer to this heuristic as GAH). The population size was set to 20 and the generations to 2000.

For each dataset, the ratio of the range of objective values for 5 trials (different starting populations) to the lowest objective value, which can be expressed by the highest objective value during the five trials divided by the lowest objective value during the five trials, was calculated. Results are reported in figure 6-1 for the twenty different datasets. It is obvious that the GAH algorithm is producing results with a higher variation (less consistent) then the OBGA heuristic. For both heuristics the average ratio was less than 15%, and thus we can conclude that the objective function values obtained are consistent for different trials.

Figure 6-2 shows the actual values (minimum and maximum value obtained from the 5 trials) of the objective function for each dataset from both heuristics (OBGA and GAH) while figure 6-3 the computational time. We can conclude that the former heuristic (OBGA) constantly outperforms the later (GAH) in terms of the minimum and maximum values, especially as the size of the problem increases, while the computational time increase is negligible.

Finally we performed a sensitivity analysis for the parameters of the OBGA and GA heuristic (population, generations, chromosomes to be optimized at each generation). We performed different experiments using the datasets described but varying the population size and generations and population size to be optimized at each generation for the OBGA. Both heuristics were proven to be robust exhibiting small changes in the variance and the minimum value of the objective function.

One Week Period Max and Min Objective Function Values Ratios



Two Weeks Period Max and Min Objective Function Values Ratios

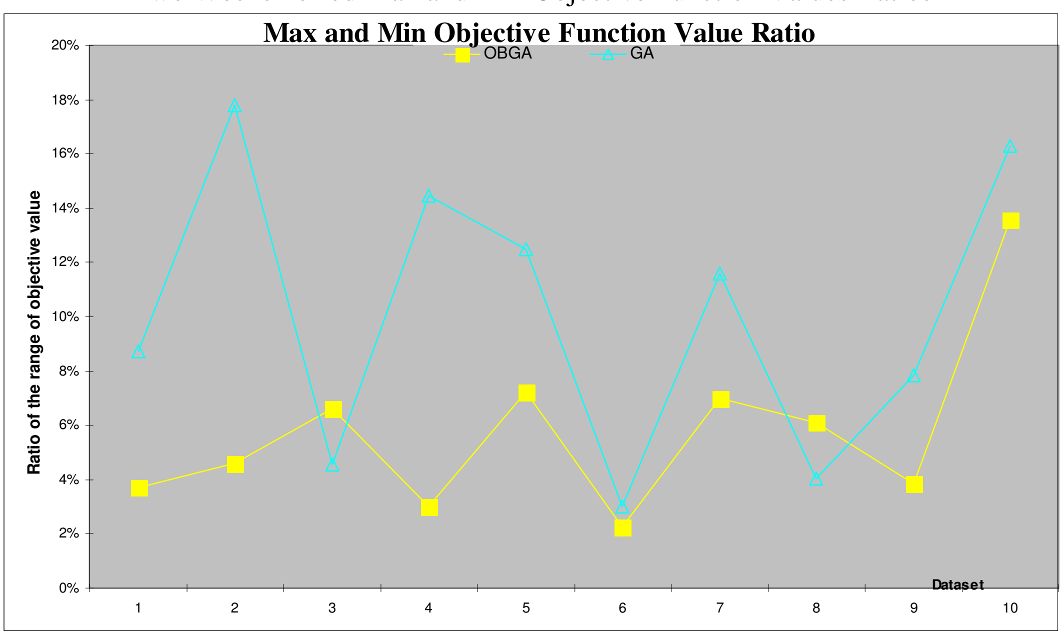
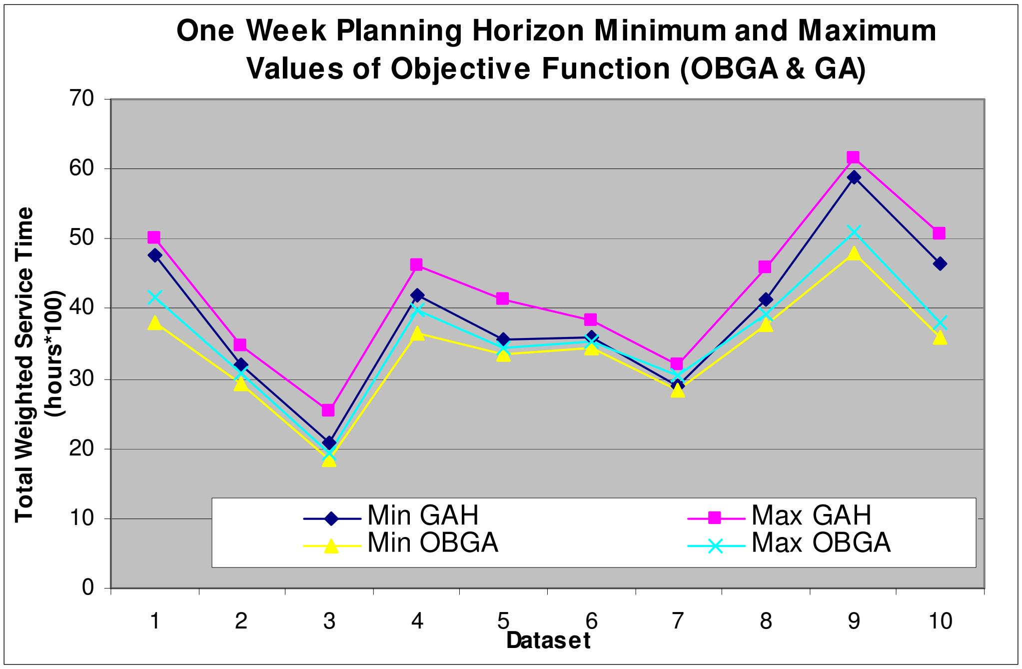


Figure 6-1 Average ratio of the range of objective values to the lowest objective value for different problem sizes (OBGA)

One Week Period Min-Max Values of Objective Function



Two Weeks Period Min-Max Values of Objective Function

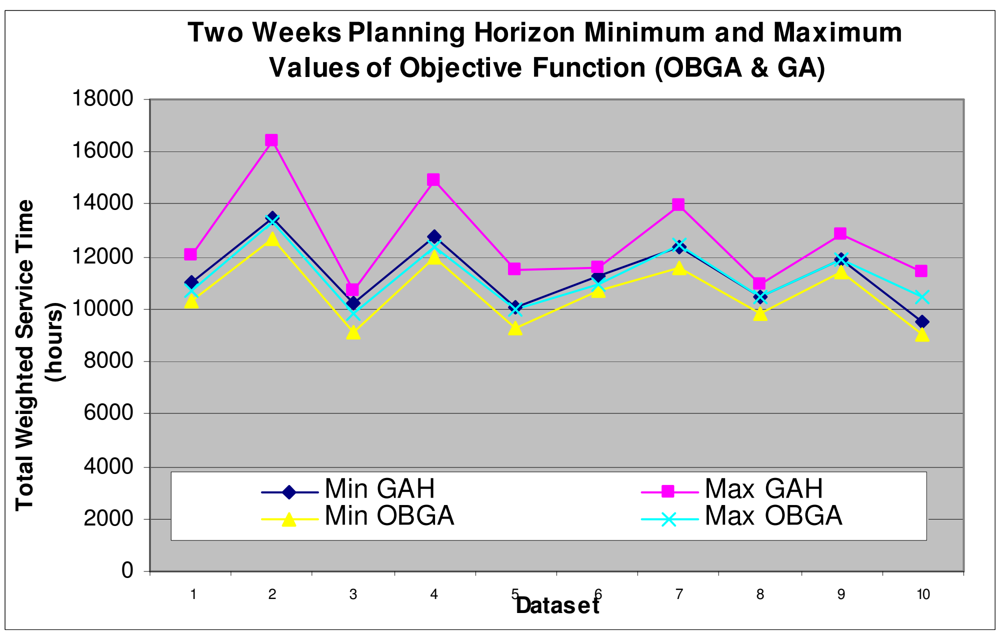
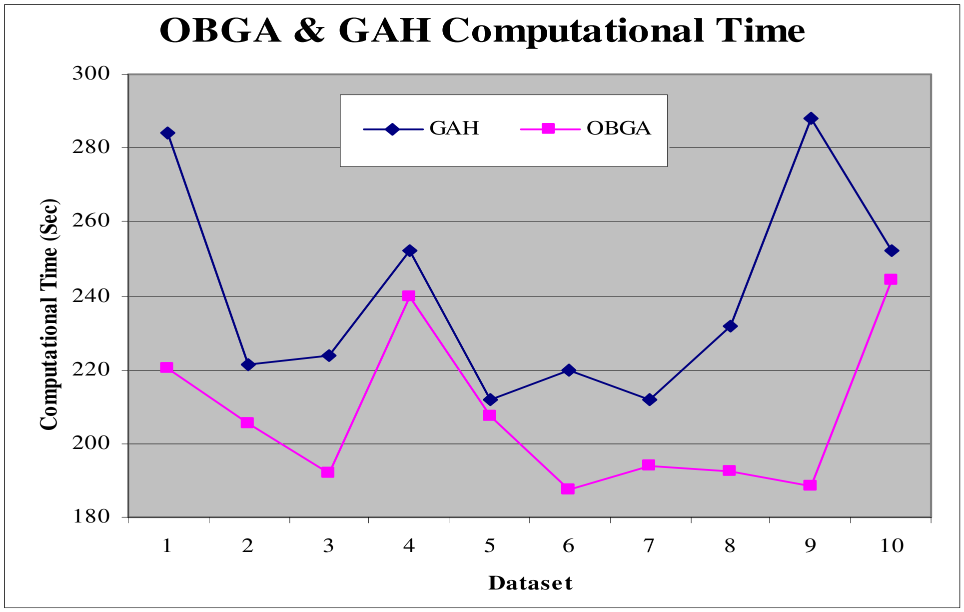


Figure 6-2 Minimum and Maximum Values of the Objective Function (GA and OBGA)

One Week Period



Two Weeks Period

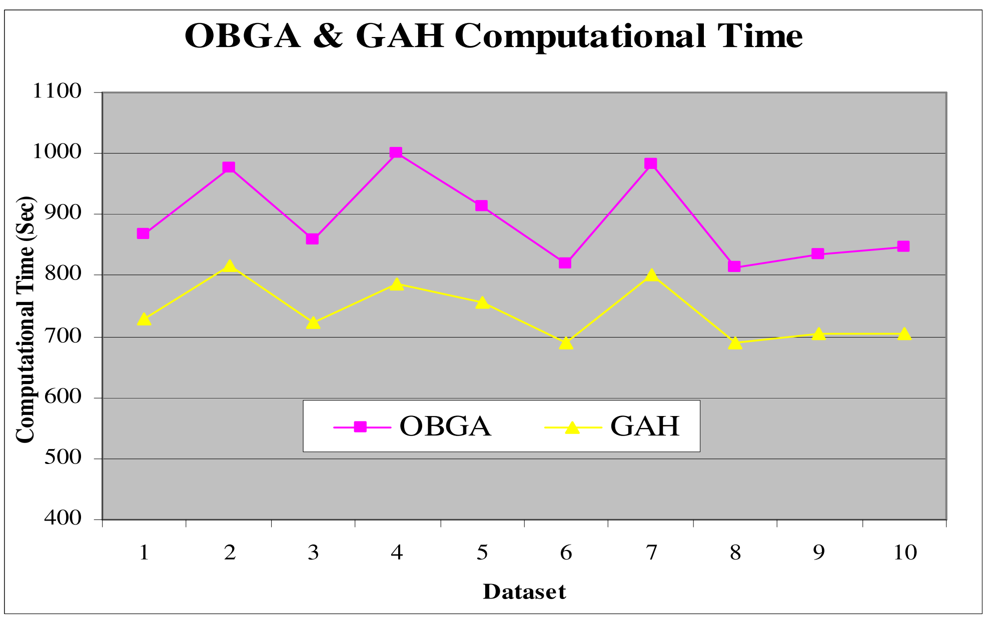


Figure 6-3 Computational Time (GA and OBGA)

Conclusions

In this chapter we presented an optimization based Genetic Algorithm heuristic for the Berth Allocation Problem (BAP). The proposed approach was evaluated by considering the problem of allocation space at a berth for vessels with the objective of minimizing the total weighted service time of all the vessels. The problem was formulated as a linear mixed integer program.

The proposed heuristic was evaluated against a GA based heuristic that lacked the optimization component. In order to decrease the computational time of the former heuristic, two additional heuristics were proposed for the single BAP. Computational experiments showed that the proposed algorithm outperformed the GA heuristic lacking the optimization component in terms of the variance and minimum values of the objective function, especially as the problem size increased. On the other hand, the increase in computational complexity due to the optimization component was negligible, and ranged between 8 to 100 sec for the one week planning horizon and 124 to 180 sec for the two week planning horizon.

7 A TWO OPT BASED HEURISTIC FOR THE DISCRETE AND DYNAMIC BERTH SCHEDULING WITH TIME WINDOWS

Introduction

In this chapter we deal with the discrete and dynamic berth allocation with time windows. Berth allocation aims to optimally schedule and assign vessels to berthing areas along a quay at a container terminal**.** The vessels arrive at the port over a period of time and normally request service and departure within a specified time window. These time windows are determined through contractual agreements between the port operator and the carrier. Based on these contractual agreements different vessels receive different service priorities varying from berthing upon arrival, to guaranteed service time window and/or guaranteed service productivity. In this chapter the discrete and dynamic BAP (DDBAP) is formulated as a linear MIP problem with the objective to simultaneously minimize the port operators' costs from vessel late departures (departure past the time window) and maximize the port operators' premiums from vessel early and timely departures (departure before and within the requested time window).

Although the discrete and dynamic berth allocation problem has been studied extensively (Imai et al. 2001, 2003, 2007a, 2007b; Imai et al. 2003, Nishimura et al., 2001; Cordeau et al., 2005; Hansen et al., 2007) all the formulations presented so far in the literature reduced the time window to a point in time while leading to NP-hard or NP-complete problems that required some sort of (meta)heuristic algorithm to be applied for a computationally acceptable solution time. All of these (meta)heuristic did not guarantee convergence of the algorithm to a local or global optimal.

In this chapter we present a 2-opt heuristic for the discrete and dynamic berth allocation that guarantees local optimality for the final solution. Our work extends and integrates the work of several previous authors and the work presented in the previous chapters, but results in a new heuristic that guarantees local optimality for the discrete and dynamic berth allocation problem. We illustrate the behavior and efficiency of the proposed heuristic using the minimum cost with time windows BSP using real world size instances. The next section provides a formal description of the problem. The third section presents the heuristic and in the fourth section the heuristic is evaluated. The last section concludes the chapter.

Problem Description

Assume that a set of vessels are set to arrive at a port over a period of time and serviced at a number of berths. We assume that each berth can handle one vessel at a time regardless of the vessel's size and that there are no physical/technical restrictions. The vessel's handling time is assumed to be dependent only on the berth where it will be assigned and on the number of containers to be loaded/unloaded. To formulate the DDBAP we define the following: $i = (1,\ldots\ldots,I) \in B$ set of berths, $j = (1,\ldots.,T) \in V$ set of vessels, $k = (1,\ldots.,T) \in O$ set of service orders, S_i = Time when berth i becomes available for the first time in the current planning horizon, A_j =Arrival time of vessel j, C_{ij} = Handling time of vessel j at berth i, y_{ijk} = idle time of berth i before vessel j is serviced as the k^{th} vessel, X_{ijk1} = 1 if vessel j serviced at berth i as the k^{th} vessel and departs or berths before the requested time window and zero otherwise, X_{ijk2}= 1 if vessel j serviced at berth i as the k^{th} vessel and departs or berths after the requested time window and zero otherwise, X_{ijk3}= 1 if if vessel j serviced at berth i as the k^{th} vessel and departs or berths within the requested time window and zero otherwise, a_{j1}= Hourly earliness departure premium for vessel j, a_{j2}= Hourly earliness berthing premium for vessel j, b_{j1} = Hourly lateness departure penalty for vessel *j*, b_{j2} = Hourly lateness berthing penalty for vessel *j*, γ_{j1} = Hourly timely departure premium for vessel *j*, γ_{j2} = Hourly timely berthing premium for vessel *j*, t_{j1} = Requested early departure/berthing time of vessel *j*, t_{j2} = Requested late departure/berthing time of vessel *j*, *WST*= 1 if vessel *j* sets a request for early, timely and late departure 0 otherwise, DT_{ijk} = Difference of early/late actual and requested finish/start time of vessel *j* serviced at berth *i* as the k^{th} vessel, $DT^{+}_{ijk} = \max(0, DT_{ijk})$, $DT^{-}_{ijk} = \min(0, DT_{ijk})$, DTT_{ijk}=Difference of early timely requested and actual timely finish/start time of vessel *j* serviced at berth *i* as the k^{th} vessel, R_{ij}=0 if vessel *j* cannot be serviced at berth *i* due to physical or technical restrictions and 1 otherwise

The DDBAP can be formulated as follows:

$$\min - \sum_{i}\sum_{j}\sum_{k}\left\langle \begin{array}{l} \{a_{j1}WST_j + a_{j2}(1-WST_j)\}DT^{+}_{ijk} \\ +\{b_{j1}WST_j + b_{j2}(1-WST_j)\}DT^{-}_{ijk} \\ +\{\gamma_{j1}WST_j + \gamma_{j2}(1-WST_j)\}DTT_{ijk} \end{array} \right\rangle, \qquad \text{(Eq. 7-1)}$$

Subject to: $\sum_{i}\sum_{j}\sum_{k}\sum_{r=1}^{3} X_{ijkr} = 1, \forall j$, (Eq. 7-2)

$$\sum_{j\in V}\sum_{r=1}^{3} X_{ijkr} \le 1, \forall i \in B, k \in O \qquad \text{(Eq. 7-3)}$$

$$\sum_{m\neq j\in T}\sum_{h<k\in O}(C_{im}\sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk} - (A_j - S_i)\sum_{r=1}^{3} X_{ijkr} \ge 0, \forall i \in B, j \in T, k \in O \qquad \text{(Eq. 7-4)}$$

$$y_{ijk} \le M\sum_{r=1}^{3} X_{ijkr}, \forall i \in B, j \in T, k \in O \qquad \text{(Eq. 7-5)}$$

$$DT^{+}_{ijk} \le (t_{j1} - WST_j C_{ij} - S_i)X_{ijk1} - y_{ijk} - \sum_{j\neq m\in T}\sum_{h<k\in O}(C_{im}\sum_{r=1}^{3} X_{imhr} + y_{imh}) + alpha_{ijk}, \qquad \text{(Eq. 7-6)}$$

$\forall i \in B, j \in T, k \in O$

$$t_{j1} X_{ijk1} \geq \sum_{j \neq m \in T} \sum_{h < k \in O} (C_{im} \sum_{r=1}^{3} X_{ijkr} + y_{imh}) + y_{ijk} + (WST_j C_{ij} + S_i) X_{ijk1} - M(1 - X_{ijk1}), \quad \text{(Eq. 7-7)}$$
$\forall i \in B, j \in T, k \in O$

$$DT_{ijk}^{-} \leq (t_{j2} - WST_j * C_{ij} - S_i) X_{ijk2} - y_{ijk} - \sum_{j \neq m \in T} \sum_{h < k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + M(1 - X_{ijk2}), \quad \text{(Eq. 7-8)}$$
$\forall i \in B, j \in T, k \in O$

$$t_{j2} X_{ijk2} \leq \sum_{j \neq m \in T} \sum_{h < k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk} + (WST_j C_{ij} + S_i) X_{ijk2}, \quad \text{(Eq. 7-9)}$$
$\forall i \in B, j \in T, k \in O$

$$alpha_{ijk} \leq M(1 - X_{ijk1}), \forall i \in B, j \in T, k \in O \quad \text{(Eq. 7-10)}$$

$$alpha_{ijk} \leq \sum_{j \neq m \in T} \sum_{h < k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk}, \forall i \in B, j \in T, k \in O, \quad \text{(Eq. 7-11)}$$

$$DT_{ijk}^{+} \leq MX_{ijk1}, \forall i \in B, j \in T, k \in O, \quad \text{(Eq. 7-12)}$$

$$t_{j1} X_{ijk3} \leq \sum_{j \neq m \in T} \sum_{h < k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk} + (WST_j C_{ij} + S_i) X_{ijk3} + M(1 - X_{ijk3}), \quad \text{(Eq. 7-13)}$$
$\forall i \in B, j \in T, k \in O$

$$t_{j2} X_{ijk3} \geq \sum_{j \neq m \in T} \sum_{h < k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) + y_{ijk} + (WST_j C_{ij} + S_i) X_{ijk3} - M(1 - X_{ijk3}), \quad \text{(Eq. 7-14)}$$
$\forall i \in B, j \in T, k \in O$

$$DTT_{ijk} \leq t_{j1} X_{ijk3} - \sum_{j \neq m \in T} \sum_{h < k \in O} (C_{im} \sum_{r=1}^{3} X_{imhr} + y_{imh}) - y_{ijk} - (WST_j C_{ij} + S_i) X_{ijk3} + M(1 - X_{ijk3}), \quad \text{(Eq. 7-15)}$$
$\forall i \in B, j \in T, k \in O$

$$\sum_{k} \sum_{r} X_{ijkr} \leq MR_{ij}, \forall i \in B, j \in T, \quad \text{(Eq. 7-16)}$$

$X_{ijk1}, X_{ijk2}, X_{ijk3} \in \{0,1\}$, $y_{ijk} \geq 0$, $DT_{ijk}^{-} \leq 0, DT_{ijk}^{+} \geq 0, DTT_{ijk} \leq 0$, $alpha_{ijk} \geq 0$, M is a large positive number

In the objective function (7-1) the first three terms correspond to the total cost from delayed departures/berthing, the second to the total premiums from early departures/berthing, and the third to the total premium from timely departures/berthing (depending on if WST_j=1 or 0).

Constraints (7-2) ensure that vessels must be serviced once; constraints (7-3) ensure that each berth services one vessel at a time; and constraints (7-4) and (7-5) ensure that each vessel is serviced after its arrival. Constraints (7-6) and (7-15) enforce the declaration of the decision and auxiliary variables. Equation (7-16) enforces different physical and technical constrains that do not allow the berthing of certain vessels at certain berths (i.e. berth depth).

Solution Approach

In this section we describe how we solve the DDBAP to local optimality. In order to solve the problem we devised two heuristics based on the lamda-optimal heuristic by Lin and Kernighan (1973). Both proposed heuristics are able to find a local minima of the problem and reduce dramatically the number of computations required. Given the nature of the problem it is always almost impossible to obtain an optimal solution using classical optimization approaches. The complexity of the problem quickly increases as the number of vessels and berths increase. On the other hand the same problem with 2 berths and a reasonable number of customers for any given time period may be solved to optimality by using a branch and bound algorithm or even by enumeration of all the feasible solutions.

2-opt based heuristics

A solution is said to be lamda-optimal if it is impossible to obtain a better solution by replacing any lamda relation instances by any other set of lamda relation instances. The lamda-optimal heuristic is based on the concept that in each trial, lamda instances of the chosen relation (mapping between problem components, e.g. vessels to berthing timeslots) in the working solution are exchanged. The trial process continues until a move that satisfies a specified acceptance criteria is found. The accepted move is then used to update the working solution.

Computational time rapidly increases for increasing values of lamda. As a result, the values lamda = 2 and lamda = 3 are the most commonly used. In many applications lamda= 2 is powerful enough to yield near optimal solutions in a fraction of the time needed for an exhaustive search.

In this subsection we first present a heuristic (Heuristic A) that starts with the parameter lamda=2. This heuristic guarantees a local optimal solution but is myopic. In order to account for this problem we extend Heuristic A and present a second heuristic (Heuristic B) where Heuristic A is used internally *n* times (where *n* are all the possible combination of the berths if at each combination we exchange the positions of two berths at a time, i.e. [Berth 1, Berth 2, Berth 3] and [Berth 2, Berth 1, Berth 3] are two different combinations while [Berth 1, Berth 2, Berth 3] and [Berth 3, Berth 1, Berth 2] are the same combination).

Heuristic 7-1: Small Neighborhood Search

STEP 0: Obtain an initial feasible solution using a Genetic Algorithm (GA) heuristic (Chapter 6). Assume the objective function value from the GA heuristic is OFV^0. Set counter n=0

STEP 1: Set counter n=n+1 and counter c=0

STEP 2: Set counter c=c+1

STEP 3: If c<|B| reassign vessels to berths c and c+1 using CPLEX or enumeration and go to Step 2, Else if c=|B| reassign vessels to berths c and 1 using CPLEX or enumeration, and go to Step 4

STEP 4: If OFV^n-OFV^{n-1} NE 0 go to Step 1else end

Heuristic 7-2: Large Neighborhood Search

STEP 0: Obtain an initial feasible solution using the Genetic Algorithm (GA) heuristic (Chapter 6). Set counter n=0

STEP 1: While n<|B|-1

STEP 2: Set counter n=n+1 and counter k=n

STEP 3: Set counter k=k+1

STEP 4: Set $B_k=B_{k+1}$, $B_{k+1}=B_k$, $C_{kj}=C_{(k+1),j}$, $C_{(k+1),j}=C_{k,j}$, $S_k=S_{k+1}$, $S_{k+1}=S_k$, and go to step 5

STEP 5: Apply Heuristic A excluding Step 0

STEP 6: If k<|B|-1 go to step 3 else go to 7

STEP 7: If n<|B|-1 go to Step 2 else end

Computational Experiments

The solution procedure was coded in SciLab 4.1[14] on a Toshiba Dual Core Intel T2250 with 2GB of RAM. Problems used in the experiments were generated randomly but systematically. The focus was in creating datasets that would be computationally challenging and reflect real life conditions. We developed forty base problem datasets where vessels are served with various handling volumes at a multi-user container terminal (MUT) with five and ten berths, with a planning horizon of one and two weeks. Vessel handling volumes ranged from 250 to 4,000 (TEU) based on a uniform distribution pattern (Table 7-1). The handling time of a vessel was dependent on the berth assigned, and was a function of the number of the cranes that may be assigned. We assume that 1 to 3 cranes operate on small sized vessels (250-2000 TEU), 2 to 4 cranes on medium sized vessels (2000-3000 TEU), and 3-6 on large mother vessels (3000-4000 TEU). The average crane productivity was assumed to be 25 TEU/hour. The average vessels per

[14] Copyright © 1989-2005. INRIA ENPC <www.scilab.org>

berth per week equivalence (VBWE) was set to five. Testing instances with lower VBWE averages would not provide a representative evaluation of the heuristics' performance .

The minimum handling time was calculated by dividing the handling volume by the average productivity of a crane multiplied by the number of cranes operates on the vessel. Random numbers were used to generate the handling time of vessels at the other berths, always in relation to the berth with the minimum handling time. The association of the minimum time with the berth was also made randomly. Physical restrictions for vessel berthing was created randomly and was restricted to a maximum of one berth for instances with berth capacity of five and two for instances with berth capacity of ten. Availability of berths was calculated using a uniform probability with a minimum of zero and a maximum of 10 hours. One of the most crucial issues was the selection of the interarrival vessel distribution. We assumed that vessels arrived randomly over a weekly period with a minimum of 40 and a maximum of 80 vessels for instances with berth capacity of five and a minimum of 80 and a maximum of 120 vessels for instances with berth capacity of ten. Figure 7-1 shows the arrival patterns for each one of the forty base problem datasets. The x-axis shows the arrival time intervals (24 hours) and the y-axis the number of vessels arriving at that interval.

Computational time

The computational time of the proposed 2-opt heuristic is mainly dependent on the computational time at Step 3 of Heuristic A. Excluding simple BSPs (minimize total service time) Heuristic A becomes computationally expensive even for small problems (five berths and 50 vessels). A compromising solution would be to use the GA heuristic proposed in chapter 5 as the optimization algorithm of step 3 of heuristic A. The main idea behind the use of the GA is that for small problems similar to the one solved at every iteration at Step 3 of Heuristic A, the GA will most probably provide the optimal solution. We evaluated this assumption for problems with planning horizon of one week and berth capacity of five with the objective to minimize the total service time. The algorithm converged to the same objective value for all the ten datasets and on average took one fifth of the computational time of the CPLEX based heuristic. These results do not guarantee the same behavior for the heuristic for different BSPs but are promising. For the computational examples in this chapter we used the GA heuristic from chapter 5 as the optimization procedure at Step 3 of Heuristic A. We used a population of 25 and 500 generations.

Table 7-1 Dataset Information

	Planning Horizon of One Week				***Planning Horizon of Two Weeks***			
	Five Berths		Ten Berths		Five Berths		Ten Berths	
Dataset	Vessels	Volume (TEU)	Vessels	Volume (TEU)	Vessels	Volume (TEU)	Vessels	Volume (TEU)
1	68	159 330	104	236 186	94	218 387	201	436 671
2	47	94 205	113	258 998	112	250 931	203	438 800
3	75	165 608	105	223 845	109	243 266	220	457 128
4	57	131 432	116	254 256	144	299 030	203	442 327
5	53	109 679	111	242 466	109	223 359	169	337 946
6	58	104 474	104	215 513	145	335 261	213	449 117
7	74	157 930	103	214 724	127	270 058	198	412 600
8	43	91 093	115	232 804	123	260 009	201	438 246
9	49	101 118	103	209 773	108	234 985	235	513 681
10	60	131 961	119	244 713	103	215 550	177	392 751
Average (Planning Horizon)	58	124 683	109	233 328	117	255 084	202	431 927
Average (Annual)	3 037	6 483 516	5 684	12 133 046	6 105	13 264 347	10 504	22 460 188

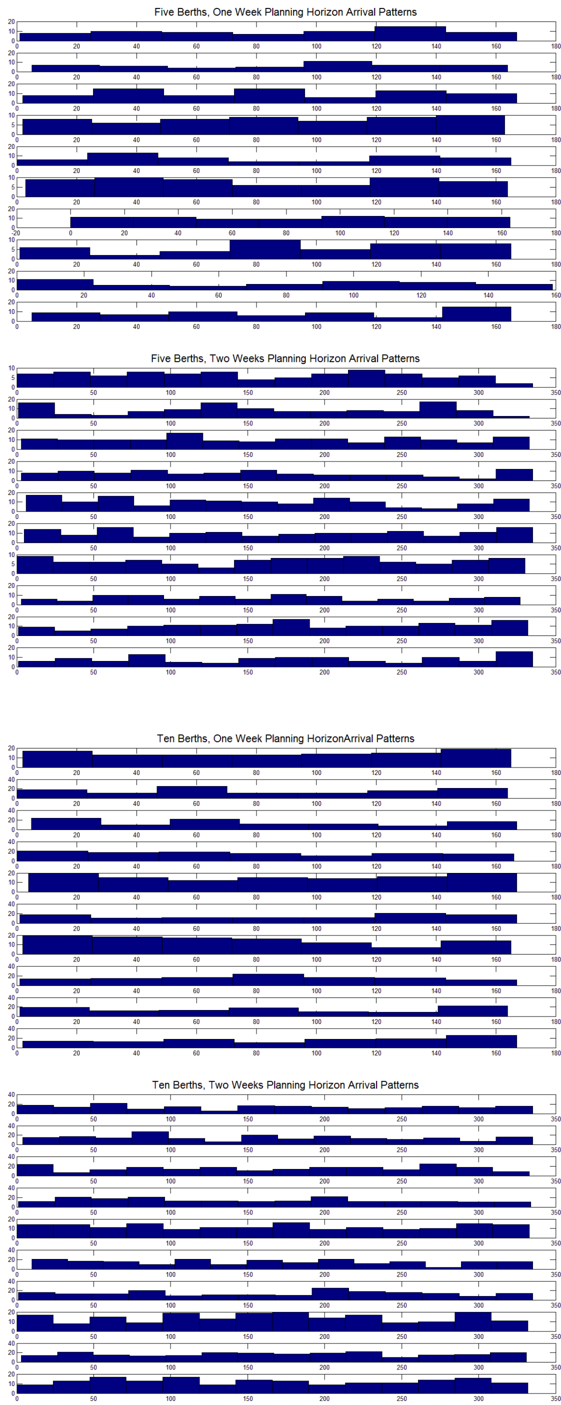


Figure 7-1 Vessel Arrival Patterns

Convergence

Figures 7-2 through 7-5 shows the convergence of the objective value for each berth capacity and planning horizon combination for all the datasets. The algorithm showed a promising rate if convergence in all the 40 cases[15].

Conclusions

In this chapter we presented a 2-opt heuristic for the discrete and dynamic berth allocation that guarantees local optimality. The proposed heuristic managed to improve the solution from the GA based heuristic, proposed in Chapter 5. The only disadvantage of the proposed approach is the increased computational time as the problem size increases as compared to the GA based heuristics from Chapter 5 and 6. A compromising but effective solution was given for this problem by using the GA based heuristic from Chapter 5 as a replacement to the CPLEX or enumeration options pf the solution algorithm within the heuristic proposed in this chapter.

[15] Negative cost means premium for the port operator

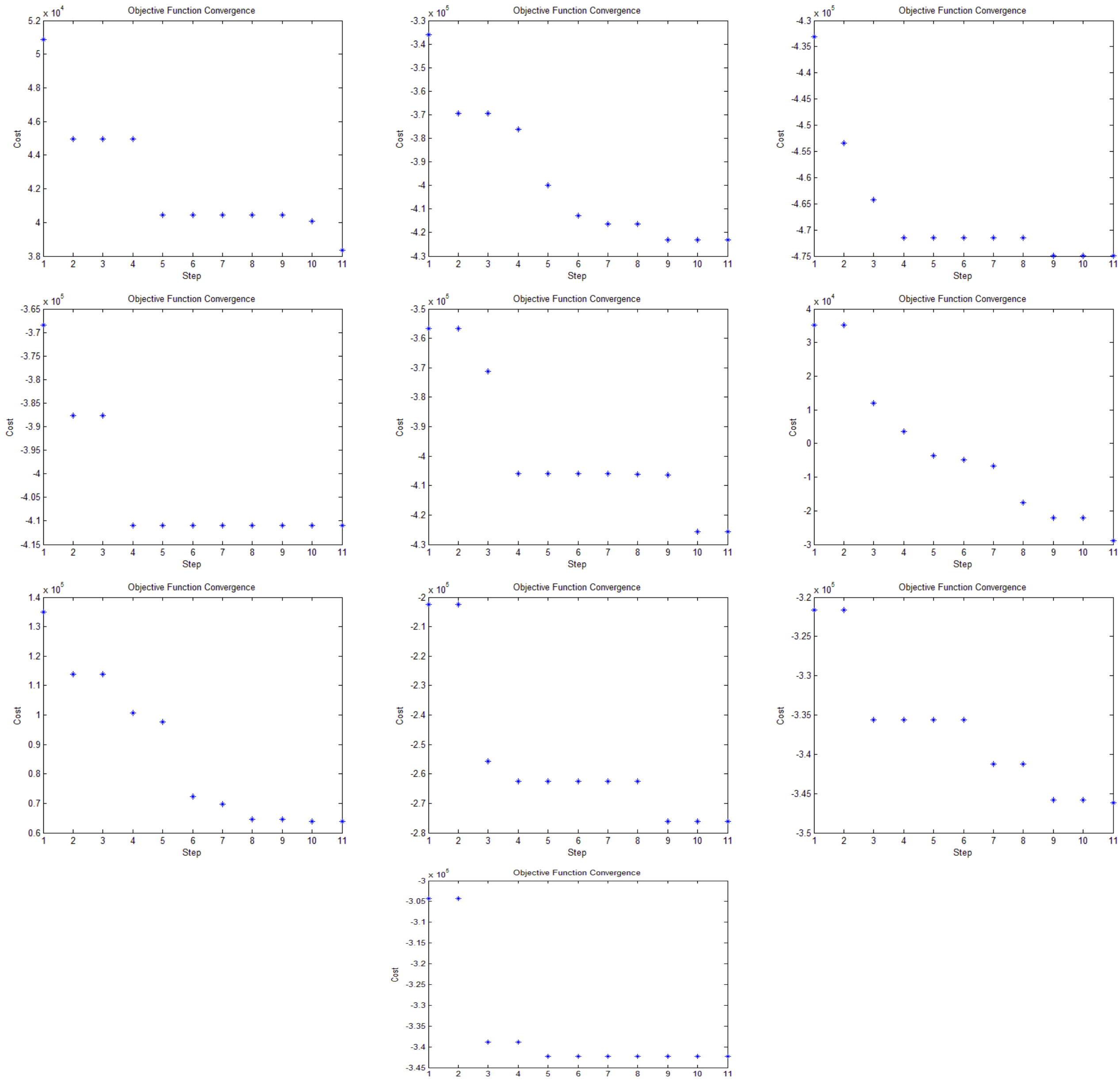


Figure 7-2: Convergence of objective function (Five Berths, One Week Planning Horizon)

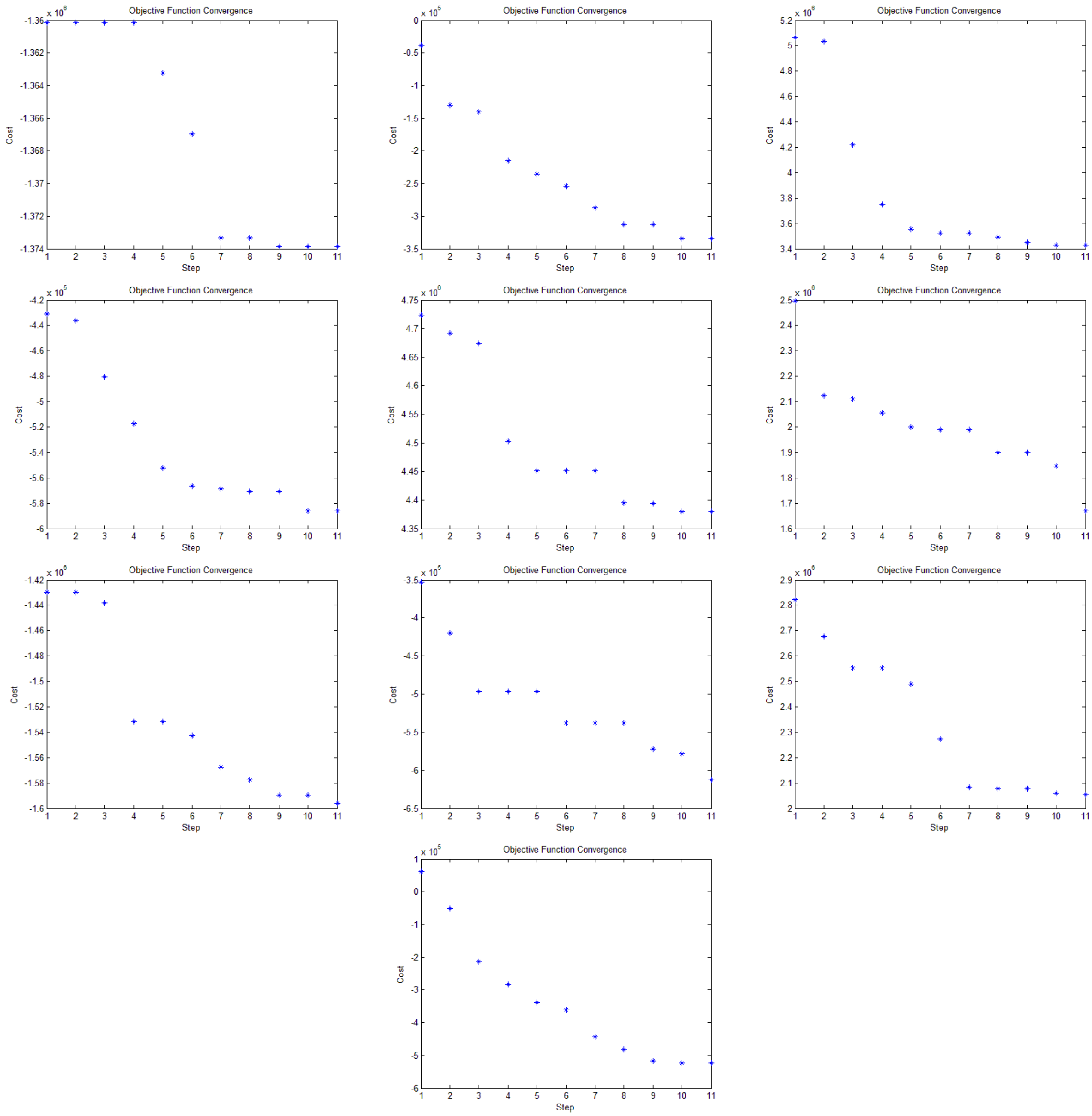


Figure 7-3: Convergence of objective function (Five Berths, Two Weeks Planning Horizon)

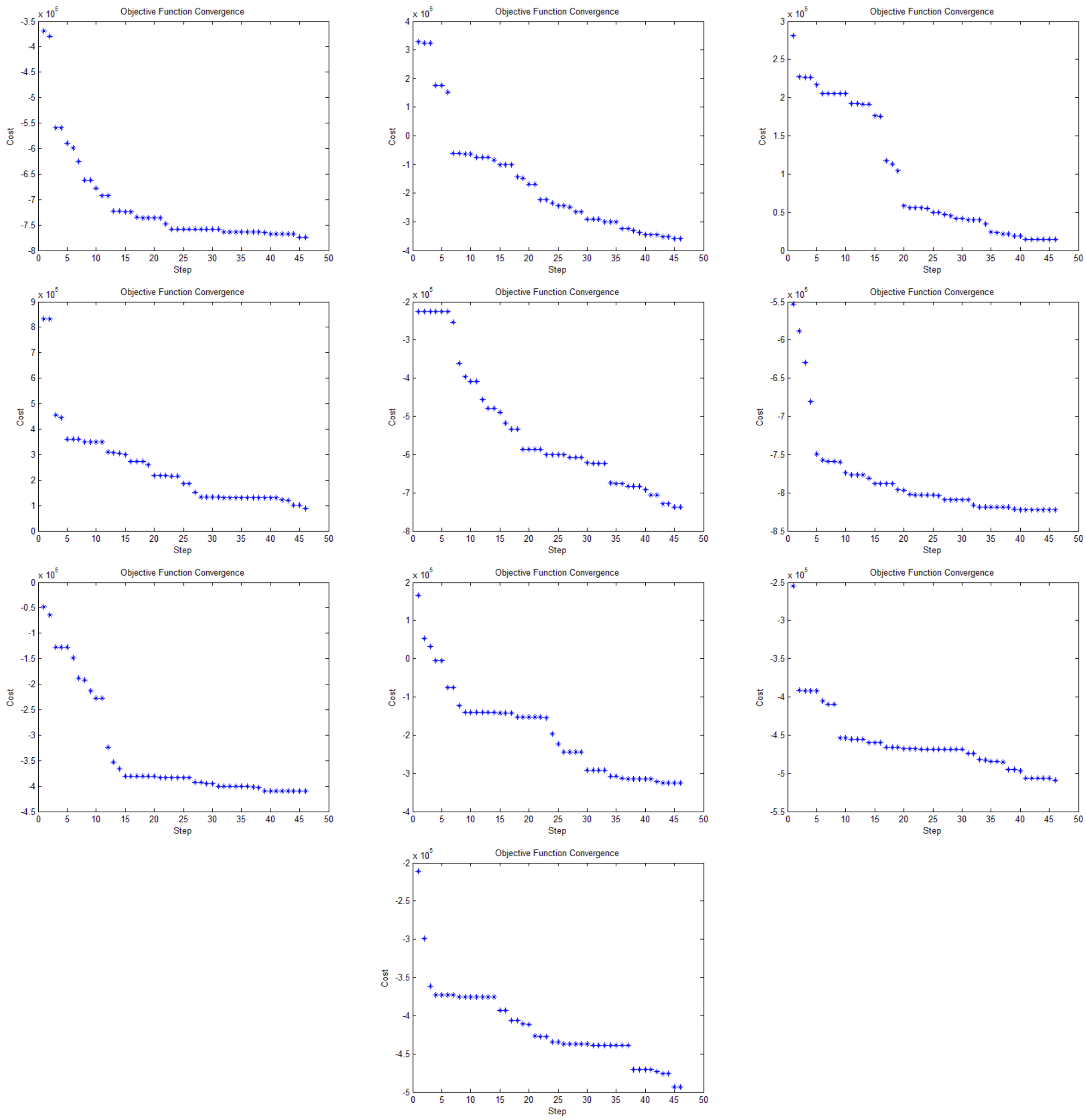


Figure 7-4: Convergence of objective function (Ten Berths, One Week Planning Horizon)

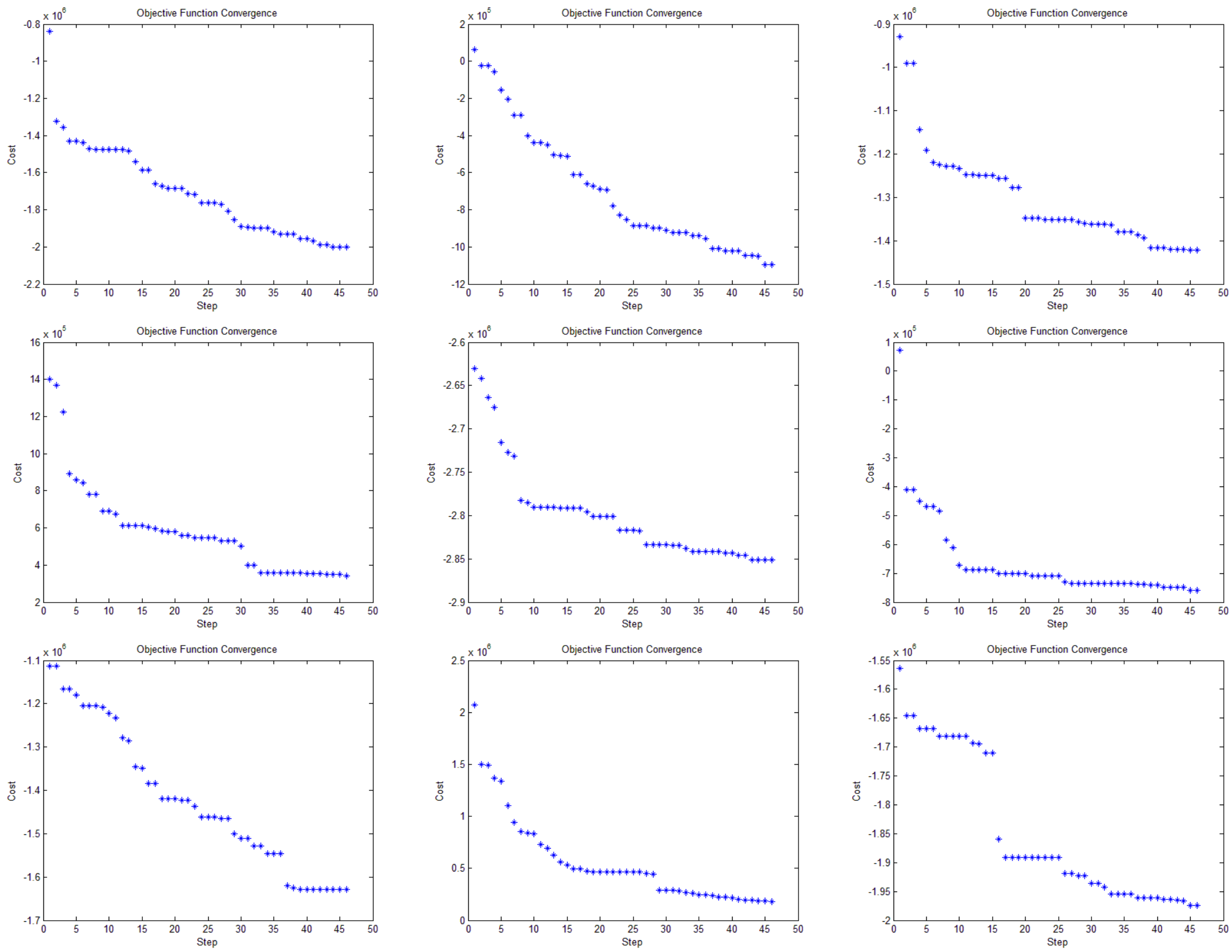


Figure 7-5: Convergence of objective function (Ten Berths, Two Weeks Planning Horizon)

8. A ROLLING TIME HORIZON HEURISTIC FOR THE BERTH ALLOCATION PROBLEM WITH WAIT TIME CONSTRAINTS

Introduction

In this chapter we study the berth allocation problem with waiting time constraints. In this problem vessels set a maximum waiting time limit beyond which heavy penalties occur to the port operator. Unlike Imai et al. (2007) our approach does not consider the alternative of vessels being serviced at a different port if the wait time exceeds a certain time limit, since it is not a good practice to redirect vessels to other ports. The problem formulation seeks to optimality assign vessels to berths so that the total service time for all the customers is minimized and the wait time constraints are not violated. To tackle infeasibility issues arising from this type of constraints in scheduling problems, we developed two time window based heuristics. The next section describes and formulates the problem, while the third section presents the solution approach. We leave the evaluation of the proposed heuristics as future research.

Problem Formulation

In order to formulate the discrete and dynamic berth allocation problem with wait time constraints (BSPWT) we need to define the following:

$i=(1,\ldots\ldots,I) \in B$ set of berths,

$j=(1,\ldots.,T) \in T$ set of vessels,

$k=(1,\ldots.,T) \in O$ set of service orders,

S_i= Time when berth becomes idle for the first time for the current planning horizon,

A_j= Arrival time,

HT_{ij}= Handling time of vessel j at berth i,

y_{ijk} = Idle time of berth i between start of service of vessel j and its immediate predecessor,

X_{ijk} = 1 if vessel j is serviced at berth i as the kth vessel and departs/berths before the requested date and zero,

$R_{ij} = 0$ if vessel j cannot be serviced at berth i due to physical or technical restrictions, and 1 otherwise

WT_j= wait time limit of vessel j.

BSPWT can then be formulated as follows:

$$\text{[BSPWT]: } \min \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} (kHT_{ij} + S_i - A_j)(X_{ijk}) + \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} ky_{ijk} \qquad \text{(Eq. 8-1)}$$

$$\text{Subject to: } \sum_{i \in B} \sum_{k \in U} X_{ijk} = 1, \forall j \in T \qquad \text{(Eq. 8-2)}$$

$$\sum_{j \in T} X_{ijk} \leq 1, \forall i \in B, k \in O \qquad \text{(Eq. 8-3)}$$

$$\sum_{m \neq j \in T} \sum_{h < k \in O} (HT_{im} X_{imh} + y_{imh}) + y_{ijk} - (A_j - S_i) X_{ijk} \geq 0, \forall i \in B, j, \in T, k \in O \qquad \text{(Eq. 8-4)}$$

$$WT_j X_{ijk} \leq \sum_{j \neq m \in T} \sum_{h < k \in O} (HT_{im} X_{imh} + y_{imh}) + y_{ijk} + (S_i - A_j) X_{ijk} + M(1 - X_{ijk}), \\ \forall i \in B, j \in T, k \in O \qquad \text{(Eq. 8-5)}$$

$$X_{ijk}, \in \{0,1\}, \; y_{ijk} \geq 0 \qquad \text{(Eq. 8-6)}$$

The objective function (8-1) minimizes the total service time of all the vessels. Constraints (8-2) ensure that vessels must be serviced once; constraints (8-3) ensure that each berth services one vessel at a time; and constraints (8-4) ensure that each vessel is serviced after its arrival and constraints (8-5) ensure that each vessels' waiting time does not exceed the desired time.

BSPWT has a set of hard constraints (8-5) that may deem the problem infeasible. In order to avoid this problem we define the following auxiliary variable: RL_{ijk}, and relax constraints (8-5). RL will be greater than zero if the vessel j exceeds its wait time serviced as the k^{th} vessel at berth i, and zero otherwise. The relaxed problem is formulated as follows:

$$\text{[BSPWTR]: } \min \sum_{i\in B}\sum_{j\in V}\sum_{k\in U}(kHT_{ij} + S_i - A_j)(X_{ijk}) + \sum_{i\in B}\sum_{j\in V}\sum_{k\in U} ky_{ijk} + RL_{ijk} \quad \text{(Eq. 8-7)}$$

$$\text{Subject to: } \sum_{i\in B}\sum_{k\in U} X_{ijk} = 1, \forall j \in T \quad \text{(Eq. 8-8)}$$

$$\sum_{j\in T} X_{ijk} \le 1, \forall i \in B, k \in O \quad \text{(Eq. 8-9)}$$

$$\sum_{m\ne j\in T}\sum_{h<k\in O}(HT_{im}X_{imh} + y_{imh}) + y_{ijk} - (A_j - S_i)X_{ijk} \ge 0, \forall i \in B, j, \in T, k \in O \quad \text{(Eq. 8-10)}$$

$$\sum_{m\ne j\in T}\sum_{h<k\in O}(HT_{im}X_{imh} + y_{imh}) + y_{ijk} - (A_j - S_i)X_{ijk} \ge 0, \forall i \in B, j, \in T, k \in O \quad \text{(Eq. 8-11)}$$

$$WT_j X_{ijk} \le \sum_{j\ne m\in T}\sum_{h<k\in O}(HT_{im}X_{imh} + y_{imh}) + y_{ijk} + (S_i - A_j)X_{ijk} + M(1 - X_{ijk}) + RL_{ijk}, \quad \forall i \in B, j \in T, k \in O \quad \text{(Eq. 8-12)}$$

$$RL_{ijk} \le MX_{ijk}, \forall i \in B, j \in T, k \in O \quad \text{(Eq. 8-13)}$$

$$X_{ijk}, \in \{0,1\}, \; y_{ijk} \ge 0 \;, \; RL_{ijk} \ge 0 \quad \text{(Eq. 8-14)}$$

In the relaxed problem the objective function (8-1) minimizes the total service time of all the vessels and the total excess of wait time.

Solution Approach

It is not likely that an efficient exact solution procedure exists for BSPWTR, leading to an optimal solution in polynomially bounded computation time. For this reason we developed the following two heuristics for solving the BSPWTR.

Heuristic 8-1: Constant Depth Myopic Heuristic

STEP 0: Sort vessels in ascending order of arrival time $S=\{S_1, S_2, \ldots\ldots, S_n, S_{n+1}, S_j\}$, where $A_n<A_{n+1}$

STEP 1: Select the first n vessels and create a set named $N=\{S_1, S_2, \ldots\ldots, S_n\}$

STEP 2: Solve BSPWTR with N using CPLEX

STEP 3: Create a set named ND and include all the vessels from N that have finished service before the arrival of vessel S_{n+1}, have exceeded their requested wait time, or finish service before the vessels that exceed their wait time

STEP 4: If $ND=\{\}$ then include in N all vessels that arrived before the finish time of the vessel finishing first from N and go to step 2

STEP 5: If $ND \neq \{\}$ reassign vessels in ND using CPLEX and remove these vessels from N

STEP 6: Add vessel S_{n+1} to N

STEP 7: Go to step 2 until $N=\{\}$;

Heuristic 8-2: Variable Depth Time Window Search Heuristic

Step 0: Obtain an initial schedule (S^0) by randomly assigning all vessels in *J*. Name the initial objective function value (OFV^o). Set t= (*Planning Horizon*/7) days. Without loss of generality assume a minimization problem

Step 1: Set n=0, VA=J

Step 2: Set n=n+1

Step 3: Select vessels that finish service within [n*(t-1), n*(t+1)] under S^n and name this set V^n

Step 4: Schedule vessels of set V^n and name this sub-schedule SS^n

Step 5: Select vessels that finish service within [n*(t-1), n*t] under SS^n and name this set VS^n

Step 6: Schedule vessels of set VS^n

Step 7: Remove VS^n from VA and update machine availability

Step 8: If VA=empty go to step 9 else go to step 2

Step 9: If OFV^n<OFV^{n-1} replace S^{n-1} with S^n, set t= (*Planning Horizon*/7) and go to step 11 else go to step 10

Step 10: By a probability of 1/n increase time step by $t/2$ and keep the S^{n-1} or replace S^{n-1} with S^n and set t= *Planning Horizon*/7. Go to step 9

Step 11: If no improvement is observed end else go to step

The proposed heuristics might be inefficient in cases where set |N| becomes too large to be solved in an acceptable time by CPLEX. In this case we may use the heuristic from Chapter 5 or Chapter 6. This compromise is acceptable since for small to medium problems the heuristic from Chapter

5 or Chapter 6 has provided solutions with small deviation from the optimal values in small computational time.

Conclusions

This chapter presented a formulation and a solution approach for the berth allocation problem with wait time constraints. Future research is focusing in implementing the proposed algorithm and testing it on real life instances.

9. BERTH ALLOCATION AND QUAY CRANE SCHEDULING

Introduction

One of the major issues that have not been studied in depth in port operations is the simultaneous assignment of vessels to berths and quay cranes to vessels; two problems that are interrelated (Steenken et al., 2004). Most of the research papers dealing with the berth allocation problem (BAP), as discussed in Chapter 3, considered handling operations of the vessel independent to the number of the quay cranes assigned to that vessel. To our knowledge only two research papers have appeared in the literature that consider the issue of the dynamic BAP and the quay crane scheduling (QCS) together (Park and Kim, 2003; Lee et al, 2006). In light of the above discussion this chapter presents a formulation for the simultaneous berth allocation and quay crane scheduling.

In this chapter we present a general formulation for the dynamic and continuous BAP with simultaneous quay crane assignment and propose a two-dimensional Genetic Algorithm based heuristic for solving the resulting problem. We consider the minimization of costs due to not meeting agreed (un)loading performance, cost for waiting and handling, and tardiness of completion for all vessels. Our research deviates from BSP presented so far in the following aspect: a) unlike Lee et al. (2006) we solve the BAP and QCS simultaneously by determining the actual position of the vessel along the wharf, the start and finish time of the vessels' handling, and the number of quay cranes to be assigned at each vessel, and b) Unlike Park and Kim (2003) we consider the minimization of costs due to not meeting agreed (un)loading performance, cost for handling as well as earliness and tardiness of completion for all vessels.

This chapter is organized as follows. The next section provides the problem formulation while the third section introduces the Genetic Algorithm (GA) based heuristic solution algorithm. The fourth section concludes the chapter.

Problem Formulation

To formulate the Berth Allocation and Quay Crane Scheduling problem (referred to from now on as the BAQCS) we make the following assumptions, some of which are similar to Park and Kim (2003):

1. The maximum number of cranes that can be simultaneously assigned to a vessel, is only limited by the length of the vessel
2. The duration of berthing of a vessel, is inversely proportional to the number of cranes assigned to the vessel and proportional to the distance from the vessels' non favorable position
3. For each vessel a cost is incurred if the committed minimum number of TEU moves per hour is not met. This can be the result of an insufficient number of crane assignment and/or the berthing of the vessel in a non favorable position
4. For each vessel, a penalty cost/premium is incurred by berthing later/earlier or later than the committed time.
5. For each vessel, a penalty cost/premium is incurred by departing later/earlier or later than the committed time.
6. For the port operator, a cost is incurred for servicing a vessel as the handling time of each vessels is decreased due to the increase of cranes

To formulate the BAQCS problem (fig. 9-1) we define the following:

s : $(1,2,3,\ldots\ldots\ldots.S)$ Number of vessels,

L_s : Length of vessel s,

Q : Total number of available cranes,

W : Length of wharf,

A_s : Arrival of vessel s,

C_{ijs} : Handling time of vessel s assigned to cranes i through j,

X_{ijkls} : 1 if vessel s covers rectangle $jikl$, where i, j ($i<j$) and ($k<l$) are the first and last crane assigned to the vessel, and k, l is the start and finish time

TB_s : Requested berthing time of vessel s

TD_s : Requested departure time of vessel s

BTB_s : Hourly premium for early berthing of vessel s

CTB_S : Hourly cost for late berthing of vessel s

BTD_s : Hourly premium for early departure of vessel s

CTD_s : Hourly cost for late departure of vessel s

HC_s : Hourly handling time cost of vessel s (depended on the number of cranes assigned to the vessel)

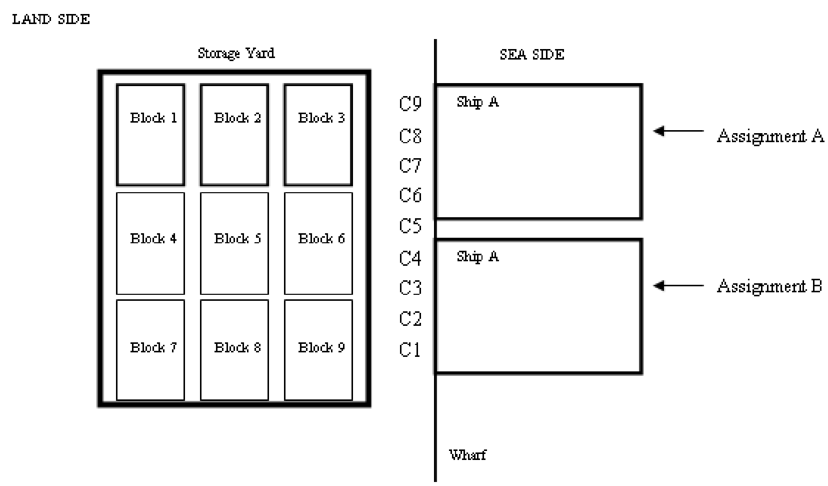
QCs : Number of quay cranes committed to operator of vessel s

QCC_s : Unit cost from not meeting quay crane quota for vessel s

V_s : Volume (in TEUs) to be (un)loaded from vessel s

ST_s : Start time of service of vessel s

FT_s : Finish time of service of vessel s



Note: C: Quay Crane

Figure 9-1 Schematic representation of the problem formulation

The problem can then be formulated as follows:

$$\min(\sum_s(-t_{1s}-t_{2s}-t_{3s}-t_{4s})+\sum_i\sum_j\sum_k\sum_l\sum_s HC_s(i-j+1)X_{ijkls}+ \\ \sum_i\sum_j\sum_k\sum_l\sum_s QCC_s(QC_s-i+j-1)X_{ijkls} \qquad \text{(Eq. 9-1)}$$

$$\sum_i\sum_j\sum_k\sum_l X_{ijkls}=1,\forall s\,, \qquad \text{(Eq. 9-2)}$$

$$\sum_i\sum_j\sum_s\sum_l L_s X_{ijkls}\le W,\forall k \qquad \text{(Eq. 9-3)}$$

$$M(1-X_{ijkls})+X_{ijkls}-\sum_{s'\neq s}\sum_{i>a}\sum_{j>b}\sum_{d>k>c}\sum_{l>d}X_{abcds'}-\sum_{s'\neq s}\sum_{i<a}\sum_{j>b}\sum_{d>k>c}\sum_{l>d}X_{abcds'} \\ -\sum_{s'\neq s}\sum_{i<a<j}\sum_{j>b}\sum_{d>k>c}\sum_{l>d}X_{abcds'}-\sum_{s'\neq s}\sum_{i>a}\sum_{j>b}\sum_{k<c}\sum_{l>d}X_{abcds'}-\sum_{s'\neq s}\sum_{i<a}\sum_{j>b}\sum_{k>c}\sum_{l>d}X_{abcds'}\,, \\ -\sum_{s'\neq s}\sum_{i<a<j}\sum_{j>b}\sum_{k>c}\sum_{l>d}X_{abcds'}-\sum_{s'\neq s}\sum_{i>a}\sum_{j>b}\sum_{k<c<l}\sum_{l<d}X_{abcds'}-\sum_{s'\neq s}\sum_{i<a}\sum_{j>b}\sum_{k<c<l}\sum_{l<d}X_{abcds'} \\ -\sum_{s'\neq s}\sum_{i<a<j}\sum_{j>b}\sum_{k<c<l}\sum_{l<d}X_{abcds'}\ge 1,\forall i,j,k,l,s \qquad \text{(Eq. 9-4)}$$

$$\sum_i\sum_j\sum_k\sum_l(k*X_{ijkls})-A_s\ge 0,\forall s\,, \qquad \text{(Eq. 9-5)}$$

$$ST_s = \sum_i \sum_j \sum_l k * X_{i\,jkls}, \forall s, \quad \text{(Eq. 9-6)}$$

$$FT_s = \sum_i \sum_j \sum_k l * X_{i\,jkls}, \forall s, \quad \text{(Eq. 9-7)}$$

$$t_{1s} \leq FT_s - DT_s, \quad \text{(Eq. 9-8)}$$

$$t_{2s} \leq DT_s - FT_s \quad \text{(Eq. 9-9)}$$

$$t_{3s} \leq ST_s - BT_s, \quad \text{(Eq. 9-10)}$$

$$t_{4s} \leq BT_s - ST_s, \quad \text{(Eq. 9-11)}$$

t_{1s}, t_{2s} ,t_{3s}, t_{4s} <0, X_{ijkls}= Binary, (Eq. 9-12)

The first term of the objective function corresponds to the total costs/premiums form early or late departures/berthing while the second term to the total handling costs. The last term corresponds to the cost from not meeting the committed productivity of (un)loading operations to a vessel, expressed in the number of cranes assigned.

According to constraints (9-2) each vessel is serviced once. According to constraints (9-3) at each point in time the vessels served at the wharf have a total length less then the length of the wharf. A grid square must be covered only by one square according to constraints (9-4). Constraints (9-5) ensure that each vessel is served after its arrival, while constraints (9-6) through (9-12) estimate the start and finish time of service and the delay/earliness of berthing/departure for each vessel.

Solution Approach

A heuristic is proposed for the problem formulation presented herein, since there is not likely that an efficient exact solution procedure exists, which finds an optimal solution in polynomially bounded computation time. The procedure we employ for the heuristic is the Genetic Algorithms (GAs).

Representation

In this chapter we use an integer chromosomal representation in order to exploit in full the characteristics of the problem. We use a two-dimensional chromosome to capture the nature of the problem (Kahng and Moon, 1995; Pargas and Jain; 1993; Al-Attar, 1994; Lin et al., 1993). For further details refer to Krzanowski and Raper (2001). An example chromosome representation is provided in Figure 9-2, using a small instance of a problem (5 vessels, 10 quay cranes, and a planning horizon of maximum of 100 hours). Each chromosome will have 1000 cells {chromosome length= (Number of Quay Cranes) x (Planning Horizon Length)}. In the chromosome in figure 9-2 vessel 1 is serviced by quay cranes 1 through 3, starts service at the beginning of the planning horizon, finishing service 3 hours later. Cells with zero value represent that no vessel is serviced at that time.

Quay Crane								
10	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
8	0	0	0	0	**4**	**4**	**4**	0
7	0	**3**	**3**	0	**4**	**4**	**4**	0
6	**2**	**3**	**3**	0	0	0	0	0
5	**2**	**3**	**3**	0	0	0	0	0
4	0	0	0	0	0	**5**	**5**	**5**
3	**1**	**1**	**1**	**1**	0	**5**	**5**	**5**
2	**1**	**1**	**1**	**1**	0	0	0	0
1	**1**	**1**	**1**	**1**	0	0	0	0
Time	0	1	2	3	4	5	… … … .	100

Figure 9-2. Chromosome Representation

Fitness/Selection

The fitness/selection criterion applied in this chapter was presented in Chapter 5.

Crossover/Mutation

Crossover and mutation will be performed in our heuristic with the objective to zero out chromosomes with infeasible solutions. A crossover may generate infeasible children i.e., a child chromosome may have no vessels served or vessels served twice. Crossover will not be allowed since this will lead to a large number of infeasible solutions that will either will be discarded or mutated to produce feasible solutions. Mutation is performed using a one dimensional chromosome that defines the order by which a vessel will be selected and included to the 2D chromosome (following a shortest start and service time rule). Figure 9-3 shows an example of the 1D chromosome used for the genetic operations. In figure 9-3 the first vessel that will be assigned in the 2D chromosome is vessel 2 and will be assigned so that it begins service as soon as possible (after its arrival date) and is assigned the maximum number of available cranes.

2	4	3	5	1

Figure 9-3 One Dimensional Chromosome Representation

Since it is unlikely that a vessel arriving at the end of the planning horizon will be serviced before a vessel arriving at the beginning of the planning horizon, we apply a tabu ruled based mutation that restricts certain mutation operations. A small example of the tabu mutation is shown in figure 9-4. Assume that vessel 8 arrives at day 1 of the planning horizon while vessel 5 at day 5. The tabu rule forbids any mutation type that will move vessel 5 to be serviced before vessel 8. To achieve the tabu mutation the 1D chromosome population is partitioned into N sub-populations of equal size, where N is equal to the number of days of the planning horizon. For each sub-population vessels are assigned a selection order (to be assigned to the 2D chromosome) within a specific time window after their arrival date, ranging from one day to the full planning horizon. The resulting 2D mutation is shown in figure 9-5.

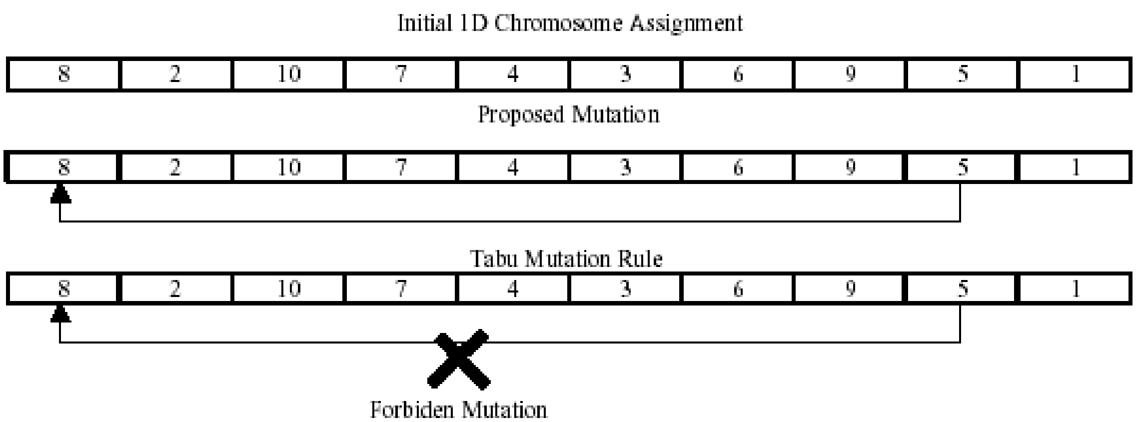


Figure 9-4 Tabu Mutation Rule

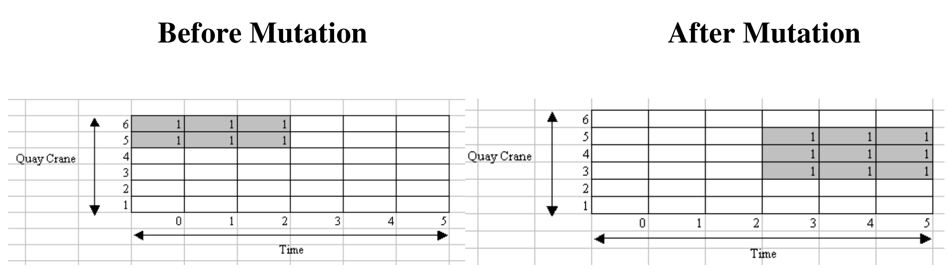


Figure 9-5 Chromosome mutation scheme in two dimensions

Four different types of mutation are applied: insert, swap, inversion, and scramble. Each of the four types of mutations is applied to all the 1D chromosomes and has its own characteristics in terms of preserving the order and adjacency information. Insert mutation picks two cells at random and moves the second one to follow the first, thus preserving most of the order and adjacency information. Inversion mutation picks two cells at random and then inverts the substring between them preserving most adjacency information (only breaks two links) but disrupting the order information. Swap mutation picks two cells from a chromosome and swaps their positions preserving most of the adjacency information but disrupting the order. Finally, scramble mutation scrambles the position of a subset of cells of the chromosome. Computational experiments showed that when all four mutations were applied, the GA heuristic converged at a faster rate and there was significant improvement in the value of the objective function. Thus in our algorithm we employed all four mutation types but as the GA progressed the weight was shifted from the Inverse and Scramble mutation to the Insert and Swap mutation. This way, in the beginning of the search the heuristic performs large *jumps* and as the objective function improves the heuristic searches in an increasing smaller region. The procedure of the full GA heuristic is outlined in figure 9-6.

Numerical Experiments

Dataset Description

Problems used in the experiments were generated randomly but systematically. We developed six test problem instances where vessels are served with various handling volumes (ranging from 500 to 4000 TEU) at a multi-user container terminal (MUT) with 12 quay cranes, with a planning horizon of one week. The number of vessels for each problem instance was 15, 20, 25, 30, 40, and 50 respectively. The random generation process was based on data from two real world container terminals with similar terminal operating systems (one in Europe and one in the US). The range of variables and parameters considered were chosen according to the data obtained from these two container terminals. To our knowledge there is limited literature on 2D GA and none that provides indicative values for the genetic algorithm parameters. In this chapter a generation of 100 and a population of 10 were used.

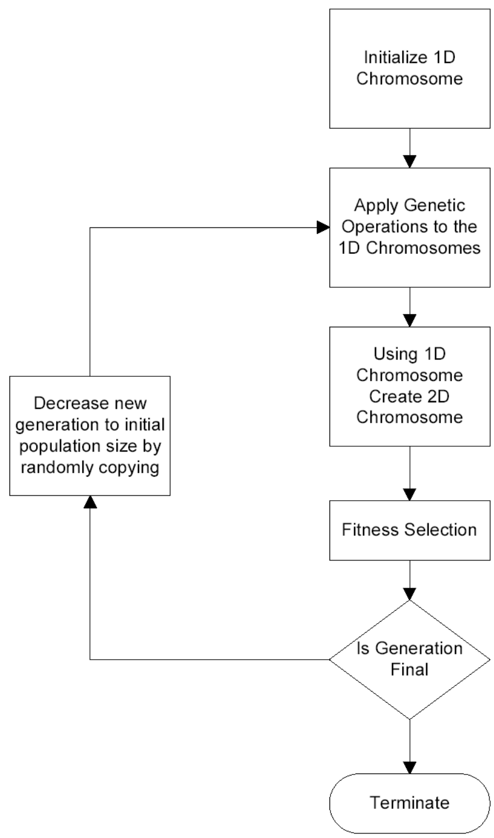


Figure 9-6 Genetic Algorithm Heuristic

A sub-case of the generic model was used for the experiments, with the objective to minimize the total costs from not meeting the committed productivity of (un)loading operations to a vessel. To formulate this problem we assumed that $t_{1s}= t_{2s}=t_{3s}=t_{4s}=HC_s=0, \forall s$. The sub-case formulation is shown in equations 9-13 through 9-18

$$\min \sum_{i}\sum_{j}\sum_{k}\sum_{l}\sum_{s} QCC_s (QC_s - i + j - 1) X_{ijkls} \quad \text{(Eq. 9-13)}$$

$$\sum_{i}\sum_{j}\sum_{k}\sum_{l} X_{ijkls} = 1, \forall s, \quad \text{(Eq. 9-14)}$$

$$\sum_{i}\sum_{j}\sum_{s}\sum_{l} L_s X_{ijkls} \le W, \forall k \quad \text{(Eq. 9-15)}$$

$$\begin{aligned} & M(1 - X_{ijkls}) + X_{ijkls} - \sum_{s'\neq s}\sum_{i>a}\sum_{j>b}\sum_{d>k>c}\sum_{l>d} X_{abcds'} - \sum_{s'\neq s}\sum_{i<a}\sum_{j>b}\sum_{d>k>c}\sum_{l>d} X_{abcds'} \\ & - \sum_{s'\neq s}\sum_{i<a<j}\sum_{j>b}\sum_{d>k>c}\sum_{l>d} X_{abcds'} - \sum_{s'\neq s}\sum_{i>a}\sum_{j>b}\sum_{k<c}\sum_{l>d} X_{abcds'} - \sum_{s'\neq s}\sum_{i<a}\sum_{j>b}\sum_{k>c}\sum_{l>d} X_{abcds'}, \\ & - \sum_{s'\neq s}\sum_{i<a<j}\sum_{j>b}\sum_{k>c}\sum_{l>d} X_{abcds'} - \sum_{s'\neq s}\sum_{i>a}\sum_{j>b}\sum_{k<c<l}\sum_{l<d} X_{abcds'} - \sum_{s'\neq s}\sum_{i<a}\sum_{j>b}\sum_{k<c<l}\sum_{l<d} X_{abcds'} \\ & - \sum_{s'\neq s}\sum_{i<a<j}\sum_{j>b}\sum_{k<c<l}\sum_{l<d} X_{abcds'} \ge 1, \forall i, j, k, l, s \end{aligned} \quad \text{(Eq. 9-16)}$$

$$\sum_{i}\sum_{j}\sum_{k}\sum_{l} (k * X_{ijkls}) - A_s \ge 0, \forall s, \quad \text{(Eq. 9-17)}$$

X_{ijkls}= Binary, (Eq. 9-18)

Results

Figure 9-7 shows the average computational time for 100 generations for each problem instance for the first and second problem respectively. Excluding the last problem instance the average computational time is within acceptable limits.

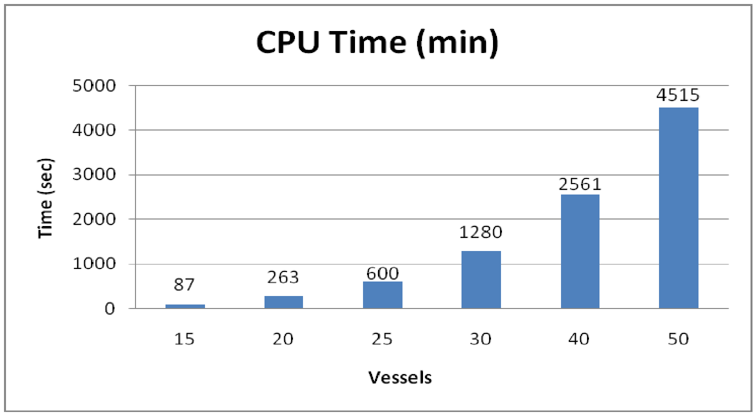


Figure 9-7 Computational Time (Case A)

Figure 9-8 shows the values of the objective function for the instances per generation. The upper part of the graph shows all the values of the objective function obtained from all the chromosomes at each generation while the lower part the best. We observe that for all the problem instances the algorithm converged immediately. This can be partially attributed to the large number of population used but mainly to the objective of the problem. Since the objective of the problem is to find the minimum berth productivity cost the solution is insensitive to the service order of the vessels and parameters such as the total service time or the finish time of each vessel. To answer this issue we take advantage of the multi-population structure of the GA heuristic and at every generation we store the finish time of the last vessel for each chromosome. Figure 9-9 shows the finish time of the last vessel for each chromosome for each generation. The solution approach is flexible to incorporate any other type of criteria (i.e. total service time, total waiting time etc).

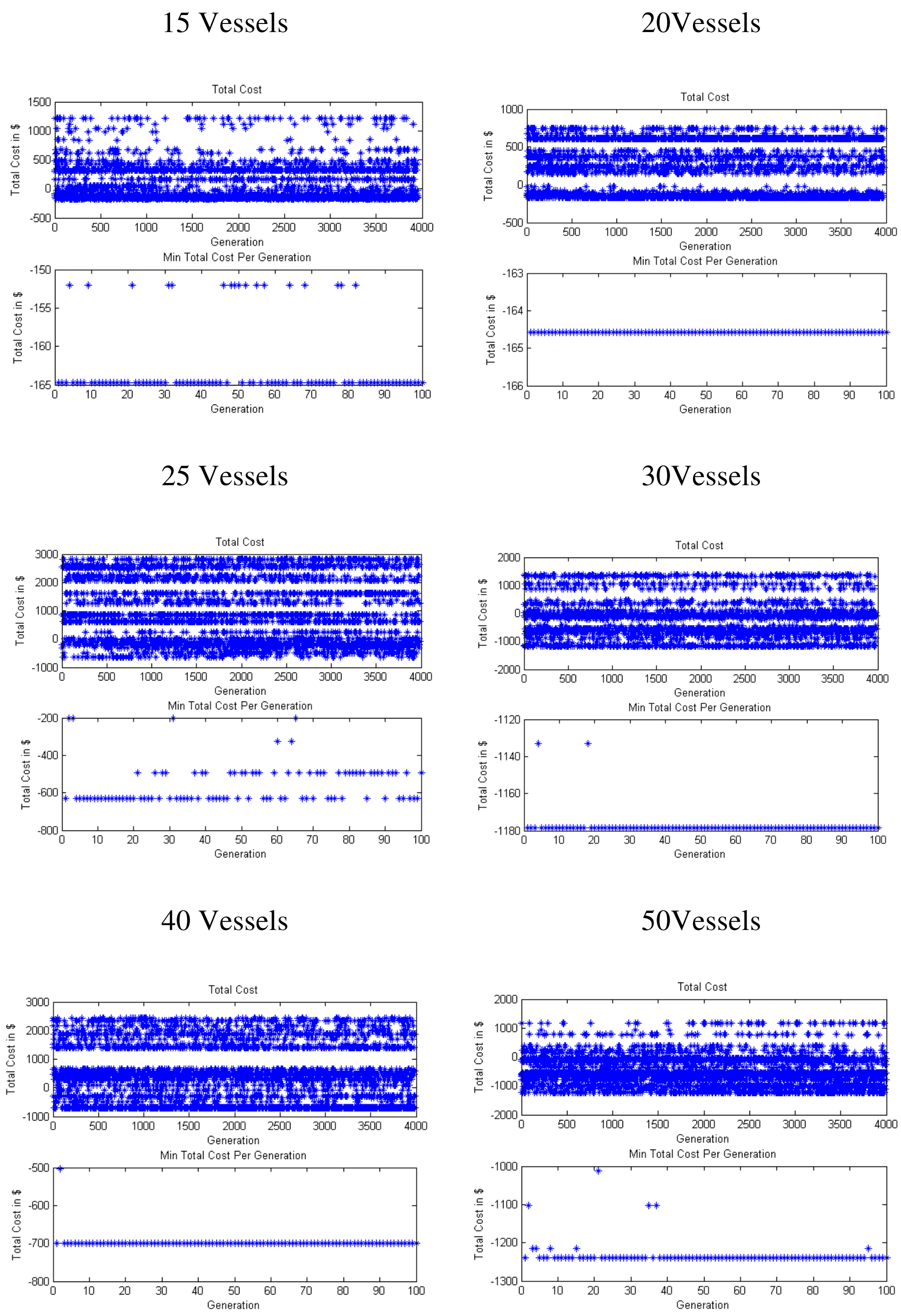
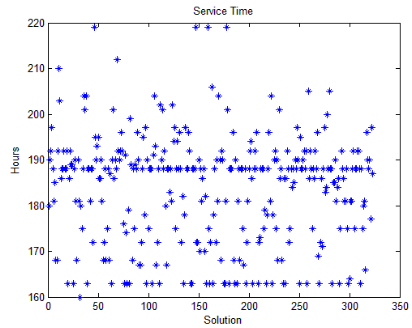
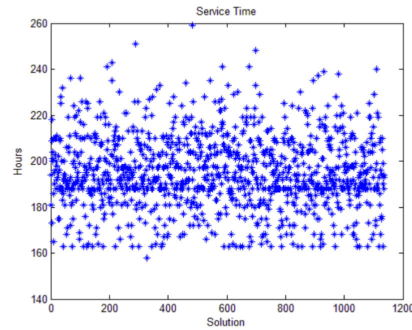


Figure 9-8 Objective Value Progression

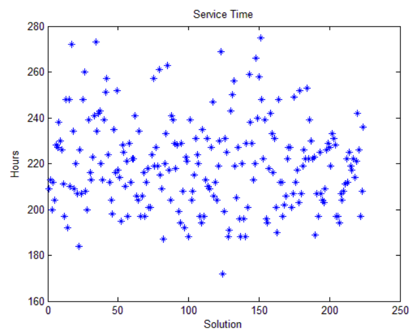
15 Vessels



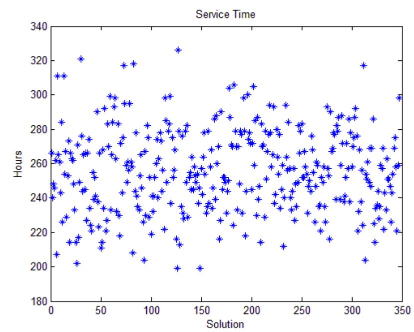
20Vessels



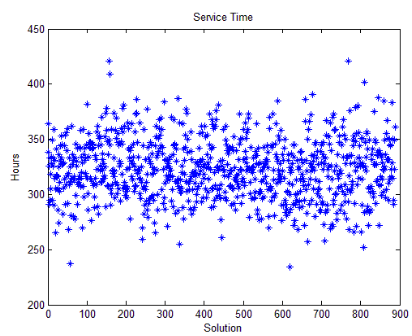
25 Vessels



30 Vessels



40 Vessels



50Vessels

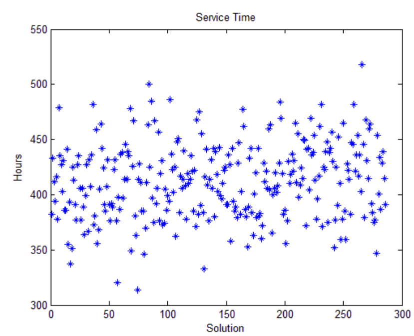


Figure 9-9 Service Time

Conclusions

In berth allocation, the calling time of vessels, favorable vessel berthing locations, and the number of available quay cranes must be considered simultaneously. The vessels arrive at the port over a period of time and normally request service and departure within a specified time window. Based on contractual agreement, carriers usually request for a minimum berth service productivity, translated to the average number of containers loaded/unloaded onto the ship per hour. Failure to meet these contractual agreements implies costs to both the port operator and the ocean carrier. Based on these contractual agreements different ships receive different service productivity levels, translated to the berth location and the number of quay cranes assigned. In this chapter the berth-allocation and quay crane scheduling problem was studied. The problem was formulated as an integer programming model with the objective to minimize costs inadequate berth productivity service levels. The model presented herein simultaneously assigned quay cranes and dynamically allocated ships along a wharf, assuming that the handling time of each ship is a function of the number of cranes assigned and the location of the vessel along the wharf, including wharf length constraints.

A two-dimensional GA based heuristic and a tabu rule mutation based heuristic procedures were developed to solve the resulting problem. The proposed approach adopted in this chapter could be beneficial for ports operated by a company different than the ocean carrier since it provides information on costs endured from meeting (or failure to meet) service contractual agreements. The proposed approach could also be valuable to terminals operated by the carrier as different ships may have different priorities for the carrier and consequently different departure deadlines by which they must complete cargo handling operations and leave for the next destination port. Finally, the formulation and solution approach allows for the model to be easily reduced and produce a number of different BSPs. Future research will focus on evaluating the proposed heuristic to real world data and improvement on the heuristics performance.

10. BERTH ALLOCATION BY CUSTOMER SERVICE DIFFERENTIATION: A MULTI-OBJECTIVE APPROACH

Introduction

Container terminal operators set several objectives when defining berth schedules (reduce vessel turnaround time, increase port throughput, increase revenues, increase competitiveness of the port, increase customers' satisfaction etc), which ideally need to be optimized simultaneously. These multiple objectives are often non-commensurable. Gaining an improvement on one objective often causes degrading performance on the other objectives.

Research on berth allocation has recognized the multi-objective nature of the problem (Steenken et al. 2004, Vis and DeCoster, 2003, Hansen et al. 2007), but has been restricted in either combining the multiple objectives into a single scalar value (Imai et al. 2003; Hansen et al., 2007) or restricting optimization to one of the objectives (the majority of the literature focuses on the minimization of the vessels total handling and waiting time). The former approach consists of using a weighted aggregate function according to preferences set by decision-makers. The weighted approach complexity and accuracy lies in the proper selection of the weights or utility functions that are used to depict the decision-maker's preferences. In practice, it can be very difficult to precisely and accurately select these weights, even for someone familiar with the problem domain (Coello Coello, 2000; Konak et al., 2006). Furthermore, in the berth allocation problem (BAP) selecting the appropriate weights for each vessels/customer in order to satisfy contractual agreements, between the port operator and the liner shipping company, may be a very cumbersome or even an impossible task.

From the computational complexity theory, berth allocation (as with most scheduling problems) is known to be NP-hard or NP-complete depending on the formulation, objective function and constraint type (Papadimitriou and Steiglitz, 1982; Pinedo, 2002, Imai et al. 2003, Imai et al., 2005). In addition, modeling different BSPs results in a number of different BAP formulations with different constraints, some hard and other soft. Hard constraints must not be violated (for example each vessel must be serviced once and each berth can service one vessel at a time, two constraints found in all the BAP formulations in the literature) while soft constraints, usually used to estimate auxiliary variables, can be relaxed (for example see Moorthy and Teo, 2006). Satisfying both types of constraints is a difficult problem itself. When different constraints cannot be satisfied simultaneously, the problem is often deemed to admit no solution. On the other hand if constraints are relaxed then the problem solution is inferior. A multi-objective formulation offers the advantage of treating these constraints as objectives, and can consistently outperform the single objective approach without a significant sacrifice in terms of performance (Coello Coello, 2000). This observation could be proven very valuable in complex berth allocation problems where a number of the constraints, that limit the feasible region of the problem, can be viewed as objectives.

In this chapter we formulate the BAP as a multi-objective mixed integer optimization problem (MOMIP). Special attention is given to customer service differentiation by the use of different objective functions. As pointed out by Imai et al. (2003), vessels with a large container handling volume typically request to be given higher priority over small vessels, leading to a decrease in berth productivity (high total service time for all the vessels at the current planning horizon). On the other hand if vessels with small container handling volume are given priority then large vessels are forced to wait, leading to customer dissatisfaction. The goal of this chapter is to use a multi-objective formulation that will provide the port operator with a variety of different berth schedules ranging from a schedule with the best overall berth performance (in terms of the total service time for all the vessels) to a schedule with minimum customer dissatisfaction (in terms of the total service time for the customers' vessels). To our knowledge this is the first time that the proposed BSP has been formulated and solved as a multi-objective optimization problem. Due to the nature of the problem a Genetic Algorithms (GA) based heuristic solution is proposed (Taboada, 2007). A number of numerical experiments are performed to evaluate the performance of the heuristic and critically discuss the benefits of the proposed approach. Results show that the proposed approach outperforms the weighted approach and the state of the art multi-objective heuristic NSGA-II (Deb et al., 2002). The rest of this chapter is organized as follows. The next section presents a brief description of the general problem and the model formulation, while the third section describes the solution approach. The fourth section presents a number of experimental results and the last section concludes the chapter.

Problem description and formulation

In our model we make several assumptions: a) The wharf is divided into a number of berths and each berth can service one vessel at a time regardless of the vessel's size, b) The handling time of the vessel is agreeable to its handling volume and depended on the berth assigned, c) Once a vessel has moored, it will remain in its location until all the required container processing is done, and d) There are no physical or technical restrictions (i.e. water depth). In our model we assume that each vessel arriving at the port requesting service, belongs to a customer (preferential or not). Each customer may define different subgroups of vessels with different priorities in terms of their total service time, since different vessels have different priorities depending on the schedule of the vessel, calling time at the next port, delays of arriving at the current port, updated information of service at the next port of call etc. These subgroups of vessels that belong to the same customer are considered as separate subgroups that may or may not be treated preferentially.

To illustrate this concept we present the example in Figure 10-1, where two customers with different priorities for their vessels request service. In this example vessels 1, and 2 of customer A, have different priorities than vessels 3 and 4, and the same stands for vessels 5, 6, 7, and 8 belonging to customer B. Thus, we can assume that customer A is represented by two customers, A1 and A2, each having two vessels and that customer B is represented by two customers, B1 and B2, with three and one vessels respectively. In this example only a portion of the vessels of customer B are preferential (i.e. A1 and A2 are both preferential, while only B1 is preferential from the B1, B2 group) and thus the problem would have four objective functions, one for minimizing the total service time and three for minimizing the service time of the vessels belonging to the three different preferential customers (A1, A2, and B1).

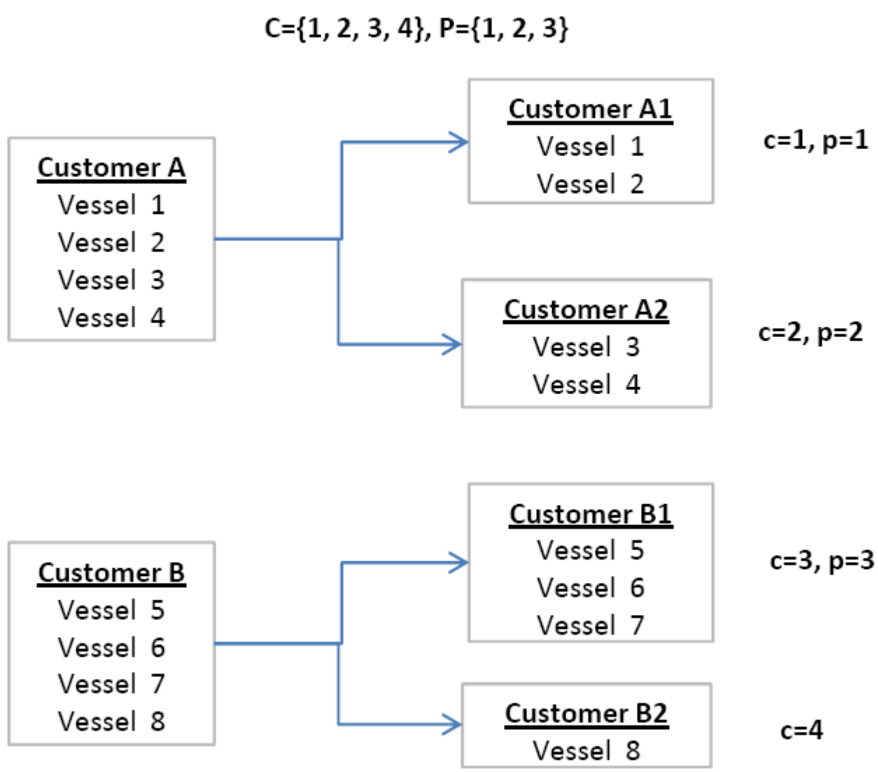


Figure 10-1. Example of Customer and Preferential Customer Sets

In order to formulate the multi-objective discrete and dynamic BAP with service priorities (MBAP) we need to define the following:

$i=(1,\ldots\ldots,I) \in B$ set of berths,

$j=(1,\ldots.,T) \in T$ set of vessels,

$k=(1,\ldots.,T) \in O$ set of service orders,

$c=(1,\ldots\ldots.,C) \in C$ set of customers,

$p=(1,\ldots\ldots.,P) \in P \subseteq C$ set of preferential customers,

S_i= Time when berth becomes idle for the first time in the planning horizon,

A_j= Arrival time, HT_{ij}= Handling time of vessel j at berth i,

y_{ijk} = Idle time of berth i between departure of vessel j service as the k^{th} vessel (from the end), and its immediate predecessor,

X_{ijk} = 1 if vessel j is serviced at berth i as the k^{th} vessel(from the end) , and zero otherwise,

WT_{ij}=wait time of vessel j serviced at berth i.

The MBAP can be formulated as follows:

[MBAP]: $\min \sum_{i \in B} \sum_{j \in V} \sum_{k \in O} (kHT_{ij} + S_i - A_j) X_{ijk} + \sum_{i \in B} \sum_{j \in V} \sum_{k \in O} k y_{ijk}$ (Eq. 10-1)

$$\min \sum_{i \in B} \sum_{j \in J} \sum_{k \in O} (HT_{ij} + WT_{ij}) X_{ijk}, \; p = 1, j \in P \qquad \text{(Eq. 10-2)}$$

$$\min \sum_{i \in B} \sum_{j \in J} \sum_{k \in O} (HT_{ij} + WT_{ij}) X_{ijk}, \; p = 2, j \in P \qquad \text{(Eq. 10-3)}$$

……………………………………………………………………

$$\min \sum_{i \in B} \sum_{j \in J} \sum_{k \in O} (HT_{ij} + WT_{ij}) X_{ijk}, \; p = P, j \in P \qquad \text{(Eq. 10-4)}$$

Subject to: $\sum_{i \in B} \sum_{k \in U} X_{ijk} = 1, \forall j \in T$ (Eq. 10-5)

$\sum_{j \in T} X_{ijk} \leq 1, \forall i \in B, k \in O$ (Eq. 10-6)

$$\sum_{m \neq j \in T} \sum_{h<k \in O} (HT_{im} X_{imh} + y_{imh}) + y_{ijk} - (A_j - S_i) X_{ijk} \geq 0,$$
$$\forall i \in B, j, \in T, k \in O$$ (Eq. 10-7)

$$WT_{ij} \geq \sum_{m \neq j \in T} \sum_{h<k \in O} (HT_{im} X_{imh} + y_{imh}) - A_j + S_i - M(1 - X_{ijk}),$$
$$\forall i \in B, j, \in T, k \in O$$ (Eq. 10-8)

$X_{ijk}, \in \{0,1\},\ y_{ijk} \geq 0, WT_{ij} \geq 0$ (Eq. 10-9)

The first objective function (10-1) minimizes the total of waiting and handling time (also known as vessel service time) for all the vessel and the idle time of the berths, while the second set of objective functions (10-2.p, p={1, 2, …..P}) minimize the total of waiting and handling time of all the vessels belonging to the preferential customer p. Constraint set (10-3) ensure that vessels must be serviced once; constraint set (10-4) that each berth services one vessel at a time. Finally, constraint set (10-5) ensures that each vessel is serviced after its arrival, while constraint set (10-6) estimates the waiting time of each vessel (fig. 10-2).

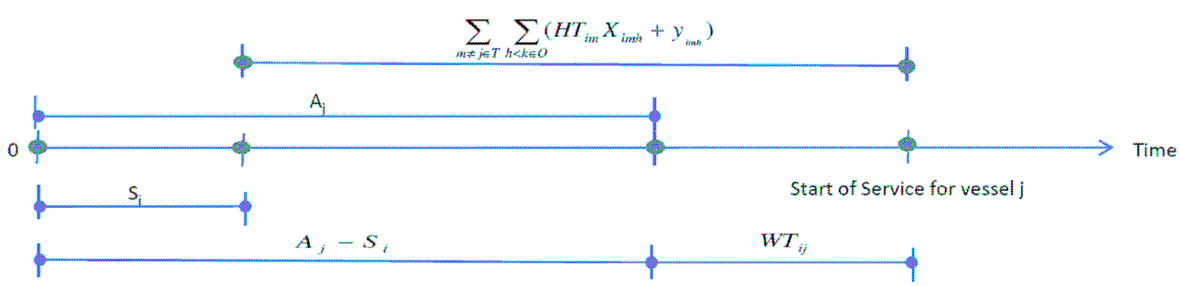


Figure 10-2. Prictorial explanation of estimation of wait time for vessel j serviced at berth i

Solution Approach

The general N-objective optimization problem (or in general the multi-objective optimization problem - MOO) can be defined in the following way (as stated by Coello Coello, 1999): Find the vector of decision variables (also called solution) $X=[x_1, x_2, \ldots, x_n]$ that optimizes (minimizes or maximizes) a vector objective function: $F(X)=[f_1(X), f_2(X), \ldots, f_n(X)]$, subject to m inequality constraints $G_i(X),\ i=(1,2,3\ldots,m)$ and k equality constraints $H_j(X),\ j=(1,2,3,\ldots,k)$. When the variables x are discrete the problem is called *Multi-Objective Combinatorial Optimization* (MOCO) problem. Due to the conflicting nature of the objectives it is usually the case that there is no unique optimal solution. It is possible to improve separately at least one (but not all) objective function of a given solution but this will usually cause the declining of its remaining objective functions (or at least one of them). Thus, several different solutions could be thought of as "optimal", because no one dominates the other. The main difficulty with the multi-objective approach lies in the comparison of the solutions.

By definition one solution outperforms another if the values of all objective functions of the first solution are better than the second. In other words if X_1 and X_2 are two solutions then $F(X_1)$ dominates $F(X_2)$ if and only if, $f_i(X_1) \geq f_i(X_2), \forall i$, and $f_i(X_1) \succ f_i(X_2)$, for at least one i. Such solutions are called "Pareto-optimal". If no solution can dominate the given solution then it can be considered to be optimal. All Pareto-optimal solutions compose a certain boundary between the space, which contains dominated solutions and the space where no solutions exist. This boundary is called the trade-off surface or Pareto-front or Pareto-set. It can be depicted as a surface in the N-dimensional space, where N is the number of objectives.

The use of exact methods to solve multi-objective optimization problems is time consuming and is often infeasible (Zitzler et al., 2002). The most common approach for solving these types of problems is the use of multi-objective metaheuristics (i.e. Evolutionary Algorithms) and usually one needs to develop custom made heuristics that take advantage of the problems domain. In this chapter we develop a multi-objective metaheuristic using Genetic Algorithms (GA). Initially, the fast elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) was considered but the algorithm could not produce any feasible solutions even for large combinations of the population and generation parameters (the maximum number of population and generations used with NSGA-II was 500 and 75000 respectively).

Representation

Although binary-coded GA are commonly used, there is an increasing interest in alternative encoding strategies, such as integer and real-valued representations. For some problem domains, like scheduling problems, it can be argued that the binary representation is, in fact, deceptive since it obscures the nature of the search (Taboada, 2007). Thus, in this chapter we use an integer chromosomal representation in order to exploit in full the characteristics of the problem. For instance, consider the following example of 5 vessels and 2 berths (fig. 10-3). For this problem each chromosome will have ten cells {chromosome length= (Number of Berths) x (Number of Customers)}.

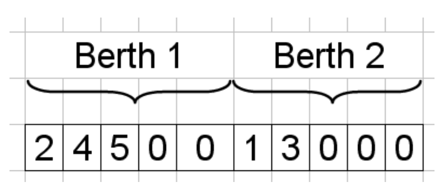


Figure 10-3. Chromosome Representation

The first 5 cells represent the 5 possible service orders in Berth1 and the last 5 cells the 5 possible service orders in Berth 2. In this assignment Vessels 2, 4, and 5 are serviced at Berth 1 as the first, second and third vessel respectively, and Vessels 1 and 3 are serviced in Berth 2 as the first and second vessel respectively. No vessel will be serviced after vessels 5 and 3 (zero value of cell).

Genetic Operations: Crossover & Mutation

In GA operations crossover can combine information from two parents while mutation can introduce new information. Crossover is explorative[16]; it makes a big jump to an area somewhere "in between" two parent areas (Eiben, and Smith, 2003). On the other hand mutation is exploitative[17]; it creates random diversions staying near or not in the area of the parent, depending on the mutation (insert, swap, inversion, and scramble). There is a debate on the use of crossover and mutation and which approach is the best to use. The main conclusion is that the performance of either mutation or crossover is highly affected by the problems' domain. In our problem, at each generation the crossover operation will generate a large number of infeasible children in terms of constraint set (2) (i.e., a child chromosome may not service all the vessels while other vessels are served twice). In the BAP literature simple heuristics were applied to eliminate this problem (Nishimura 2001; Imai 2003; Imai 2006 & 2007). Running several computational examples with and without crossover results showed that problems solved with crossover returned worse solutions than problems using only mutation and were computationally more expensive. We do acknowledge that complex crossover techniques (partially mapped crossover, cycle crossover, and edge recombination) could eliminate the former insufficiency of the crossover operation. This could result though to a significant increase of the computational time and was not implemented within this chapter.

---

[16] ***Exploration***: Discovering promising areas in the search space, i.e. gaining information on the problem
[17] ***Exploitation***: Optimizing within a promising area, i.e. using information

Instead of crossover we experimented with four different types of mutation: insert, swap, inversion, and scramble mutations that were applied to all the chromosomes at each generation. Each of the four types of mutations has its own characteristics in terms of preserving the order and adjacency information. Insert mutation picks two cells at random and moves the second one to follow the first, thus preserving most of the order and adjacency information. Inversion mutation picks two cells at random and then inverts the substring between them preserving most adjacency information (only breaks two links) but disrupting the order information. Swap mutation picks two cells from a chromosome and swaps their positions preserving most of the adjacency information but disrupting the order. Finally, scramble mutation scrambles the position of a subset of cells of the chromosome. Computational experiments showed that when all four mutations were applied, the GA heuristic converged at a faster rate and there was significant improvement in the value of the objective function. Thus in our algorithm we employed all four mutation types but as the GA progressed the weight was shifted from the Inverse and Scramble mutation to the Insert and Swap mutation. This way, in the beginning of the search the heuristic performs large *jumps* and as the objective function improves the heuristic searches in an increasing smaller region.

Fitness/Selection

The MBAP is a multi-objective minimization problem; thus the smaller the values of each objective function are, the higher the fitness value will be. In order to find the best solution for each objective and at the same time retain a variety of different solutions in the Pareto-set we use a multi-population approach. At every generation, after the genetic operations are completed, the mutated generation is split into two sets of equal size used to select the parents of the next generation using two different fitness techniques. Under the first technique, and using the first copy, parents of the next generation are selected based on the Pareto set. If the selected parents are less than the population in the first copy their number is increased by randomly copying from the current parents. If the selected parents are more than the population in the first copy their number is decreased using the selection algorithm known as Roulette Wheel Selection (Goldberg, 1989).

Under the second technique, and using the second copy is used in an elitist way and the best chromosome based on each objective function is selected and copied until their cumulative number is equal to half of the initial population. For example for the case of two preferential customers each best chromosome would be copied until its number is equal to one sixth of the initial population (or one third of the size of the second set).

The purpose of using two separate selection techniques is that the first will increase the variety of solutions in the final Pareto set (figure 10-4) while the second will provide better minimum values for each objective function.

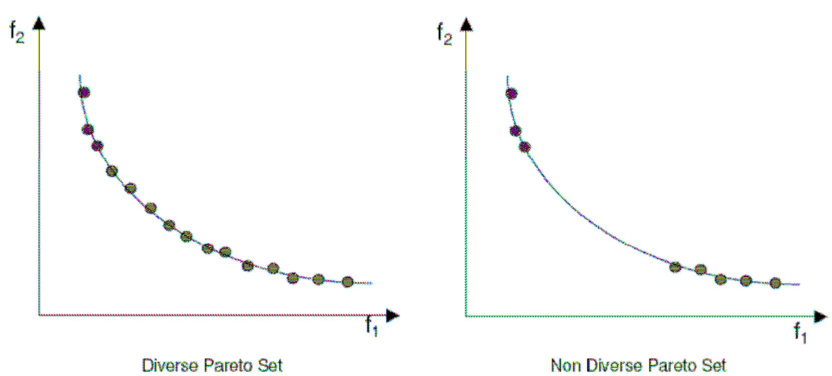


Figure 10-4. Pareto set diversity

The procedure of the GA heuristic is outlined in figure 10-5.

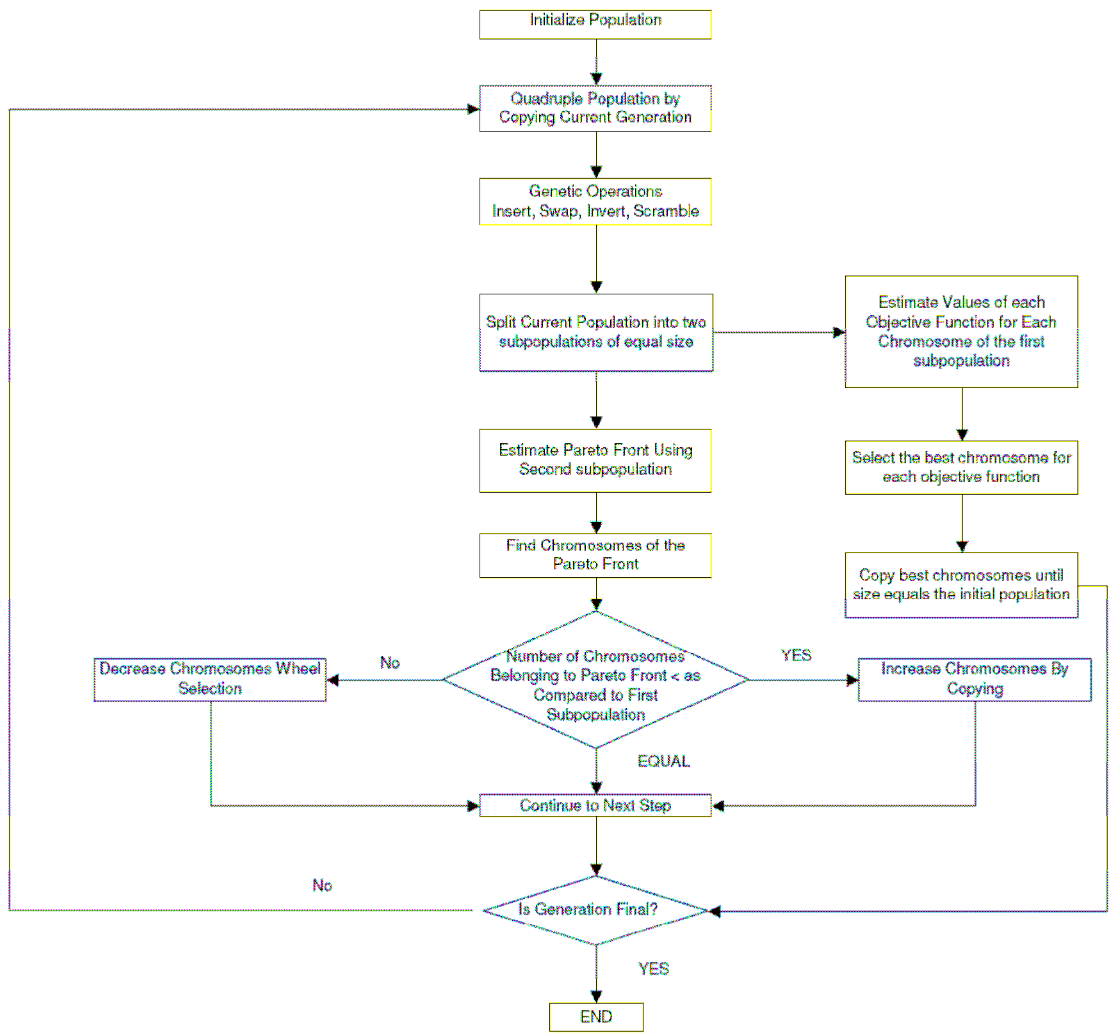


Figure 10-5. Multi-Objective GA Heuristic

Optimality for each objective function in isolation

For small size problems the optimal values for each objective function in isolation (i.e. excluding the rest of the objectives and unnecessary constraints depending on the objective) were obtained using CPLEX while for medium and large problems by the 2-opt heuristic presented in Chapter 7. Any other (meta) heuristic maybe used to obtain the optimal values for each objective function separately, before applying the multi-objective heuristic.

Computational Examples

Dataset Description

Problems used in the experiments were generated randomly but systematically. When creating the experimental data the focus was in obtaining computationally challenging datasets that portray real life conditions. We developed forty base problem instances where vessels are served with various handling volumes at a multi-user container terminal (MUT) with five and ten berths, with a planning horizon of one and two weeks. These are shown in table 10-1, which presents ten datasets for each planning horizon and number of berths considered. The random generation process was based on data from two real world container terminals with similar terminal operating systems (one in Europe and one in the US). The range of variables and parameters considered were chosen according to the data obtained from these two container terminals.

In the dataset used in the experiments, vessel handling volumes (loading and unloading) range from 250 to 4,000 (TEU/vessel), based on a uniform distribution pattern. The handling time of a vessel is dependent on the berth assigned, and is a function of the number of the cranes that may be assigned. We consider that 1 to 3 cranes operate on small sized vessels (<2000 TEU of handling volume), 2 to 4 cranes on medium sized vessels (<3000 TEU of handling volume), and 3-6 on large mother vessels (<4000 TEU of handling volume). The average crane productivity is taken to be 25 TEU/hour. The average vessels per berth per week equivalence (VBWE) is 5 while the minimum, average and maximum handling time is 10, 18, and 30 hours. Testing instances with lower VBWE averages would not provide a representative evaluation of the heuristics performance.

The minimum handling time is calculated by dividing the handling volume by the average productivity of a crane multiplied by the number of cranes operating on the vessel. The handling time of vessels at the other berths is generated in relation to the berth with the minimum handling time. Availability of berths is calculated using a uniform probability with a minimum of zero and a maximum of 10 hours.

One of the most crucial issues in these experiments is the selection of the interarrival vessel distribution. Vessel interarrival patterns were based on the scheduled vessel arrivals at the two container terminals over a period of a year. To test the effectiveness of the approach under highly congested conditions, the peak periods for the two terminals were selected for the purpose of this application. Based on these data we generated vessel arrivals with a minimum of 40 and a maximum of 80 vessels per week for the problems with a berth capacity of five and a minimum of 80 and a maximum of 120 vessels per week for the problems with a berth capacity of ten. Arrival times of vessels within the week period are randomly generated.

Without loss of generality and to graphically present experimental results, we restricted the preferential customer sets to one and two (p={1}, and p={1, 2}). From each dataset we generated four different subsets of vessels belonging to preferential customers using four beta distributions with parameters (2,5), (2,4), (2,3), and (2,2) respectively. In the case of the two preferential customers a beta distribution (2,2) is used to select the number of vessels that belong to each preferential customer. In total 320 problem datasets were generated.

The solution procedure was coded in SciLab 4.1[18] on a Toshiba Satellite Dual Core Intel T2250 with 2GB of RAM. The number of chromosomes was set to 25 and the number of generations

---

[18] Copyright © 1989-2005. INRIA ENPC <www.scilab.org>

was set to 1000. The average computational time per generation was less than two seconds for the larger datasets (ten berths and two weeks planning horizon).

Pareto set

Figures 10-6 to 10-9 show the feasible and Pareto solution space for all the datasets. The upper part of each graph shows the feasible solution space while the lower the Pareto set. As the total service time increased the preferential customers total service time decreased and vice versa, as expected. Minimum values for each objective function were estimated by solving the single objective problem using a 2-opt heuristic algorithm (Chapter 7). Figures 10-10 to 10-13 show similar results for the case of two preferential customers. The Pareto set can be further reduced by exclusion from the Pareto set solutions that do not satisfy preferences of the port operator in terms of the total service time i.e. certain values of the objective functions might be considered unacceptable due to their high values.

Solution Space VS Number of Preferential Customers

Figures 10-14 to 10-21 show the feasible and Pareto solution space. Both the single preferential and the dual preferential customer instances, with the percentage of the total vessels belonging to the preferential customers varying from 20% to 50%, with an increase step of 10%, were considered. We observe that the solutions are robust in terms of the curve shape. As we increase the percentage of preferential customers and the problem size, the Pareto set switches from a stepwise like function to a smoother curve. Furthermore, as we decrease the number of the vessels of the preferential customer the Pareto set does not decrease but retains the number of solutions to an acceptable number (over twenty and under one hundred).

We observe that as we increase the number of preferential customers, the maximum values for the first objective function (total service time for all the customers) in the Pareto set does not always increase. This can be explained as follows: The maximum value of the total service time (TST) for all the vessels in the Pareto set depends on the size of the vessels of the preferential customers. By definition as we increase the number of preferential customers, and thus the number of preferential vessels, the total service time of a schedule focusing on accommodating the needs of the preferential customers will provide a low total berth productivity (high total service time) only if the preferential customers have large handling volumes. This of course is translated to large vessels being serviced before small vessels and large idle berth times, and as a consequence larger service times for all the vessels.

In this chapter preferential customer vessels are not defined by their volumes but rather by the owner shipping line and were chosen in random and not based on vessel handling volumes. Preferential customers are based on total service agreements and do not necessarily involve large container vessels. In practice service of a preferential customer may involve main liner as well as feeder vessels. Although the most preferred solutions for the terminal operator (i.e. the ones that balance the objectives) are usually found around the knee of the curve, all the solutions should be kept, since extreme solutions may, under certain circumstances fit the terminal operator's objectives.

The proposed heuristic's performance was also evaluated in terms of its consistency. For each one of the 320 problems, the ratio of the range of the objective function values for 5 trials (different starting populations) to the lowest objective value, which can be expressed by the highest objective function value during the five trials divided by the lowest objective function value during the five trials, was calculated, for each objective function. The average ratio was less than 7%[19], and thus we can conclude that results obtained from the GA heuristic are consistent for different trials.

Optimality

The multi-objective heuristic, as well as any other multi-objective (meta)heuristic that has appeared in the literature to date cannot guarantee optimality for the solutions in the Pareto set. The same can be stated for the heuristics that have been presented in literature for the BAP. In order to test the quality of the Pareto set the weighted approach was used to solve small instances of the problem using a number of different weights[20], where optimality can be obtained using CPLEX 9.0. All the solutions obtained from the weighted approach were already present in the Pareto set, obtained from the proposed multi-objective heuristic.

Conclusions

In this chapter the discrete and dynamic BAP was formulated and solved for the first time as a multi-objective combinatorial problem. There are two general approaches for the solution of a multi-objective problem, requiring either the aggregation of the objectives into an overall objective function or the determination of a Pareto set. In this chapter the second approach was adopted and a genetic algorithms based heuristic was proposed as a solution approach for the resulting problem. Computational examples showed that the heuristic performed well even for large instances of the problem. The proposed heuristics has two main advantages over the classical weighted approach, traditionally used to solve these types of problems in container terminal operations research (Imai et al., 2004). In terms of the computational complexity while

---

[19] The average ratio increased with the problem size

[20] The weights were sampled from a uniform distribution U(0,1)

the proposed heuristic required a single run to evaluate all possible berth schedules, doing the same with the weighted approach would require an enormous amount of time consuming computations. In terms of usability, the proposed heuristic allows the derivation of a large list of different schedules without the need for precise knowledge of the objective functions priorities and relative importance, which can be very difficult to determine even with a very detailed knowledge of the system (Taboada, 2007). One disadvantage of the heuristic is its inability to guarantee optimality for the Pareto set, a problem faced by all the heuristics proposed for the BAP up-to-date.

Future research is focusing on applying the multi-objective approach to different BSPs and exploring the computational efficiency of the algorithm presented in the next chapter and one by Taboada (2007), two heuristics that guarantee optimality for the Pareto set and can be applied as a second step to the heuristic presented herein.

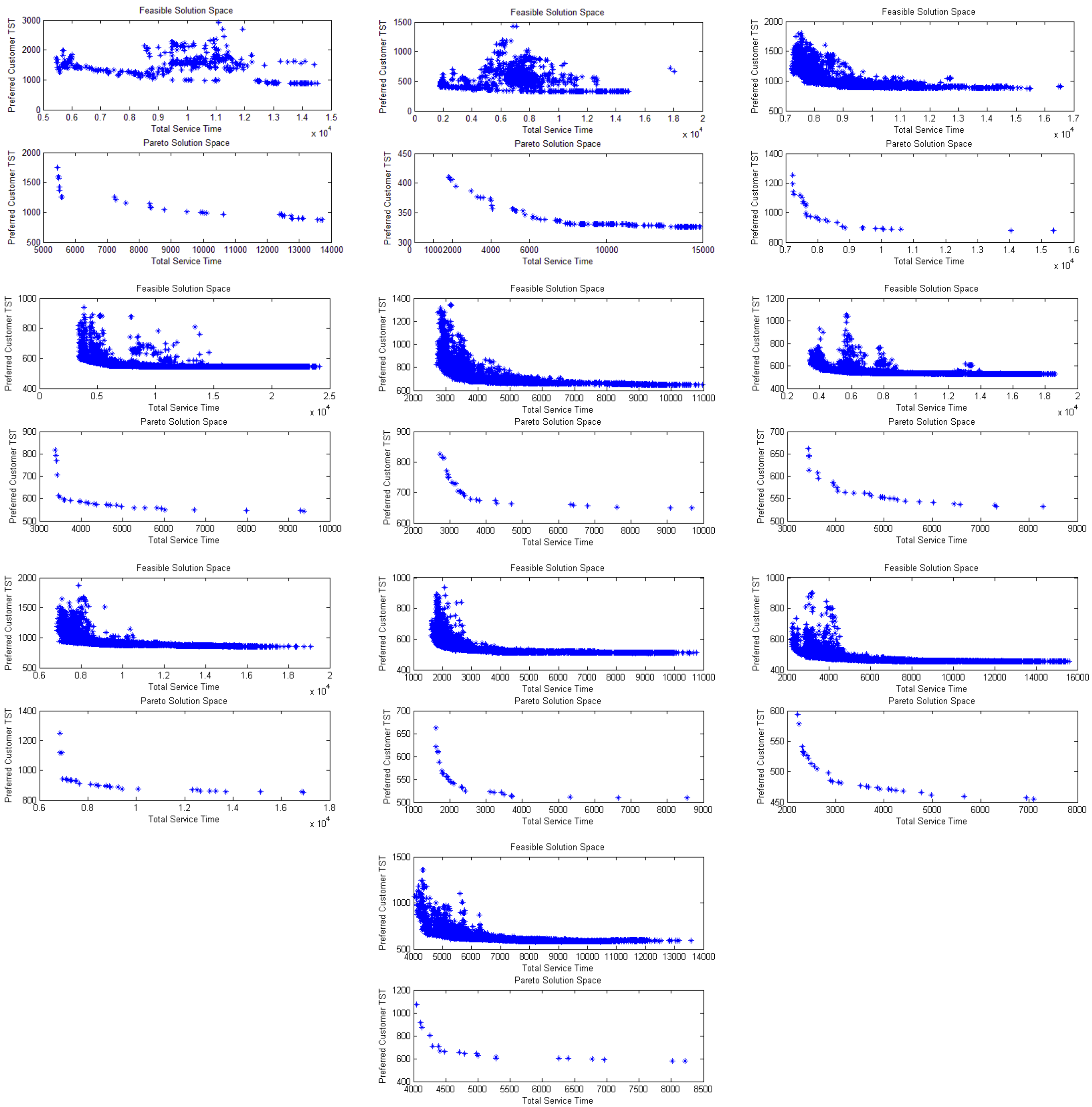


Figure 10-6 Feasible and Pareto Front for Five Berths and One Week Planning Horizon, 50% of Total Vessels Belong to Preferential Customers, One Preferential Customer (Dataset 1 through 10)

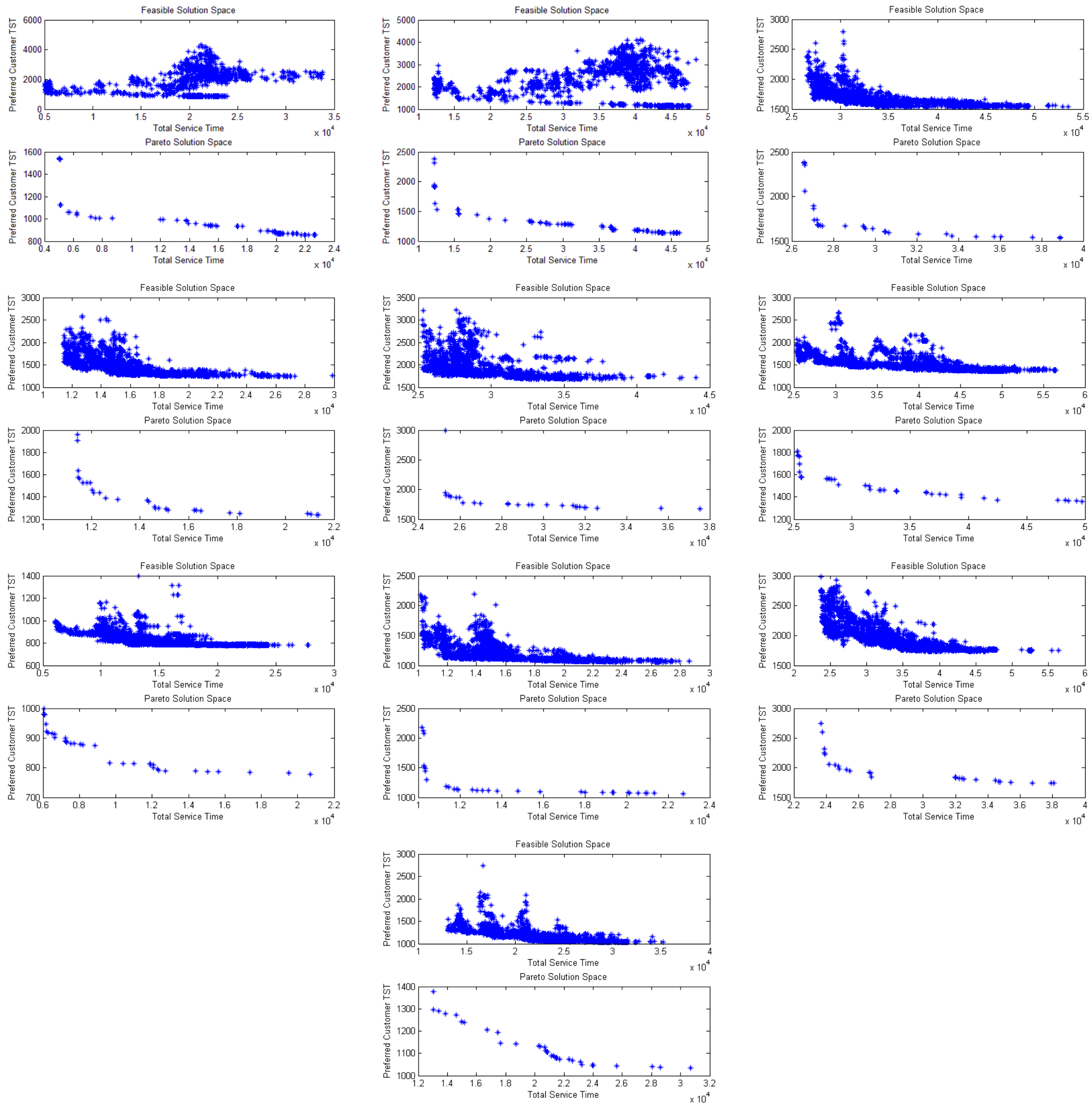


Figure 10-7 Feasible and Pareto Front for Five Berths and Two Weeks Planning Horizon, 50% of Total Vessels Belong to Preferential Customers, One Preferential Customer (Dataset 1 through 10)

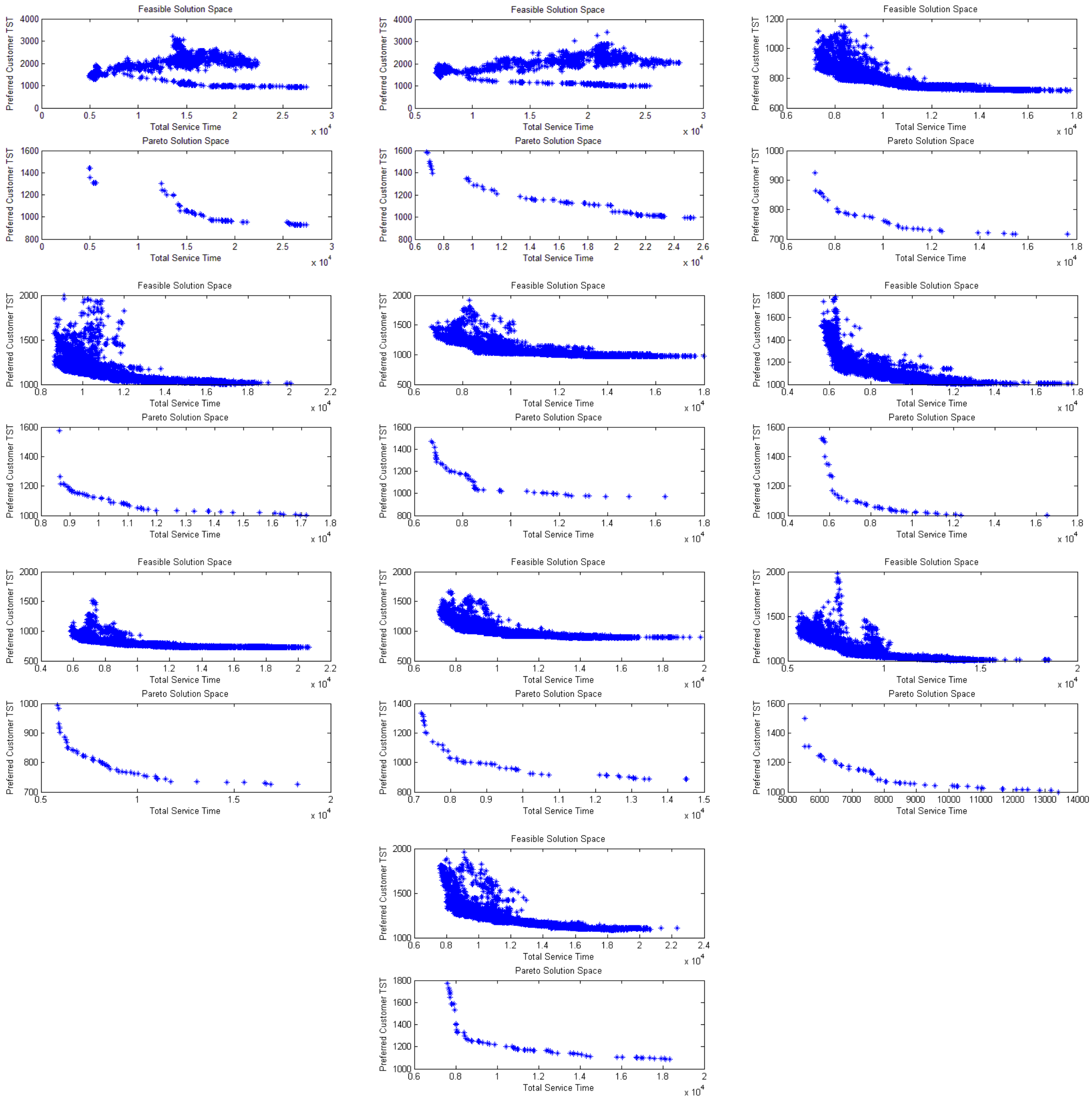


Figure 10-8 Feasible and Pareto Front for Ten Berths and One Week Planning Horizon, 50% of Total Vessels Belong to Preferential Customers, One Preferential Customer (Dataset 1 through 10)

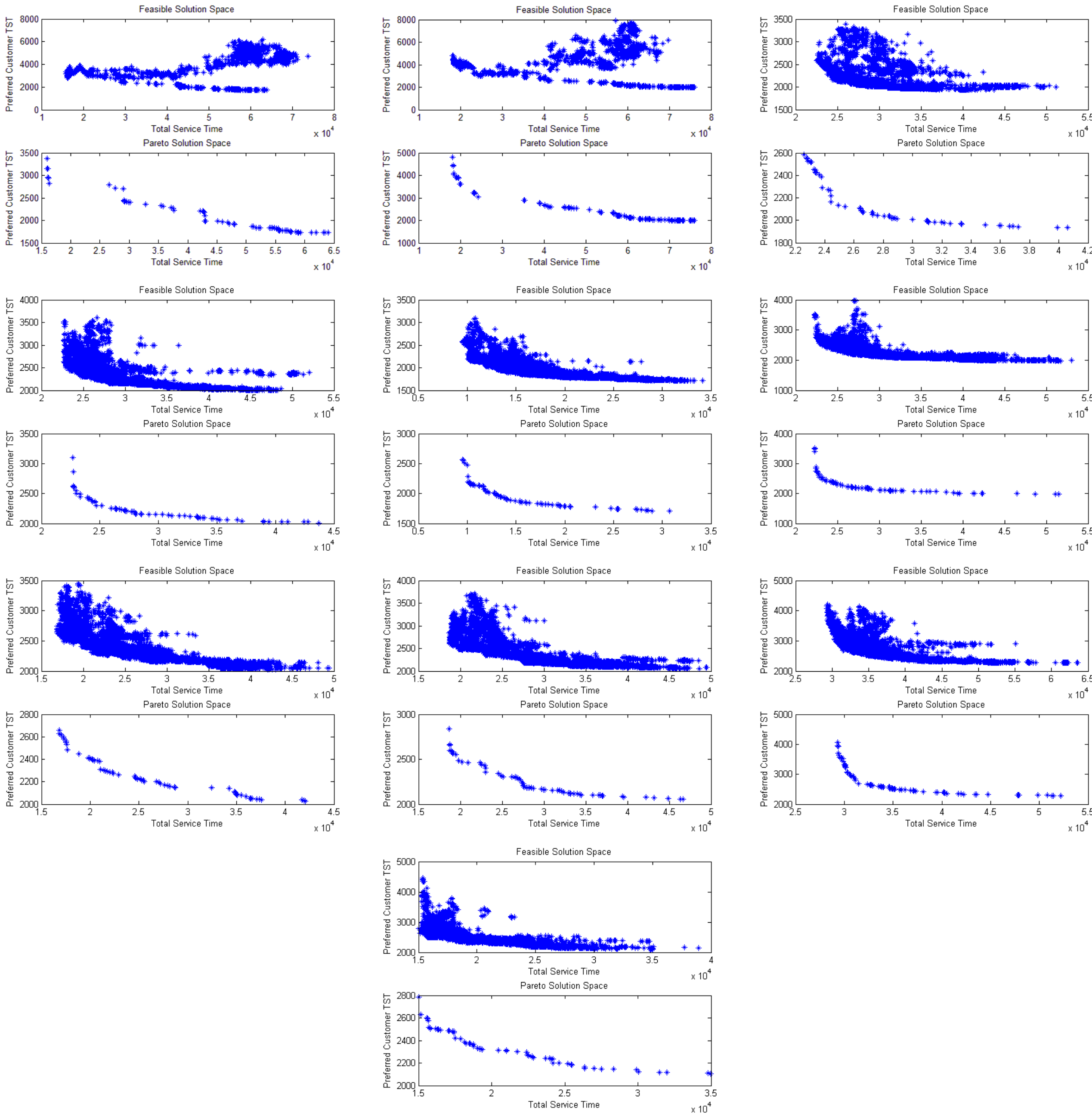


Figure 10-9 Feasible and Pareto Front for Ten Berths and Two Weeks Planning Horizon, 50% of Total Vessels Belong to Preferential Customers, One Preferential Customer (Dataset 1 through 10)

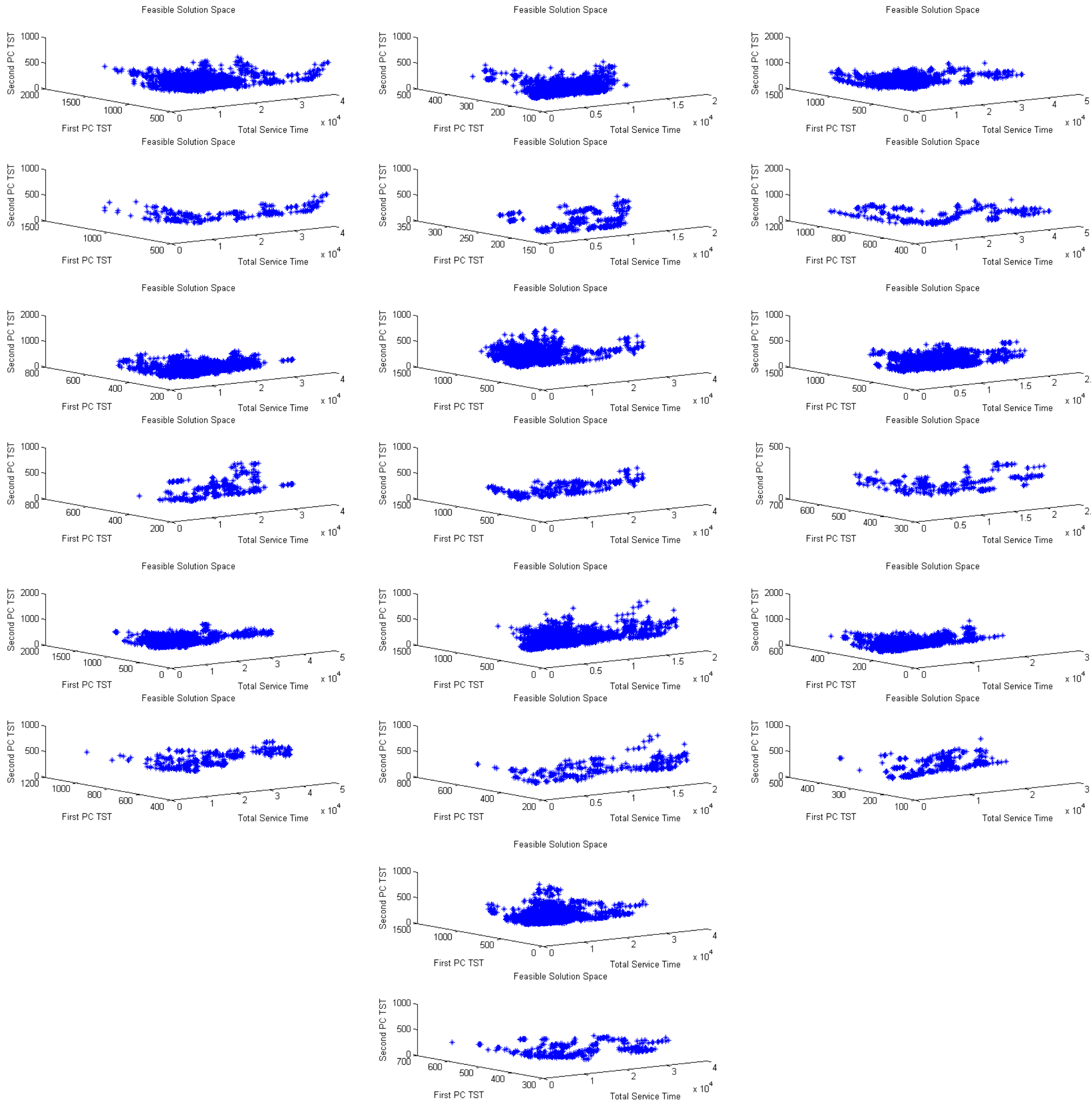


Figure 10-10 Feasible and Pareto Front for Five Berths and One Week Planning Horizon, 50% of Total Vessels Belong to Preferential Customers, Two Preferential Customers (Dataset 1 through 10)

Note: PC=Preferential Customer

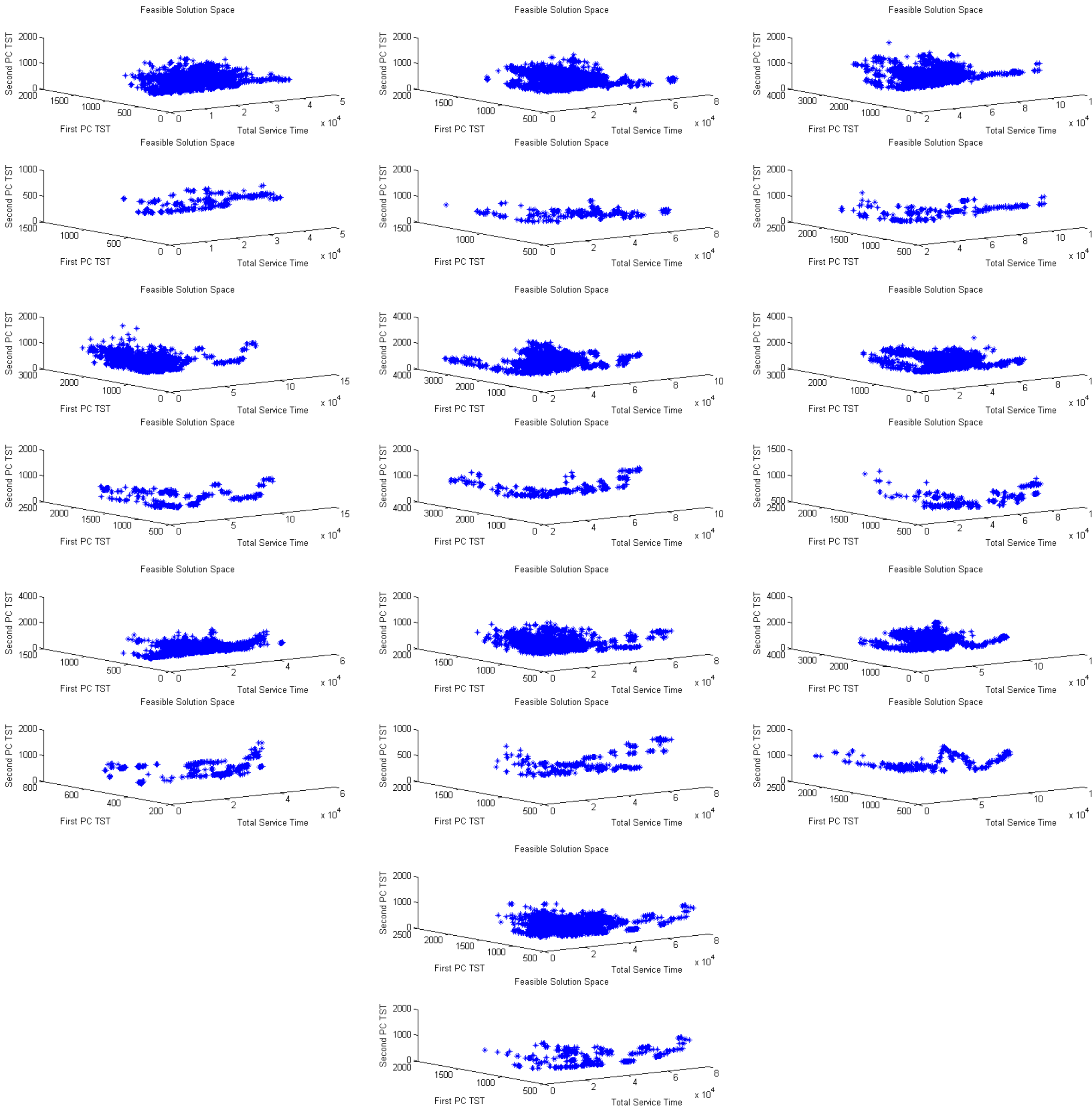


Figure 10-11 Feasible and Pareto Front for Five Berths and Two Weeks Planning Horizon, 50% of Total Vessels Belong to Preferential Customers, Two Preferential Customers (Dataset 1 through 10)

Note: PC=Preferential Customer

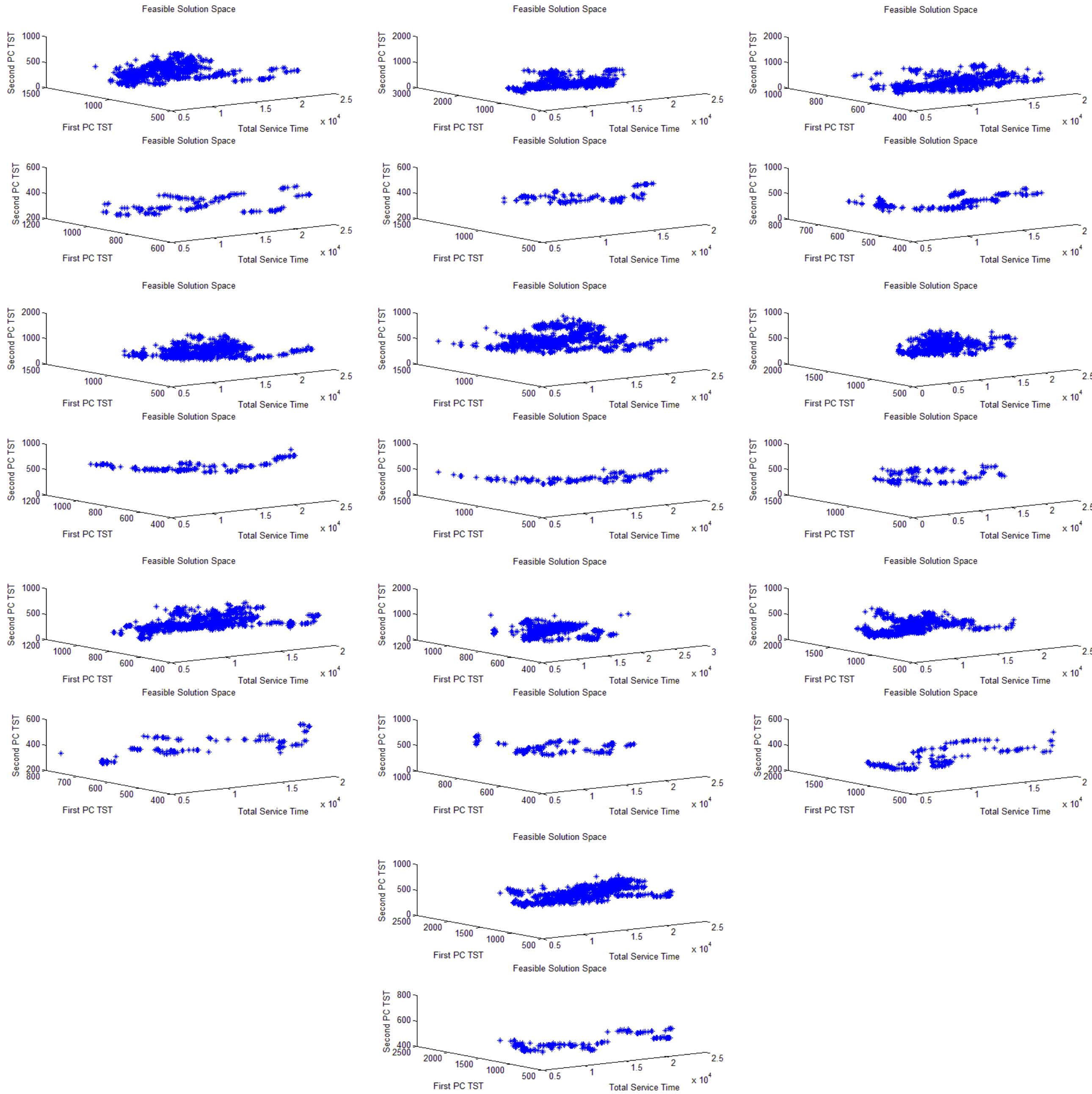


Figure 10-12 Feasible and Pareto Front for Ten Berths and One Week Planning Horizon, 50% of Total Vessels Belong to Preferential Customers, Two Preferential Customers (Dataset 1 through 10)

Note: PC=Preferential Customer

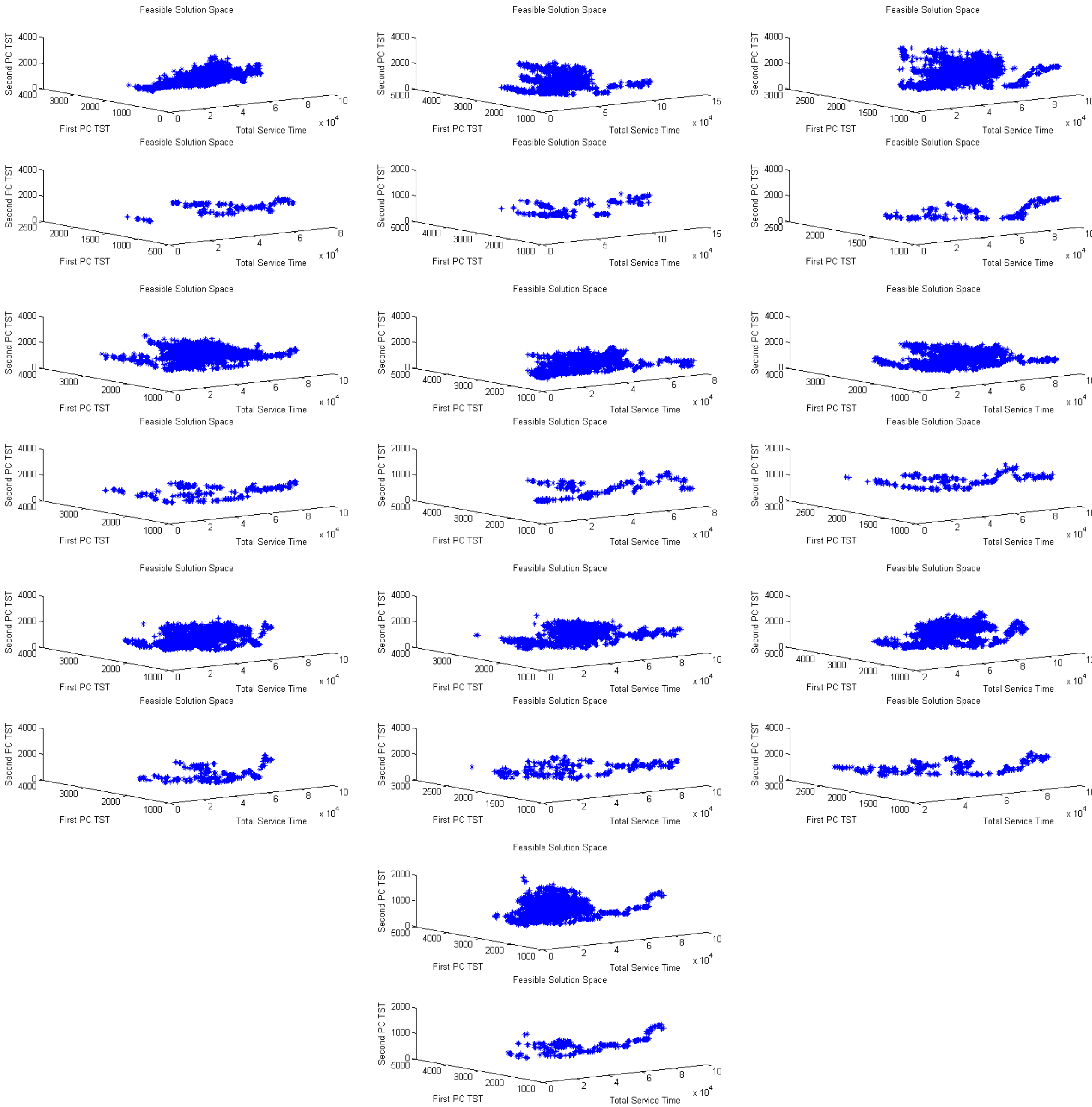
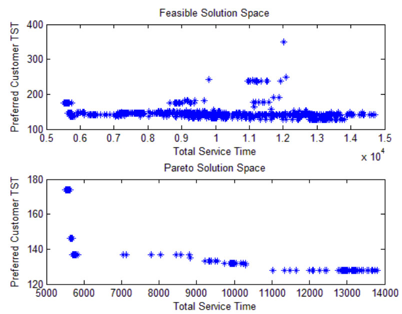
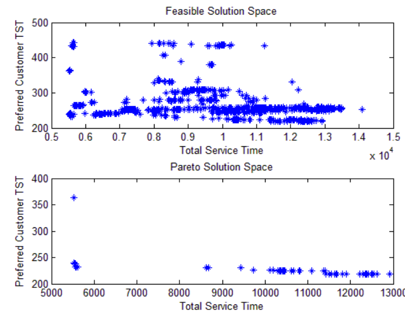


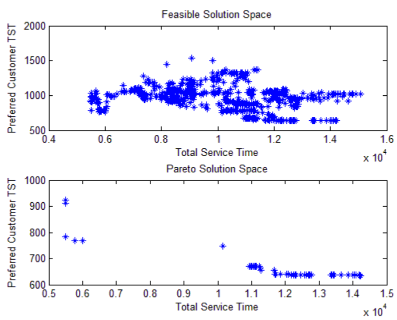
Figure 10-13 Feasible and Pareto Front for Ten Berths and Two Weeks Planning Horizon, 50% of Total Vessels Belong to Preferential Customers, Two Preferential Customers (Dataset 1 through 10)

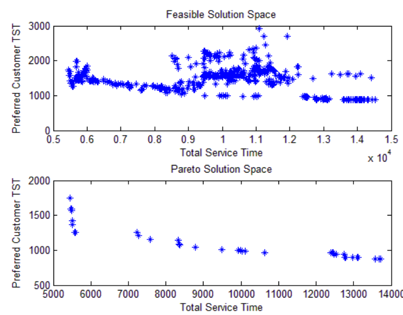
Note: PC=Preferential Customer





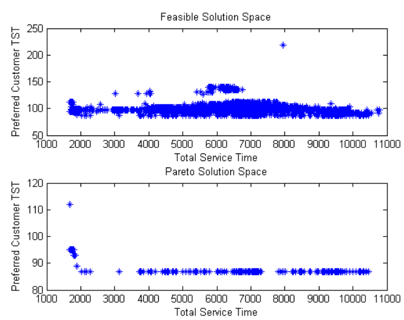
Preferential Customer Vessels=20% Preferential Customer Vessels=30%

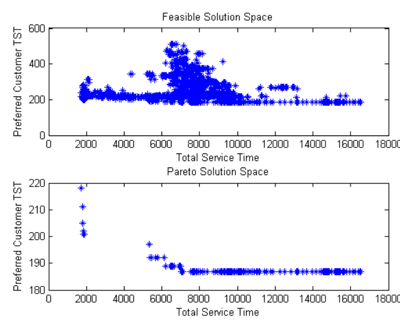




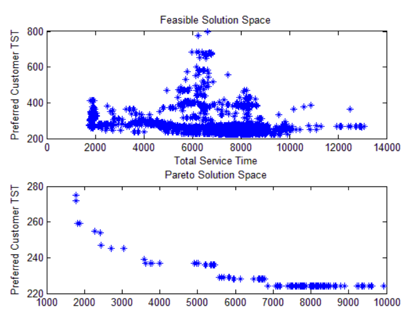
Preferential Customer Vessels=40% Preferential Customer Vessels=50%

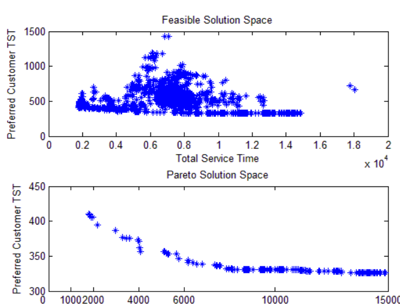
Dataset 1





Preferential Customer Vessels=20% Preferential Customer Vessels=30%

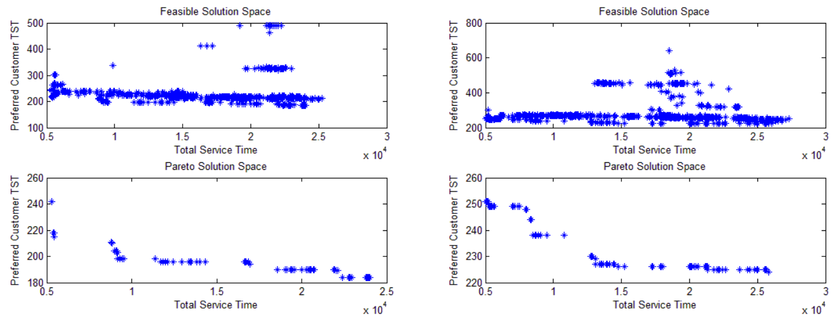




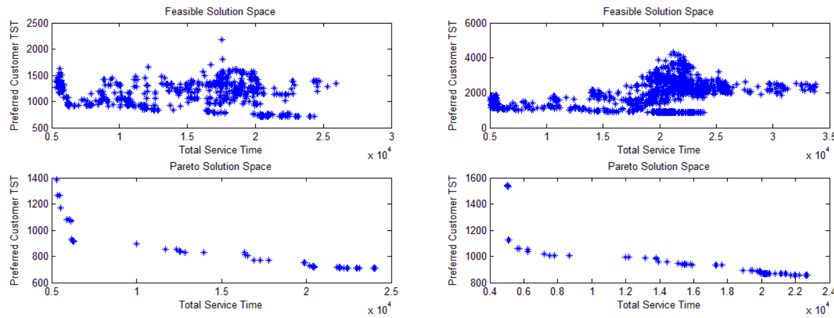
Preferential Customer Vessels=40% Preferential Customer Vessels=50%

Dataset 2

Figure 10-14 Feasible and Pareto Front for Five Berths and One Week Planning Horizon, 20% through 50% of Total Vessels Belong to Preferential Customers, One Preferential Customer (Dataset 1 and 2)

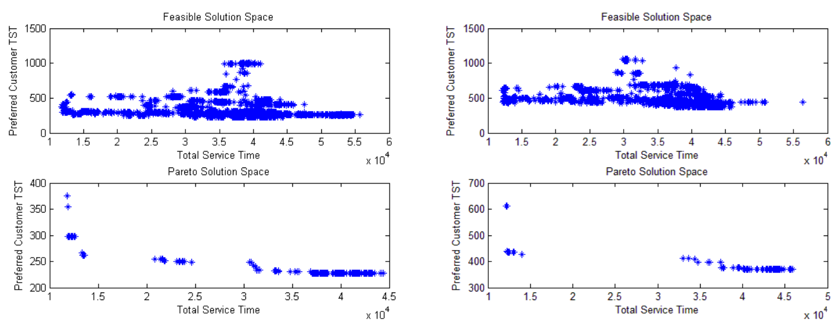


Preferential Customer Vessels=20% Preferential Customer Vessels=30%

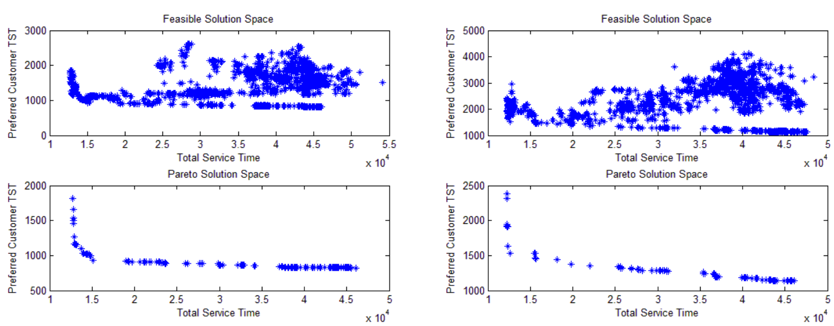


Preferential Customer Vessels=40% Preferential Customer Vessels=50%

Dataset 1



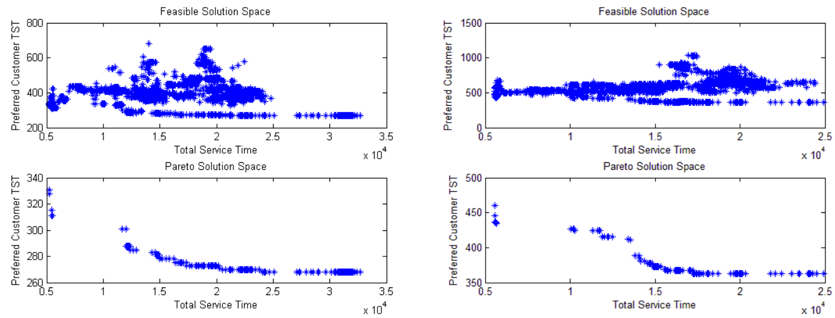
Preferential Customer Vessels=20% Preferential Customer Vessels=30%



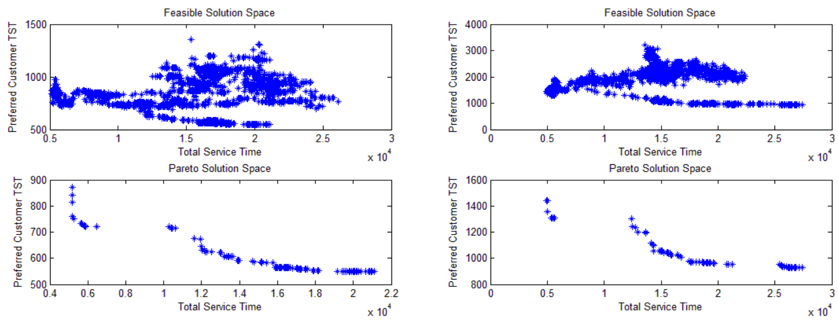
Preferential Customer Vessels=40% Preferential Customer Vessels=50%

Dataset 2

Figure 10-15 Feasible and Pareto Front for Five Berths and Two Weeks Planning Horizon, 20% through 50% of Total Vessels Belong to Preferential Customers, One Preferential Customer (Dataset 1 and 2)

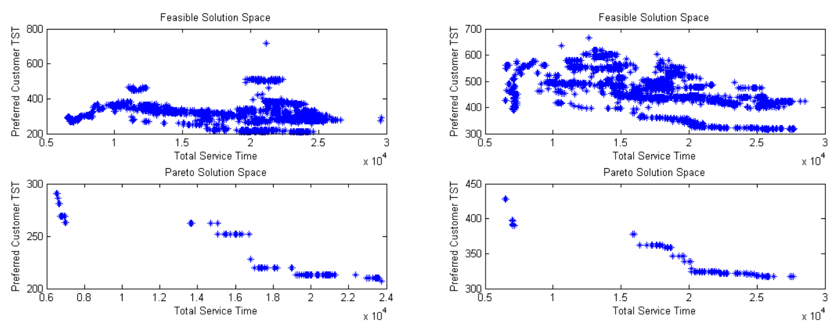


Preferential Customer Vessels=20% Preferential Customer Vessels=30%

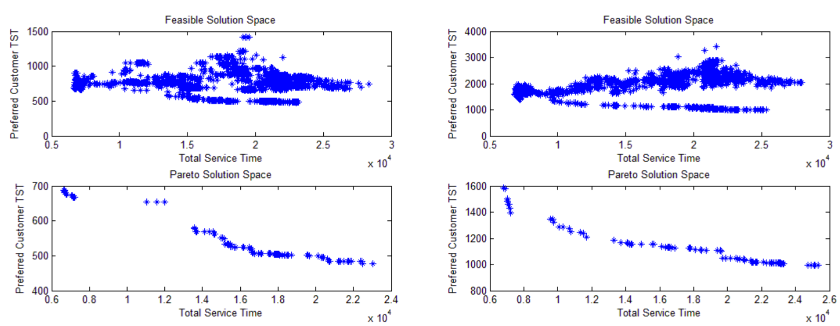


Preferential Customer Vessels=40% Preferential Customer Vessels=50%

Dataset 1



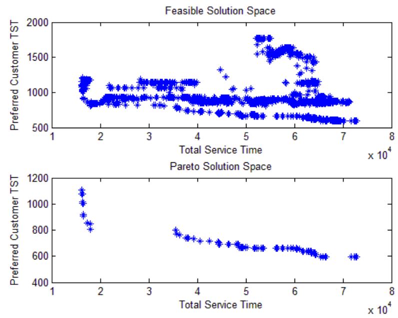
Preferential Customer Vessels=20% Preferential Customer Vessels=30%



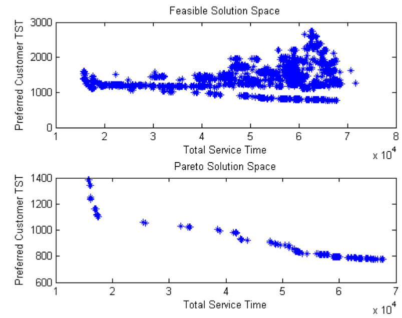
Preferential Customer Vessels=40% Preferential Customer Vessels=50%

Dataset 2

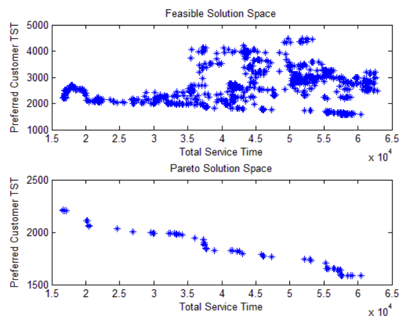
Figure 10-16 Feasible and Pareto Front for Ten Berths and One Week Planning Horizon, 20% through 50% of Total Vessels Belong to Preferential Customers, One Preferential Customer (Dataset 1 and 2)



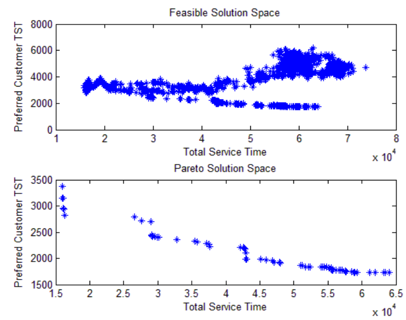
Preferential Customer Vessels=20%



Preferential Customer Vessels=30%

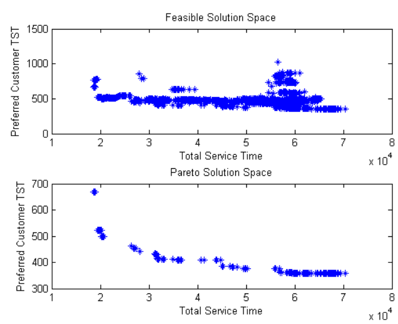


Preferential Customer Vessels=40%

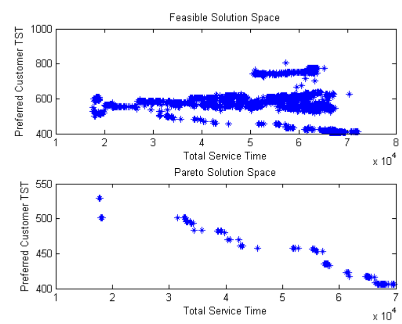


Preferential Customer Vessels=50%

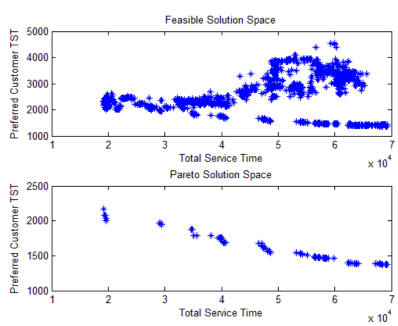
Dataset 1



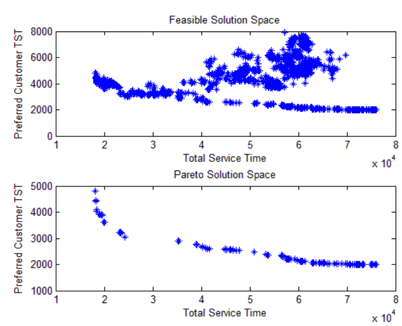
Preferential Customer Vessels=20%



Preferential Customer Vessels=30%



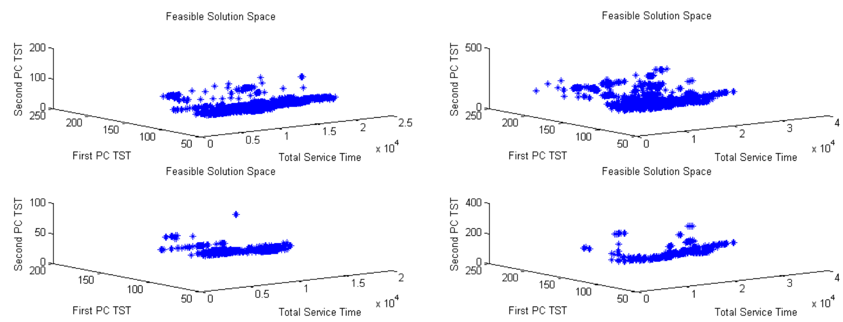
Preferential Customer Vessels=40%



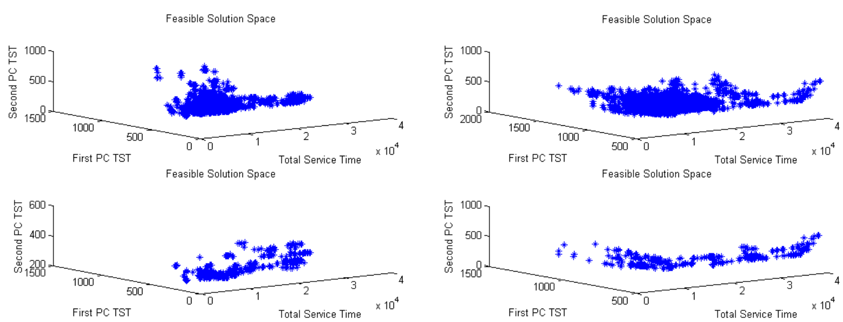
Preferential Customer Vessels=50%

Dataset 2

Figure 10-17 Feasible and Pareto Front for Ten Berths and Two Weeks Planning Horizon, 20% through 50% of Total Vessels Belong to Preferential Customers, One Preferential Customer (Dataset 1 and 2)

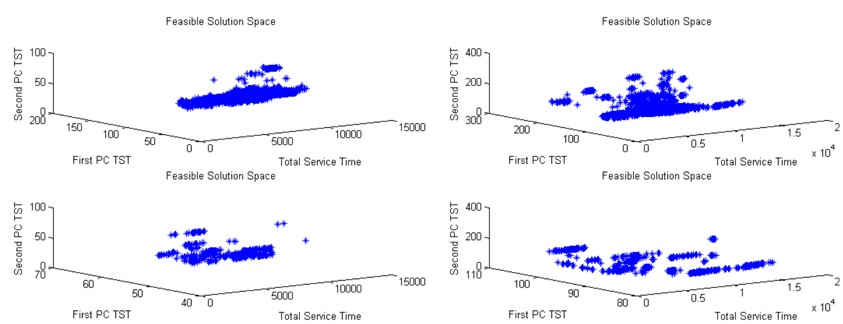


Preferential Customer Vessels=20% Preferential Customer Vessels=30%

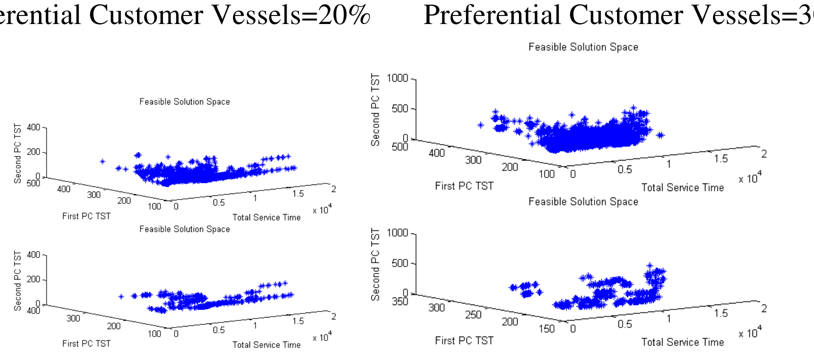


Preferential Customer Vessels=40% Preferential Customer Vessels=50%

Dataset 1



Preferential Customer Vessels=20% Preferential Customer Vessels=30%

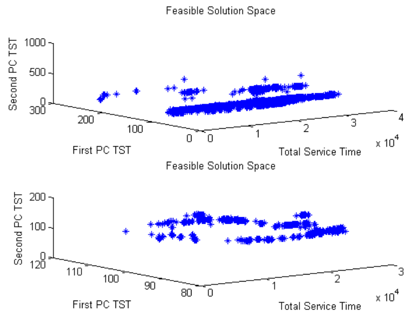


Preferential Customer Vessels=40% Preferential Customer Vessels=50%

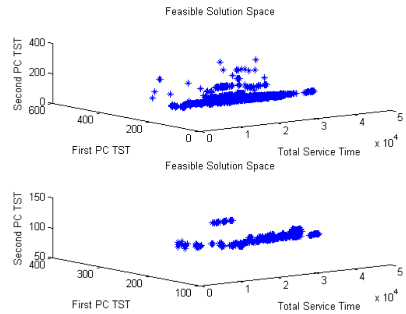
Dataset 2

Figure 10-18 Feasible and Pareto Front for Five Berths and One Week Planning Horizon, 20% through 50% of Total Vessels Belong to Preferential Customers, Two Preferential Customers (Dataset 1 and 2)

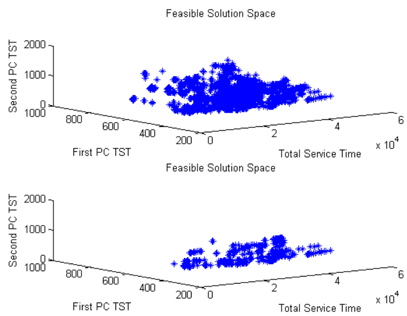
Note: PC=Preferential Customer



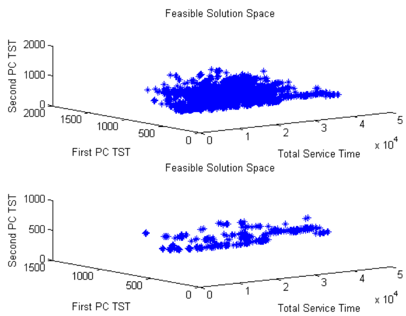
Preferential Customer Vessels=20%



Preferential Customer Vessels=30%

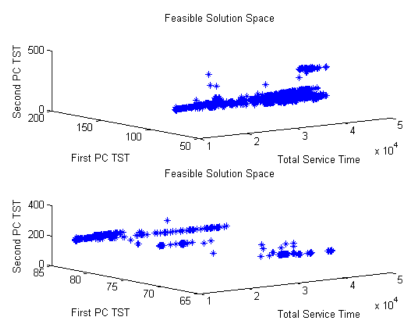


Preferential Customer Vessels=40%

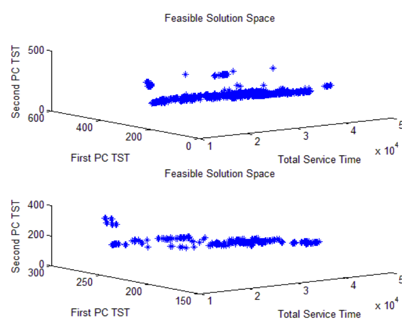


Preferential Customer Vessels=50%

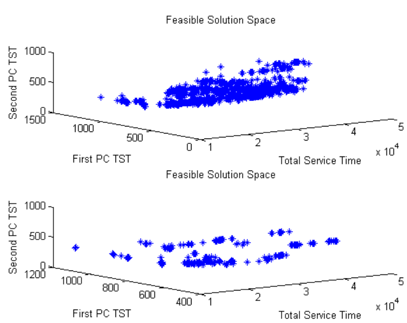
Dataset 1



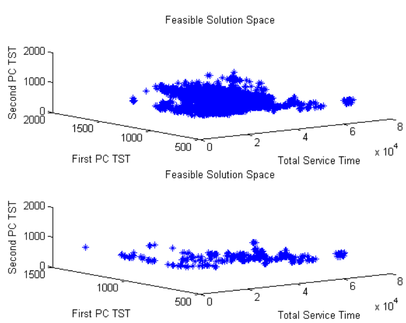
Preferential Customer Vessels=20%



Preferential Customer Vessels=30%



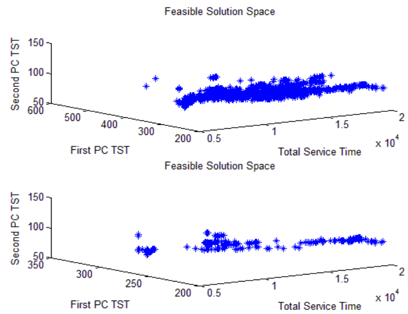
Preferential Customer Vessels=40%



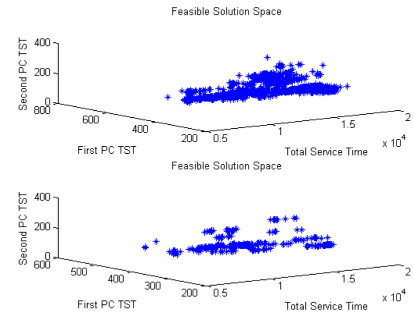
Preferential Customer Vessels=50%

Dataset 2

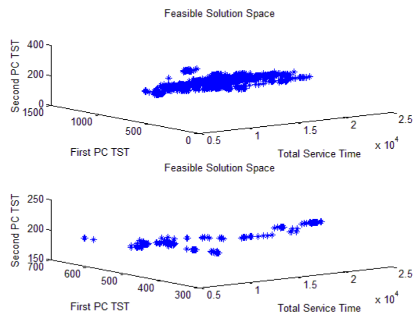
Figure 10-19 Feasible and Pareto Front for Five Berths and Two Weeks Planning Horizon, 20% through 50% of Total Vessels Belong to Preferential Customers, Two Preferential Customers (Dataset 1 and 2)
Note: PC=Preferential Customer



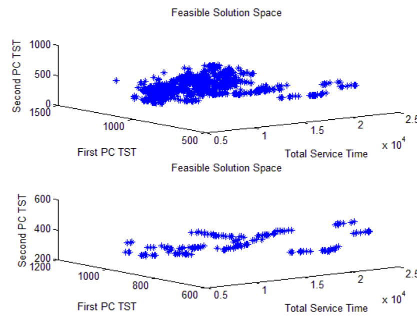
Preferential Customer Vessels=20%



Preferential Customer Vessels=30%

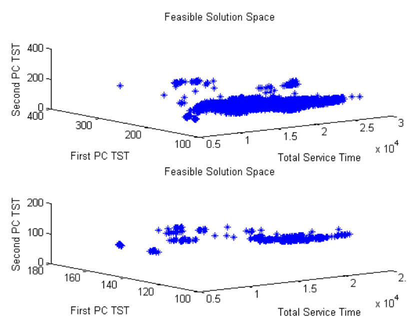


Preferential Customer Vessels=40%

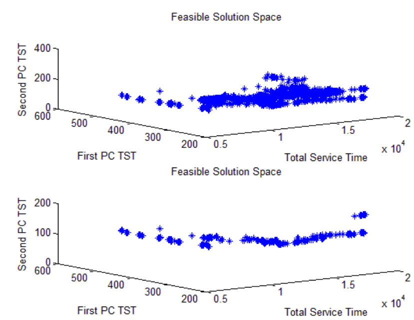


Preferential Customer Vessels=50%

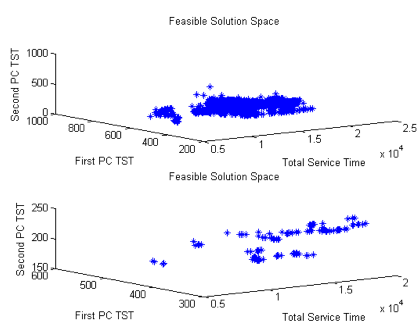
Dataset 1



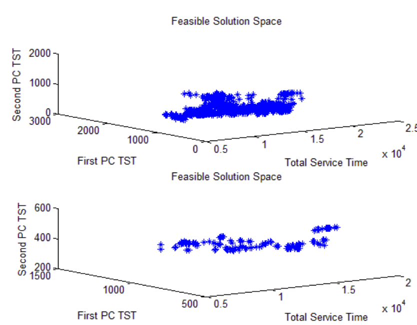
Preferential Customer Vessels=20%



Preferential Customer Vessels=30%



Preferential Customer Vessels=40%

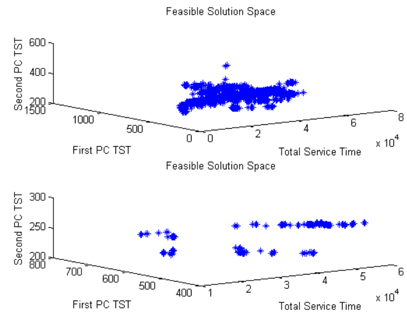


Preferential Customer Vessels=50%

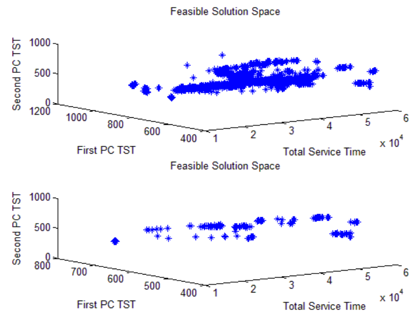
Dataset 2

Figure 10-20 Feasible and Pareto Front for Ten Berths and One Week Planning Horizon, 20% through 50% of Total Vessels Belong to Preferential Customers, Two Preferential Customers (Dataset 1 and 2)

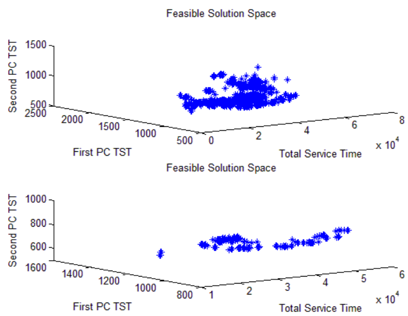
Note: PC=Preferential Customer



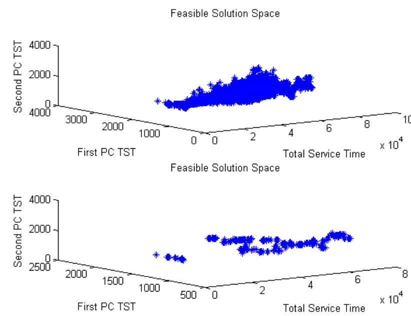
Preferential Customer Vessels=20%



Preferential Customer Vessels=30%

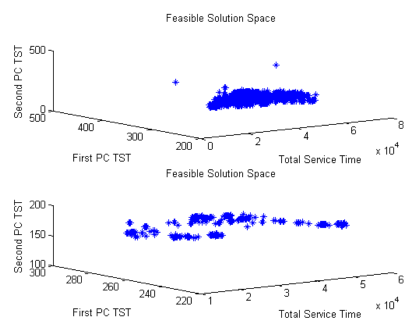


Preferential Customer Vessels=40%

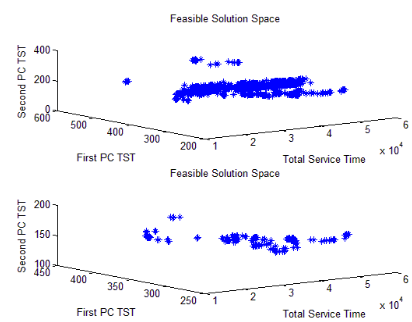


Preferential Customer Vessels=50%

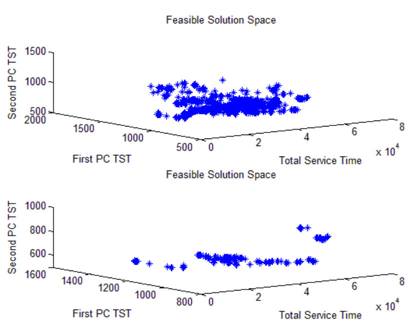
Dataset 1



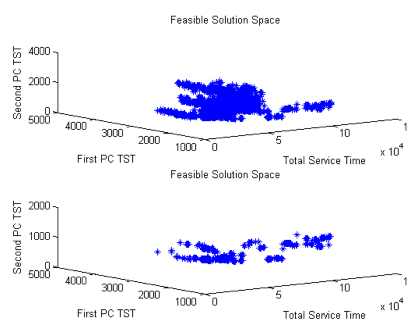
Preferential Customer Vessels=20%



Preferential Customer Vessels=30%



Preferential Customer Vessels=40%



Preferential Customer Vessels=50%

Dataset 2

Figure 10-21 Feasible and Pareto Front for Ten Berths and Two Weeks Planning Horizon, 20% through 50% of Total Vessels Belong to Preferential Customers, Two Preferential Customers (Dataset 1 and 2)
Note: PC=Preferential Customer

11 A 2-OPT BASED HEURISTIC FOR THE MULTI-OBJECTIVE BERTH SCHEDULING

Introduction

Evolutionary algorithms have been applied extensively as a solution approach to multi-objective problems. These algorithms do not guarantee optimality of the Pareto set. In this chapter we present a 2-opt based genetic algorithm heuristic that guarantees that the final solutions will belong to the true Pareto set.

Heuristic Description

Heuristic 11-1

STEP 0: Obtain the best local optimal for each objective function using the 2-opt heuristic presented in Chapter 7. Name this set of values for the objective functions *LOP*

STEP 1: Obtain an approximation of the true Pareto Set using the multi-objective GA presented in Chapter 10. Name this matrix *APS*. Create an empty matrix named *TPS*

STEP 2: Create a weight matrix *W* of equal size to *APS*. Use any type of weight formulation desired

STEP 3: Set n=1

STEP 4: Apply the 2-opt heuristic from Chapter 7, with an aggregate weighted function. As weights use the values from the n^{th} row of matrix *W*

STEP 5: Add the solution to *TPS*

STEP 6: If n<|APS|+1 set n=n+1 and go to Step 4 else go to Step 7

STEP 7: End

STEP 8: Combine *LOP*, *APS*, and *TPS* solutions to find the final Pareto Set

A graphical representation of the heuristic is shown in figure 11-1.

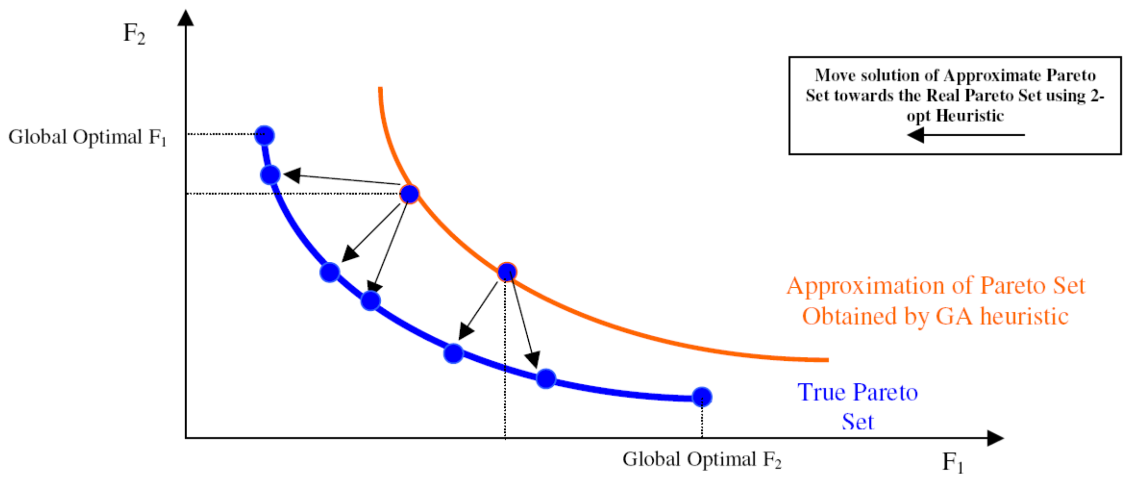


Figure 11-1. Graphical representation of heuristic

Proposition 11-I: Heuristic 11-1 will always produce solutions that belong to the true Pareto Set

Proof: Any solution obtained using a single objective formulation with an aggregate weighted objective function belongs to the true Pareto Set by definition. Thus, solutions obtained from Heuristic 11-1 will satisfy the following inequality:

$$f_1^{'} + f_2^{'} + \dots\dots\dots + f_n^{'} \leq f_1 + f_2 + \dots\dots\dots + f_n \,, \qquad \text{(Eq. 11-1)}$$

where n is the number of objective functions, f_n is the value of objective function n obtained using the procedure from Chapter 10 and $f_n^{'}$ is the value of objective function n obtained using Heuristic 11-1

12 STOCHASTIC BERTH SCHEDULING

Introduction

In this chapter we study the discrete and dynamic BAP (DDBAP) where vessel arrival and handling times are considered as stochastic variables (SDDBAP). We present and conceptually compare three different heuristic solution approaches: a) a Markov Chain Monte Carlo simulation based heuristic b) an Online Stochastic Optimization based heuristic, and c) a deterministic solution based heuristic. We also present a generic Genetic Algorithms based heuristic that will be used within the former two (a and b) heuristics. Several conclusions are drawn on the complexity of the problem and the solutions approaches and the possible benefits and drawbacks of the consideration of a stochastic environment for the DDBAP.

A conceptual formulation for the Stochastic DDBAP (SDDBAP)

In this chapter we present a conceptual formulation of the SDDBAP. The objective is to provide the port operator with a model that considers uncertainty in the vessels arrivals and handling times. Usually vessels provide the port operator with a time window in which they may arrive at the port and request service (berthing, loading/unloading, and departure). These time windows are not known with certainty until few hour of a vessels arrival. On the other hand as soon as a vessel is moored a number of quay cranes are assigned to operate on the vessel. The total handling time of a vessel is directly connected to the productivity of the quay cranes (usually measures in TEU moves per hour), which is not know with certainty and depends on a number of deterministic (relative berth position of the vessel to the storage yard, number of internal transport vehicles assigned to the vessel, etc) and stochastic (quay crane breakdowns, internal transport vehicle productivity etc) parameters and variables. For an excellent discussion on the parameters and how they affect the productivity of quay cranes (i.e. vessel loading/unloading, downtime etc) we refer to Steenken et al. (2004).

Assume that i=(1,……,I) ∈ B set of berths, j=(1,….,T) ∈ V set of vessels, A_j=Arrival time of vessel j, C_j= Handling time of vessel j at berth i. In the SDDBAP the vessels arrivals and handling times are no longer considered as deterministic problem parameters, but rather follow as random variables. For the DDBAP it is sufficient to assume a discrete distribution for both the vessel arrival and handling times. Thus we can assume that A_j and C_{ij} follows a discrete probability distribution $A_j \sim (A_{j1}, A_{j2}, \ldots, A_{jn})$, $C_{ij} \sim (C_{ij1}, C_{ij2}, \ldots, C_{ijn})$ where A_{j1} and A_{jn}, and C_{ij1} and C_{ijn}, are the upper and lower value of the expected arrival times and handling times of vessel j, and $P_{aj}=(P_{aj1}, P_{aj2}, \ldots, P_{ajn})$, $P_{cj}=(P_{cj1}, P_{cj2}, \ldots, P_{cjn})$ are the probabilities of the arrival and handling times of vessel j ($\sum_n P_{a\,jn} = 1, \sum_n P_{cjn} = 1, \forall j \in T$). The only constraint is on the form of the distributions. It must be one that we can sample from (Fishman, 2006). Without loss of generality we can assume that A_j, *and* C_{ij} are discrete distributions for all the vessels.

In the next subsection we will present the conceptual formulation for the stochastic berth allocation problem and proceed with the presentation of four solution approaches.

SDBAP Conceptual Formulation

The conceptual formulation for the SDDBAP is as follows:

[SDDBAP]:

$$\min E\{F(X_{ijk}, A_j, C_{ij})\} \quad \text{(Eq. 12-1)}$$

Subject to**:**

$$G_1(X_{ijk}) \leq m, \qquad \text{(Eq. 12-2)}$$

$$G_2(X_{ijk}) = n, \qquad \text{(Eq. 12-3)}$$

$$P\{H(X_{ijk}, A_j, C_{ij})\} \leq a \qquad \text{(Eq. 12-4)}$$

The first equation (12-1) minimizes the expected value of the objective function. The terms that the objective function will consist of depend on the port operators' objectives and can take different forms. Constraint sets (12-2) and (12-3) describe the physical properties of the problem (i.e. each vessel serviced once, each berth services one vessel at a time etc). The last sets of equations represent probability constraints that again depend on the port operators' objectives. The problem formulation leads to an integer or mixed integer stochastic program, which traditional integer-programming techniques are not suited to exploit the special problem structure and deal with the problem of dimensionality. For this purpose we propose several heuristics that can be used to solve the problem by applying different modifications of MCMC simulations on the deterministic problem. It is highly unlikely that a solution that utilizes probability densities within the optimization framework will be efficient even for toy problems.

Proposed Solution Heuristics

Markov Chain Monte Carlo Simulation Based Solution Approach

MCMC Module

STEP 0: Generate *N* numbers of unique arrival and handling times patterns by sampling from the arrival and handling time distributions

STEP 1: Solve the deterministic DDBAP using each one of the sampled arrival patterns

Heuristic 12-1

STEP 0: Obtain a set of feasible/optimal solutions using the MCMC Module

STEP 1: Set h=1

STEP 2: Update A_1 and A_2, where A_1 are the realized arrival times of vessels already in the port and A_2 are the updated future arrivals of vessels not yet in the port, at time step t_h

STEP 3: Find the *best* schedule that minimizes deviations from A_1 and A_2

STEP 4: If all vessels are in the port end else set h=h+1 and go to Step 2

The time step (t_h) increases every time new information on vessel arrivals becomes available (including the actual arrival of a vessel). Heuristic 12-1 is easily adopted and applied to any type of BSPs. The main drawback is that the full DDBAP has to be solved N times, during the MCMC module. Thus, an efficient solution approach needs to be adopted for solving the DDBAP, as Step 1 of the MCMC module. Unfortunately, there is no unified heuristic solution for the DDBAP, although a number of heuristic approaches have been presented in the literature , and the problem needs to be addressed per case.

Stochastic Online Scheduling Based Solution Approach

Heuristic 12-2

For *h=1:H (H: time steps)*

STEP 0: Select end[21] of new time period t_h (time step maybe variable)

STEP 1: Estimate S_{ih} (berth i becoming idle for time period h)

STEP 2: For vessels that have not finished service up to beginning of t_h sample arrivals and handling times and obtain a sample size of N arrival and handling patterns for each vessel

STEP 3: Select all vessels arriving before the end of t_h

STEP 4: Solve the DDBAP for each sample

STEP 5: Select the schedule that minimizes the deviation from all the feasible schedules at the current time step

STEP 6: If all vessels have been scheduled end otherwise go to Step 0

Heuristic 12-2 is based on the idea of the stochastic online scheduling by Van Hentenryck and Bent (2006) and the rolling time window heuristic for the DDBAP (Chapter 8). The main advantage of Heuristic 12-2 is that the DDBAP solution as Step 4 will consist of a much smaller instance then the original problem since not all vessels will be present at each time step. Heuristic 12-2 though provides a volatile solution, sensitive to vessel arrival and handling time changes.

Deterministic Based Solution Approach

Heuristic 12-3

STEP 0: Solve the DDBAP deterministically using the expected values of the arrival and handling times for all the vessels

[21] The beginning of the time step time period t_h *is the end of* t_{h-1}

STEP 1: When a new arrival and/or new handling times are realized re-optimize the current schedule if the new realization of arrivals and handling times are different the expected values, used to obtain initial schedule

Heuristic 3 is more efficient in terms of the computational complexity as compared to Heuristic 1 and 2 and has zero volatility. The main question is on the additional penalty that occurs when the assignment of the vessels is rescheduled, every time new vessels arrivals and/or handling times become available. This penalty occurs from the new berth availability time for the new schedule, from the vessels already in service at the time the new information becomes available.

Genetic Algorithm Solution Approach for the Stochastic BAP

Heuristic 12-4

STEP 0: Generate *N* numbers of unique arrival and handling times patterns by sampling from the arrival and handling time distributions

STEP 1: Initialize GA population to a size equal to *N*, and set *counter*=0

STEP 2: Set *counter = counter+1*

STEP 3: Apply genetic operations

STEP 4: Evaluate the genetically altered population using the *N* different instanced and select the new population

STEP 5: If *counter*< Generation Limit go to Step 2 else end.

From the four presented heuristics the GA heuristic seems as the most promising solution approach; mainly due to the multi-population attribute inherit in genetic algorithm heuristic and the autonomy from an additional heuristic to solve the DDBAP. The advantage of the multiple chromosome and simultaneous solution evaluation of a number of different input instances (different arrival and handling time patterns) in a single run reduces the computational complexity of the problem, but does not guarantee that results will be stable to the realization of the input data.

Conclusions

In this chapter we presented a conceptual formulation for the stochastic dynamic and discrete berth allocation problem. Traditional integer-programming techniques are not suited to exploit the resulting problem's structure and four heuristics were presented as possible solution approaches. The complexity of the heuristics depends on the complexity of the deterministic formulation and on the approach adopted for the stochastic sampling. Depending on the size of the problem and the complexity of the deterministic formulation one of the four heuristics may be chosen based on the time restriction needs and the distribution of the stochastic variables. Future research should focus on the comparison of the proposed heuristics on different BSPs.

13 CONCLUSIONS AND FUTURE RESEARCH

In this dissertation we presented a number of new models portraying different BSPs, in an attempt to capture the operational environment of a container terminal and include some of the attributes of the system that current models lacked. The models presented herein captured a significant portion of current container terminal operator practices while minimizing assumptions made about real world conditions. Accompanying the formulations were solutions algorithms providing feasible or optimal schedules for vessel berthing in any type of a container terminal (PT or DT). A number of computational examples were presented for a number of the proposed solutions approaches using different BSPs, while for the rest of the problems this task is left as future research. One of the main contributions of this dissertation, that should be pointed out, is that the presented models and solution algorithms are not limited to the berth allocation problem and can, with minor or no modifications, be applied to the parallel and unrelated machine-scheduling, the vehicle routing, and the facility location problem. Furthermore, the problem formulations and heuristics can be used as building blocks (lower heuristics) for constructing hyper-heuristics, an emerging technology in search and optimization with a scope to create general systems that can handle a wide range of problems.

Future research is directed in the following areas:

a. Evaluate the benefits of a berth scheduling formulation with stochastic vessel arrivals and extend to include stochastic handling times
b. Complete the coding of the two dimensional genetic algorithm heuristic proposed in Chapter 9
c. Evaluate the heuristic proposed in Chapters 8, 11 and 12
d. Evaluate and compare all the proposed algorithms on different berth scheduling policies

e. Apply and evaluate the proposed algorithms to a number of different parallel and unrelated machine-scheduling problems

f. Evaluate the applicability and performance of the algorithms in construction management problems

g. Create a software application that Implements the proposed algorithms

We believe that this dissertation has only touched upon some of the problems that port operators face when dealing with the allocation of vessels to berths. This area of research is vast and we would like to conclude by indicating a number of future research possibilities. These include but are not limited to: a) Improved algorithms for the berth scheduling problem that can guarantee local/global optimality, b) Effects of port security initiatives to berth and quay crane performance, and c) Simultaneous scheduling of internal transport vehicles, quay cranes, and berth allocation.

REFERENCES

Bent R., and Van Hentenryck P. (2004a) The Value of Consensus in Online Stochastic Scheduling. Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling

Bent R., and Van Hentenryck P. (2004b) Regrets Only! Online Stochastic Optimization Under Time Constraints. American Association for Artificial Intelligence

Bent, R., and Van Hentenryck, P. (2001). Scenario Based Planning for Partially Dynamic Vehicle Routing Problems with Stochastic Customers. (http://www.cs.brown.edu/~pvh/ORsr.ps)

Birge, J.R, Louveaux, F. (1997). Introduction to stochastic programming. New York, NY: Springer.

Briano C, Briano E., Bruzzone A. G., Revetria R. (2005) Models for Support Maritime Logistics: A Case Study for Improving Terminal Planning. 19th European Conference on Modeling and Simulation. June 1st - 4th, 2005 Riga, Latvia

Brown, G.G., Cormican, K.J., Lawphongpanich, S., and Widdis, D.B. (1997). Optimizing submarine berthing with a persistence incentive. Naval Research Logistics 44, 301-318.

Brown, G.G., Lawphongpanich, S., and Thurman, K.P. (1994). Optimizing vessel berthing. Naval Research Logistics 41, 1-15.

Chang, H.; Givan, R.; and Chong, E. On-line Scheduling Via Sampling. In AIPS'2000, 62–71.

Chang, P-C, Chen S-H and Lin, K-L. (2005) Two-phase sub population genetic algorithm for parallel machine-scheduling problem. Expert Systems with Applications, Volume 29, Issue 3, pp. 705-712

Chen Y, Narita M, Tsuji M, Sa S. (1996) A genetic algorithm approach to optimization for the radiological worker allocation problem. Health Physics, 70:180-6.

Chou, M.C., Liu, H., Queyranne, M. and Simchi-Levi, D. (2005). On the asymptotic optimality of a simple online algorithm for the stochastic single machine weighted completion time problem and its extensions. Operations Research, (to appear)

Chung-Piaw, T. Dai, J., and Moorthy, R. L. (2003) On Optimizing PSA Berth Planning System. Appearing in High Performance Computation for Engineered Systems (HPCES) (http://hdl.handle.net/1721.1/3720)

Cieniawski SE, Eheart JW, Ranjithan S. (1995) Using genetic algorithms to solve a Multi-objective groundwater monitoring problem. Water Resources Research;31, 399-409.

Cochran, K.J., Horng,S-M., Fowler J.W. (2003) A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines, Computers & Operations Research, 30, 1087-1102

Coello Coello, A.C. and Pulido, G.T. A micro-genetic algorithm for Multi-objective optimization. in Evolutionary Multi-Criterion Optimization. First International Conference, EMO 2001, 7-9 March 2001. 2001. Zurich, Switzerland: Springer-Verlag.

Coello Coello, A.C., and D. Corne, editors, Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001), volume 1993 of Lecture Notes in Computer Science, pages 213-225, Berlin, 2001. Springer-Verlag.

Coello Coello, A.C.. An updated survey of evolutionary multi-objective optimization techniques: State of the art and future trends. Congress on Evolutionary Computation 1, 3-13,Washington, D.C., July 1999.

Cohon, J.L., Multi-objective Programming and Planning, Academic Press, New York, 1978.

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

Corne, D., Jerram, N.R., Knowles, J., and Oates, J. PESA-II: Region-based Selection in Evolutionary Multi-objective Optimization. in Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001). 2001. San Francisco, California.

Corne, D.W., Knowles, J.D., and Oates, M.J. The Pareto envelope-based selection algorithm for Multi-objective optimization. in Proceedings of 6th International Conference on Parallel Problem Solving from Nature, 18-20 Sept. 2000. 2000. Paris, France: Springer-Verlag.

Czyzak P., Jaszkiewicz A. (1998) Pareto Simulated Annealing - a Metaheuristic for Multiple-Objective Combinatorial Optimization, Journal of Multi-Criteria Decision Analysis, Vol. 7, No. 1, pp. 34-47.

Czyzak, P. and Jaszkiewicz, A. (1998) Pareto simulated annealing – A metaheuristic technique for multiple-objective combinatorial optimization, Journal of Multi-Criteria Decision Analysis 7, 34-47.

Daganzo, C.F. (1989). The crane scheduling problem. Transportation Research Part B 23B (3), 159-175.

Daganzo, C.F. (1990). Crane Productivity and Vessel Delay in Ports. Transportation Research Record, 1251, 1-9.

Dai J., Lin W., Moorthy R. , Teo C-P. Berth Allocation Planning Optimization in Container Terminal. 2004. <www.bschool.nus.edu.sg/staff/bizteocp/berthplanningjuly2004.pdf >

De Castilho B, Daganzo C F (1993) Handling strategies for import containers at marine terminals. Transportation Research-B 27: 151-166

Dean C B. (2005) Approximation Algorithms for Stochastic Scheduling Problems. PhD Dissertation (www.cs.clemson.edu/~bcdean/bdean_phd_thesis.pdf)

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002) A fast and elitist Multi-objective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6(2), 182-197.

Dragović, B., Park N-K, Radmilović Z. (2006) Ship-berth link performance evaluation: simulation and analytical approaches. Maritime Policy & Management, Vol. 33, 3, pp. 281-299

Edmond E. D., and Maggs R. P., (1978). How useful are queue models in port investment decisions for container berths? Journal of the Operational Research Society Vol. 29, 741–750.

Ehrgott M., Gandibleux X. (2000) A Survey and Annotated Bibliography of Multi-objective Combinatorial Optimization, OR Spectrum, Vol. 22, No. 4, Springer, pp. 425-460.

Eiben, A.E. and Smith, J.E. Introduction to Evolutionary Computing, Springer, Berlin. 2003.

Eliiyi, D.T. and Kocer U.U. Stochastic Online Scheduling. In Proceedings: International Conference on Mathematical and Statistical Modelingin Honor of Enrique Castillo. June 28-30, 2006.

Fishman S.G. A First Course in Monte Carlo. Thomson Higher Education, CA, 2006.

Fonseca, C.M. and Fleming, P.J. Multi-objective genetic algorithms. In IEE Colloquium on `Genetic Algorithms for Control Systems Engineering' (Digest No. 1993/130), 28 May 1993. 1993. London, UK: IEE.

Forrest, S. (1993) Genetic algorithms: principles of natural selection applied to computation. Science, vol.261, p.872-878.

Fowler JW, Horng S-M, Cochran JK. (2000) A hybridized genetic algorithm to solve parallel machine problem scheduling problem with sequence dependent setups. International Journal of Manufacturing Technology and Management, 1:3-4.

Funk K.M., Cheng K.L, and Qiu D.L. (2003) The impact of terminal handling charges on overall shipping charges: an empirical study. Transportation Research Part A, 37: 703-716.

Gambardella L M, Rizzoli A E, Zaffalon M (1998) Simulation and planning of an intermodal container terminal. Simulation 71(2): 107-116.

Goldberg D.E. (1989). Genetic algorithms in search, optimization, and machine-learning. Reading, MA: Addison-Wesley.

Graham-Rowe D. (2002) Radio emerges from the electronic soup. New Scientist, Vol.175, no.2358.

Grunert da Fonseca, V., Fonseca, M.C., and Hall, O.A.. Inferential performance assessment of stochastic optimisers and the attainment function. In E. Zitzler, K. Deb, L. Thiele, C. A.

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

Guan, Y., and Cheung, K. R. (2004). The Berth Allocation Problem: Models and Solution Methods. Operations Research Spectrum, 26: 75-92, Springer-Verlag

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

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

Hansen M.P., Tabu Search for Multi-objective Optimization: MOTS, Technical Report Presented at 13th International Conference on MCDM, Technical University of Denmark, 1997

Hansen P.M and Jaszkiewicz A. Evaluating the quality of approximations of the non-dominated set. Technical report, Institute of Mathematical Modeling, Technical University of Denmark, 1998. IMM Technical Report IMMREP- 1998-7.

Hansen P.M. Tabu Search for Multi-objective Optimization: MOTS. Proceedings of the 13th International Conference on Multiple Criteria Decision Making (MCDM'97), 1997

Hansen, P., and C. Oguz. (2003) A Note on Formulations of Static and Dynamic Berth Allocation Problems. Report, Les Cahiers du GERAD.

Hansen, P., C. Oguz, and N. Mladenovic. (2007) Variable neighborhood search for minimum cost berth allocation. European Journal of Operational Research, In Press.

Haupt, R. and Haupt S-E. Practical Genetic Algorithms. John Wiley & Sons, 1998.

Hertz A., and Klober D. (2000) A Framework for the Description of Evolutionary Algorithms, European Journal of Operational Research, Vol. 126, No.1, pp.1-12.

Holland, J. (1992) Genetic algorithms. Scientific American, 267 (1), pp. 66-72.

Horn, J., Nafpliotis, N., and Goldberg, D.E. A niched Pareto genetic algorithm for Multi-objective optimization. in Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, 27-29 June 1994. 1994. Orlando, FL, USA: IEEE.

Hulten L A R (1997) Container logistics and its management. PhD thesis, Chalmers University of Technology: Department of Transportation and Logistics

Hyun C.J, Kim Y, Kin YK. (1998) A genetic algorithm for multiple objective sequencing problems in mixed model assembly. Computers & Operations Research

Imai A., H.C. Chen, E. Nishimura, and S. Papadimitriou. (2007) The simultaneous berth and quay crane allocation problem. Transportation Research Part E., In Press.

Imai A., K. Nagaiwa, and C-W. Tat. (1997) Efficient planning of berth allocation for container terminals in Asia. Journal of Advanced Transportation, Vol. 31, pp. 75–94.

Imai A., Sasaki, K., Nishimura, E., and Papadimitriou, S. (2006). Multi-objective simultaneous stowage and load planning for a container vessel with container rehandle in yard stacks. European Journal of Operational Research. (In Press).

Imai, A., E. Nishimura, and S. Papadimitriou. (2001) The dynamic berth allocation problem for a container port. Transportation Research Part B, Vol. 35, pp. 401–417.

Imai, A., Nagaiwa, K., and Chan, W.T. (1997). Efficient planning of berth allocation for container terminals in Asia. Journal of Advanced Transportation 31, 75-94.

Imai, A., Nishimura, E., and Papadimitriou, S. (2003) Berth allocation with service priority. Transportation Research Part B, Vol. 37, pp. 437-457.

Imai, A., Nishimura, E., and Papadimitriou, S. (2007) Berthing vessels at a multi-user container terminal with a limited quay capacity. Transportation Research Part E, In Press.

Imai, A., Nishimura, E., and Papadimitriou, S., (2001). The dynamic berth allocation for a container terminal. Transportation Research Part B 35, 401-417.

Imai, A., Nishimura, E., and Papadimitriou, S., (2003). Berth allocation with service priority. Transportation Research Part B 37, 437-457./

Imai, A., Nishimura, E., M. Hattori, and S. Papadimitriou. (2007) Berth allocation at indented berths for mega-container vessels. European Journal of Operations Research, Vol.179, No. 2, pp. 579-593.

Imai, A., Sun, X., Nishimura, E., and Papadimitriou, S., (2005). Berth Allocation in a Container terminal: Using Continuous Location Space Approach. Transportation Research Part B 39, 199-221.

Imai, A., X. Sun, E. Nishimura, and S. Papadimitriou. (2005) Berth Allocation in a Container Port: Using Continuous Location Space Approach. Transportation Research Part B, Vol. 39, pp. 199–221.

Jaszkiewicz A. (1997) A metaheuristic approach to multiple objective nurse scheduling. Foundations of Computing and Decision Sciences. 22,169-83.

Jaszkiewicz A. (2001) Multiple objective metaheuristic algorithms for combinatorial optimization, Habilitation thesis, 360, Poznan University of Technology, Poznan.

Jaszkiewicz A. (2002) Genetic Local Search for Multi-objective Combinatorial Optimization. European Journal of Operational Research, Vol. 137, No. 1, pp.50-71.

Jones, D.F., Mirrazavi, S.K., and Tamiz, M. (2002) Multi-objective meta-heuristics: an overview of the current state-of-the-art, European Journal of Operational Research 137(1), pp. 1-9.

Kalyanmoy Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer et al., editors, Parallel Problem Solving from Nature - PPSN VI, pages 849-858, Berlin, 2000. Springer.

Kim K, Park Y. (2003). A crane scheduling method for port container terminals. European Journal of Operations Research, Volume 156, Issue 3, pp. 752-768.

Kim, K.H., and K.C. Moon. (2003) Berth scheduling by simulated annealing. Transportation Research Part B, Vol. 37, pp. 541–560.

Kleywegt A. J., Shapiro, A., and Homem-De-Mello T. (2001) The sample average approximation method for stochastic discrete optimization. SIAM Journal of Optimization, 12, pp. 479–502.

Knowles J. and Corne D. M-PAES: A Memetic Algorithm for Multi-objective Optimization (2000) Proceedings of the 2000 Congress on Evolutionary Computation CEC00

Knowles J. D., Corne D. W, and Oates M. J. On the assessment of Multi-objective approaches to the adaptive distributed database management problem. In M. Schoenauer et al., editors, Parallel Problem Solving from Nature - PPSN VI, pages 869-878, Berlin, 2000. Springer.

Knowles J.D. and Corne D. On metrics for comparing non-dominated sets. In Congress on Evolutionary Computation (CEC 2002), pages 711-716, Piscataway, NJ, 2002. IEEE Press.

Knowles J.D. Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization. PhD thesis, Department of Computer Science, University of Reading, UK, 2002.

Knowles J.D., Corne D.W., M-PAES A Memetic Algorithm for Multi-objective Optimization, Proceedings of the 2000 Congress on Evolutionary Computation CEC 2000, pp. 325-332, 2000223-247, 2000

Lai, K.K, and K. Shih. (1992) A study of container berth allocation. Journal of Advanced Transportation, Vol. 26, pp. 45–60.

Lee D-H, Song L., and Wang H. A genetic algorithm for a bi-level programming model of berth allocation and quay crane scheduling. Proceeding of the 2006 Annual Transportation Research Board Meeting. Washington D.C., 2006.

Legato P., and Mazza R. M. (2001) Berth planning and resources optimization at a container terminal via discrete event simulation. European Journal of Operational Research 133: 537-547

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

Lim A. Rodrigues B, Xiao F, Zhu Y. (2003) Crane Scheduling with Spatial Constraints. Naval Research Logistics Quarterly Volume 51 (3)

Lim, A. (1998) The berth planning problem. Operations Research Letters, Vol. 22, pp. 105–110.

Liu J, Wan Y-W, Wang L. (2005) Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. Naval Research LogisticsVolume 53, Issue 1, pp 60 - 74

Lokuge P. and Alahakoon D. (2006) Improving the adaptability in automated vessel scheduling in container ports using intelligent software agents. European Journal of Operational Research, (In Print)

Lu, H. and Yen, G.G., (2003) Rank-density-based Multi-objective genetic algorithm and benchmark test function study, IEEE Transactions on Evolutionary Computation 7(4), 31, 325-343

Marett R, Wright M. (1996) A comparison of neighborhood search techniques for multi-objective combinatorial problems.Computers andOperations Research, 23, 465-83.

Megow N., Uetz M. and Vredeveld T., (2005). Stochastic online scheduling on parallel machines. In G. Persiano and R. Solis-Oba, eds., Lecture Notes in Computer Science, 3351:67-180.

Mehnen J. Multi-objective Evolutionary Algorithms: A Practical Presentation of MOEA with Examples from Mechanical Engineering. 4th Workshop on Quality Improvement, Bommerholz, 2005

Mitchell, M. An Introduction to Genetic Algorithms. MIT Press, 1996.

Monaco F.M. and M. Sammarra. (2007) The Berth Allocation Problem: A Strong Formulation Solved by a Lagrangean Approach. Transportation Science, Vol. 41No. 2, pp. 265-280.

Moon K.C. (2000). A Mathematical Model and a Heuristic Algorithm for Berth Planning. Unpublished thesis. Pusan National University.

Moorthy, R. and C-P. Teo. (2006) Berth management in container terminal: The template design problem. OR Spectrum, DOI 10.1007/s00291-006-0036-5.

Muller G (1995) Intermodal freight transportation, 3rd edn. Eno Foundation for Transportation, Westport, CN

Murata T, Ishibuchi H, Tanaka H. (1996) Multi-objective genetic algorithm andits application to job shop scheduling. Computers and Industrial Engineering ;30:957-968.

Nikolaou N.S. (1967) Berth planning by evaluation of congestion and cost. Journal of Waterways Highways Div. Proc. Am. Soc. Civ. Engrs. pp. 107-132.

Nishimura, E., A. Imai, and S. Papadimitriou. (2001) Berth allocation planning in the public berth system by genetic algorithms. European Journal of Operational Research, Vol. 131, pp. 282–292.

Nishimura, E., Imai, A., and Papadimitriou, S., (2001). Berth allocation planning in the public berth system by genetic algorithms. European Journal of Operational Research 131, 282-292.

Omori R, Sakakibara Y, Suzuki A. (1997) Applications of genetic algorithms to optimization problems in the solvent extraction process for spent nuclear fuel. Nuclear Technology;118:26-31.

Park K.T. and Kim, K.H.. (2002) Berth scheduling for container terminals by using a sub-gradient optimization technique. Journal of the Operational Research Society, Vol. 53, pp. 1054–1062.

Park, M.Y., and Kim, H.K.A. (2003) A scheduling method for berth and quay cranes. Operations Research Spectrum, Vol. 25, pp. 1-23.

Peter Sanders, Naveen Sivadasan, and Martin Skutella Online Scheduling with Bounded Migration. J. Diaz et al. (Eds.): ICALP 2004, LNCS 3142, pp. 1111-1122, 2004.Springer-Verlag Berlin Heidelberg 2004

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

Pinedo M., 2002. Scheduling: Theory, Algorithms, and Systems. Prentice-Hall Inc.

Pirlot, M. (1992). General Local Search Heuristics in Combinatorial Optimization: A Tutorial. Belgian Journal of operations reseach, statistics and computer science, 32 (1-2), 7-68.

Portworker Development Programme. The ILO's global training strategy for your port. 2000

Sabria F. and Daganzo F.C. (1989). Approximate Expressions for Queueing Systems with Schedule Arrivals and an Established Service Order", Transportation Science, 23(3), pp. 159-165.

Sarker, R., Liang, K.-H., and Newton, C., (2002) A new Multi-objective evolutionary algorithm, European Journal of Operational Research, 140(1), 12-23.

Sbalzarini F.I., Muller S., and Koumoutsakos P.. Multi-objective optimization using evolutionary algorithms, Online document <ctr.stanford.edu/Summer00/sbalzarini.pdf >

Schaffer J.D. (1985), Multiple objective optimization with vector evaluated genetic algorithms, in: J.J. Grefenstette (ed.), Genetic Algorithms and Their Applications: Proceedings of the Third International Conference on Genetic Algorithms, Lawrence Erlbaum, Hillsdale, NJ, 93-100

Serafini Paolo, Simulated Annealing for Multi-objective Optimization Problems, Procceedings of the 10th International Conference on Multiple Criteria Decision Making, Taipei Taiwan, pp. 87-96, 1992

Serafini, P. (1994) Simulated annealing for multi-objective optimization problems. In Multiple criteria decision making. Expand and enrich the domains of thinking and application, pp. 283--292.

Shapiro, A. (2005) On Complexity of Multistage Stochastic Programs. Online Document: {www.optimization-online.org}

Sideris, A., Boilé M.P. and Spasovic L. (2002) “Using On Line Information to Estimate Container Movements for Day-to-Day Marine Terminal Operations” Proceedings of the 81st Transportation Research Board annual meeting.

Silva et al. 2004

Srinivas, N. and Deb, K. (1995) Multi-objective Optimization Using Nondominated Sorting in Genetic Algorithms, Journal of Evolutionary Computation 2(3) 221-248.

Steenken D, Voss S., and Stahlbock R (2004) Container Terminal Operation and Operations Research- A Classification and Literature Review. OR Spectrum 26:3-49

Steenken, D., Voss, S., Stahlbock, R. (2004). Container terminal operation and operations research – a classification and literature review. OR Spectrum 26, 3–49.

Suppapitnarm A., Seffen A., Parks G.T., Clarkson P.J. (2000) A Simulated Annealing Algorithm for Multi-objective Optimisation Engineering Optimization, Vol. 33, No. 1, pp. 59-85.

Suppapitnarm, A., Seffen, K., Parks, G., and Clarkson, P. (2000) A simulated annealing algorithm for multi-objective optimization. Engineering Optimization, vol. 33, pp. 59--85.

Taboada, H. A. Multi-Objective Optimization Algorithms Considering Objective Preferences and Solution Clusters. PhD Dissertation. Rutgers University, 2007.

Taleb-Ibrahimi, M., De Castilho, B., and Daganzo F.C. (1993) Storage Space VS Handling Work in Container Terminals. Transportation Research Pt. B., 27B, No.1, 13-32.

Tan K. C., Lee T. H., and Khor E. F. Evolutionary algorithms for multi-objective optimizaton: Performance assessments and comparisons. In Proceedings of the 2001 Congress on Evolutionary Computation CEC2001, pages 979-986, COEX, World Trade Center, 159 Samseongdong, Gangnam-gu, Seoul, Korea, 27-30 May, 2001. IEEE Press.

Tong, J.C., Lau, C.H., and Lim, A. (1999). Ant Colony Optimization for the Vessel Berthing Problem. P.S. Thiagarajan, R. Yap (Eds.): ASIAN'99, LNCS 1742, pp. 359-370, Springer-Verlag Berlin Heidelberg

Ulungu E., Teghaem, J., Fortemps, P., and Tuyttens, D. (1999) MOSA method: a tool for solving multi-objective combinatorial decision problems," Journal of multi-criteria decision analysis, vol. 8, pp. 221--236.

Ulungu E.L., Teghem J. Fortemps P.H., Tuyttens D. (1999) MOSA Method: A Tool for Solving Multi-objective Combinatorial Optimization Problems, Journal of Multicriteria Decision Analysis, 8, 221-236.

United Nations Conference on Trade and Development (UNCTAD). 2004. Review of maritime transportation. Technical report, New York.

Van Hentenryck P. and Bent R. (2006) Online Stochastic Combinatorial Optimization. The MIT Press. (http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=11014)

Van Hentenryck P., Bent R., and Upfal E. (2005) Online Stochastic Optimization Under Time Constraints. Tenth International Conference on Principles and Practice of Constraint Programming Toronto Marriott Downtown Eaton Center Toronto, Canada, September 27 - October 1, 2004

Van Veldhuizen A. D. and Lamont B. G. (2000) Multi-Objective Evolutionary Algorithms: Analyzing the State of the Art. Evolutionary Computation 8(2): 125-147

Van Veldhuizen, A.D., and Lamont, B.G.. On measuring Multi-objective evolutionary algorithm performance. In A. Zalzala and R. Eberhart, editors, Congress on Evolutionary Computation (CEC 2000), volume 1, pages 204-211, Piscataway, NJ, 2000. IEEE Press.

Van Veldhuizen. A.D. Multi-objective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. PhD thesis, Graduate School of Engineering of the Air Force Institute of Technology, Air University, June 1999.

Vis I F A, de Koster R (2003) Transshipment of containers at a container terminal: an overview. European Journal of Operational Research 147: 1-16

Wang, F., and A. Lim. (2007) A stochastic beam search for the berth allocation problem. Decision Support Systems, Vol. 42 No. 4, pp. 2186-2196.

Weiss, G. and M. Pinedo (1980). Scheduling tasks with exponential service times on nonidentical processors to minimize various cost functions. Journal of Applied Probability 17, 187-202.

Yen, G.G. and Lu, H. (2003) Dynamic Multi-objective evolutionary algorithm: adaptive cellbased rank and density estimation, IEEE Transactions on Evolutionary Computation, 7(3), 253-274.

Yim SJ, Lee DY. Multiple objective scheduling for flexible manufacturing systems using petri nets and heuristic search. Proceedings IEEE International Conference on Systems, Man, and Cybernetics 1996; 4:2984-9.

Zhu Y. and Lim A. Crane Scheduling with Spatial Constraints: Mathematical Model and Solving Approaches. Eighth International Symposium on Artificial Intelligence and Mathematics Fort Lauderdale, Florida, January 4-6, 2004

Zitzler E., L. Thiele, M. Laumanns, M.C. Fonseca, and V. Grunert da Fonseca. Performance assessment of multi-objective optimizers: An analysis and review. TIK-Report No. 139. Institut for Technische Informatik and Kommunikationsnetze, ETH Zurich, Switzerland, 2002.

Zitzler E.and Thiele L. Multi-objective optimization using evolutionary algorithms - a comparative case study. In Agoston E. Eiben, Thomas B¨ack, Marc Schoenauer, and Hans-Paul Schwefel, editors, Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V), pages 292-301, Berlin, Germany, 1998. Springer.

Zitzler, E. and Thiele, L. (1999) Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Transactions on Evolutionary Computation 3(4), pp. 257-271.

Zitzler, E., Laumanns, M., and Thiele, L., SPEA2: Improving the strength Pareto evolutionary algorithm. Swiss Federal Institute Techonology: Zurich, Switzerland. (2001).

APPENDIX A

Dissertation Bibliography

Published

Theofanis S., Boilé M., Golias M. *An Optimisation Based Genetic Algorithm Heuristic for the Berth Allocation Problem*. IEEE Conference on Evolutionary Computation, Singapore, 2007.

Theofanis, S., Boilé, M. and Golias, M. *Continuous and Dynamic Berth Allocation with Simultaneous Quay Crane Scheduling*. ModSim World, September 10-13, Virginia Beach Convention Center, 2007.

Theofanis S., Boilé M., Golias M. *Berth and Quay Crane Scheduling A Formulation Reflecting Service Deadlines and Productivity Agreements*. Second Transtec Conference, Prague, 13-15 September, 2007.

Golias M., Boilé M., Theofanis S. *The Stochastic Berth Allocation Problem*. Second Transtec Conference, Prague, 2007.

Boilé M., Theofanis S., Golias M., and Coit D. *Berth Planning By Customer Differentiation A Multi-Objective Approach.* 11th World Conference on Transportation Research, Berkeley, 2007.

Boilé M., Theofanis S., Golias M. *The Berth Allocation Problem A Formulation Reflecting Service Priority Arrangements With Time Windows.* 2nd International Conference on Transportation Logistics, Shenzhen, P. R. China, 2007.

Theofanis S., Boilé M., Golias M. *Container Terminal Seaside Operations Scheduling Does Current Research Address The Practical Challenges of Berth Planning? – A Critical Review.* 48th Transportation Research Forum Annual Meeting, Boston, Massachusetts, 2007.

Golias M., Boilé M., and Theofanis S. *The Berth Allocation Problem A Formulation Reflecting Service Time Window Service Deadlines*. 48th Transportation Research Forum Annual Meeting, Boston, Massachusetts, 2007.

Golias M., Boile M., Theofanis S. The Berth Allocation Problem A Formulation Reflecting Service Delay Penalties and Early Premiums. 86th Transportation Research Board, Washington D.C., 2007.

Boilé M., Theofanis S., Golias M. *Berth Allocation with Service Priorities A Linear Model Formulation.* 5th WSEAS/IASME International Conference on System Science and Simulation in Engineering (ICOSSSE 2006), Tenerife, Canary Islands, Spain, 2006.

Under Review

Boile M., Theofanis S., Golias M. *A 2-opt based heuristic for the minimum service and minimum weighted service time berth allocation problem.* 10th International Conference on Applications of Advanced Technologies in Transportation, 2008.

Golias M., Theofanis S., Boile M., Taboada H.A *A non-numerical ranking preferences approach for the discrete and dynamic multi-objective berth allocation problem.* 87th Transportation Research Forum Annual Conference, Washington D.C., 2008.

Boile M., Theofanis S., Golias M. *The Minimum Cost and Service Time Berth Allocation Problem*. 87th Transportation Research Forum Annual Conference, Washington D.C., 2008.

Golias M., Boile M., and Theofanis S. (2007) Berth Planning by Customer Service Differentiation A Multi-Objective Approach. *Transportation Research Part E.*

Golias M., Theofanis S., Boile M. *Berth and Quay Crane Scheduling A Formulation Reflecting Start and Finish of Service Deadlines and Productivity Agreements*. 2nd Annual National Urban Freight Conference, Long Beach, CA, 2007.

Theofanis S., Golias M., Boile M. *An Adaptive Local Search Heuristic for the Discrete Berth Allocation Problem*. 2nd Annual National Urban Freight Conference, Long Beach, CA, 2007.

CURRICULUM VITA

Michail D. Gkolias

Colleges Attended

August 2004 - October 2007	Ph.D. in Civil and Environmental Engineering. Rutgers, The State University of New Jersey
August 2002 - October 2004	M.Sc. in Civil and Environmental Engineering. Rutgers, The State University of New Jersey
August 2002 - January 2004	Graduate Diploma in Transportation Studies. Rutgers, The State University of New Jersey
October 1996 - October 2001	Diploma in Civil and Environmental Engineering. Aristotle University of Thessaloniki

Positions Held

August 2002-August 2007	Graduate Research Assistant, Rutgers, The State University of New Jersey
May 2004-December 2005	Transportation Engineer, The Center for Advanced Infrastructure & Transportation (CAIT), Rutgers, The State University of New Jersey
January 2002 - May 2002	Civil Engineer, Tsompanoglou S.A Civil Engineering Firm, Thessaloniki, Greece