

Evolutionary Optimisation Approach for the Single and Multiple-Port Berth Allocation and Quay Crane Assignment Problem



By

Tamer El-Boghdadly

The thesis is submitted in partial fulfilment of the requirement

for the award of the degree of

Doctor of Philosophy

School of Computing, Faculty of Technology

University of Portsmouth

United Kingdom

2018

“Research is to see what everybody else has seen
and to think what nobody else has thought”

Albert Szent-Gyorgyi (1937 Nobel Prize)

COPYRIGHT

Copyright © 2018 Tamer El-Boghdadly. All rights reserved.

The copyright of this thesis rests with the Author. Copies (by any means) either in full, or of extracts, may not be made without prior written consent from the Author.

Abstract

Container terminals play an indispensable role in loading/unloading containers from/to container vessels, since more than 80% of international trade volume in goods is transported by sea. This research aims to improve container terminal operational efficiency, particularly in the seaside area. Three major problems occur in this area during the planning and scheduling of incoming vessels. The first problem is the Berth Allocation Problem (BAP), which is associated with allocating berthing space and time to vessels. The second is the Quay Crane Assignment Problem (QCAP), which is on assigning a number of cranes (QCs) to vessels such that all required transshipments of containers can be fulfilled. The third is the Quay Crane Scheduling Problem (QCSP), which is on scheduling the sequence transshipment operations of the assigned QCs to the appropriate vessels. These problems can be solved independently. However, the first two problems are dependent on one another. The number of available QCs depends on when and where the vessel is berthed, and the berthing handling time varies depending on the number of QCs that can be or are assigned.

This research addresses the integrated Berth Allocation and Quay Crane Assignment Problem (BACAP). The research has developed three models. The first model solves the BACAP with a large-scale benchmark. The second model solves the BACAP with the desired berthing position. These models consider the setting to be a single port with a single terminal. The third model solves the BACAP with desired berthing position by considering there to be multiple ports with multiple terminals. The common objectives are to minimise the total service time for the vessels (waiting time plus handling time), to minimise the total terminal costs and to maximise the utilisation of the quay cranes. The BACAP has been proven to be an NP-hard problem. Consequently, the large-scale instances are difficult to solve using exact methods. Therefore, heuristic methods are essential to solve this type of problem in an acceptable computational time.

This research provides a novel self-adaptive constructive meta-heuristic algorithm using genetic programming (GP) and a genetic algorithm (GA) for solving the continuous dynamic BACAP. The vessel priority rule has a crucial impact on the scheduling processes in order to achieve the vessel operators and terminal operators' goals. As a result, this research uses GP to evolve effective and robust composite dispatching rules to select the priority of the incoming vessels automatically with regard to the problem's constraints and berth layout. The outcome will be a solver for the BACAP rather than a solution that can find an optimal or near optimal solution.

The research solves the problem with different constraints and well-known benchmarks for large and small-scale instances. Comparative studies based on extensive computational experiments of the model and the literature were performed to verify its performance. Furthermore, the research extends the current state-of-the-art by innovating a new mathematical model to integrate the operational planning levels and solve the BACAP with multiple ports, in which one terminal operator owns multiple ports with multiple terminals and the incoming vessels have no restriction concerning berthing in any of them.

The computational results of all proposed models show high performance when it comes to solving the BACAP in both single and multiple ports. The results indicate that the frameworks are quite competitive with other techniques and outperform other heuristic-based frameworks on many occasions. Finally, the research presents future work areas and proposes an approach to unify the literature benchmarks and to develop a dataset generator.

*For my beloved parents,
Mohammed & Amina.*

*For my children,
Mohanned & Hossam.
I miss you so much.*

*For my adored wife,
Yasmin.*

Acknowledgements

Thanks to God Almighty for giving me the strength, knowledge, ability and opportunity to undertake this research and to complete it satisfactorily.

This research would never have been completed without the support of many people who have enriched me through their wisdom, friendship and love.

I would like to express my sincerest thanks to my supervisors, Dr Mohamed Bader-El-Den, Prof. Dylan Jones, and Dr Alex Gegov, for their support, guidance, and advice. I would also like to thank all of my colleagues, friends and the staff at the University of Portsmouth who helped me and advised me during my studies.

Special thanks go to my beloved wife Yasmin for her endless support, encouragement and understanding, and for putting up with me through the toughest moments of my life. This work would never have been done without you in my life. May God bless you, and I wish for all of our other future dreams to come true.

I would also like to extend my appreciation to my colleagues at the “Arab Academy for Science, Technology and Maritime Transport” and the colleagues of the “Mediterranean Memorandum of Understanding on Port State Control” for their continuous support over the past five years. I would especially like to express my gratitude to my previous manager, Cap. Emad Islam. I have no words to thank you for your kindness and for putting up with me during my PhD period. You were always there for me when I needed you.

My acknowledgement would be incomplete without thanking and sending love to my parents and family for their almost unbelievable support; they encouraged, funded and helped me at every stage of my personal and academic life, and longed to see this achievement come true. They are the most important people in my world, and I dedicate this thesis to them.

Declaration

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.

Word Count: 46,061 words

Tamer Elboghdadly

November 2018

Table of Contents

COPYRIGHT	iii
Abstract	iv
Acknowledgements	vi
Declaration	vii
Table of Contents	viii
List of Tables	xiii
List of Figures	xiv
Abbreviation	xvii
Dissemination	xviii
Chapter 1 : Introduction.....	1
1.1 Motivation and Scope of Research.....	1
1.2 Research Aim and Objectives.....	4
1.2.1 Aim	4
1.2.2 Objectives.....	4
1.3 Research Methodology.....	5
1.4 Research Framework.....	6
1.5 Research Structure and Thesis Organisation.....	8
Chapter 2 : Container Terminal Management.....	10
2.1 Introduction.....	10
2.2 Container Terminal Overview	12
2.3 Container Terminal: Systems and Processes	15
2.3.1 Seaside Area	17
2.3.1.1 Arrival of vessels	17
2.3.1.2 Unloading and loading containers	18
2.3.2 Transport Area	19
2.3.2.1 Transfer containers from vessel to stack and vice versa	19
2.3.3 Yard Area.....	20
2.3.3.1 Stacking of containers.....	20
2.3.4 Landside Area	21
2.3.4.1 Inter-terminal transport and other modes of transportation	21
2.4 Container Terminal: Planning Levels and Problems	21

2.4.1	Strategic planning.....	24
2.4.2	Tactical Planning.....	24
2.4.3	Operational Planning.....	25
2.4.3.1	Seaside Operational Problems.....	27
2.4.3.1.1	Berth Allocation Problem (BAP)	27
2.4.3.1.2	Quay Crane Assignment Problem (QCAP)	29
2.4.3.1.3	Quay Crane Scheduling Problem (QCSP).....	30
2.4.3.1.4	Stowage Planning Problem (SPP)	31
2.4.3.2	Seaside operational Problems' constraints	32
2.5	Conclusions.....	36
Chapter 3 : Literature Review.....		37
3.1	Overview	37
3.2	Related Work on Seaside Operational Problems	39
3.2.1	BAP Literature Review.....	39
3.2.1.1	Discrete layout	40
3.2.1.2	Continuous layout.....	41
3.2.1.3	Hybrid layout	42
3.2.2	QCAP Literature Review	43
3.2.3	QCSP Literature Review	44
3.3	Related Work on the Integrated BAP and QCAP (BACAP).....	45
3.3.1	Single Port BACAP	45
3.3.1.1	Benchmarks	48
3.3.1.2	Optimisation Methods.....	50
3.3.1.2.1	Exact methods.....	50
3.3.1.2.2	Approximation method.....	50
3.3.2	Multiple Ports BACAP.....	52
3.4	Research Gaps and Conclusion	56
Chapter 4 : A Genetic Programming Algorithm for the Berth and Quay Crane Allocation Problem		58
4.1	Introduction.....	58
4.2	Problem Description.....	60
4.3	Mathematical Model Formulation	61
4.3.1	Model Assumption	62
4.3.2	Notations.....	63

4.3.3	Mathematical Model.....	64
4.4	Dispatching rules.....	64
4.5	Genetic Programming for BACAP (<i>BACAP_GP</i>)	68
4.5.1	<i>BACAP_GP</i> Framework.....	69
4.5.1.1	<i>BACAP_GP</i> Representation	71
4.5.1.2	Generate a set of initial solutions	73
4.5.1.3	Fitness function.....	73
4.5.1.4	Crossover Operation	73
4.5.1.5	Mutation Operation.....	74
4.5.2	BACAP Scheduler Algorithm (<i>BACAP_Scheduler</i>).....	74
4.5.2.1	Numerical example	75
4.6	Computational Experiments and Analysis	77
4.6.1	Datasets and parameter settings	77
4.6.2	Results of SPRs and <i>BACAP_GP</i>	79
4.7	Conclusion	87

Chapter 5 : Enriched meta-heuristic approach and cost-based model for the

BACAP with desired berthing position.....		89
5.1	Introduction.....	89
5.2	Real BACAP problem discussion.....	90
5.2.1	DP World Sokhna General Information.....	90
5.2.2	DP World Sokhna Study	92
5.2.3	Sensitivity Analysis	94
5.3	Problem Description.....	96
5.4	Mathematical Model Formulation	98
5.4.1	Model Assumption	99
5.4.2	Notations.....	100
5.4.3	Mathematical Model.....	101
5.5	Genetic Programming for the BACAP with desired berthing position	
(BACAP_GP_DP).....		103
5.5.1	BACAP_GP_DP Framework	103
5.5.1.1	BACAP_GP_DP Representation.....	104
5.5.1.2	Generate a set of initial solutions	106
5.5.1.3	Fitness function.....	106
5.5.1.4	Crossover & Mutation Operation	106

5.5.2	<i>BACAP_Scheduler_DP</i>	106
5.5.3	Numerical Example	110
5.6	Computational Experiments and Analysis	114
5.6.1	Datasets and parameter settings	114
5.6.2	Results of SPRs and the <i>BACAP_GP_DP</i>	115
5.7	Conclusion	121
Chapter 6 : Berth and Quay Crane Optimisation Model with Multiple Ports..		122
6.1	Introduction.....	122
6.2	Multiple Ports Operations	123
6.3	Problem Description.....	125
6.4	Mathematical Model Formulation	127
6.4.1	Model Assumptions	127
6.4.2	Notations.....	128
6.4.3	Mathematical Model.....	129
6.4.4	Multiple Ports BACAP Plan Framework.....	131
6.5	Genetic Algorithm for Multiple Ports BACAP (MultiP_BACAP_GA)	134
6.5.1	MultiP_BACAP_GA Framework.....	136
6.5.2	Chromosome Representation	138
6.5.3	Generate a set of initial solutions	138
6.5.4	Fitness function	138
6.5.5	Crossover Operation	139
6.5.6	Mutation Operation	139
6.6	Computational Experiments and Analysis	140
6.6.1	Datasets and parameter settings	140
6.6.1.1	Datasets and instances	140
6.6.1.2	Parameters.....	141
6.6.2	Results of Multi-Port BACAP, CPLEX and GA comparison	141
6.6.2.1	Performance Analysis	141
6.6.2.2	Evolution Analysis	149
6.6.2.3	Heuristic Analysis	152
6.7	Conclusions.....	154
Chapter 7 : Conclusions and Future Work.....		155
7.1	Research Summary and Conclusions	155

7.2	Research Contributions	158
7.3	Future Work	159
7.3.1	Framework Scope and Functionality.....	160
7.3.2	Related to Literature Benchmarks	160
7.3.3	Single/Multi-port Dataset Generator.....	161
	References	163
	Appendix A.....	177
A.1	Sample of the instance produced by Mario Rodriguez-Molins, Ingolotti, et al., 2014 to solve BACAP using BACAP_GP.....	177
	Appendix B.....	179
B.1	DP World Sokhna Vessel Map (Baplie file).....	179
B.2	Sample of CDR created by BACAP_GP_DP to solve instance number 1	180
	Appendix C.....	181
C.1	Example of multiple ports BACAP in chapter 6	181
C.2	CPLEX code for Multiple Ports BACAP	184
	Appendix D.....	190
D.1	Ethical Statement.....	190

List of Tables

Table 2-1: Top 20 busiest container ports in the world as of 2016, (The Journal of Commerce, 2016).	11
Table 3-1: Research papers published on the BACAP from 2000 to 2018.	47
Table 3-2: Summary of the BACAP literature benchmarks.	49
Table 4-1: Overview of research applied Dispatching Rule in seaside container terminal problems.....	66
Table 4-2: Terminal set for the <i>BACAP_GP</i> tree representation.	71
Table 4-3: Function set for the <i>BACAP_GP</i> tree representation.....	72
Table 4-4: <i>BACAP_GP</i> Parameters.....	78
Table 4-5: Results of SPRs to solve BACAP.	80
Table 4-6: Comparing the best T_s results from SPR with <i>BACAP_GP</i>	82
Table 5-1: Terminal set for the <i>BACAP_GP_DP</i> tree representation.....	105
Table 5-2: Function set for the <i>BACAP_GP_DP</i> tree representation.	106
Table 5-3: A sample of instance with 20 vessels produced by (Meisel & Bierwirth, 2009) and solved using <i>BACAP_GP_DP</i>	110
Table 5-4: <i>BACAP_GP_DP</i> Parameters	115
Table 5-5: Performance comparison between the dispatching rules (SPRs)	117
Table 5-6: Performance comparison between the evolved GP scheduling and the other heuristics.	120
Table 6-1: Vessel-Quay Matrix cost list	133
Table 6-2: Multi-Port BACAP, GA and CPLEX comparison (2 Quays)	143
Table 6-3: Multi-Port BACAP, GA and CPLEX comparison (3 Quays)	145
Table 6-4: Proposed model results, assuming random costs	148
Table 6-5: Results of 100 vessels with 5 Quays.	149

List of Figures

Figure 1-1: Global containerised trade, 1996-2016 (Millions of TEUs and percentage annual change), (UNCTAD, 2016).....	2
Figure 1-2: Research Framework.....	7
Figure 1-3: Research Structure and Flow.....	8
Figure 2-1: TCB Barcelona container terminal photo APM Terminals. Source (Fonseca, 2016).....	13
Figure 2-2: Quay Crane. Source (Kalmar global, 2018).....	13
Figure 2-3: The 20 feet Container (TEU)	13
Figure 2-4: Maximum container-vessel capacity from 1956 to 2017, TEU. Source (Saxon & Stone, 2017).....	14
Figure 2-5: A container vessel layout. Source (Kim, Kang, & Ryu, 2004)	15
Figure 2-6: Container Terminal Operation Scenario (Cordeau et al., 2005).	16
Figure 2-7: Container terminal system for transshipment process.	17
Figure 2-8: (a) Yard truck. (b) Forklift trucks. (c) Straddle carriers. (d) AGV. Source (Kalmar global, 2018)	19
Figure 2-9: (a) RMGC. (b) RTGC. Source (Kalmar global, 2018).....	21
Figure 2-10: Decision problems in a container terminal. Source (C. Iris et al., 2016).	23
Figure 2-11: Seaside operational planning tasks at a container terminal. Source (Meisel & Bierwirth, 2013).....	31
Figure 2-12: Static arrival time	34
Figure 2-13: Dynamic arrival time.....	34
Figure 2-14: Types of berth layouts. (a) Discrete layout. (b) Continuous layout. (c, d) Hybrid layout. (Bierwirth & Meisel, 2010)	34
Figure 3-1: BAP classification scheme, (Bierwirth & Meisel, 2015).	40
Figure 3-2: QCSP classification scheme, (Bierwirth & Meisel, 2015).....	44
Figure 3-3: Number of research papers for the BACAP by year of publication...	48
Figure 4-1: The BACAP	59
Figure 4-2: The single-quay BACAP problem description. QC: Quay Crane, ETA: Expected time of arrival	61

Figure 4-3: Vessel to berth and time-space representation	61
Figure 4-4: Types of Scheduling Algorithms.	65
Figure 4-5: GP Engine and <i>BACAP_Scheduler</i> relationship	68
Figure 4-6: <i>BACAP_GP</i> Flowchart.....	70
Figure 4-7: Example of a <i>BACAP_GP</i> tree with defined functions and terminals.	72
Figure 4-8: Crossover representation for the <i>BACAP_GP</i> operation.....	73
Figure 4-9: Mutation representation for the <i>BACAP_GP</i> operation	74
Figure 4-10: Two-Dimensional array representation for the BACAP	75
Figure 4-11: Comparing the results of best SPRs with the <i>BACAP_GP</i> for 100 instances	84
Figure 4-12: The GP performance of one instance and three runs over 50 generations.	85
Figure 4-13: Box-plots of the distribution of the solutions for six randomly selected instances over ten runs.	86
Figure 4-14: The GP trees, representing the worst (A) and best (B) CDRs evolved for a random instance.	86
Figure 5-1: DP World Sokhna. (A) Location. (B) Masterplan	91
Figure 5-2: Post Panamax Crane.....	92
Figure 5-3: The impact of changing a vessel from its desired position on its handling time and cost.	97
Figure 5-4: Structure of the service cost of a vessel.(Meisel & Bierwirth, 2009)	98
Figure 5-5: Definition of <i>Overlap Time</i>	110
Figure 5-6: Definition of <i>Overlap Space</i>	110
Figure 5-7: Example of the best CDR evolved by <i>BACAP_GP_DP</i> to solve the BACAP instance.	112
Figure 5-8: Example of complete berth Plan with QCs.	113
Figure 5-9: Proportion of the <i>best SPR</i> by dispatch rule performance.	118
Figure 5-10: Compare GP1 performance with the Best SPR.....	119
Figure 6-1: The Multi-Port BACAP problem description	126
Figure 6-2: The Multi-Port BACAP Relationships.....	127
Figure 6-3: Multi-Port BACAP Plan Framework	132

Figure 6-4: Distributing vessels to Ports/Quays (Case1)	134
Figure 6-5: Distributing the vessels to the Ports/Quays (Case2)	135
Figure 6-6: Distributing the vessels to the Ports/Quays (Case3)	135
Figure 6-7: The MultiP_BACAP_GA flowchart	137
Figure 6-8: GA chromosome encoding.....	138
Figure 6-9: Crossover / Mutation operators.	139
Figure 6-10: Box-plot for the objective function (z-values) of five runs, 2 Quays.	144
Figure 6-11: Box-plot for the objective function (z-values) of five runs, 3 Quays.	146
Figure 6-12: Evolution Analysis, 2 Quays, fixed cost	150
Figure 6-13: Evolution Analysis, 3 Quays, fixed cost	150
Figure 6-14: Evolution Analysis, 2 Quays, random cost	151
Figure 6-15: Evolution Analysis, 3 Quays, random cost	151
Figure 6-16: Counts of dispatching rules used in the GA generations	153

Abbreviation

BACAP	Integrated berth allocation and quay crane assignment problem
BACAP_GP	Algorithm for solving the BACAP using the GP method
BACAP_GP_DP	Improved algorithm for BACAP_GP to solve BACAP with desired position.
BACAP_Scheduler	Algorithm for searching the best schedule for each vessel in a quay and assigning suitable QCs dynamically to it.
BACAP_Scheduler_DP	Improved algorithm for BACAP_Scheduler to solve BACAP with desired position.
BAP	Berth allocation Problem
CDR	Composite Dispatching rule
CT	Container Terminal
FCFS	First-come first-served
GA	Genetic Algorithm
GP	Genetic Programming
MultiP_BACAP_GA	Algorithm for solving the Multi Ports BACAP using the GA method
QCAP	Quay crane assignment problem
QCs	Quay cranes
QCSP	Quay crane scheduling problem
SPR	Standard Priority Rule
TEU	Twenty-foot equivalent unit
TO	Terminal Operators
VO	Vessel Operators

Dissemination

Published Papers:

- El-boghdadly, T., Bader-El-Den, M., & Jones, D. (2016b). Evolving local search heuristics for the integrated berth allocation and quay crane assignment problem. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2880–2887). IEEE. <http://doi.org/10.1109/CEC.2016.7744153>
- El-boghdadly, T., Bader-El-Den, M., & Jones, D. (2016a). A Genetic Programming Algorithm for the Berth and Quay Crane Allocation Problem. In *2016 5th International Conference on Advanced Logistics and Transport (ICALT)*. IEEE.

Presentations:

- “Modelling and Solving the Integrated Berth Allocation and Quay Crane Assignment Problem in Container Terminals”, Alexandria Port, Egypt, 2016
- MARLOG 4, The International Maritime Transport and Logistics Conference “A sustainable Development Perspective for Mega Projects”, Egypt, 29-31 March 2015
- IMTTC 2015, International Maritime Transport & Technology Conference, “Towards seaside operations optimization in container terminal”, 28-29 November 2015
- “Container terminal operations optimisation: a classification and literature review”, DP World Sokhna port, Egypt, 5 December 2014
- SCOR 2014, 4th Student Conference on Operational Research, “Multi-objective hyper-heuristic algorithm for solving the Dynamic Berth Allocation Problem”, Nottingham, United Kingdom, 2-4 May 2014

Poster

- *Best Poster Awarded*

Faculty of Technology Research Conference, “Multi-objective hyper-heuristic algorithm for solving the integrated Berth and Quay Crane Allocation Problem”, July 2014, University of Portsmouth, United Kingdom

Chapter 1 : Introduction

This chapter introduces the context of this research. It highlights the importance of the research by providing the motivation and scope of the research. Furthermore, the chapter presents the research framework and the direction followed in order to achieve the research objectives. Finally, it demonstrates the research structure and the organisation of the thesis.

1.1 Motivation and Scope of Research

The port industry has a significant economic impact on the global economy. This impact is expected increase over the next few years (European Commission, 2013). Ports, and container terminal ports in particular, are considered to be vital accelerators of local trade development in the age of globalisation. Therefore, global containerised trade volumes are experiencing a rapid increase. In 2015, they reached 1.69 billion tons, equivalent to 175 million twenty-foot equivalent units (TEUs) as seen in Figure 1-1 (UNCTAD, 2016). A recent European commission study on Europe's Seaport 2030 key facts (European Commission, 2013) found that the cost and quality of port services are major factors for future European business to consider. Moreover, the cost of handling cargo, port dues and nautical port services can account for between 40%-60% of the total logistics chain costs concerning transported goods. However, Europe's ports face three significant challenges:

- A 50% growth of the amount of cargo handled in European Union seaports is expected by 2030.
- The newest generation of container ships can carry more containers.
- The performance gap between Europe's ports produces enormous inefficiencies – longer routes, longer sea and land trips, more transport emissions and more congestion of the economy.

Container terminal seaports play an essential role in the maritime transportation life cycle, as they are the intersection point between sea and land global transport chains. Ships might have to wait for a berthing location and other terminal logistical services. Time is money; ports may not become attractive for shippers if the waiting time for ships to be served increases even if the port charges are low. Vessel operators expect the service at any port to be handled in a short amount of time in order to save on their running expenses. On the other hand, terminal operators expect to reduce their terminal costs, making sure to meet the customers' satisfaction standards and to increase the productivity of their terminals. Therefore, terminal operators might have the facilities to find optimal choices to respond to any conceivable scenario. This has opened up various challenging combinatorial optimisation problems in the field of maritime transportation.

Computer technology has become an essential part of intelligent management systems used in container terminal operations. Operations research (OR) methods seek to innovate ways to enable the port operators to come up with cost-effective scheduling plans for vessels visiting their ports. The field of OR has significantly evolved over the years, and the problems encountered have become increasingly challenging and complex. An overview of the different terminal operations and the impact of OR has been described by Carlo, Vis, & Roodbergen, (2013); Voß, Stahlbock, & Steenken, (2004).

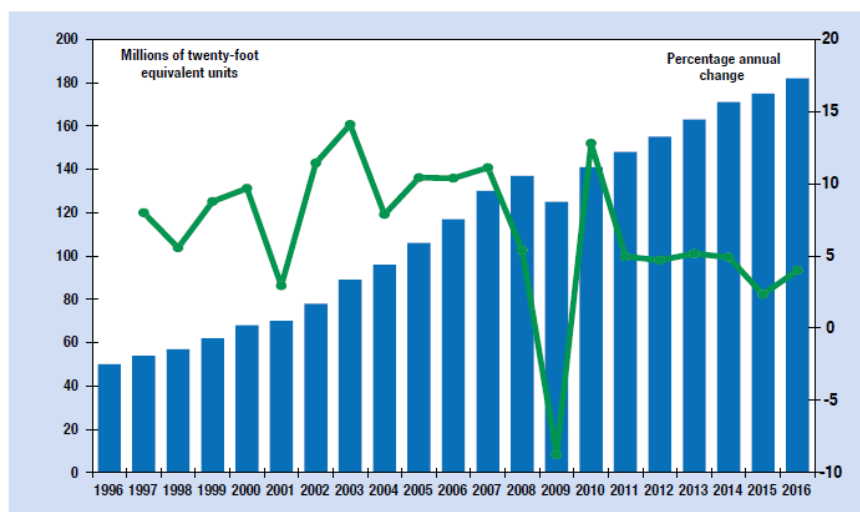


Figure 1-1: Global containerised trade, 1996-2016 (Millions of TEUs and percentage annual change), (UNCTAD, 2016)

Volume and time in the context of port operations are the two critical aspects involved in measuring performance. Volume is a measurement of throughput or a port's output and is expressed in either TEUs or weight (tons). Service time, which is waiting time plus handling time, is how long vessels spend in a port which also affects the port's performance and productivity (UNCTAD, 2013).

For a port operator handling a large volume of containers and ship calls, the productivity for such a terminal is high. Therefore, port operators might seek to improve productivity by using the port's resources efficiently in order to cope with the rapid increase of incoming vessels. The scheduling of incoming vessels should be optimised in order to minimise the total service time/total service costs of all vessels.

A container terminal has three main areas; the seaside, yard side and land side. Each area has various problems that need to be solved. The seaside area, which is the major focus of this research, has three interrelated operational planning problems namely the berth allocation problem (BAP), the quay crane assignment problem (QCAP) and the quay crane scheduling problem (QCSP). The decisions related to these problems have a crucial impact on container terminal operational efficiency.

In the literature, we have disclosed a few gaps that this research needed to fill in. Previous studies have been tackled the seaside problems separated, and there is still an insufficient number of them have tackled them in an integrated manner. We also noted the impact of the dispatching rules and give different priorities for ordering incoming vessels while scheduling them. The studies have been used a single rule while solving the problems or used different rules and applying them one by one each time. To the best of our knowledge, a few studies solve these problems in a way of composite dispatching rules which consider multiple rules at the same time. Moreover, the studies were seeking to solve the problem and not to find a general solver which can apply it in similar problems with similar constraints. Most of the studies have solved the problems considering them in a single quay while a few of them consider multiple quays.

From previous studies, we can understand that container terminal ports need to adapt with aim of meeting the industry's future needs, and to handle the increase in the number and depth of container ships in order to serve them faster than before.

This research investigates the above problems, and an integrated solution approach has been developed to solve the first two problems, namely berth allocation and quay assignment problem (BACAP). The research studies the seaside problems from a different perspective and analyses the problems' constraints. Moreover, this study extends the state-of-the-art solution by presenting a novel model to solve the seaside problems in the case of multiple ports that have multiple terminals owned by one owner, as commonly found nowadays.

1.2 Research Aim and Objectives

1.2.1 Aim

This research aims to contribute to the existing knowledge to improve the operational efficiency of single and multiple container terminal ports, particularly in the seaside area as it is the bottleneck of container terminal operations. The increase of quay crane utilisation, the optimisation of container ships turnaround time (waiting time and handling time) and the minimisation of container terminal costs will lead to improvements.

1.2.2 Objectives

This research will examine the optimisation methods which will be applied to the operational planning problems that arise in the seaside container terminal area. We will use different approaches to model the integration between these problems by using novel self-adaptive hyper-heuristic algorithms to solve the problem. The main objectives of this research are as follows:

- Identify the main combinatorial optimisation problems (COP) in container terminals and to classify the different methods used for solving each problem.

- Define the container terminal characteristics and berth layouts which have a significant impact on the container terminal seaside operational problems.
- Identify the main vessel and QC scheduling challenges in real-world ports.
- Develop new/pioneering intelligent algorithms for COP to optimise the total service time of the incoming vessels and to minimise the container terminal's total costs.
- Examine the developed algorithm in relation to the integrated terminal's seaside problems. The study will focus on the integrated Berth Allocation and Quay Crane Assignment problem (BACAP).
- Develop a new model to solve the BACAP in the case of multiple ports being available for berthing incoming vessels.
- Compile and update a set of benchmarks that are used in the literature for the container terminal COP, focusing on the BACAP.
- Evaluate and validate the performance of the developed algorithms against both exact and state-of-the-art heuristic methods.

1.3 Research Methodology

The proposed research study is quantitative and data-based. It uses experimental studies for the validation of the proposed novel methods and the analysis of the results. The research study also provides tools and techniques for solving the seaside operational planning problems of container terminals.

This research will develop intelligent combinatorial optimisation algorithms to evolve the solutions. A meta-heuristic approach using a genetic algorithm (GA) and genetic programming (GP) was used to solve the problems. The analysis of the modelling efficiency was measured by comparing the results with state-of-the-art results for large instance benchmarks, and with a commercial software package such as CPLEX for smaller instances. A visit to a real container terminal port was conducted to study and analyse the practical seaside operational problems in

addition to the new challenges and circumstances faced during the scheduling process related to incoming vessels.

1.4 Research Framework

The research framework and direction followed to achieve the research objectives has been illustrated in Figure 1-2. The research starts with the motivation to improve the situation caused by the maritime logistics problems, and then found that the container terminal is a major part of the maritime logistics life cycle. Container terminals have three main areas as described before, which are the seaside, yard, and landside area. The research progressed to the first area, which was the seaside. The seaside area has three sequential problems, which are the berth allocation problem (BAP), the quay crane assignment problem (QCAP) and the quay crane scheduling problem (QCSP). This research focuses on the integration between the first two problems, which are berth allocation and quay crane assignment problem (BACAP). The BACAP was tackled using different constraints and objectives for comparing with the existing literature models and to prove the effectiveness of the proposed model on a single port with one container terminal (one quay), before extending it to solve the BACAP with multiple quays/Ports.

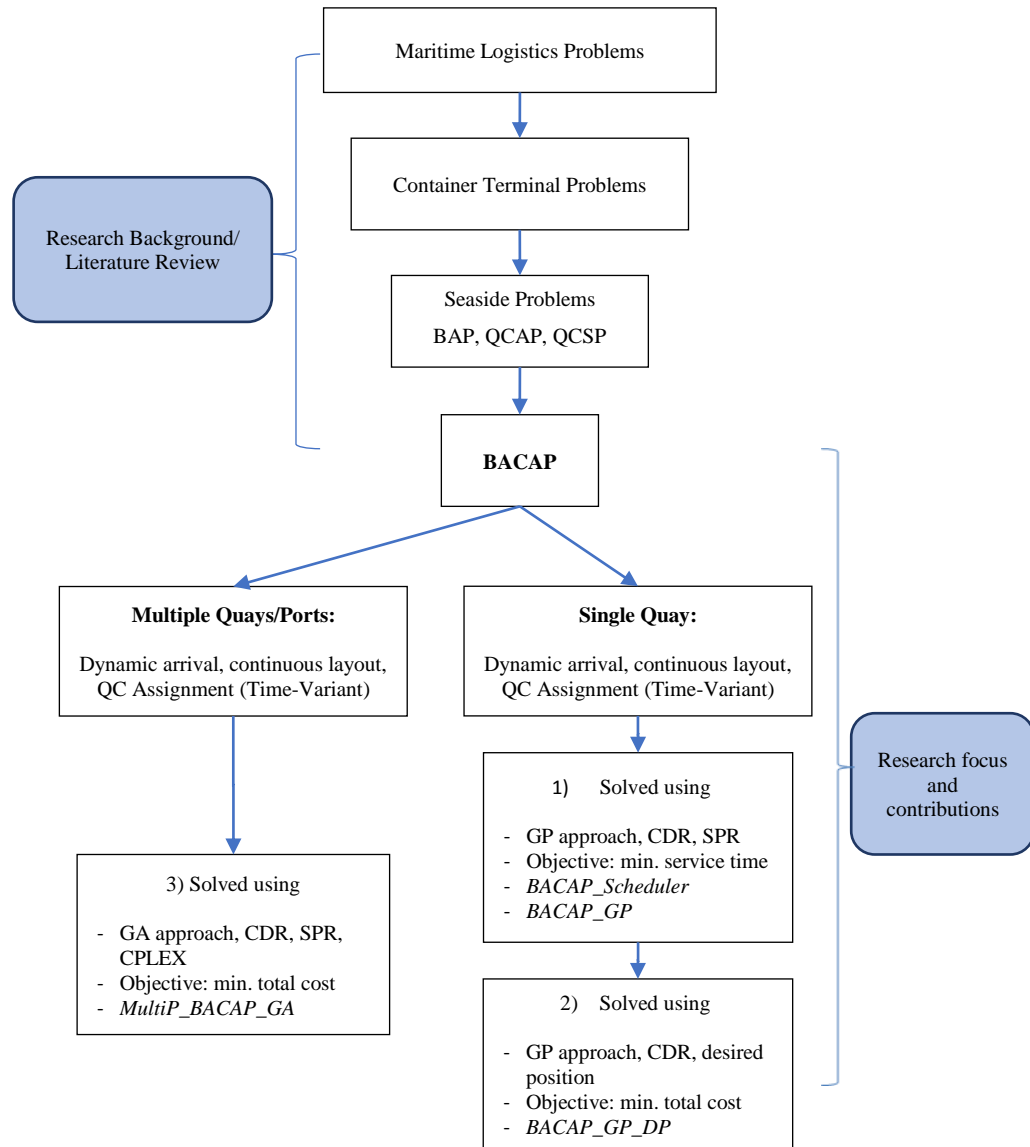


Figure 1-2: Research Framework.

BACAP: Berth Allocation Quay Crane Assignment Problem, GP: Genetic Programming, GA: Genetic Algorithm, CDR: Composite Dispatching Rules, QC: Quay Crane, SPR: Standard Priority Rules.

1.5 Research Structure and Thesis Organisation

The overall research structure and flow illustrated in Figure 1-3.

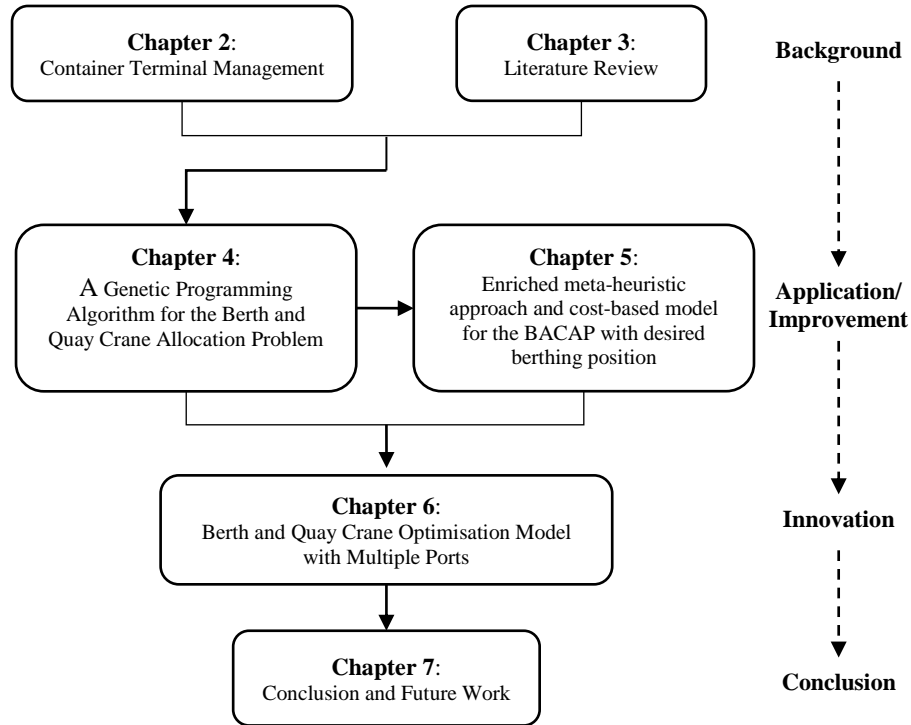


Figure 1-3: Research Structure and Flow.

Thesis Organisation:

The outline of the thesis with a summary of each chapter is as follows.

Chapter 2 provides an overview of the importance of the container terminal. It introduces the container terminal management life cycle regarding its systems and processes. It shows the types and layout of a container terminal, the different areas and its associated problems and classifications, as well as the different types of equipment needed during transshipment.

Chapter 3 presents a comprehensive literature review on container terminal seaside operational problems including, in particular, the BACAP. It presents the BACAP classification, benchmarks, and the optimisation methods that have been used to

solve the problem in the existing literature. It reviews the literature for the BACAP in a single port and investigates the BACAP when considering the case of multiple ports. Finally, it points out the research gaps that will be covered in this thesis.

Chapter 4 develops a genetic programming approach to solve continuous dynamic BACAP (*BACAP_GP*). It presents the proposed independent algorithm for scheduling vessels (*BACAP_Scheduler*). It reviews the dispatching rules and composite dispatching rules (CDRs), and studies the efficiency of using CDRs for solving the BACAP on a large-scale benchmark.

Chapter 5 studies a real BACAP in the “DP World-Sokhna” port. It adapts the proposed model in Chapter 4, *BACAP_GP* and improves the *BACAP_Scheduler* to solve the BACAP with the desired berthing position (*BACAP_GP_DP*). It provides a comparison study with the results of the state-of-the-art solution. The main difference between chapter 4 and chapter 5 as follows:

Model difference/features	Chapter 4	Chapter 5
Vessel can moor before its expected time of arrival	X	√
Vessel can moor in any position with cost	X	√
Results compared with the state-of-the-art	X	√
Evaluated on a large-scale dataset	√	X
The objective is to minimise the total service	time	costs
The handling time calculated using the number of quay crane assigned	√	√+ vessel’s shifting from its desired berthing position
Support quay space utilisation in advance	X	√

Chapter 6 presents a new problem, which is the BACAP with multiple ports. It develops a genetic algorithm model to solve the problem (*MultiP_BACAP_GA*). It integrates the strategic planning level with the operational level in order to solve the problem. It solves the problem via an exact method using CPLEX and presents a comparison study with the proposed heuristic method to prove its efficiency.

Chapter 7 concludes this research and provides further research directions.

Chapter 2 : Container Terminal Management

This chapter introduces the main operations of maritime container terminals. It provides the reader with an overview of container terminal management, its systems and processes and the equipment needed during the transshipment life cycle. This chapter also demonstrates the main container terminal's areas and its problems as classified by the different planning/decision levels. Furthermore, it covers the seaside operational planning problems and its constraints. It also insights into the research gaps to fill in this study and the challenges encountered when handling the container terminals' operations.

2.1 Introduction

Container terminal ports are gateways in the global containerised shipping industry. They are essential for completing the global supply chains which allow for links between the different means of freight transportation. Due to the importance of this industry, seaport competition has increased considerably as containerised trade volumes have rapidly increased in recent years. Port managers have to ensure that (i) the vessels are berthed as soon as possible after arrival to achieve a fast turnaround, (ii) the quay cranes load and unload the required containers in the shortest possible time and (iii) the cost of container transshipment is minimised. Table 2-1 presents the world's 20 busiest container ports in the world in terms of the TEUs handled (The Journal of Commerce, 2016). We noticed that Shanghai, China, ranked as the busiest container terminal in the world in 2016 with a growth rate of 1.6% up from 2015. The volume reached 37.13 million TEU in 2016. Moreover, there is an increase parallel to this in almost all ports throughput.

Table 2-1: Top 20 busiest container ports in the world as of 2016, (The Journal of Commerce, 2016).

Rank	Port, Country	Volume 2016 (Million TEU)	Volume 2015 (Million TEU)	2015-2016 % VOLUME CHANGE
1	Shanghai, China	37.13	36.54	1.6%
2	Singapore	30.90	30.92	-0.1%
3	Shenzhen, China	23.97	24.2	-1.0%
4	Ningbo-Zhoushan, China	21.60	20.63	4.7%
5	Hong Kong, S.A.R., China	19.60	20.07	-2.3%
6	Busan, South Korea	19.45	19.46	-0.1%
7	Guangzhou Harbor, China	18.90	17.22	9.8%
8	Qingdao, China	18.00	17.47	3.0%
9	Jebel Ali, Dubai, United Arab Emirates	15.73	15.6	0.8%
10	Tianjin, China	14.49	14.11	2.7%
11	Port Klang, Malaysia	13.20	11.89	11%
12	Rotterdam, Netherlands	12.40	12.23	1.4%
13	Kaohsiung, Taiwan, China	10.46	10.26	1.9%
14	Antwerp, Belgium	10.04	9.65	4.0%
15	Dalian, China	10.00	9.45	5.8%
16	Xiamen, China	9.60	9.18	4.6%
17	Hamburg, Germany	8.90	8.82	0.9%
18	Los Angeles, U.S.A.	8.80	8.16	7.8%
19	Tanjung Pelepas, Malaysia	8.28	9.1	-9.0%
20*	Keihin Ports, Japan	7.61	7.52	1.2%

Container terminal management is increasingly becoming a complex task due to the number of incoming vessels and consequently, the increasing number of containers that need to be served. Container terminals have many integrated systems that work together to serve and track the containers from the moment that they arrive until they are released from the port. Moreover, there are other systems in place to communicate with both shipping lines and customers in order to pre-plan the containers' collection and delivery. This include different systems to manage the equipment and vehicles, keep track of their performance and recording their maintenance records.

All of the above systems must be managed efficiently to handle the container terminal operations. These operations have various planning problems that need to be solved, and decisions that must be taken. To manage such an industry and to cope with the rapid increase of the world's level of containerised trading, terminal managers are forced to use computer technology and operations research techniques to remain competitive in global trade. These techniques are crucial for reducing the terminal costs and increasing the terminal equipment's utilisation parallel to meeting shipping line satisfaction.

2.2 Container Terminal Overview

In this section, we start with the terms that are commonly used in the context of a container terminal and the corresponding definition. A *Container Terminal* (CT) is a type of port where cargo containers are transhipped between ships, or between ships and land vehicles (trucks or trains) (Salido, Rodriguez-Molins, & Barber, 2011). Figure 2-1 shows a visual illustration of the “TCB Barcelona container terminal”. The figure shows the *waiting area*, which is the area where incoming vessels waiting in order to start being served. A *quay* is a structure alongside a harbour where vessels may dock to load and unload cargos or passengers. The mooring location of a vessel on the quay is called the *berth*. A quay can include one or more berths.

Container terminals use *cranes* of different types to hold and transfer containers from one place to another, or from one mode of transportation to another. The essential type of crane in container terminals is the Quay Crane (QC); see Figure 2-2. The QC is located and lined up alongside the quay and equipped with trolleys, which makes it easy to move along the quay but they cannot pass each other. The structure and design allows for the moving and transferring of containers from a moored vessel to a vehicle, or vice versa. We will provide more of a description of the types of crane that are commonly used in container terminals later on in this chapter.

Containers are large metal boxes of a standard design and size for the transportation of goods; the standard sizes possible for a container is 20 foot (6.09 meters), 40 foot (12.18 meters), 45 foot (13.7 meters), 48 foot (14.6 meters), and 53 foot (16.15 meters) (Worldshipping, 2014). The two most commonly used sizes today are the 20-foot and 40-foot lengths. The Twenty-foot-Equivalent Unit (TEU) is used to refer to one container with a length of twenty-foot, while 2TEU or the Forty-foot Equivalent Unit (FEU) indicates a container of 40-foot length. Figure 2-3 illustrates a typical TEU container.



Figure 2-1: TCB Barcelona container terminal photo APM Terminals. Source (Fonseca, 2016)



Figure 2-2: Quay Crane. Source (Kalmar global, 2018)



Figure 2-3: The 20 feet Container (TEU)

A *Container Ship* (CS) is a type of ship or vessel that carries containers in order to transfer goods from one place to another. As a result of globalisation, international trade has dramatically increased, and container ships have become larger and larger. In 2017, containerised trade continued to face the upsizing of container ships. Figure 2-4 shows the rapid increase in the size of container vessels from 1956 to

2017. The largest container vessel in the world can hold more than 21,000 TEU, and its operator is OOCL (Hong Kong). The length of the largest vessel is 400 meters with a depth of 16 meters. Since the creation of modern water channels, for example the Suez Canal with a depth of 24 meters, ports have continued to improve their depth, and it is expected for there to be a further growth in vessel capacity up to 50,000 TEU in the next 50 years (Saxon & Stone, 2017).

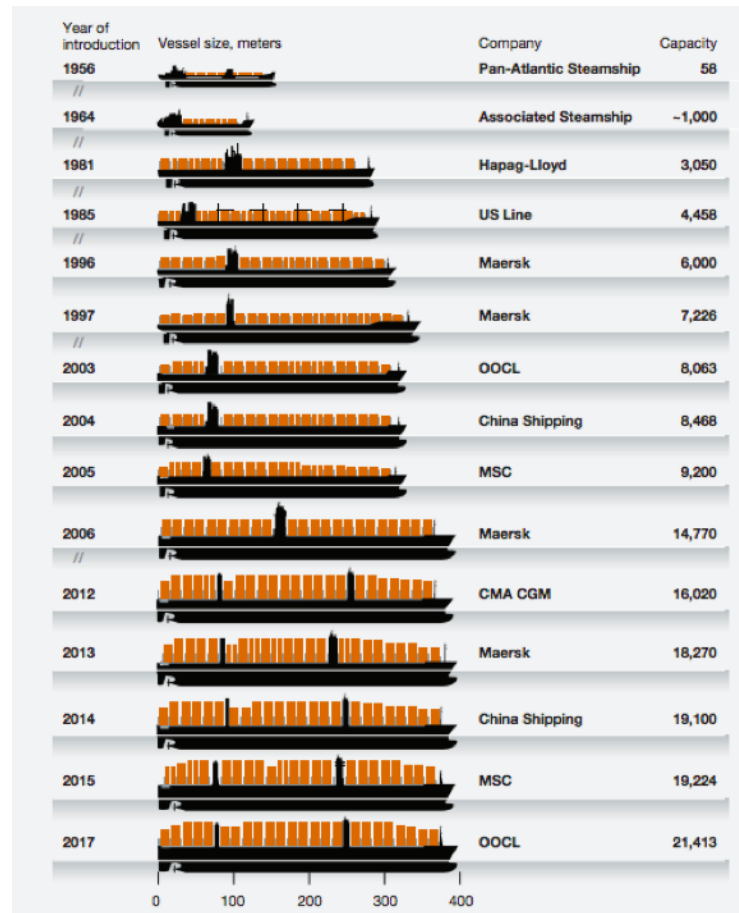


Figure 2-4: Maximum container-vessel capacity from 1956 to 2017, TEU. Source (Saxon & Stone, 2017)

The layout of a container vessel has been presented in Figure 2-5. Containers are sorted on *bays*, which consist of many *stacks* in *hold* and on *deck*. Each bay has a number of *stacks* and *tiers*. *Hatch covers* separate the stacks in the hold from those on the deck. The unloading/loading containers processed to/from a container vessel should consider in advance the placement of the container, such as if it is either on deck or in the hold. For the unloading process, the containers on deck should be

removed first before the containers in the hold. As for the loading process, containers in the hold should be loaded first before loading the containers on the deck.

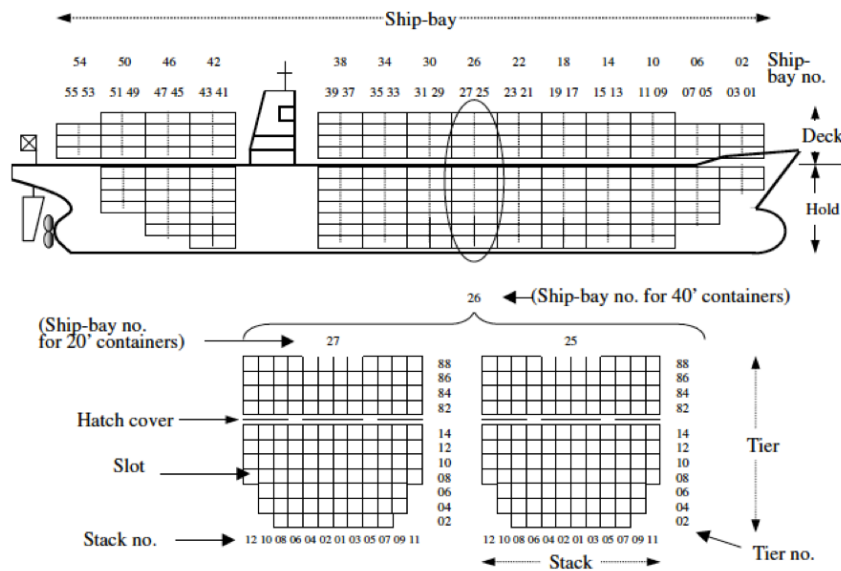


Figure 2-5: A container vessel layout. Source (Kim, Kang, & Ryu, 2004)

The continuous increase of vessel capacity has had a crucial impact on container terminal traffic. This has led to more significant pressure being placed on cargo handling services and the associated operational costs (UNCTAD, 2016). Moreover, it forces container terminals to expand their structure or to build a new one in addition to updating its equipment to cope with the increase. Container terminals are expensive to build and difficult to operate.

2.3 Container Terminal: Systems and Processes

The container terminal's main process starts after the arrival of vessels at the port, followed by the allocation of vessels to the berths equipped with quay cranes (QCs). The QCs are responsible for loading and unloading the containers to and from the vessels. These containers are then transferred by trucks or trains to the yard area where they are temporarily stored until either transported inland or transferred to

other vessels (Vis & de Koster, 2003). Figure 2-6 shows a common CT operational scenario (Cordeau, Laporte, Legato, & Moccia, 2005).

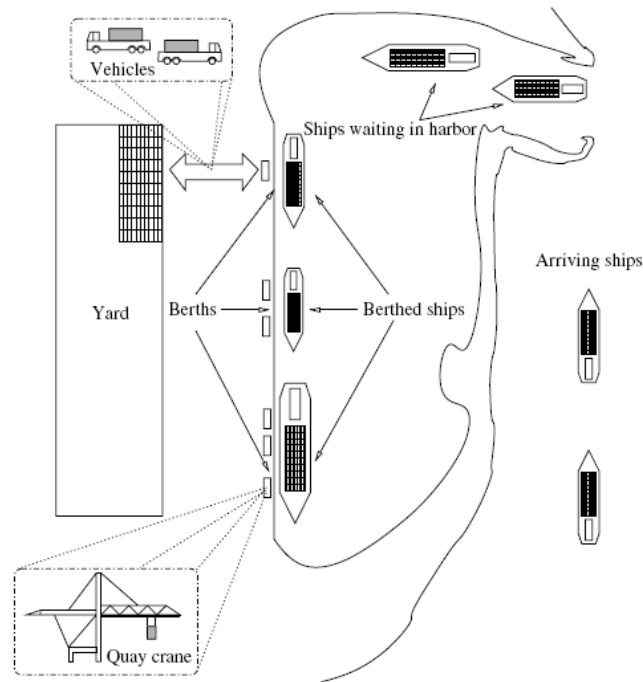


Figure 2-6: Container Terminal Operation Scenario (Cordeau et al., 2005).

Container terminals can be divided into three main areas, namely the *seaside* area, the *yard*, and the *landside* area. Figure 2-7 illustrates a typical modern container terminal system used in the transshipment process. The *seaside* provides the services required by the container vessels. The *yard* area is responsible for storing containers temporarily until they are imported or exported while the *landside* area connects the container terminal to the hinterland. The CT uses trucks, trains and other means of transportation to transfer the containers between these areas. The flow of the involved processes and the interconnection between these areas working in two directions depends on importing or exporting containers. Each area has many problems that need to be optimised.

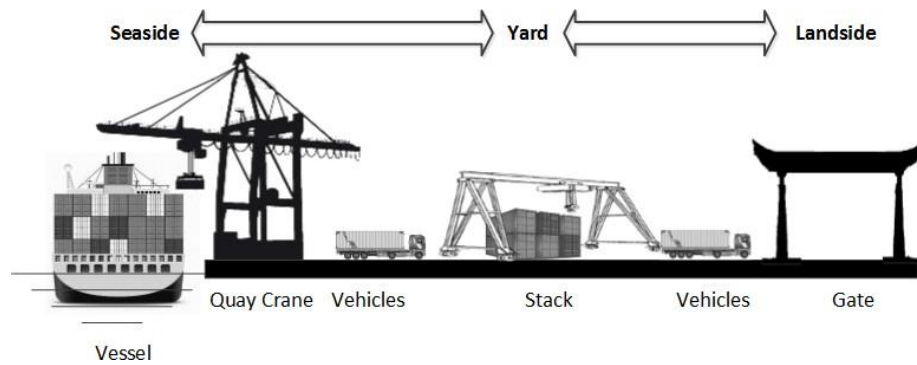


Figure 2-7: Container terminal system for transshipment process.

The following sub-processes can summarise the transshipment systems and processes incorporated into CT:

1. Arrival of vessels.
2. Unloading and loading of containers.
3. Transferring containers from vessel to stack, and vice versa.
4. Stacking of containers.
5. Inter-terminal transport and other modes of transportation.

2.3.1 Seaside Area

2.3.1.1 Arrival of vessels

There is a direct communication link between the vessel operator and the terminal operators. This link is used to exchange information, such as the vessel's expected time of arrival, the number of containers to be loaded/unloaded, the specific services needed etc. This information should be received in advance, sent by the vessel operators to the terminal operator for pre-planning and scheduling purposes.

The time of a vessel being due to arrive in port is also referred to as the expected time of arrival (ETA). However, the vessel might face delays. The exact time that the vessel arrives at the port and is ready to be served is the actual time of arrival (ATA). Moreover, if the vessel arrives earlier than the ETA and the terminal is ready to serve it, then it is denoted as the earlier starting time (EST).

When the vessel arrives at the port, the terminal operator must allocate a berth on the quay to let the vessel moor. Moreover, it is important that the terminal operator finds a berth available at the requested vessel's ETA. In addition, if the vessel operator requests a desired berthing position on the quay, then the terminal operator can consider this while finding a berth for that vessel. The desired berthing position is usually defined as the closest berth to the stacking area. The stacking area is the place where the containers should be located for a specific vessel. We can disclose the problems that might happen if the vessel arrival time changes from ETA to ATA, or from ETA to EST. The terminal operator might also consider other constraints concerning allocating a berth to a vessel, such as the depth of the berth should be higher than the vessel's draft, the berth should have available quay cranes to serve the vessel etc. In this stage, the vessel is berthed and is ready to start loading or unloading its containers.

2.3.1.2 Unloading and loading containers

It is crucial for terminal operators to know how many containers are to be moved and their position onboard for either unloading or loading purposes to/from the vessels. The terminal operator needs to be ready with a suitable schedule plan to handle all of the movements in advance. The QCs are the handling equipment that the terminal uses at this stage. There are various types of QCs that might be used at either automated and manned ports. Some of QCs can move two or more containers at the same time, which will reduce the vessel's service time.

The terminal operator might need to decide on the number of QCs that will be assigned to the vessel to serve it. The time needed to move all containers mainly depends on the number of assigned QCs. Moreover, the operators might consider that the QCs move from one place to another on the quay using a rail track alongside the quay, which restricts them in passing one another. In addition, only one QC can work in a vessel's bay area at a time, and there is a safety distance that should be considered between two adjacent quay cranes.

There are two methods of assigning QCs to a vessel. The first method is the QC static assignment (time-invariant). In this case, the QC that is assigned to a vessel

will continue to be reserved to that vessel even if it has finished its work, and it will be released once the vessel has finished its services and is ready to move. The second one is the QC dynamic assignment (time-variant). In this case, the QC can be moved from one vessel to another before the first one has finished and moved. The terminal operator may use the dynamic assigning strategy to save more time and to increase the utilisation of the terminal's QCs.

2.3.2 Transport Area

2.3.2.1 Transfer containers from vessel to stack and vice versa

In this stage, the containers are transferred from vessel to stack and vice versa. In the previews stage, the quay crane unloads containers from the vessel and loads them onto one of the transportation means that are used to transfer containers to the destination of the stack. There are many types of transportation that can be used such as yard trucks, forklift trucks, multi-trailer system and straddle carriers, as in Figure 2-8.



Figure 2-8: (a) Yard truck. (b) Forklift trucks. (c) Straddle carriers. (d) AGV. Source (Kalmar global, 2018)

At an automated container terminal, there are automated guided vehicles (AGVs) Figure 2-8.(d) and automated lifting vehicles (ALVs) that can be used to transfer containers from vessels to stack and vice versa. AGVs are robotic vehicles that travel along a predefined path that are automatically controlled by the terminal operators. This type of transport is only practical for ports with high costs regarding maintaining and operating the AGVs. Currently, AGVs are used in the container terminals of Rotterdam in the Netherlands and of Hamburg in Germany. For ALVs, they are used at the container terminal of Brisbane, in Australia.

The terminal operator might decide on the type and the number of vehicles to be used and which vehicle transports which container, defining its destination in the stacking area.

2.3.3 Yard Area

2.3.3.1 Stacking of containers

The yard area is the space in the container terminal where the containers are stacked and stored for a certain period until they are either transferred to a vessel or to the hinterland. The yard area can be divided into many sections depending on the type of container or if it contains dangerous goods, as two examples. The stack is organised into multiple blocks or lanes, each of which consists of a number of rows. In this method of organisation, the terminal operator can identify each container and store its information, such as where it came from (origin), where it should be moved to (destination), when will it be transferred and what its content is.

It is crucial for the container terminal to sort the containers within the blocks, such as keeping the earliest transferred ones available and the most accessible to move before the ones that are to stay longer in the terminal. Moreover, the terminal operator might store empty containers separately. In this area (yard area), the terminal uses a type of cranes called a rail-mounted gantry crane (RMGC) or a rubber-tired gantry crane (RTGC) to handle the sorting operations, as in Figure 2-9. They can use forklift trucks, reach stackers and straddle carrier systems to support the operation of stacking and sorting the containers.

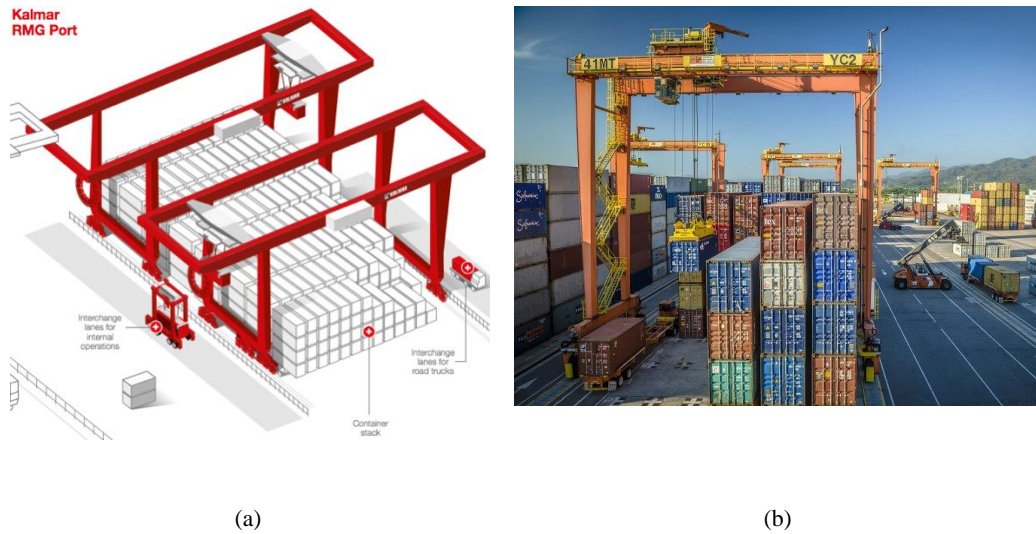


Figure 2-9: (a) RMGC. (b) RTGC. Source (Kalmar global, 2018)

2.3.4 Landside Area

2.3.4.1 Inter-terminal transport and other modes of transportation

All containers in the yard area have to be transferred, either to vessels if they are for exporting purposes or to the hinterland if they are for importing purposes. If they are imported, then the final area - which is called the landside area in the terminal - will be used to transport the containers from the stack to inland by other modes of transportation. The mode of transportation may be a truck or train, which is used to transfer the containers away from the port to their final destination.

2.4 Container Terminal: Planning Levels and Problems

A container terminal consists of very complicated logistics systems. The previous sections illustrate the transshipment systems, processes and the life-cycle of and within the container terminal. All of these processes include many problems and decisions that need to be solved either in advance or in the moment. It is very complicated to try and solve all of the problems at once. Therefore, the planning

and scheduling problems of a container terminal can be distinguished either on a planning level and/or in the specific areas of the container terminal. Based on the planning time horizon level, there are three planning levels; strategic, tactical and operational. For the container terminal areas, the three main areas as has been described; seaside, yard and landside. The following is how the literature classified the problems and decisions inside the container terminals.

Henesey (2006) classified the decisions of the container terminal's operations into two decision types; control and planning, derived from Rushton, Croucher, & Baker (2010). A literature survey conducted by Henesey (2006) was organised into a framework with four categories: (1) container terminal subsystems including Ship-to-Shore, Storage, Transfer, and Delivery/Receipt, (2) decision type (Planning and Control), (3) time frame (Strategic, Tactical and Operational) and the (4) typical issues which illustrate the type of problems that container terminal managers face when making a decision.

Vis (2009) made a webpage for the research that they conducted in a container terminal. The researcher proposed there to be four decisions related to the logistics processes in the container terminal; the arrival of the ship, the unloading and loading of the ship, the stacking of containers, and the transportation of containers from ship to stacking area and vice versa. These decisions were studied at the strategic, tactical and operational levels.

Based on the classification provided by Bierwirth & Meisel (2010), the various decision problems encountered in a container terminal and the direct link between the problems has been illustrated by Iris, Larsen, Røpke, & Pacino (2016) in Figure 2-10.

The following sub-sections provide a description on the decision planning levels, and explain in more detail, the seaside operational planning which is the main focus of this research.

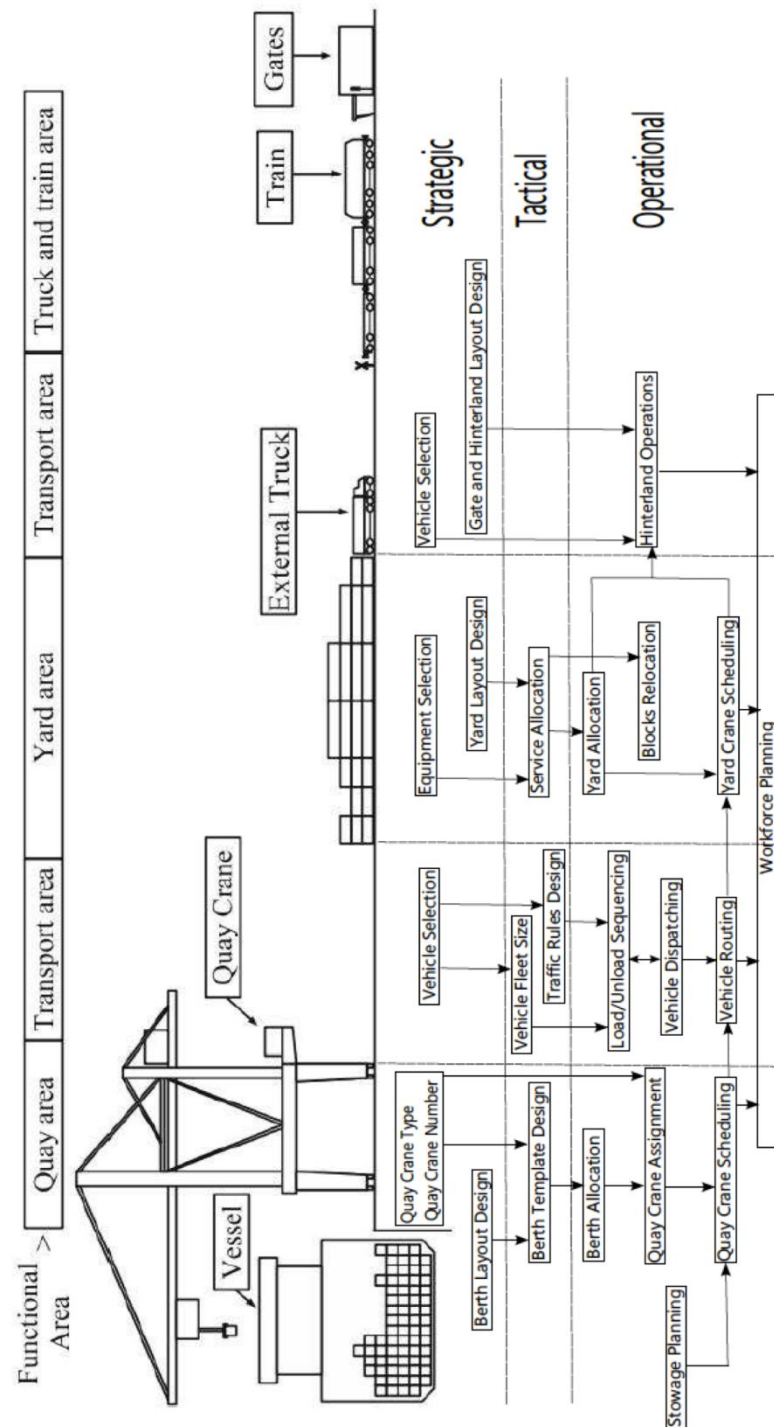


Figure 2-10: Decision problems in a container terminal. Source (C. Iris et al., 2016).

2.4.1 Strategic planning

Strategic planning problems have the highest costs to solve and have the most significant impact on the system's performance. The decisions conducted at the strategic level are long-term decisions that last for years and can lead to the definition of a set of constraints for subsequent tactical and operational decisions. The problems are mostly related to a new terminal design (location, size, and layout), the resources that are available to use, and the strategic networks within the shipping lines, including contracts.

For instance, strategic planning problems are:

(1) *Design of a container terminal (Berth, Yard and Hinterland layout design)*. Bockstael-Blok, Mayer, & Valentin (2003) studied the design of an inland container terminal through visualisation, simulation and gaming. De Castillo & Daganzo (1993) and Hwan Kim & Bae Kim (1999) studied the strategies used to locate containers in the yards and the suitable design/size of the blocks.

(2) *Transportation types/numbers inside a container terminal*. Vis & Harika (2004) studied two different types of automated vehicle (automated lifting vehicles and automated guided vehicles) used to transport containers from the stack to the ship and vice versa. Murty, Liu, Wan, & Linn (2005) used the pooling strategy to minimise the number of vehicles required for transport operations.

(3) *Selection of equipment*. Cartenì & Luca (2012) and Vis (2006) studied the strategic decisions that have to be made in order to determine the specific equipment (type/number) investments required and its properties related to the container terminal.

2.4.2 Tactical Planning

The tactical level of planning, and the associated problems and decisions, is considered to be short-term planning that lasts for months or weeks. The problems encountered are mostly related to space utilisation within the terminal, e.g. berth templates, yard templates and storage management. The layout of traffic courses for the horizontal transport system is also considered to be a tactical decision.

For instance, some of the tactical planning problems are:

(1) *Berth template design*. M. P. M. Hendriks, Lefeber, & Udding (2013), Jin, Lee, & Hu (2015) and Moorthy & Teo (2007) studied the berth-windows (berthing locations/times for services) within a fixed length of planning horizon in order to maximise the service objective. This problem is also referred to as *Tactical berth scheduling* (Imai, Yamakawa, & Huang, 2014).

(2) *Yard template design*. This problem also referred to as the *Service allocation problem*. Shipping companies, while contracting with the container terminal, usually ask them to reserve a space in the yard storage. The objective of this problem is to minimise yard reorganisation, which is the amount of reshuffling (Cordeau, Gaudioso, Laporte, & Moccia, 2007; D. H. Lee & Jin, 2013).

(3) *Vehicle fleet size*. This is the number of necessary transport vehicles to transport all containers at a point in time (Vis, de Koster, & Savelsbergh, 2005; Vis, de Koster, Roodbergen, & Peeters, 2001).

2.4.3 Operational Planning

At the operational level of planning, the decisions are related to real-time operational planning which lasts from days down to seconds. The problems encountered at this level of planning are mostly related to daily work plans, process management and the scheduling of the container terminal resources. The common goals for container terminal operators while planning and scheduling are to minimise the transshipment time of containers in the terminal and to minimise the cost of operations. This is as well as increasing the utilisation of the container terminal's resources.

The following sections will describe the operational planning of the seaside area and the literature will be further detailed in the next chapter. Regarding the operational planning problems encountered in other areas as described by C. Iris et al. (2016) in Figure 2-10, we have:

(1) *Horizontal Transport Operations*, which is comprised of the *Vehicle Dispatching and Routing Problems*. The first problem is responsible for assigning

vehicles to quay cranes for either the sequence of loading or unloading containers (exclusive assignment), or where each vehicle serves different quay cranes for the loading and unloading of containers (pooled assignment) (Meisel, 2009a). The aim is to minimise the waiting time of the quay cranes by reducing the empty travel of vehicles from the ship to a yard and vice versa (Bish et al., 2005; Bose, Reinert, Steenken, & Voss, 2000). The second problem is responsible for choosing the vehicle's travel route and minimising terminal traffic (Kim & Kim, 1999a, 1999b).

(2) *Yard Management Planning* is comprised of the following problems. (i) *Yard Allocation Problem*. This problem is responsible for the reservation of the yard's capacity and the selection of storage locations for the individual containers of incoming vessels. The yard space should be organised into different container types such as export, import, transshipment containers etc (Kim, Park, & Ryu, 2000; L. H. Lee, Chew, Tan, & Han, 2006). This problem aims to minimise the time taken in the storage yard operations for storing, retrieving and reshuffling containers. (ii) *Block Relocation Problem (re-marshalling/pre-marshalling)* is the problem of repositioning containers within the yard to resolve unsuitable storage locations and stacking orders (Expósito-Izquierdo, Melián-Batista, & Moreno-Vega, 2012; Y. Lee & Hsu, 2007). Abbas, Al-Bazi, & Palade (2018) proposed a fuzzy knowledge-based system integrated with a neighbourhood heuristic algorithm to optimise the stacking and retrieval operations of the containers in the yard. Their problem and solution considered the yard itself, the existence of pre-existing containers and the unknown container departure time. Their results showed the effectiveness of the proposed system when it came to optimising the number of containers relocated by 5%, 6.6% for the improved container relocation time and amount of yard utilisation and minimising by 42% the average waiting time per third-party logistics truck.

(3) *Yard Crane Scheduling*. This problem deals with the deployment of yard cranes to yard blocks, and the schedule stacking and retrieval operations of and for containers (Cheung, Li, & Lin, 2002; Wu, Li, Petering, Goh, & Souza, 2015; Chuqian Zhang, Wan, Liu, & Linn, 2002). The objective is the minimisation of the total delays related to the yard crane workload.

(4) *Hinterland Operations*. The hinterland area is the region where export/import containers are transferred to/from the terminal. The hinterland uses trucks or trains to connect the terminal to the inland region (Ambrosino, Caballini, & Siri, 2013; G. Chen, Govindan, & Yang, 2013). The common objective of this problem is to minimise the hinterland workload operations and the waiting times for yard transport and equipment.

2.4.3.1 Seaside Operational Problems

Seaside operations are critical and have a major impact on a container terminal's operational performance (M. Z. Li, Jin, & Lu, 2015). Seaside problems are considered to be the bottleneck operations in most CTs around the world (Carlo et al., 2015), this is because they are the primary problems to be dealt with, as they restrict the container terminals to reduce the turnaround time of the vessels and the operation costs.

The seaside operational area has three main sequential problems related to container terminals planning: the Berth Allocation Problem (BAP), the Quay Crane Assignment Problem (QCAP), and the Quay Crane Scheduling Problem (QCSP) see Figure 2-11. Furthermore, for linear shipping companies in the seaside area, there is a problem called the Stowage Planning Problem (SPP). The following sections will describe the definition of each problem, and we will then discuss more related works in the next chapter.

2.4.3.1.1 Berth Allocation Problem (BAP)

For busy container terminals, several incoming vessels arrive per week at different times. All of the vessels have to moor in the quay(s) of the terminal. The cost of constructing a quay which includes multiple berths is very high compared to the investment cost of other facilities in the terminal (Y.-M. Park & Kim, 2003). Therefore, the BAP is considered to be a critical resource for determining the

capacity of the container terminal. The problem question is, ‘when and where do incoming vessels have to moor at the quay?’

The BAP is the first problem that terminal operators need to solve when a vessel calls into a port. This is used to allocate a quay space (berth position) and service start time to an arriving vessel. In the BAP, we have the container terminal quay(s) layout and a set of incoming vessels that have to be served within a given planning horizon. The vessel’s information, such as length, draft, the expected time of arrival (ETA), desired berthing position, and the number of containers to be served/minimum cranes needed, can be given in advance by the shipping lines. Port planners need to schedule all incoming vessels by allocating a berthing position and time for each vessel such that all vessels must be moored within the quay length. It is not allowed for more than one vessel to moor in the same berth at a time. Figure 2-11.(a) represents a feasible berth plan to solve the BAP (Meisel & Bierwirth, 2013); the x-axis represents the time horizon and the y-axis represents the quay length, while the rectangles are the vessels scheduled to moor at a specific time and berthing position.

The BAP is also known as the *Berth Scheduling Problem*; see (Golias, Boile, & Theofanis, 2010; Golias, Portal, Konur, Kaisar, & Kolomvos, 2014; Kim & Moon, 2003; Y. Lee & Chen, 2009; Saharidis, Golias, Boile, Theofanis, & Ierapetritou, 2010; Y. Xu, Chen, & Quan, 2012), and also known as (*Berth Planning Problem*) see (Hendriks, Laumanns, Lefeber, & Udding, 2010; Legato & Mazza, 2001; Moon, 2000; Ruiz, Batista, & Vega, 2013; Song, Cherrett, & Guan, 2012; Theofanis, Boile, & Golias, 2009). This problem is considered to be an NP-hard problem by relating it to the set partitioning problem (Lim, 1998).

Various constraints might be considered when solving the BAP, which will be explained in section 2.4.3.2. The common objective function is minimising the vessel port stay time (vessel turn-around, also referred to as the vessel’s service time), which is the sum of the waiting and handling time of the vessel, the workload of the terminal resources, and the number of rejected vessels to be served at the terminal.

2.4.3.1.2 Quay Crane Assignment Problem (QCAP)

One of the most critical container terminal resources is the quay cranes. This is due to their high purchase and maintenance costs. The quay crane is responsible for loading and unloading containers to/from vessels. The volume of containers to be moved is known in advance for each vessel and should be sent by the ship operator to the terminals for each visit. The minimum number of quay cranes that can serve a vessel simultaneously might be contracted. The ship's operators have to ensure that the correct number of containers are moved per hour, as well as the number of assigned quay cranes to a vessel, as both directly affect the expected departure time of the vessel. The problem question is, 'how many and which quay cranes will be serving each vessel?'

The QCAP is to assign a number of quay cranes to vessels such that the required transshipment of containers can be fulfilled. In this problem, the planners have two decisions to make. The first decision is to determine the sufficient number of quay cranes to be assigned to each vessel, taking into consideration the availability of the minimum cranes needed at the berth for the berthing time of each vessel. The second decision is to determine the specific QCs (index) assigned in the first decision, as in Figure 2-11.(c), due to the fact that QCs are mounted on a rail which can move alongside the quay but cannot pass with another. The QCAP is also known as the *Crane Split Problem*, for the reason of the distribution of cranes to vessels (Voß et al., 2004).

Figure 2-11.(b) and Figure 2-11.(c) demonstrate the QCAP in the case where the quay's layout is continuous (described in section 2.4.3.2). The grey squares within a rectangular vessel indicate the assignment of QCs to the vessel in each time period. The total number of QCs serving berthed vessels in each time segment must not exceed the total QCs that the quay has. Moreover, this problem will not occur in the case of a discrete berth layout (described in section 2.4.3.2), where each berth is already fitted with a fixed number of cranes. The common objective function of this problem is to minimise the vessels' delays and to maximise the utilisation of the QCs.

2.4.3.1.3 Quay Crane Scheduling Problem (QCSP)

The QCSP is the third problem in the sequence of seaside operational planning problems. After a vessel is allocated a berth, moored and assigned a number of QCs to the vessel, the planners have to schedule the sequential operations of the QCs assigned for loading and unloading specific containers. The QCAP and the stowage plan (described next section) are the input of this problem. As mentioned before, each vessel has a container storage plan, identified by the tier and stack numbers. The unloading operation should identify the location of the containers to discharge, and similarly, the loading operation should also indicate the container's destination, weight and type. Therefore, the QCs scheduling tasks usually describe the bay(s) in which the workload is location. The problem question is, 'how to schedule the assigned QCs to load/unload containers from a vessel?'

The QCSP consists of scheduling the transshipment operations of a vessel by loading and unloading the containers using the QCs that are scheduled to it. In this problem, we are given a set of tasks that together represent the transshipment operations of a vessel, and a set of assigned quay cranes. The mission is to schedule the quay cranes that are assigned to the vessel in order to finish the operations as early as possible (Kim & Park, 2004).

Figure 2-11.(d) shows the QCSP representation, which illustrates each QCs scheduling workload in each vessel's bay in each time period. The crossing of QCs and the safety margins between them should be considered in the solving of this problem, and this reflects that if more QCs are assigned to a vessel, then there is more crane interference and less crane productivity. Practically, each QC can serve only one bay at a time. This problem is considered to be an NP-hard problem (Monaco & Sammarra, 2007) and the common objective function is to minimise the span of the QCs scheduling operations, which reflects the minimisation of vessel handling time.

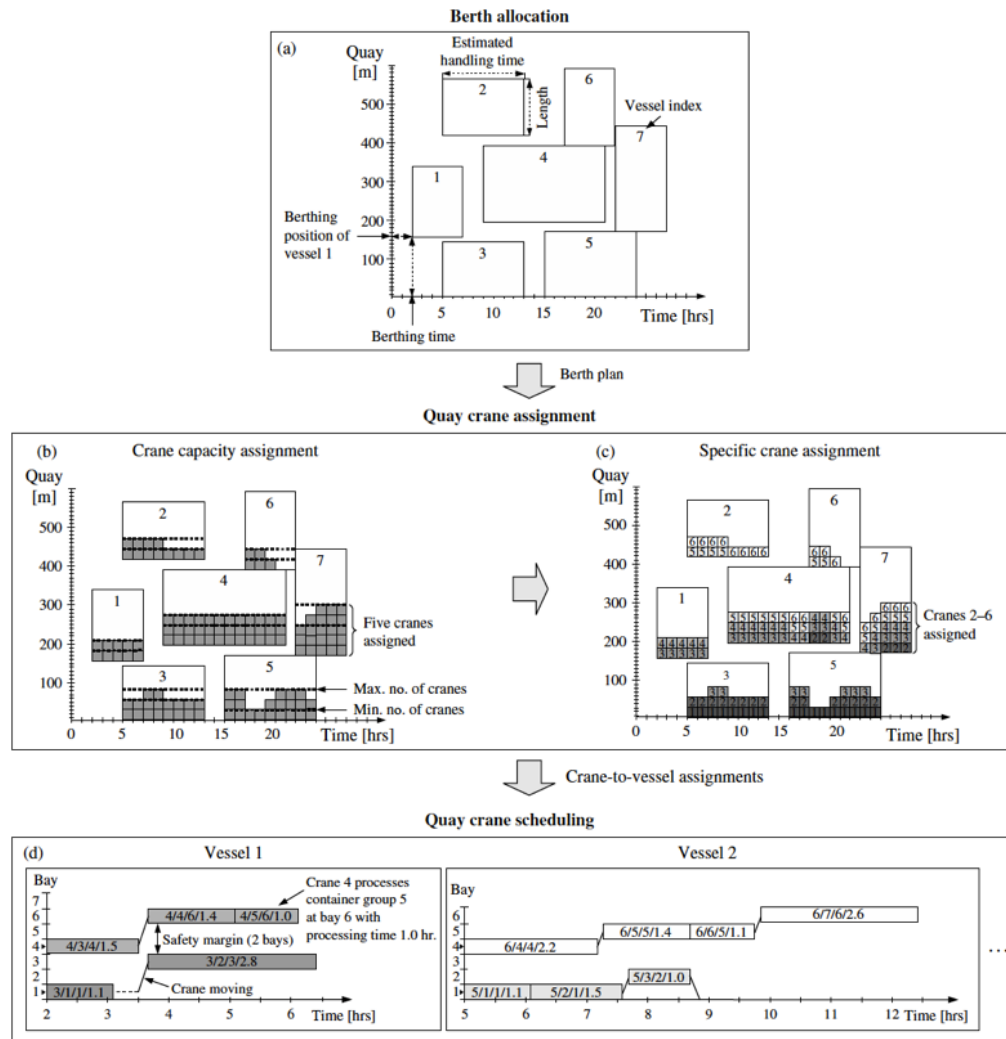


Figure 2-11: Seaside operational planning tasks at a container terminal. Source (Meisel & Bierwirth, 2013)

2.4.3.1.4 Stowage Planning Problem (SPP)

The SPP is focused on determining the exact position/slot (defined by a stack and tier number) of a vessel and assigning an export container to it. Shipping lines companies can be involved in this problem as they have to consider the following:

- (i) various measures should be considered for the stability of the vessel throughout its journeys such as trim, draft, and metacentric light;
- (ii) the information regarding the sequence of the port visit plan is known in advance and the expected number of containers and their types when it comes to loading or unloading for each port. The containers should be sorted and organised onboard to minimise the reordering of

the containers (reshuffling/re-handling); (iii) The information regarding the current containers onboard, their class/type and how they are organised considering the required restrictions. Containers should be categorised according to their class, type, size, and destination. The common objective of this problem is to maximise the utilisation of the vessel's capacity and to minimise the number of required reshuffling.

Ambrosino, Sciomachen, & Tanfani, 2004 (2006) studied the stowage planning problem and denoted it as the Master Bay Plane Problem (MBPP). Sciomachen & Tanfani (2007) formulated the MBPP as a three-dimensional bin packing problem; the objective is to minimise the total loading time and maximise the utilisation of the quay equipment. Imai, Sasaki, Nishimura, & Papadimitriou (2006) proposed a multi-criteria optimisation method for the problem of stowage, taking into consideration the number of containers that are re-handled and the ship's stability.

2.4.3.2 Seaside operational Problems' constraints

In the previous sections, we have presented the seaside operational problems. These problems can be solved individually or in an integrated manner. Therefore, in this section, we have collected information on most of the constraints that might affect the seaside operational problems, which should be considered while solving the problems themselves. It is worth stating that the problems in the seaside operational area are to be solved hierarchically, which means that the output of the BAP will set the constraints for the next problems (QCAP and QCSP), and that the output of the BAP and QCAP will set the constraints for the next problem, which is QCSP.

There are four common main constraints encountered in seaside operational problems: constraints related to vessel information, constraints related to timing, constraints related to berth layout and constraints related to the QCs.

- **Constraints related to vessel information:**
 - *Vessel length*: depending on the vessel's length, port operators allocate the vessel to the correct berth length in the quay (affects BAP).

- *Vessel draft*: this must be less than the berth depth. We should also consider the meteorological and tidal changes, and their impact on the berth depth (affects BAP).
 - *Expected Time of Arrival (ETA)*: the vessel's time of arrival could change depending on any delays that may occur (affects BAP).
 - *Expected Finishing Time (EFT)*: this is the time when the vessel should be expected to finish its services in the berth and be ready to leave. The EFT is estimated depending on the service time, which is the maximum waiting time plus the handling time for each vessel (described later) (affects BAP).
 - *Services required*: this could include the number of containers to be loaded or unloaded to/from the vessel (affects QCAP/QCSP).
 - *Quay cranes needed*: the number of quay cranes needed, which might be requested by the vessel operators in advance (affects QCAP/QCSP).
 - *Desired berthing position*: this is the specific berth place on the quay that the vessel operators might request to moor the vessel in (affects BAP).
- **Constraints related to timing:**
- Starting Time (Berthing time) (affects BAP).
 - *Static arrival*: there are no arrival times given for the vessels. It is assumed that vessels are already waiting in the port and can berth immediately; see Figure 2-12.
 - *Dynamic arrival*: fixed arrival times are given for the vessels, so then the vessels berth depending on their ETA; see Figure 2-13.
 - Handling Time: this is the time that will be taken to serve the vessel, i.e. loading/unloading the containers. In the literature, the researchers deal with this amount of time in different ways (Meisel, 2009a): (1) known in advance (fixed), (2) depends on the berthing position, (3) depends on the number of cranes assigned to the vessel, (4) depends on the work schedule of the assigned cranes and (5) a combination of (2), (3), and (4) (affects BAP).
 - Time window: the time horizon that the planner has assumed while scheduling and solving a problem (affects BAP/ QCSP).

- Ending Time (Completion Time): the exact time that the vessel should be finished and ready to leave the berth (affects BAP).

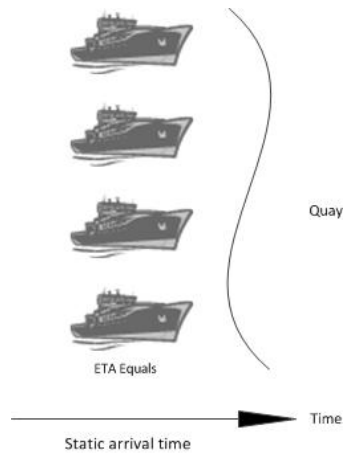


Figure 2-12: Static arrival time

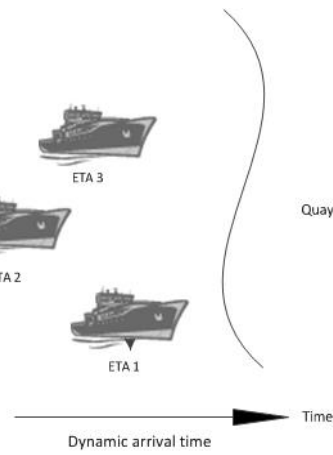


Figure 2-13: Dynamic arrival time

- **Constraints related to berthing layout:** (affects BAP)
 - *Discrete layout:* The quay could be partitioned into a number of sections called berths. Only one vessel can be served in a single berth at a time; see Figure 2-14.(a).
 - *Continuous layout:* There is no partitioning of the quay. Vessels can berth depending on their total length and the quay length. In this case, berth planning is more complicated than for a discrete layout; see Figure 2-14.(b).
 - *Hybrid layout:* like a discrete quay, but the vessels can be served in one or more berths depending on their length as shown in Figure 2-14.(c) and Figure 2-14.(d).

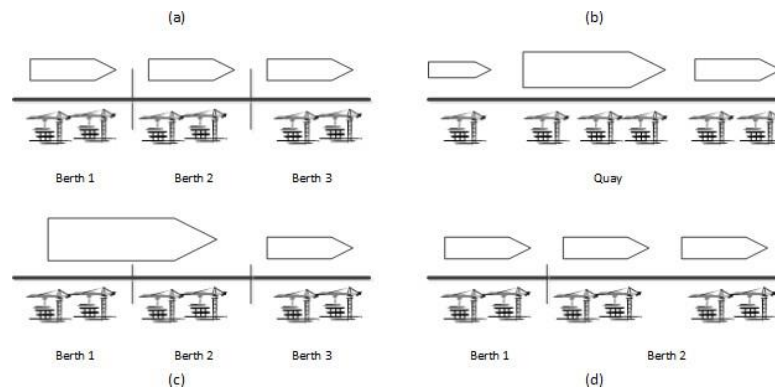


Figure 2-14: Types of berth layouts. (a) Discrete layout. (b) Continuous layout. (c, d) Hybrid layout. (Bierwirth & Meisel, 2010)

- **Constraints related to QCs:**

- *Time-Invariant QCs assignment:* assigning a constant number of QCs to a vessel during the handling time (see Figure 2-11.(b) Vessels 1 and 4) (affects QCAP/QCSP).
- *Time-Variant QCs assignment:* assigning a number of QCs to a vessel, which can change during the handling time (see Figure 2-11.(b) Vessels 2, 3, 5 and 6). The Time-Variant method of assignment is commonly used in practice due to its higher service quality and better utilisation of QCs. However, this problem is harder to solve (affects QCAP/QCSP).
- *QC tasks.* This refers to the transshipment operations needed by a QC due to a specific vessel's position (initial and final) with the starting and ending time expected. We also have to consider that each QC can serve only one bay at a time, and that there is a safety margin between any two adjacent cranes. Moreover, indexing the assigned QCs is essential, since the crossing of QCs over one another in the quay is impossible (Non-crossing). The task in the literature (Meisel, 2009a) can be defined with regard to the bay area or single bays, or with regard to the container stacks, container groups, or individual containers (affects QCSP).
- *QC timing:* this constraint is related to the crane timing attributes taken into account while scheduling tasks. It is defined by Meisel (2009a) as Ready times, QC Time windows, and the travel time between bays (affects QCSP).

2.5 Conclusions

In this chapter, we introduced the importance of the container terminal nowadays when it comes to facilitating large container transshipment around the world. We started by defining the most common terminologies used in the field, and then explained the systems and processes involved, the life cycle, and routine operations conducted in a container terminal. These processes raise many problems that need to be solved. Therefore, we provided an overview of the decision planning levels that the terminal operators follow in order to manage and control the container terminals. Regarding the complexity of the operations inside a container terminal, previous researches classified the problems for a more natural understanding and did so in order to solve the problems related to the classifications as introduced in this chapter.

For the decision planning levels (strategic, tactical and operational), we gave a brief description of the problems regarding each level as classified by the areas inside the container terminal. The main areas as described are the seaside, yard and landside. The seaside area is our focus on this research. Therefore, we gave more detail on this area and introduced the main specific problems which are the BAP, QCAP and QCSP. These problems have been tackled in the literature, either in an integrated or individual manner. As we have focused on tackling the problems using the integrated method, we collected the typical constraints that have a profound impact in order for them to be considered while solving the problems.

In the next chapter, we will focus on providing an in-depth literature review on seaside operational problems. This will enable us to highlight any gaps and weakness points that will be tackled later on, which in return will shed a light on our contributions in this research.

Chapter 3 : Literature Review

In the previous chapter, we defined the seaside operational problems and explained that these problems can be solved individually or in an integrated manner. There are many studies dealing with the stated problems, which means that the problems have attracted the interest of many researchers. In this chapter, we have provided a comprehensive literature review on the seaside operational problems. This research focuses on the integrated berth allocation and quay crane assignment problem (BACAP) with different constraints and methods involved. Therefore, we have given more attention to this problem in the literature related to its classifications, benchmarks, and the optimisation methods that have been used to solve it.

The chapter is organised as follows. In Section 3.1, an overview of the literature surveys conducted on the container terminal problems is presented. Section 3.2 provides a comprehensive literature review of the individual seaside operational problems classified by berth layout. The integrated BACAP-related work classified by single port or multiple ports is provided in Section 3.3. The most available benchmarks and the optimisation methods used to solve the single BACAP have been provided in Section 3.3.1.1 and Section 3.3.1.2 respectively. Finally, Section 3.4 has determined the literature gaps and concludes the chapter.

3.1 Overview

By exploring the surveys that have been published on container terminal problems, we have found (Stahlbock & Voß, 2008; Voß et al., 2004), described and classified the main logistics processes and operations in container terminals, and presented the solution methods used for optimisation. Rashidi & Tsang (2013) provided a survey on the container terminal problems and classified them into five scheduling decisions. They formulated the decisions as constraint satisfaction and optimisation problems respectively. Other studies presented surveys of container terminals but in the context of specific issues such as that by Angeloudis & Bell (2011), who provided a review of container terminal simulation models. Vis (2006) presented a

literature survey on the planning problems related to automated guided vehicle (AGV)-based terminals. Wiese, Kliewer, Suhl, & Str (2009) conducted a survey on container terminal characteristics and the modern equipment used for stacking operations in the yard area in different container terminals around the world.

To the best of our knowledge, there are a few surveys that have been published on seaside operational problems such as that by Bierwirth & Meisel (2010, 2015), Carlo, Vis, & Roodbergen (2014a, 2014b, 2013) and Theofanis, Boile, & Golias (2009). A comprehensive survey of BAP, QCAP and QCSP has given by Bierwirth & Meisel (2010), who proposed a classification scheme for the problem formulation based on that by Cordeau, Laporte, Legato, & Moccia (2005) and Imai, Sun, Nishimura, & Papadimitriou, (2005), which is similar to the scheme by Theofanis et al. (2009). Carlo et al., 2014a (2013) continued the work of Bierwirth & Meisel (2010). They provided a review and classification of the journal papers published between 2004 and 2012 on container terminal seaside operations using the keywords: *container*, *container terminal* and *port*, filtering them to the ones solving the BAP, QCAP, and QCSP. The paper also provides more attributes for classification and good future trends for further researchers to examine to improve seaside operational problems, including seaside layout and material handling equipment. They presented a similar review and classifications on the storage yard operations in Carlo et al. (2014b)

A follow-up survey to Bierwirth & Meisel (2010) has been presented in Bierwirth & Meisel (2015). The authors continued their work by classifying the new literature between 2009 and 2014 according to the features of the models considered for berth allocation, quay crane scheduling and integrated approaches. They used similar classification schemes to those proposed in their earlier study (Bierwirth & Meisel, 2010). We can observe from their survey that the numbers of research papers published between 2009 and 2014 increased with more than 79 new models proposed to solve the BAP and 52 models for QCSP, which means that these problems attracted the interest of many researchers. The scope of the problem has also increased, with an additional range of different constraints that reflects the reality faced by modern ports. This increase into the research of seaside operational

problems indicates its growing importance within the field of container terminal operations optimisation.

3.2 Related Work on Seaside Operational Problems

3.2.1 BAP Literature Review

There have been numerous studies on the BAP. Imai, Nagaiwa, & Tat (1997) modelled the BAP for the commercial ports in Asia and extended the model in their second study (Imai, Nishimura, & Papadimitriou, 2001). In the literature, the BAP was typically formulated by D. Xu, Li, & Leung (2012) as a combinatorial optimisation problem such as a parallel-machine scheduling problem with multiprocessor tasks. It has also been formulated as a 2D bin-packing problem (Zhi a. Hu, 2010). However, the formulation of the problem leads to NP-Hard or NP-complete problems, which require the use of heuristics and meta-heuristics to obtain solutions in an acceptable computational time (Golias, Boile, & Theofanis, 2006). Several heuristics and meta-heuristic approaches developed for the BAP have attracted enormous attention in academic research. For instance, Genetic Algorithms (GA) are used by Golias, Portal, Konur, Kaisar, & Kolomvos (2014), Simulated Annealing is used by Kim & Moon (2003) and Y. Xu, Chen, & Quan (2012), and Tabu Search is used by Lalla-Ruiz, Melián-Batista, & Marcos Moreno-Vega (2012) and Lee, Jin, & Chen (2012).

The most common goal of the BAP is to minimise the time for vessel turnaround and/or the total terminal costs. Due to the different terminal layouts and the different strategies used in port operations, researchers have started to classify the BAP (Imai et al., 2005). Figure 3-1 illustrates the latest classification for the BAP as presented by Bierwirth & Meisel (2015). The *spatial attribute* concerns the berth layout, *temporal attribute* describes the arrival process of vessels, while *handling time attribute* describes the way how the handling time of vessels is given as an input of the problem. Finally, the *performance measure attribute* considers the performance measures of the model. Most of these attributes have been explained in detail as seaside problems, constraints in the previous chapter.

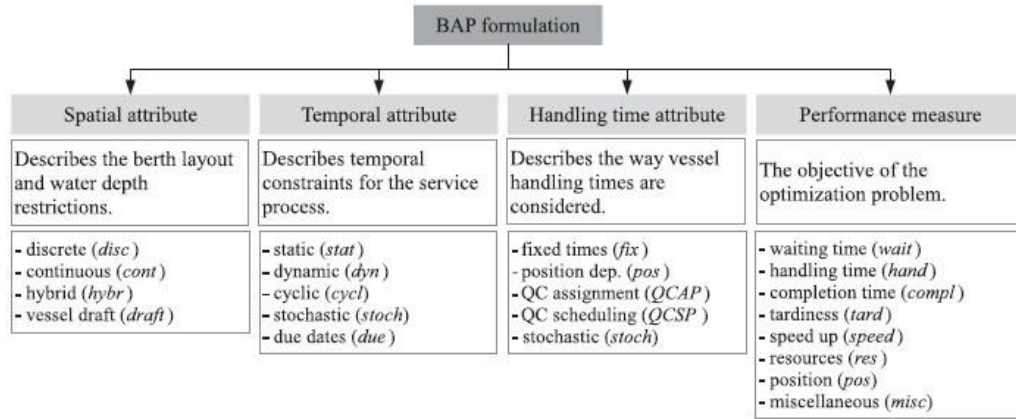


Figure 3-1: BAP classification scheme, (Bierwirth & Meisel, 2015).

The following sub-sections are the classification of the BAP in terms of the berth layout and its relationship with the other attributes. Berth layout has a crucial impact related to solving the BAP since it is the first constraint to scheduling incoming vessels. Moreover, we can introduce, from this point, the new BAP with multiple ports that will be described later on in section 3.3.2. We have also shown that this new problem has encountered less solutions and contributions from researchers in the literature.

3.2.1.1 Discrete layout

Emde, Boysen, & Briskorn (2014) studied the BAP as a discrete layout with a static vessel arrival time. They considered the handling time depending on the berthing position, and their objective was to minimise the completion time when serving vessels. They proposed an exact Branch-and-Bound method as well as developing a heuristic and Tabu search meta-heuristic approach to solve the problem. The exact method can optimally solve up to 18 vessels in a reasonable time. M. M. Golias, Boile, & Theofanis (2009) and Mihalios Golias et al. (2014) presented the BAP with similar constraints as Emde et al. (2014), but with a dynamic vessel arrival time and their objective function was to minimise the total waiting and handling time serving the vessels. They developed a GA to solve the problem. Imai et al (2001) studied the discrete BAP in the context of both static and dynamic vessel arrival time. The objective was to minimise the waiting and handling times of the vessels. A

Lagrangean relaxation-based heuristic was presented to solve the problem. Pengfei Zhou, Kang, & Lin (2006) considered vessel draft as a constraint for solving the discrete BAP in relation to the stochastic arrival and handling time of vessels. The objective was to decrease the total vessel waiting time. The authors proposed a GA to solve the problem.

3.2.1.2 Continuous layout

For continuous quay layout with dynamic vessel arrival, Ganji, Babazadeh, & Arabshahi (2010) proposed a GA to solve the BAP for a small and large-sized problem. The handling time was calculated depending on the berthing position, and the objective function was to minimise the total waiting and handling time. Lee, Chen, & Cao (2010) studied the problem with similar constraints but the handling time was considered to be fixed and their objective was to minimise the total weighted waiting and handling time. They developed two versions of the greedy randomised adaptive search procedure (GRASP). GRASP consists of two phases in each iteration, which are construction and a local search improvement to look for a near-optimal solution. Their method proposed identifying the possible berthing positions for the next vessel in the schedule. Z. H. Hu, Han, & Ding (2009) proposed a non-linear programming model and further proposed an immune algorithm to solve the continuous BAP. Guan & Cheung (2004) tackled similar problem constraints and developed a tree search procedure to minimise the total weighted port stay time of vessels. Moreover, F. Wang & Lim (2007) presented a stochastic beam search algorithm capable of solving instances with up to 400 vessels for this type of problem.

Continuous static BAP has been presented by Guan, Xiao, Cheung, & Li (2002) with fixed vessel handling times. They proposed a priority rule-based heuristic to minimise the weighted completion time of vessels.

Giallombardo, Moccia, Salani, & Vacca (2010) concluded that the continuous layout is much better when compared to the other quay layouts in terms of berth

space utilisation. In this regard, Lin, Ting, & Wu (2017) and Ting, Lin, & Wu (2013) solved the BAP as a continuous layout using simulated annealing.

3.2.1.3 Hybrid layout

For a hybrid quay layout such as Gioia Tauro port, (Cordeau et al., 2005) study focused on the BAP in this port where the nature of the quay in the middle was discontinued. They proposed a heuristic method to solve the problem. Imai, Nishimura, & Papadimitriou (2013) demonstrated the BAP of the indented terminal of the Amsterdam container terminal. The berth can accommodate one large vessel (mega-vessel) or several small ones as well as handling the mega-vessel from both sides. They proposed a channel terminal, which is similar to the indented terminal layout for handling mega-vessels from both sides, but it is also capable of avoiding the complexity of berthing small vessels. A GA was proposed to solve and compare the different layouts with the objective function of minimising the waiting and handling time of the vessels. They concluded that the channel terminal layout outperforms the indented terminal.

Kordić, Davidović, Kovač, & Dragović (2016) addressed the BAP through the static arrival of vessels and the presence of a fixed handling time in a hybrid quay. Their study was based on the model presented by Rashidi & Tsang (2013). They proposed an exact algorithm based on the mixed integer programming model in order to solve the problem. The algorithm can solve up to 60 vessels within the time limit of 1800 seconds.

C. Y. Cheong, Tan, Liu, & Lin (2010) studied the BAP using the Multi-Objective of makespan, waiting time and degree of deviation from the predetermined priority schedule. The dataset was generated randomly, and the solution was based on the Multi-Objective evolutionary algorithm incorporated with the concept of Pareto optimality. They used a fixed length chromosome which represented a fixed number of berths. In addition, they used five methods to solve the BAP which were berthing order decoding, assignment order decoding (which accepts the vessels assigning an earlier time than their ETA), berth exchange crossover, berth exchange

mutation and finally, local search exploration. They concluded that the three objectives used play an essential role in optimisation performance.

3.2.2 QCAP Literature Review

In practice, according to vessel length and the number of containers that are to be served, terminal operators need to determine the number of QCs to assign to a vessel. It may be contracted in advance with the vessel operators, as mentioned before. This number can vary between a minimum and maximum number of QCs. When the terminal operator decides to assign a number of QCs to a vessel, there are two ways to manage the operation of the QCs. The first is the number of assigned quay cranes which are fixed to the vessel handling time (time-invariant). This means that the fixed number of quay cranes assigned to a vessel is unchangeable during the vessel handling time. The second is time-variant, which means that the number of cranes assigned to a vessel can change during the handling time period. In practice, there are two significant reasons why the terminal operator cannot increase the number of QCs. The first one is the high cost of the purchase and construction of new QCs. The second is related to their structure, since they move on the same rail and cannot cross each other, so the interference between them will be very high.

In the literature, we found that this problem has barely received any attention from researchers concerning seeking to solve it individually, as it is not considered to be a difficult problem if it solved by the rule of thumb (Bierwirth & Meisel, 2010). However, it is usually solved when integrated with either the BAP (Chang, Jiang, Yan, & He, 2010; Z.-H. Hu, 2010) or with the QCSP (Diabat & Theodorou, 2014; Theodorou & Diabat, 2015) due to its high impact on the vessels' handling time. The main objective of the QCAP is to minimise crane productivity losses and to maximise their utilisation.

3.2.3 QCSP Literature Review

The QCSP was proven to be an NP-complete problem by Lee, Wang, & Miao (2008). The authors provide a mixed integer programming model and proposed a GA in order to solve the problem. In the literature, the QCSP was also addressed as a machine scheduling problem, where the QCs are the machines and the containers to be served are the jobs. This was done by considering a particular constraint to avoid crossing the cranes (Bierwirth & Meisel, 2010). Figure 3-2 illustrates the latest classification for the QCSP. *Task attribute* describes the aggregation of a vessel's containers into crane tasks, *crane attribute* is related to the proprieties of the crane resource as a whole, *interference attribute* indicates the restrictions on the movements of the cranes, and *performance measures* define the performance measures of the QCSP model.

In this research, we are not considering this type of problem. However, our future research may integrate this problem into our models. For more details on the QCSP classification scheme and the literature overview, see Bierwirth & Meisel (2015).

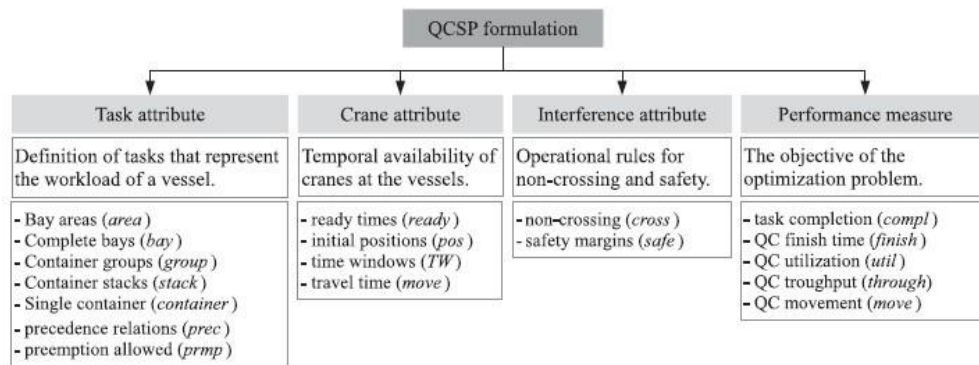


Figure 3-2: QCSP classification scheme, (Bierwirth & Meisel, 2015).

3.3 Related Work on the Integrated BAP and QCAP (BACAP)

3.3.1 Single Port BACAP

Individual seaside problems have been the subject of intensive research over the last few decades. However, in the literature, there is a limited amount of studies that seek to solve the integrated Berth Allocation Problem and Quay Crane assignment problem (BACAP). There is a direct impact of the distribution of cranes to vessels on the vessels processing time (Bierwirth & Meisel, 2010; Carlo et al., 2015). In the literature, a few studies refer to the BACAP as the Tactical Berth Allocation Problem (TBAP) (Giallombardo et al., 2010; Lalla-Ruiz, González-Velarde, Melián-Batista, & Moreno-Vega, 2014; Melián-Batista, Expósito-Izquierdo, Lalla-Ruiz, Lamata, & Moreno-Vega, 2013). These studies indicate that the problem is the NP-hard combinatorial optimisation problem, as its computational complexity increases with the increasing number of arriving vessels as shown by Cordeau et al. (2005) and Pinedo (2012).

Y.-M. Park & Kim (2003) solved the integrated BAP, QCAP and QCSP; their model suggests a two-phase solution procedure. The first phase determines the berthing position, berthing time, and the setting of the cranes at each vessel in the BACAP. The second phase, quay crane scheduling, is then constructed based on the results found from the first phase. A Lagrangean relaxation-based heuristic is used at the first decision level, and dynamic programming is applied in the second level to solve the integrated approach. Meisel & Bierwirth (2013) solved the integrated problem in another way. Since the container terminal process starts with BAP, and then QCAP and QCSP, the problem can be solved from back (QCSP) to front (BAP & QCAP), taking the productivity of the available cranes as the input data and using the constraints to solve the BAP and QCAP. The output of the BAP and QCAP is then used as the input data to resolve and adapt the final QCSP.

Container terminals are dynamic environments that are subject to uncertainty, with risks imposed on them by external factors due to changeable weather conditions, the breakdowns of QCs, changes in the vessel's expected time of arrival etc. Models

and algorithms can be adapted to consider these unexpected situations since they cannot be predicted in advance. (Mario Rodriguez-Molins, Ingolotti et al. (2014) formulated two conflicting objectives to solve the BACAP, which were minimising the total service time and maximising the robustness buffer time in order to accept uncertain situations. Their problem was solved by using the Mixed Integer Linear programming (MILP) model to minimise the service time using CPLEX, and using a GA model to maximise robustness using C++.

Ji, Zhu, Wang, Zhao, & Yang (2015) investigated the impact of changing the objectives of the terminal operator (TO) and vessel operator (VO). The VO needs their service to be completed in the minimum amount of time possible while the TO needs to ensure that there is a sufficient amount of profit for their port in order to maintain their market position goal in the wider context of port competition. In this regard, we can find that for the different sizes of incoming vessels and their requirements for service handling, the terminal berthing plan might be reliable and flexible enough to achieve both the TO and VO's goals. The different berth allocation scenarios based on the service strategies, which are first come first serve (FCFS), giving priority to small vessels (PTSV), and giving priority to large vessels (PTLV), can profoundly impact both the port's timespan and operational efficiency. In this research, we used Genetic Programming as the method to optimise the schedule of completed service time considering the TO and VO's goals, which results from the effective and robust composite dispatching rules used to solve the BACAP, as described in detail in Chapters 4, 5 and 6.

To the best of our knowledge, most of the research papers published on the BACAP have been shown in Table 3-1 and Figure 3-3, sorted by the date of publication starting from 2003 through to 2018. We used the latest surveys related to container terminal problems and the following keywords in the most publication search engines; "*Berth allocation*", "*Berth scheduling*", "*BAP*", "*BACAP*" and "*Quay crane assignment*". The search engines included Elsevier, Informs, Interscience, Palgrave, Springer, IEEE, Taylor and Francis, and Google Scholar. Moreover, we included the research studies that tackled the BACAP either separately or integrated with other problems. We found 102 research papers from 2000 to 2018; 46 of them

were published before 2013 when we started this thesis, and the remaining 56 were published in later years.

Table 3-1: Research papers published on the BACAP from 2000 to 2018.

Year	Reference	Year	Reference
2018	(Canrong Zhang, Wu, Qi, & Miao, 2018)	2013	(He, Huang, Chang, & Zhang, 2013)
2018	(Y. Liu, Wang, & Shahbazzade, 2018)	2013	(Dong, Hu, & Tao, 2013)
2018	(Agra & Oliveira, 2018)	2013	(Meisel & Bierwirth, 2013)
2018	(K. Wang, Zhen, Wang, & Laporte, 2018)	2013	(E. L. Ruiz, Izquierdo, Batista, & Moreno-Vega, 2013)
2018	(Krimi et al., 2018)	2013	(Vacca, Salani, & Bierlaire, 2013)
2018	(Z. Xu & Lee, 2018)	2012	(M. P. M. Hendriks, Armbruster, Laumanns, Lefebvre, & Udding, 2012)
2018	(Yuping, Yangyang, Yuanhui, & Tianyi, 2018)	2012	(X. Chen & Yang, 2012)
2017	(Hsu, Wang, Chou, Lee, & Wen, 2017)	2012	(Le, Wu, & Zhang, 2012)
2017	(Salhi, Alsoufi, & Yang, 2017)	2012	(Chunxia Yang, Wang, & Li, 2012)
2017	(Ç. Iris, Pacino, & Ropke, 2017)	2012	(Mario Rodriguez-Molins, Barber, Sierra, Puente, & Salido, 2012)
2017	(C. Iris & Lam, 2017)	2012	(J. H. Chen, Lee, & Cao, 2012)
2017	(Zhen, Liang, Zhuge, Lee, & Chew, 2017)	2012	(X. Liang, Li, Zhao, & Li, 2012)
2017	(Correcher & Alvarez-Valdes, 2017)	2012	(Salido, Rodriguez-Molins, & Barber, 2012)
2016	(El-boghdadly, Bader-El-Den, & Jones, 2016b)	2012	(C. Liang, Hwang, & Gen, 2012)
2016	(El-boghdadly, Bader-El-Den, & Jones, 2016a)	2011	(H. L. Ma, Chan, Chung, & Wong, 2011)
2016	(Türkoğulları, Taşkın, Aras, & Altinel, 2016)	2011	(Zhen, Chew, & Lee, 2011)
2016	(Shang, Cao, & Ren, 2016)	2011	(Ali, Abouelseoud, & Elwany, 2011)
2016	(Changchun, Canrong, & Li, 2016)	2011	(Xiaotao, Yuquan, & Qishuang, 2011)
2016	(Cavalcante, Oppen, Samer, & Urrutia, 2016b)	2011	(Raa, Dullaert, & Schaeren, 2011)
2016	(Cavalcante, Oppen, Samer, & Urrutia, 2016a)	2011	(Meisel, 2011)
2016	(J. Yang, Gao, Liu, & Liu, 2016)	2011	(Blazewicz, Cheng, Machowiak, & Oguz, 2011)
2016	(He, 2016)	2011	(C. X. Yang, Wang, & Yang, 2011)
2015	(F. Li, Sheu, & Gao, 2015)	2011	(C. Liang, Guo, & Yang, 2011)
2015	(Karam & Eltawil, 2015)	2011	(Zeng, Hu, Wang, & Fang, 2011)
2015	(C. Iris & Pacino, 2015)	2011	(Zeng, Yang, & Hu, 2011)
2015	(Alsoufi, Yang, & Salhi, 2015)	2011	(Lu, Han, & Xi, 2011)
2015	(Said & El-Horbaty, 2015)	2010	(Canrong Zhang, Zheng, Zhang, Shi, & Armstrong, 2010)
2015	(Pan & Xu, 2015)	2010	(Vacca, Salani, & Bierlaire, 2010)
2015	(M. Z. Li et al., 2015)	2010	(M. Hendriks et al., 2010)
2015	(Frojan, Correcher, Alvarez-Valdes, Koulouris, & Tamarit, 2015)	2010	(Bierwirth & Meisel, 2010)
2015	(Zhi-hua Hu, 2015)	2010	(Chang, Jiang, et al., 2010)
2015	(Hsu, 2015)	2010	(Giallombardo et al., 2010)
2015	(Ji et al., 2015)	2010	(Zhi a. Hu, 2010)
2014	(Lalla-Ruiz et al., 2014)	2010	(Chun Yew Cheong, Habibullah, Goh, & Fu, 2010)
2014	(Türkoğulları, Taşkın, Aras, & Altinel, 2014)	2010	(Chang, He, & Zhang, 2010)
2014	(Q.-M. Hu, Hu, & Du, 2014)	2010	(Chun Yew Cheong et al., 2010)
2014	(Ursavas, 2014)	2010	(Han, Lu, & Xi, 2010)
2014	(Karam, ElTawil, & Harraz, 2014)	2009	(C. Liang, Huang, & Yang, 2009)
2014	(Aras, Türkoğulları, Taşkın, & Altinel, 2014)	2009	(Meisel & Bierwirth, 2009)
2014	(H. Ma, Chan, & Chung, 2014)	2009	(Meisel, 2009a)
2014	(Mario Rodriguez-Molins, Ingolotti, et al., 2014)	2009	(C. Liang, Lin, & Jo, 2009)
2014	(M. Rodriguez-Molins, Salido, & Barber, 2014)	2009	(Na & Zhihong, 2009)
2014	(Türkoğulları, Taşkın, Aras, & Altinel, 2014)	2009	(He, Mi, Chang, & Yan, 2009)
2014	(Gao, Cao, & Zhao, 2014)	2008	(Peng-fei Zhou & Kang, 2008)
2014	(Xiao & Hu, 2014)	2008	(Giallombardo, 2008)
2013	(Shen & Ko, 2013)	2008	(Maarten Hendriks, Marco Laumanns, Erjen Lefebvre, 2008)
2013	(Zampelli, Vergados, Van Schaeren, Dullaert, & Raa, 2013)	2008	(Legato, Gulli, & Trunfio, 2008)
2013	(Rashidi & Tsang, 2013)	2007	(Lokuge & Alahakoon, 2007)
2013	(Elwany, Ali, & Abouelseoud, 2013)	2006	(Meisel & Bierwirth, 2006)
2013	(Chutian Yang, Wang, & Zheng, 2013)	2006	(J. Liu, Wan, & Wang, 2006)
2013	(Sheikholeslami, Ilati, & Hassannayebi, 2013)	2003	(Y.-M. Park & Kim, 2003)

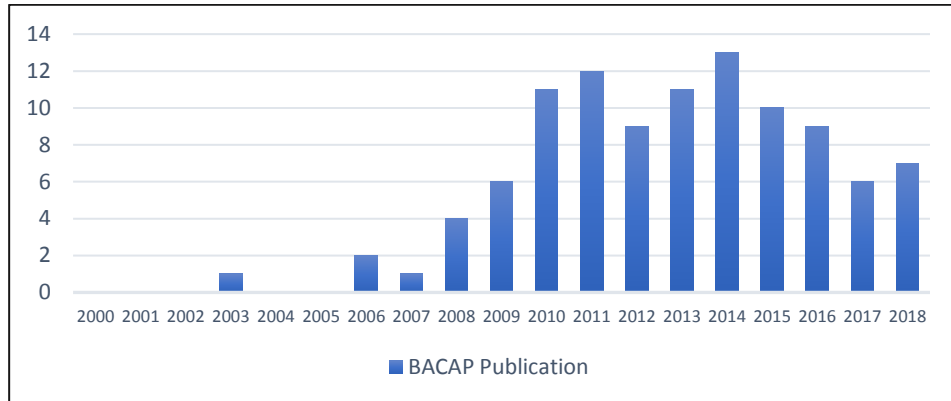


Figure 3-3: Number of research papers for the BACAP by year of publication.

3.3.1.1 Benchmarks

In the literature, there have been different benchmarks used by the researchers. Most of the benchmarks are either (i) real data imported from a port as a case study or (ii) generated data based on real data from a container terminal, considering the standard measures and average number of vessels arriving either per day or per week. In this section, Table 3-2 shows a summary of the benchmarks that have been used in the literature or updated by the researchers and that are available for the BACAP.

In our research, we selected (i) The benchmark provided by Mario Rodriguez-Molins, Ingolotti, et al. (2014) in order to solve the continuous dynamic BACAP. This is because it includes a large number of vessels and fewer constraints. We used this benchmark to verify the performance of the proposed model, and to test the composite dispatching rule concept generated by genetic programming as presented in Chapter 4; (ii) the benchmark provided by Meisel & Bierwirth (2009) in order to solve the continuous dynamic BACAP with desired berthing position as presented in Chapter 5; (iii) we generate a benchmark based on the benchmark in (ii) in order to solve the multi-port continuous dynamic BACAP with the desired berthing position as presented in Chapter 6.

Table 3-2: Summary of the BACAP literature benchmarks.

Author	(Mario Rodriguez-Molins, Ingolotti, et al., 2014)	(Meisel & Bierwirth, 2009)	(Y.-M. Park & Kim, 2003)	(Lalla-Ruiz et al., 2014)	(Giallombardo et al., 2010)
Benchmark	- Generated.	- Real data: based on (Y.-M. Park & Kim, 2003) - Generated.	- Real Data: Pusan Eastern Container Terminal (PECT) in Pusan (Korea). - Generated.	- Real Data: based on (Giallombardo et al., 2010);(Vacca et al., 2013) - Generated.	- Real Data: Medcenter Container Terminal of Gioia Tauro (Italy)
Size	Includes 100 vessels /day, for 100 days instances	Includes 30 instances, 20 vessels,30 vessels, and 40 vessels with 10 instances for each.	Real Data: 25 instances; (13-20 vessels). Randomly generated: 50 instances (20-40 vessels).	Generated: 3 sets of 5 instances with 10 possible profiles per vessel.	60 vessels, 12 instances for 6 classes of instances
Input Data	- Vessel ID. - Length of the vessel. - Arrival time of the vessel. - Number of movements (or containers) to be unloaded/loaded. - Priority of the vessel ($1 \leq p \leq 10$ where 10 is the highest priority, and 1 is the lowest one).	- Vessel ID. - Length of the vessel. - Desired berthing position of the vessel. - Crane capacity demands of the vessel (in QC-hours). - Expected time of arrival. - Earliest starting time. - Expected finishing time. - Latest finishing time. - Minimum number of cranes to assign. - Maximum number of cranes to assign. - Vessel's costs	- Expected time of arrival. - Length of the vessel. - Total operation time of cranes to handle the vessel. - Expected finishing time. - The least-cost berthing location. - The container handling cost per unit distance between berth and yard. - The penalty cost of the vessel per unit time of arrival.	- Set of vessels - Set of berths - Set of time step - Set of quay crane profile - Service time of vessel - Maximum number of QCs available - Housekeeping cost/container - Expected time of arrival. - Earliest starting time. - Start/end of the availability of a berth.	- Expected time of arrival. - Minimum number of cranes to assign. - Maximum number of cranes to assign. - Number of movements (or containers) to be unloaded/loaded.
Assumptions	- Quay length: 700 meters - There is a safe distance between two moored ships. Assume that each vessel has a 2.5% of this length at each side as a safe distance. - Available Quay Cranes: 7 - The maximum number of assigned QCs by a vessel depends on its length and not exceed the maximum number of QCs that the container terminal allows per vessel (5 QC), - Safe distance between two adjacent QCs (35 meters). - Movements of the QC per time unit: 2.5 moves/time unite	- Quay length: 1000 meter. - Time Horizon = 168 hours - Number of available cranes = 10 - QCs interference exponent = 0.90 - Berth deviation factor = 0.01 - Cost per QC-hour = 0.10	- Quay length: 1200 meter - Time Horizon = 300 hours - Total number of cranes = 11 - The maximum number of cranes = 5 - The minimum number of cranes = 2	- The service time of each vessel is determined by the QC profile assigned to it. - The vessels is divided into a number of classes and subclasses with a percentage of exchange number of containers.	- Quay length: 3395 meters. - Quay cranes available: 25 - Quay crane productivity: 24 containers/hour. - They consider six classes of instances: o 10 ships and 3 berths, 1 week, 8 quay cranes; o 20 ships and 5 berths, 1 week, 13 quay cranes; o 30 ships and 5 berths, 1 week, 13 quay cranes; o 40 ships and 5 berths, 2 weeks, 13 quay cranes; o 50 ships and 8 berths, 2 weeks, 13 quay cranes; o 60 ships and 13 berths, 2 weeks, 13 quay cranes.

3.3.1.2 Optimisation Methods

There are a variety of methods that are able to be used to solve the BACAP. Since the BACAP is widely known to be an NP-Hard problem (Y.-M. Park & Kim, 2003), we found in the literature that a few researchers have proposed an exact method for optimisation. They also use this method to test a small number of instances for the purpose of verifying their models. In contrast, several heuristic and meta-heuristic methods (approximation methods) have been developed to solve the BACAP for a large number of instances.

3.3.1.2.1 Exact methods

In computer science and operations research, the exact methods are well-known and used for solving optimisation problems. There are algorithms that allow for the finding of an optimal solution to a problem, but they are time-consuming when the problem becomes more complicated, such as problems with hardly constrained or time-varying problems (Festa, 2014).

Vacca et al. (2013) proposed a model to solve the BACAP based on an exponential number of variables, solved via column generation. They implemented an exact branch-and-price algorithm in order to produce optimal integer solutions along with several accelerating techniques developed to solve the problem. Ursavas (2014) studied discrete dynamic BACAP. The branch-and-cut algorithm was used to solve the real problem for the port of Izmir in Turkey.

Mixed Integer Linear Programming (MILP) is mostly used as an exact method to solve the BACAP using mainly CPLEX as the solver. For instance, Aras et al., (2014); Q.-M. Hu et al., (2014) solved the continuous dynamic BACAP; while M. Hendriks et al. (2010) solved the continuous cyclic BACAP using MILP.

3.3.1.2.2 Approximation method

Heuristic and meta-heuristic methods are approximated methods, since they seek to produce good quality solutions to difficult problems in a reasonable amount of computation time. They are powerful methods and flexible to use to search for near-optimal solutions (Festa, 2014). The main difference between these two methods is

that meta-heuristics consider the information collected during the search for the solution space, to further direct the search process. They have a mechanism to avoid getting stuck in local optima.

The most frequently used meta-heuristic method in the literature for solving the BAP and BACAP is the Genetic Algorithm (GA). C. Liang et al. (2011) and C. Liang, Huang, et al. (2009) proposed a GA to solve the discrete dynamic BACAP with the aim of minimising vessel handling time, waiting time, and delay time. Chang, Jiang, et al. (2010), Mario Rodriguez-Molins et al. (2012) and Chunxia Yang et al. (2012) solved the continuous dynamic BACAP using GA. Han et al., (2010); He et al. (2009); Lalla-Ruiz et al. (2014); Lu et al. (2011); Peng-fei Zhou & Kang (2008) have also used a GA to solve the BACAP.

Giallombardo et al. (2010) introduced the so-called Tactical Berth Allocation Problem. They developed a heuristic algorithm which combined Tabu search methods and mathematical programming techniques in order to solve the discrete dynamic BACAP. Moreover, Zeng, Hu, et al. (2011) and Zeng, Yang, et al. (2011) used the same Tabu search method to solve the continuous dynamic BACAP. Meisel & Bierwirth (2009) proposed a squeaky wheel optimisation and Tabu search method in order to solve the BACAP with a variable-in-time or time-variant QC related to vessel assignment. They considered that crane productivity depends on the berthing position of the vessels. The objective is to minimise the service costs plus the operational costs of the utilised QC-hours. The outcomes deliver significant improvements against the solutions reported by Y.-M. Park & Kim (2003).

Elwany et al. (2013) proposed an integrated heuristics-based solution methodology using Simulated Annealing (SA) in order to solve the BAP and QCAP simultaneously. They assumed there to be a continuous berth layout with variable water depth along the quay, with dynamic vessel arrival. The results show high quality in relation to the reasonable computational time when compared with CPLEX as an exact method.

The Greedy Randomised Adaptive Search Procedure (GRASP) is a meta-heuristic algorithm that has also been used by researchers in the literature to solve the

BACAP, like those used by Mario Rodriguez-Molins, Salido, & Barber (2014) and Salido et al. (2012).

3.3.2 Multiple Ports BACAP

In this section, we investigated the BACAP considering the setting of multiple ports/terminals under the control of one-port operator. The rapid increase in the number of containers transhipped between countries and the number of container vessels overall has lead to port managers increasing the number of container terminals, either in the same port or by building new ports. Furthermore, container terminal congestion may happen in the case of uncertainty of quay crane failure, insufficient berthing capacity or when shutting down of one terminal for any reason. Therefore, port operators must have an alternative plan for such a situation.

Most of the published studies in the literature on the BAP or BACAP consider there to be a single terminal as described in the previous sections. Some research studies were case studies on a specific port, which have a specific layout. However, a few of these studies consider multiple terminals managed by one port operator for the BAP. In this case, the problem becomes more complicated, since the port operator might have to allocate a berth space, starting time, and a specific terminal/quay for every vessel. This problem is known as multiple ports BAP.

We widely investigated this phenomenon by using online search engines to look for papers, research studies and thesis' that contained the following keywords: "*Berth allocation*", "*Berth scheduling*" and "*Terminal operations*" in combination with "*Multi/multiple*" and "*terminal/port/quay*". The online search engines included Elsevier, Informs, Interscience, Palgrave, Springer, IEEE, Taylor and Francis, and Google Scholar. We also searched the citations of the previous relevant papers. We have included the survey papers related to the subject as well. To the best of our knowledge and to discern the outcome of the above investigation, we found that the BACAP in the context of multiple terminals/quays has not been studied enough in the literature. We classified these studies according to the planning decision levels as follows.

At the strategic level, M. P. M. Hendriks et al. (2012) considered multi-terminals within the same port in order to solve the cycling of calling vessels in the BACAP. Their objective was to minimise the costs associated with each QC in order to balance the QC workload and to further minimise the inter-terminal container transportation cost. Their solution was based on mixed-integer programming. They implemented the solution for real data through the terminal operator of PSA Antwerp.

At the tactical level, Lee et al's (2012) work resembled the study of M. P. M. Hendriks et al. (2012). They included the storage yard allocation for transshipment flows rather than the QC workload. Moreover, they extended the study of Moccia & Astorino (2007), where the aim was to minimise the total inter-terminal and intra-terminal handling costs that resulted from transshipment flows. They developed an integer programming model and a two-level heuristic algorithm to solve the problem. For the continuous BAP with multiple quays at the tactical level, Huang, Suprayogi, & Ariantini (2016) developed an integer programming model to determine the berthing windows of calling vessels within a planning horizon in a cyclical way. The aim was to minimise the sum of the service starting time deviations related to the desired time. They developed two heuristic methods in the case where there was increasing port demands.

At the operational level, multiple terminals in the same port was considered by Imai, Nishimura, & Papadimitriou (2008). They formulated an additional terminal only in the case of the main terminal no longer being able to handle the expected number of incoming vessels at a point in time. In this case, a number of vessels were reallocated to a nearby terminal to minimise the total vessel service time. Frojan et al. (2015) studied continuous BAP in a container terminal with multiple quays. Their objective was to minimise the total assignment cost (waiting cost, delay cost, vessel assignment cost to the quay, and the vessel deviation cost if it was moved from its desired berthing position in the quay). They assumed that the handling time for each vessel was known and independent of the berthing position. Moreover, they assumed there to be a high cost for the condition preventing the vessel from mooring at a given quay. The experimental study was based on the dataset provided

by K. T. Park & Kim (2002) and Cordeau et al. (2005), with the updated version able handle multiple quays. They solved the problem for large instances, using GA and priority rules to build the initial population. The algorithm that they used was based on two constructive algorithms (exploratory and analytic). Finally, the local search algorithm was used to improve the solution.

Dadashi, Dulebenets, Golias, & Sheikholeslami (2017) proposed a new mathematical model for berth allocation and the scheduling of vessels at multiple marine container terminals in the same port. They modelled the multiple terminals by combining the available berthing space of each terminal into a single wharf. Their model extends the model developed by Kim & Moon (2003) with the objective of minimising the total weighted delayed departure time of the vessels. They considered the problem to be a continuous berthing problem and studied the tidal effects on berth scheduling. They solved the problem using CPLEX. The benchmark used was from the port of Bandar Abbas (Iran), which has two terminals that differ in length and depth operated by one port operator. They grouped the data into 27 instances and conducted three experiments. The first one assumed there to be a first-come-first-serve priority weight to compare it with the current operations ongoing in Bandar Abbas port. The outcome was a reduction in all 27 instances in the delay of the departure. They observed that many factors affect the delay of vessels such as vessel size, storage yard utilisation and average quay crane production. The second experiment studied four sets of weights and their impact on the results. The final experiment evaluated the effect of increasing access channel depth on a vessel's delayed departure. In their research, we can observe that the authors did not explain the method of dividing the vessels between the multiple terminals and the impact of sending one vessel to another terminal in the case of there being a tidal effect. Their objective was to minimise the total delay time only while not considering the cost of changing the vessel's desired berthing position, which has a high impact on the results.

Zhen, Wang, & Wang (2016) investigated the BACAP for one port with multiple terminals in the transshipment hubs. The model was formulated to minimise the bunker consumption and inter-terminal transfer costs for the containers. A local

branching-based method and particle swarm optimisation were developed to solve the problem.

As explained before, berth layout can be divided into three types: continuous, discrete, and hybrid. If we study the hybrid layout, we can find that this type is quite similar to the multi-quay problem. This is as the quay is divided into berths, and each berth can handle two small vessels. One large vessel can occupy two berths. However, in the case of multiple ports, the problem is much more complicated since each port has multi-quay/terminal, and each quay in a different port has its own respective characteristics and an associated number of QCs. The quay crane in the multi-quay is independent, and there is a space for each vessel to moor in different berth other than its desired. Each container has a desired yard position (inter-terminal). Therefore, there is a cost to transferring containers from the vessel's berthing position in one terminal to another, or from one port to another. All of these constraints might be considered when solving this type of problem.

The work of Imai et al. (2013) and S.-W. Lin & Ting (2014) is an excellent example of a hybrid dynamic BAP. The work was done by Türkoğullari et al. (2014b) on a hybrid BAP while considering the QCAP is another great example. Türkoğullari et al. (2014b) solved the BACAP and extended the work in order to handle berth allocation and quay crane specifics, which determines the specific cranes that will work on a vessel. They considered the hybrid layout, where the berth is divided into sections so then the vessel can be allocated to one or more of the berth sections. Moreover, they considered dynamic vessel arrival. Their objective was to minimise the costs of waiting, position and tardiness. They assumed there to be a time-invariant assignment to the QCs, while the number of QCs did not change during the vessel's stay in the berth.

Chun Yew Cheong et al. (2010) considered the BACAP through the use of multi-objective optimisation in order to minimise the handling and waiting time. They also used a hybrid layout port, which consisted of 23 berths and 87 QCs. By solving the problem with multiple objectives, the solution emphasised on the search for a Pareto-optimal set of solutions. The authors used the same method as used in C. Y.

Cheong et al's study (2010), including referencing the QCAP solved using constructive and improvement heuristics.

3.4 Research Gaps and Conclusion

Although seaside operational problems have received much attention in the literature, we have identified a few gaps that this research can fill in; the rest can be future research trends. In this section, we have provided the conclusion of the studied problems in the literature and an insight into the research gaps.

It was noted that seaside operational problems consist of many constraints that may vary and change from one port to another. These constraints have a crucial impact on the solution method and how the problem is solved. Therefore, in this chapter, we have started to provide an overview of the literature, which aims to classify the problems concerning the constraints. These classifications can allow future researchers to identify their problems and constraints clearly and to facilitate their comparisons with similar studies.

Several studies have tackled seaside operational problems independently. However, these problems are interrelated, and there is still an insufficient number of studies that has sought to solve them in an integrated manner. This research consisted of a deep integration of BAP and QCAP, solving them dependently.

As we have focused this thesis on the BACAP, we reviewed the literature on the BAP as classified by berth layout, such as discrete, continuous, and hybrid. Consequently, we have provided a comprehensive body of literature on the BACAP regarding its classifications, benchmarks and the methods that have been used to solve the problem. We can observe that most of the previous studies tackle the BACAP as a single port/terminal. Therefore, we have reviewed the literature for this problem in the context of a single port, and then progressed to analysing it as if in multiple ports.

We have found in common that most of the researchers have concluded that the current benchmarks that are available for the BAP/BACAP are not general enough. They advised finding general benchmarks that include different information to

handle different types of problems. Moreover, it is preferred that the benchmarks are quite similar to real container terminal data. These benchmarks will provide the best method and a better solution when the researchers go to compare their work with one another. We have presented the five most common benchmarks that were used in the literature and the ones that we used in our experiments in this thesis.

Moreover, it can be observed that the BACAP was solved using different methods, either exact or heuristic. The heuristic methods have more of a share of the literature regarding the complexity required to solve the problem with the exact methods. We have discussed both methods, and found that the preferable way to verify any model is to use the exact method for small instances and then to compare the outcome with the results of the heuristic methods. This is as the heuristic methods do not guarantee optimality like the exact ones. We found that although some methods can provide excellent solutions, they are not flexible enough to cope with the practical requirements of TO and VO concerning giving priority to incoming vessels.

In the literature, we noticed that most of the researchers tried to find a better solution using different methods, but no-one considered finding a better solver instead of finding a better solution, which is not capable of handling future constraints and different port layouts.

We considered most of these issues in our research, by innovating an algorithm in order to find better dispatching rules using genetic programming to solve the BACAP. The resulting outcomes of this algorithm are the best solver, and thus can produce a better solution. This solver can easily be used to solve similar problems in the future with different constraints, providing a near optimal solution within an acceptable amount of computational time.

Finally, we explored the literature for the works that have been conducted related to the new problems that arise in the container terminals, which is the multiple ports BACAP. We have found that this type of problem is not considered often enough in the literature and this motivates us to study this type of problem in detail.

Chapter 4 : A Genetic Programming

Algorithm for the Berth and Quay Crane Allocation Problem

4.1 Introduction

In container terminals, most of the operations are strongly interdependent. As mentioned previously, seaside operations face three main problems, BAP, QCAP and QCSP. These problems can be solved independently with a small number of instances (e.g. a small number of vessels with few constraints) as this may only have a slightly negative impact on the performance of the solution. However, this is not the case for highly constrained large instances, as the first two problems are highly dependent on one another. The number of available QCs depends on when and where the vessel is berthed, and the berthing handling time varies depending on the number of QCs assigned (W. Li, Wu, & Goh, 2015). Bierwirth & Meisel (2010) determined that treating the BAP and QCAP as an integrated problem could improve overall performance by 34%.

This chapter focuses on the first two integrated problems, the Berth Allocation problem and Quay Crane Assignment Problem (BACAP) as in Figure 4-1 with no desired berthing position known in advance. The objective is to minimise the total service time of all vessels. The overall goal is to optimise the seaside container terminal operations by solving the BACAP as one problem, which in turn improves the container terminal throughput.

Finding an effective vessel dispatching rule for a given problem is not a trivial task, as it is time consuming, and requires expert knowledge. Current constructive methods used to solve the BACAP are based on simple/standard priority rules, and the methods used to find a solution rather than finding a better solver. Our objective is to develop a new intelligent algorithm which uses a new priority-based

scheduling method to solve the problem using the Genetic Programming (GP) approach. GP is used to automatically evolve dispatching rules for the BACAP depending on the problem constraints and circumstances and it is able to find a good solver that can cope with different situations. In addition, the GP has advantages over traditional fixed-length chromosomes and the limitations of the genetic algorithm (GA) approaches (Koza, 1992).

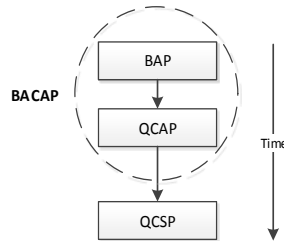


Figure 4-1: The BACAP

A comparative study of Standard Priority Rules (SPRs) and Composite Dispatching Rules (CDRs) has been presented in this chapter. CDR has been shown to be more efficient and flexible when it comes to meeting the needs of both terminal and vessel operators adequately.

The contributions of this chapter are the following. (1) A novel genetic programming-based approach to evolve the dispatching rules for the BACAP problem that outperform other standard priority rules. (2) The “self-adaptability” of the proposed method; since almost all container terminal ports have distinctive characteristics, and so the performance of DR-based schedules varies significantly from port to another, and therefore, it is important to tune and select the best-performing DRs manually. The proposed GP approach is “self-adaptable”, in which it automatically discovers/evolves high performing DR using different sets of variables based on what is available in each berth (3) We have developed an independent scheduler for the BACAP (*BACAP_Scheduler*) that can be combined with any appropriate optimisation method. (4) Provided an analysis of a wide range of DRs and compared the GP approach with a well-known large BACAP benchmark.

The chapter is organised as follows. In Section 4.2, the problem description has been described. The mathematical model with its notations and assumptions has been presented in Section 4.3. The complete description and survey of the used dispatching rules and composite dispatching rule to solve the BACAP has been presented in Section 4.4. In Section 4.5, the proposed algorithm *BACAP_GP* to solve BACAP has been presented. The extensive computational results have been presented and analysed in Section 4.6. In Section 4.7, we concluded the chapter and presented the future research directions.

4.2 Problem Description

In the BACAP, there are three main factors: (1) a set of incoming vessels (vessel list), where each vessel has several attributes such as length, expected time of arrival, the number of containers to be loaded/unloaded, and the minimum QC contracted between the Terminal Operators (TO) and Vessel Operators (VO) to serve the vessel; (2) the container Terminal (quay), where the quay has spaces (berths) to accommodate the vessels and the individual characteristics vary from port to another, such as quay length and depth and (3) the number of available QCs on the quay to load and unload containers to/from the vessel.

To solve the integrated BACAP for a single quay in a port, we are required to allocate a time slot within a planning horizon and a berthing space on the quay to incoming vessels, taking into consideration that at least the minimum number of their QCs needed are available. Once a berth is set as being occupied for a vessel, no other vessel can occupy the same berth at the same time. Then, we assign a set of QCs to serve the vessel during its stay in the berth.

Figure 4-2 demonstrates the BACAP problem. We have five vessels; the third and fifth one have the same expected time of arrival, and the rest have different times. These vessels should be scheduled to berth and be serviced by the QCs. The quay is shown as a single quay with four QCs. A quay could have one or more berths depending on the quay length and layout.

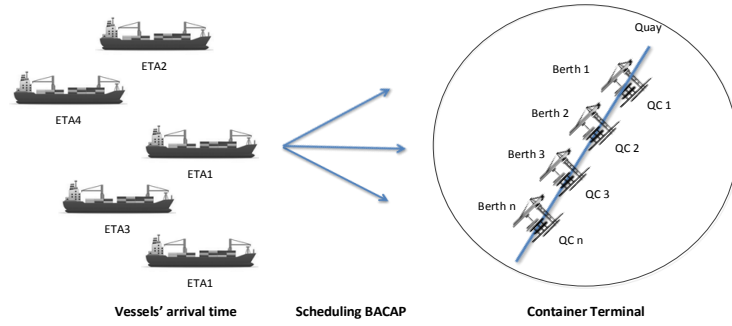


Figure 4-2: The single-quay BACAP problem description. QC: Quay Crane, ETA: Expected time of arrival

4.3 Mathematical Model Formulation

The general goal of container terminal's berth planning is to provide fast and reliable services of vessels. This is reflected in the literature by various objective functions. The most common objective is to minimise the sum of the waiting and handling time of vessels (service time). Further objectives are, for instance, the minimisation of the workload of terminal resources and minimisation of the vessels rejected to be served at a terminal (Meisel, 2009b).

In this study, the BACAP was modelled as a single objective function with the aim of minimising the total service time for all vessels as shown in equation (4.1). Figure 4-3 represents a vessel coming in to berth and the associated time-space diagram. The notations used in the diagram will be explained in detail in the following sections.

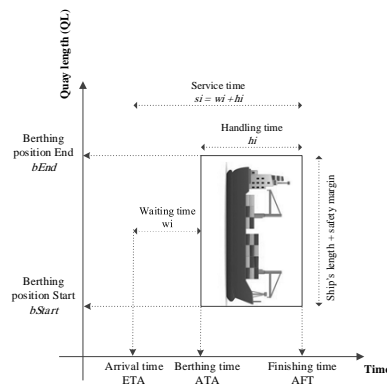


Figure 4-3: Vessel to berth and time-space representation

According to the classification scheme in Bierwirth & Meisel (2015), our approach is represented by the (BAP, QCAP(number)), which is the integration between the

berth allocation problem and quay crane assignment problem in order to decide on the berthing position, berthing time, and the number of cranes to assign to each vessel. Moreover, the problem is defined as *cont / dyn / QCAP / Σ (wait + hand)*, which is described as follows:

- *Spatial attribute: Continuous layout:* The quay is of a continuous layout, with no partitioning so then the vessel can berth depending on its length within the boundaries of the quay.
- *Temporal attribute: Dynamic arrival:* All fixed arrival times are known in advance for all incoming vessels, so then the arrival times restrict the earliest berthing times by adding more costs
- *Handling time attribute: unknown in advance:* The handling time of a vessel is unknown in advance, and it depends on the number of assigned QCs (QCAP) and the moves required.
- *Performance measure: wait and handling times:* The objective function is to minimise the total sum of the waiting time and the total sum of the handling times (total service time) of all vessels V .

4.3.1 Model Assumption

The proposed model was established based on the following assumptions:

- The vessel can be moored on the quay if there is a space greater than or equal to the vessel length, and if there is at least one QC available to start with. Moreover, we assume that all vessels can moor in any position on the quay (there is no desired berthing position).
- The number of QCs assigned to the vessels is greater than or equal to the minimum number of allowed quay cranes and less than or equal to the maximum number of allowed quay cranes.
- The QCs can be assigned to vessels using a dynamic approach (time-variant) so then the QCs can be assigned to another vessel before the original vessel departs. For more details on the differences between static and dynamic assigning for QCs, see Rodriguez-Molins, Salido, & Barber (2014).

- Information related to the incoming vessels is known in advance including the length of the vessel, expected time of arrival, and the movement/number of containers to be loaded/unloaded.
- Every vessel has a draft that is lower than the depth of the quay.
- The time for the QCs movements along the quay as well as the berthing and departure times of vessels to berth was not considered since it supposes there to be a constant penalty time (cost) for all vessels.
- Vessel length includes the required safety margins, which is the safe distance between two moored vessels.

4.3.2 Notations

The following are the notations used in the proposed approach:

Input data:

V	Set of vessels (Vessel list) to be served, each vessel denoted as $i \in V$.
vl_i	Length of the vessel i including the safety margins.
QC	Number of available QCs in the quay. It is assumed that all QCs are homogeneous.
$QCmovements$	Number of movements of containers per time unit (hour).
$safeQC$	Safe distance required between two continuous QCs including QC width.
q_i^{max}	Maximum number of QCs that container terminal allows per vessel.
q_i^{max}	Maximum number of QCs that can be assigned to vessel i , and this depends on the vl_i and $safeQC$, calculated by Equation (4.2).
q_i^{min}	Minimum number of QCs that can be assigned to vessel i .
QL	length of the quay.
ETA_i	Expected time of arrival of vessel i .
EFT_i	Expected finishing time of vessel i .
m_i	Number of required movements to load/unload containers to/from vessel i .

Decision variables:

ATA_i	Actual time of mooring vessel i .
AFT_i	Actual finishing time of vessel i .
q_i	Number of assigning QCs to vessel i .
w_i	Waiting time of vessel i .
h_i	Handling time of vessel i .
T_s	Total service time for V .
v_iMinHT	Minimum handling time needed to service vessel i , in this case, we set $q_i = q_i^{max}$, and this is the ideal number of QCs to be assigned to vessel i .
s_i	Service time of vessel i .
$bStart_i$	The start point of the berth on the quay for vessel i .
$bEnd_i$	The end point of the berth on the quay for vessel i .
v_iOrder	The order of vessel i in the V , obtained from the CDR tree generated by the GP.

4.3.3 Mathematical Model

In this section, the mathematical model is based on (M. Rodriguez-Molins, Salido, & Barber, 2014; Mario Rodriguez-Molins, Ingolotti, et al., 2014) with minor modifications to overcome the proposed assumptions such as the QCs being assigned to a vessel in a dynamic approach (time-variant), which maximise the QCs utilisation and minimising the total service time. The following are the main equations used.

$$\text{minimize } T_s = \sum_{i \in V} (w_i + h_i) \quad (4.1)$$

$$q_i^{\max} = \max \left(1, \min \left(q^{\max}, \frac{vl_i}{\text{safeQC}} \right) \right) \quad \forall i \in V \quad (4.2)$$

$$w_i = ATA_i - ETA_i \quad \forall i \in V \quad (4.3)$$

$$h_i = AFT_i - ATA_i \quad \forall i \in V \quad (4.4)$$

$$h_i = \left(\frac{m_i}{q_i \cdot QCmovements} \right) \quad \forall i \in V \quad (4.5)$$

$$v_i \text{MinHT} = \left(\frac{m_i}{q_i^{\max} \cdot QCmovements} \right) \quad \forall i \in V \quad (4.6)$$

$$s_i = w_i + h_i \quad \forall i \in V \quad (4.7)$$

$$bEnd_i = bStart_i + vl_i \quad \forall i \in V \quad (4.8)$$

4.4 Dispatching rules

When solving scheduling optimisation problems, there are three major types of scheduling algorithm as illustrated in Figure 4-4 and explained in detail by Pinedo (2012).

- *Exact algorithms* are the algorithms used to find and solve a problem to optimal. This type of algorithm cannot solve more sophisticated problems such as NP-hard Problems. In the literature, we found that this method applied in only 24% of the approaches for solving the BAP (Bierwirth & Meisel, 2015) with small instances (20 to 30 vessels benchmark).

- *Approximation algorithms* which are the algorithms that produce solutions approximate to NP-hard optimisation problems that are guaranteed to be within the distance of the actual optimum.
- *Heuristic algorithms* - this type of algorithms is commonly used to solve NP-hard problems. There is no guarantee that the solutions will be close to the optimum. However, it is a fast way to obtain solutions in an acceptable computational time. The performance is evaluated empirically.

For heuristic algorithms, there are two main types of heuristics.

- *Construction Heuristics* start to solve a problem from the beginning without a pre-scheduled solution. The algorithm schedules the jobs one by one and builds the solution as it progresses.
- *Improvement Heuristics* - this type of algorithm starts to solve a problem with a predefined and scheduled solution, and tries to improve it or seeks a better solution.

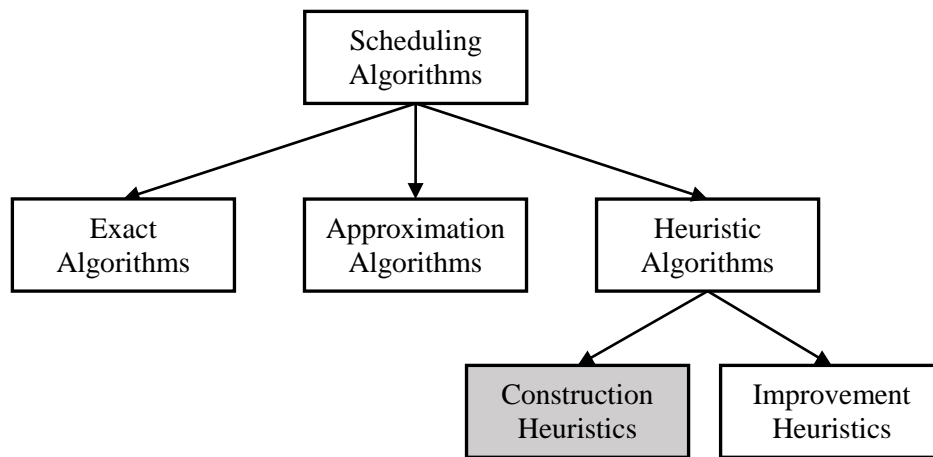


Figure 4-4: Types of Scheduling Algorithms.

Dispatching rules (DRs) are defined by Pinedo (2012) as examples of construction heuristics. DRs in scheduling have received attention from researchers over the past few decades. In the field of job-shop-scheduling problems, DR in general, is a rule

that prioritises all of the jobs that are waiting for processing; whenever a machine is freed, a job with the highest priority in the queue is selected to be processed.

The Standard priority rule (SPR), or dispatching rule (DR) in our approach is the method of how to tackle the vessel list in order to schedule them. In the literature, most of the researchers use a first-come-first-serve rule (FCFS) to order the vessel list depending on its ETA. Table 4-1 shows that fewer researchers consider the DR, while some of them use DR indirectly in their solutions. They also noted the priority-based rules used to solve the seaside CT problems. Priority-based rules either give a fixed value and assigned it to a vessel depending on the many strategies and policies related to the terminal managers. They may also assign priority to a vessel as a variable in the case of congestion in the berth.

Table 4-1: Overview of research applied Dispatching Rule in seaside container terminal problems.

Year	Reference	Rule(s)
2017	(Correcher & Alvarez-Valdes, 2017)	CDR, and Random priority rules
2017	(Hsu et al., 2017)	Different heuristic rules for BAP and QCAP such as, give priority depend on ETA, the least workload, QCs load balance
2017	(De León, Lalla-Ruiz, Melián-Batista, & Marcos Moreno-Vega, 2017)	FCFS, compared 12 algorithms
2017	(Expósito-Izquierdo, Lalla-Ruiz, de Armas, Melián-Batista, & Moreno-Vega, 2017)	Four DRs: Random, FCFS, WSPT, EDD
2016	(Shang et al., 2016)	Random priority rule
2016	(Türkoğulları et al., 2016)	EDD
2016	(Ursavas & Zhu, 2016)	Different priorities measured by the waiting cost.
2016	(He, 2016)	Eight DRs
2016	(Huang et al., 2016)	Three different priorities: given to vessels depends on existing in the berth template.
2016	(El-boghdadly et al., 2016b)	CDRs
2016	(El-boghdadly et al., 2016a)	CDRs
2015	(Ursavas, 2015)	Priority control mechanism
2015	(Frojan et al., 2015)	Twenty-six DRs
2015	(M. Z. Li et al., 2015)	Priority factor for vessels with important clients.
2015	(Ji et al., 2015)	Three DRs: FCFS, small vessels, big vessels
2014	(Basri & Zainuddin, 2014)	Ignored FCFS
2014	(Mario Rodriguez-Molins, Salido, et al., 2014)	Four DRs: FCFS, FCMP, MWWT, EWMT
2013	(Elwany et al., 2013)	CDR
2013	(Chutian Yang et al., 2013)	Priority given to vessels
2012	(D. Xu et al., 2012)	WSPT
2012	(Y. Xu et al., 2012)	Construction rule

2012	(Zhen & Chang, 2012)	Give priority to vessels with weak performing
2012	(X. Liang et al., 2012)	The vessels ordered by sequence optimised by Practical Swarm Optimization
2011	(Böse, 2011)	Two DRs: FCFS, EDD
2011	(C. Liang et al., 2011)	Random priority rule
2011	(Ali et al., 2011)	Random priority rules
2011	(Guldogan, Bulut, & Tasgetiren, 2011)	Priority given to vessels
2010	(Han et al., 2010)	Priority given to vessels
2010	(C. Y. Cheong et al., 2010)	Priority given to vessels
2009	(C. Liang, Huang, et al., 2009)	Random priority rules
2009	(Meisel & Bierwirth, 2009)	Priority given to vessels
2008	(Tang & Dai, 2008)	Give priority to vessels with high capacity of containers
2008	(Imai, Chen, Nishimura, & Papadimitriou, 2008)	Ignored FCFS
2006	(Pengfei Zhou et al., 2006)	Ignored FCFS
2003	(Imai, Nishimura, & Papadimitriou, 2003)	Ignored FCFS

Rodriguez-Molins, Salido, et al. (2014) applied three different DRs, plus the FCFS rule. (1) They used First Come Maximum Priority (FCMP): this is similar to FCFS, where the next vessel is chosen according to the arrival order but, in this case, there is no restriction on the time that the vessels can moor. (2) Maximum Weighted Waiting Time (MWWT): the vessel list is ordered according to their weighted waiting time. The vessel with the highest value is moored first. (3) Earliest Weighted Mooring Time (EWMT): among the vessels that can moor earlier, the operator chooses the vessel with the highest priority. Ji et al. (2015) applied three rules, FCFS, giving priority to small vessels and giving priority to large vessels depending. Böse (2011), proposed a simulation model that used two selection approaches; FCFS and the Earliest Due Date (EDD) rule. (D. Xu et al. (2012) proposed a weighted shortest processing time first (WSPT) rule as a selection priority method for incoming vessels. Meisel & Bierwirth (2009) proposed two meta-heuristic approaches, which enable changes in the priority list in order to improve the quality of the berth plans. Elwany et al. (2013) applied a criterion to give a higher priority to larger vessel with a later EFT. Ali et al. (2011) and C. Liang et al. (2011, 2009) used random priority rules to order the vessel list. Frojan et al. (2015) used 26 priority rules randomly generated as the initial population using a genetic algorithm. The first group of rules corresponds to the individual characteristics of the vessels, while the second group combines the features into more complex rules. The rest of the research studies shown in Table 4-1 either

ignored the FCFS rule and applied a priority list to the vessels, or they used FCFS only.

The composite dispatching rule (CDR) is a combination of a number of SPRs used as one rule to evaluate the priorities of the jobs waiting in the queue for processing. To the best of our knowledge, this is the first work that has solved the BACAP using the concept of CDRs. The following sections will explain how we employed the CDRs to solve the BACAP.

4.5 Genetic Programming for BACAP (*BACAP_GP*)

The proposed algorithm *BACAP_GP* consists of two main parts. The first part is the GP approach, which is responsible for discovering the best dispatching rule for a given port scenario to optimise the T_s . The second part is the *BACAP_Scheduler*, which is responsible for applying the ordered vessel list by GP and finding the best mooring time and berth location for a given vessel v_i , in addition to assigning QCs to it. Figure 4-5 shows the relationship between the GP engine and the *BACAP_Scheduler*. More details will be explained in the following sections.

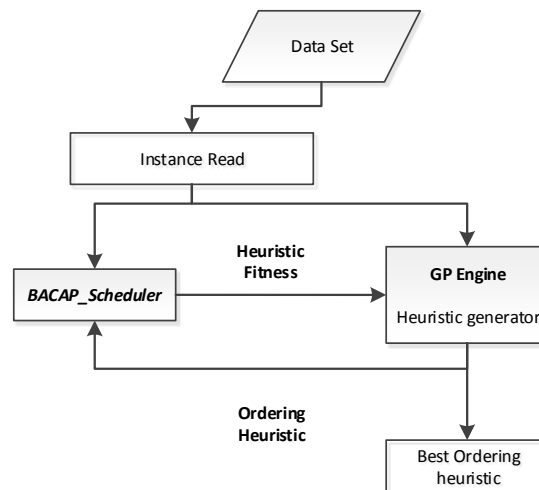


Figure 4-5: GP Engine and *BACAP_Scheduler* relationship

4.5.1 *BACAP_GP* Framework

In this section, we have investigated the use of GP for evolving effective and robust CDRs to solve the BACAP. GP, as shown in our literature review, is not a conventional method to use to solve the BACAP. Nguyen, Zhang, Johnston, & Chen Tan (2013) applied the GP to solve QCSP, and they concluded that the GP outperformed the GA in terms of the average and best fitness. Therefore, it will be interesting to solve the integration problem, BACAP, using the GP.

During the scheduling process in a container terminal, it is crucial to prioritise and order the incoming vessels so as to solve the BACAP. Instead of using SPRs, we have tried to evolve the priority functions based on the SPRs and created a CDR that can determine the best possible ordering for the vessel list, which in turn achieves our objective function. Figure 4-6 shows the *BACAP_GP* flowchart. The GP starts by creating a random initial population of CDRs using the terminal set and function set. The GP was used for evolving the vessel dispatching rules and ordering the vessel list accordingly, before sending it to the *BACAP_Scheduler* (as described in Section 4.5.2). Each CDR was implemented using the *BACAP_Scheduler* which, in return, used the GP to evaluate the fitness function of the schedule. If the maximum number of generations or the stopping condition is not reached, then the GP reproduces another population by using the crossover and mutation method, generating other CDRs. The process will continue until the stopping condition is reached, and then the GP will obtain the best CDR that determines the best ordering heuristic used.

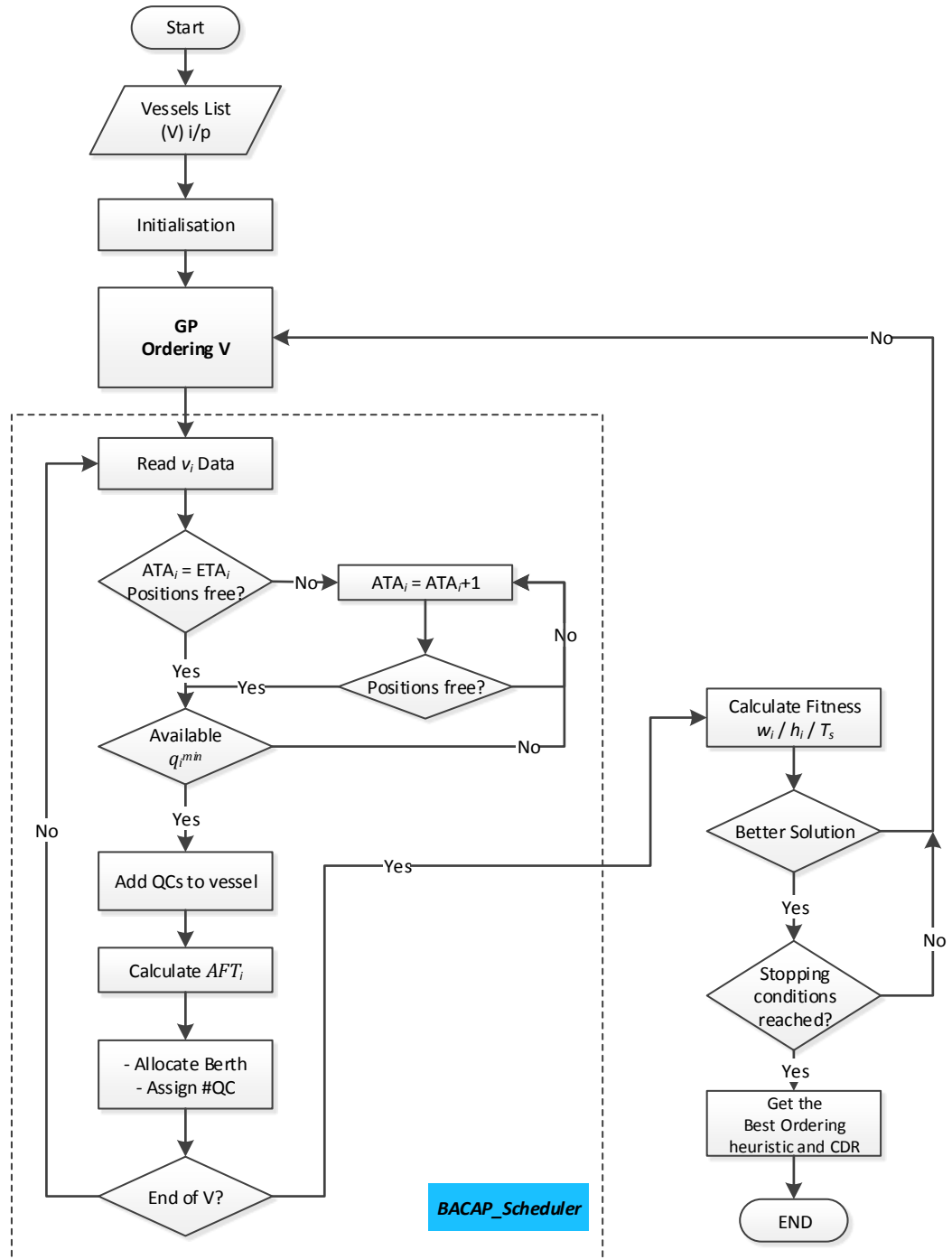


Figure 4-6: BACAP_GP Flowchart

4.5.1.1 BACAP_GP Representation

Genetic programming is a technique used to evolve computer programs to solve complex computational problems. The representation of GP individuals is a tree-based form. Each individual presents a dispatching rule/equation constructed by a terminal set and a function set. In this section, we will describe the GP representation proposed for generating individuals using both the terminal and function set.

1) Terminal set:

The terminal set (leaf nodes) will be chosen as shown in Table 4-2, as they have a profound influence on the quality of the BACAP solution. These values can be found in the input data or calculated in the initialisation process as described before.

Table 4-2: Terminal set for the *BACAP_GP* tree representation.

Notation	Description
vl_i	Vessel length
ETA_i	Vessel expected time of arrival for the given vessel i
EFT_i	Vessel expected finishing time for the given vessel i
q_i^{Max}	The maximum number of QCs that can be assigned to a vessel
m_i	Number of movements (loading/unloading)
$v_i MinHT$	Time for handling a vessel while it is working with q_i^{Max}

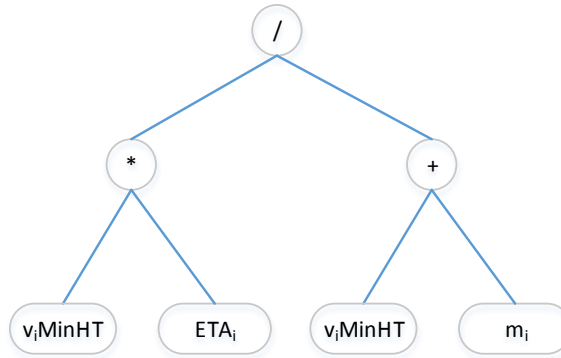
2) Function Set:

The function set (internal nodes) will consist of the standard mathematical operators that are commonly used in the GP literature (addition, subtraction, multiplication, and division). Furthermore, we used other functions which often occur in evolving dispatching rules. Table 4-3 shows the chosen function set and its description.

Table 4-3: Function set for the *BACAP_GP* tree representation.

Notation	Description
<i>Add</i> (<i>x</i> , <i>y</i>)	Addition
<i>Sub</i> (<i>x</i> , <i>y</i>)	Subtraction
<i>Mul</i> (<i>x</i> , <i>y</i>)	Multiplication
<i>Div</i> (<i>x</i> , <i>y</i>)	Safe division, returns 1 if the denominator equals to 0
<i>Avg</i> (<i>x</i> , <i>y</i>)	Returns the average value of the input variables.
<i>Min</i> (<i>x</i> , <i>y</i>)	Returns the minimum of the two inputs.
<i>Max</i> (<i>x</i> , <i>y</i>)	Returns the maximum of the two inputs.
<i>Abs</i> (<i>x</i>)	Returns the absolute value of variable <i>x</i> .
<i>Ceiling</i> (<i>x</i>)	Returns the ceiling value of the input.
<i>Floor</i> (<i>x</i>)	Returns the floor value of variable <i>x</i> .

An illustration of an individual with the tree representation generated by the *BACAP_GP* has been provided in Figure 4-7, which provides a possible CDR. The above chosen terminal and function sets are well-defined and closed, allowing for any combination of arguments that it may encounter. They have the important property of solving the BACAP problem, which is known as *closure* (Koza, 1992). The terminal set was chosen with different types, the GP works efficiently and there is no need for normalisation. Furthermore, the proposed fitness function is the objective function of the problem as shown in Equation (4.1). The following formula (4.9) represents a CDR (determines the order of a given vessel *i*) obtained from the tree shown in Figure 4-7. It gives a weight for each vessel which in return the GP used them to order the vessel list.

**Figure 4-7:** Example of a *BACAP_GP* tree with defined functions and terminals.

$$v_iOrder = \frac{v_iMinHT \cdot ETA_i}{v_iMinHT + m_i} \quad (4.9)$$

4.5.1.2 Generate a set of initial solutions

The *BACAP_GP* generates an initial population randomly by creating individuals as illustrated before. Each represents an ordering vessel and one possible solution with its corresponding fitness function. The *BACAP_GP* determines the best individual in the initial population and evolves it to the next generation.

4.5.1.3 Fitness function

In this study, our objective is to find effective CDRs for solving the BACAP with a minimum total service time T_s for all vessels. The GP fitness function chosen to be the problem objective function has been shown in Equation (4.1). Therefore, we have proposed a method to form a CDR from the tree-based result of the GP. This CDR is then, with the minimum service time, used to evaluate the fitness value of the BACAP.

4.5.1.4 Crossover Operation

For crossover operations in the GP system, which is a tree-based individual (Koza, 1994), the new individuals for the next generation are created by randomly recombining sub-trees from two selected parents. Figure 4-8 illustrates the crossover operations of the *BACAP_GP* approach.

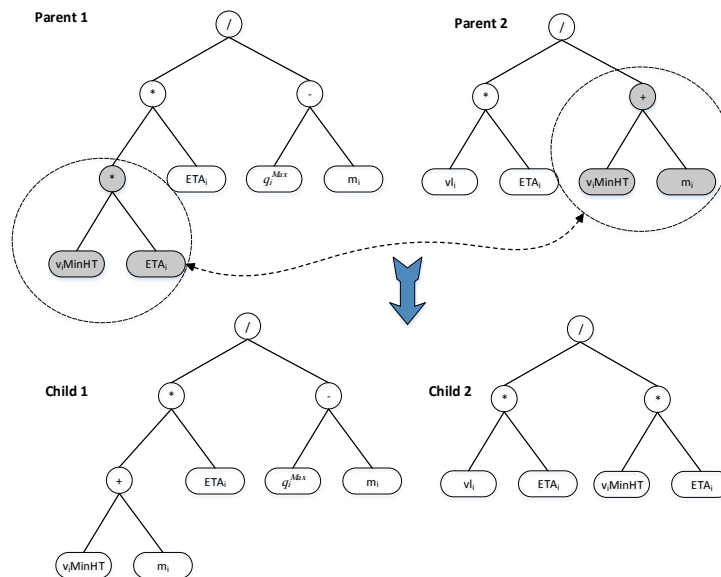


Figure 4-8: Crossover representation for the *BACAP_GP* operation

4.5.1.5 Mutation Operation

The mutation operation in the GP system was implemented by randomly selecting the node of a chosen parent individual and replacing the rest of the sub-tree by also randomly generating another sub-tree. Figure 4-9 illustrates the mutation operation *BACAP_GP* approach.

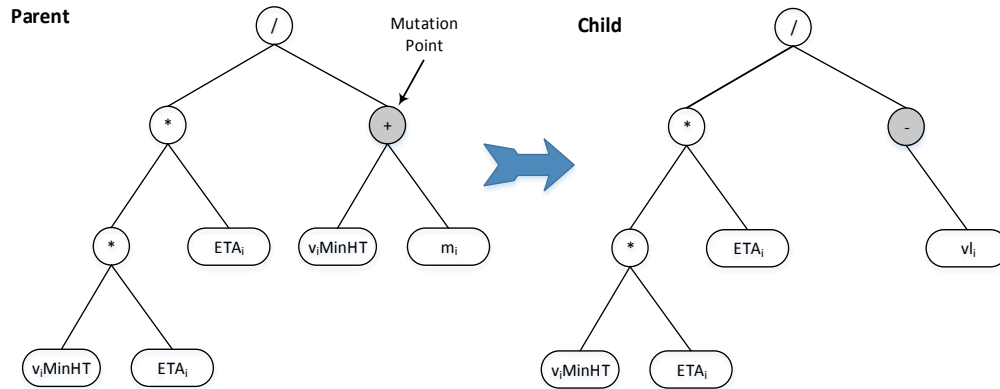


Figure 4-9: Mutation representation for the *BACAP_GP* operation

4.5.2 BACAP Scheduler Algorithm (*BACAP_Scheduler*)

The *BACAP_Scheduler* is the algorithm that we proposed to use to schedule the vessels in the plan according to the dispatching rules presented from the GP. It returns the total service time of the schedule plan for evaluation by the GP. To solve the BACAP, the *BACAP_Scheduler* represents the problem as a two-dimensional array, the x-axis demonstrates the time horizon, and the y-axis demonstrates the quay length. The goal is to assign all incoming vessels V in the array without violating the constraints mentioned earlier. We developed a software program using java to handle the assignment process for the vessel list one-by-one in the array, which will describe its function as (Scheduler). The order of the vessels is determined by the GP, which has been described in the previews section.

After the Scheduler places a vessel in the array, it fills in the location with the correspondent Vessel ID, which shows the location of the vessel in the schedule. Empty spaces are denoted by zeros. A rectangle demonstrates the vessel with its position on the horizon starting from ATA and ending in AFT for the time horizon

(x-axis). The length of the rectangle shows the length of the vessel v_i on the quay (y-axis). The following example demonstrates the representation of the scheduler array and how it works

4.5.2.1 Numerical example

An instance has been shown in Figure 4-10, which is the output of the Scheduler for four vessels with lengths of 400, 300, 300, 300 and ETA 1, 1, 2, 2 respectively. The Scheduler will reserve the array with the vessel's ID, which indicates its mooring time and the spaces in the quay. We can also notice that there is available space on the quay at time 2, position 400 to 500 - therefore the Scheduler fills it with a zero. The QC array shows the remainder of the QCs. If the container terminal has 7 QCs, then the remainder of QCs at time 1 and 2 are zero, while the remainder at time 3 and 4 is 1. Starting from time 5 until the end of the time horizon, the remaining QCs is 7, which means that they are not occupied. We noticed that all of the vessels have $ATA = ETA$, while vessel number 4 is $ETA = 2$. However, there is no availability for it to berth at this time, so it waits for 1 hour and is assigned to the time $ATA = 3$.

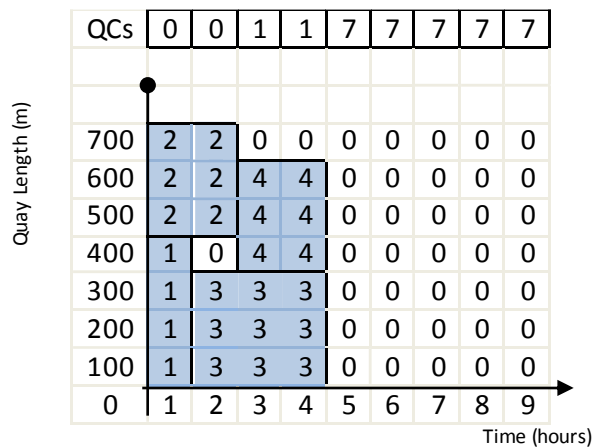


Figure 4-10: Two-Dimensional array representation for the BACAP

The following is how the *BACAP_Scheduler* works as shown in flowchart Figure 4-6:

- Step 1. *Initialisation*: in this step, we initialise the vessel list V by reading the given data (vessel ID, vessel length, expected time of arrival, number of movements). We will also initialise the quay as a two-dimensional array and initialise the QC array. Using the vessel list data, we can calculate the q_i^{max} , $v_i MinHT$ and EFT .
- Step 2. *Berth Allocation Problem (BAP)*: The Scheduler reads the vessel data list one-by-one and investigates the availability of both the times and spaces (berths) in order to moor vessels in the quay. The Scheduler checks the schedule at the ETA for the vessel, starting from quay position zero until the end of the quay length. If there is a free space in the quay (quay array cells equal to zero) greater or equal to vl_i , then we assign $ATA = ETA$ and $AFT = EFT$ before going to step 3. If there is no free space for v_i at the ETA , then the Scheduler increases the ETA by one-time unit ($ATA = ETA + 1$, $AFT = EFT + 1$) and searches the schedule again at new ATA for free space in the quay until a free space is found.
- Step 3. *Quay Crane Assignment Problem (QCAP)*: after a suitable location and time for the given vessel is found, then the Scheduler checks the availability of the QCs between time ATA and AFT that will serve the vessel while considering the given constraints of the minimum and maximum number of QCs required to serve the vessel. The Scheduler will accept the place if the q_i^{min} is available in this period. If so, then we will go to step 4; if not, then the Scheduler will increase the ATA by one-time unit and let $ATA = ATA + 1$ and $AFT = AFT + 1$ before starting again from step 2.
- Step 4. *Dynamic QCs assignment (time-variant)*: after scheduling the vessel with at least the minimum number of QCs available, the Scheduler will try to add more QCs if possible to the vessel and recalculate the handling time needed, decreasing the EFT to AFT .

Step 5. *Final plan assignment and updates:* The Scheduler will place the vessel in the schedule (filling the quay array with the vessel IDs in the cells occupied by the vessel) and then update the QCs array with the number of remaining QCs available. These processes (step 2 to step 5) will continue until the end of the vessel list is reached.

4.6 Computational Experiments and Analysis

In this section, computational experiments were conducted to evaluate the performance of the proposed algorithm. We defined the standard SPRs commonly used in the literature and then produced a comprehensive study and comparison of the SPRs along with the evolved CDRs obtained from the *BACAP_GP*. The *BACAP_Scheduler* and the *BACAP_GP* were developed using Java and included the ECJ22 library (a Java-based Evolutionary Computation Research System, n.d.) in order to implement the GP.

4.6.1 Datasets and parameter settings

The experiments were performed using the same benchmark introduced in Rodriguez-Molins, Ingolotti et al. (2014) and Rodriguez-Molins, Salido, et al. (2014), published online. Due to the difficulties encountered when trying to find the subset of the vessels that the authors used in their experiments, we did not compare our results with them. A sample of this benchmark can be found in Appendix A. The benchmark contains 100 instances generated randomly that follows the suggestion of the container terminal operators; each instance is composed of a queue of 100 vessels to be scheduled over almost three months (2,160 hours). We chose this very large benchmark in order to investigate the performance of using the proposed algorithm and the GP method to solve the problem. The data for each vessel in the benchmark includes vessel ID, the length of the vessel, expected time of arrival and the movements (number of containers) to be unloaded/loaded. In this study, the following assumptions have been considered:

- Quay length $QL = 700$ meters
- The number of QCs on the quay $QC = 7$ QCs
- Maximum number of QCs that the container terminal allows per vessel $q^{max} = 5$ QCs
- Minimum number of QCs that can be assigned to a vessel $q_i^{min} = 1$ crane
- QC Movements per time unit $QCmovements = 25$ containers/hour.
- $safeQC = 35$ metres

From the above assumptions, the input data and using Equation (4.2) and Equation (4.6), we can initially calculate the best values for the maximum number of QCs that the vessel can work with q_i^{max} for the entire vessel list. We can then calculate the best handling time for the vessel $v_i MinHT$.

In this experiment, the GP parameters were chosen as shown in Table 4-4, which shows the common values used in the literature. We implemented the *Ramped half and half* (full and grow method) to generate the initial population of the GP as proposed by Koza (1992), which produces a wide variety of trees of various sizes and shapes. In this method, the initial population is divided into two parts. Half of the trees were generated randomly with a maximum depth proposed (full method) and the second half contained the randomly generated trees with variable depth values ranging from one to the maximum depth proposed (growth method).

Table 4-4: *BACAP_GP* Parameters

Parameter	Value
Number of iteration	50
Population size	50
Number of generation	50
Creation Type	Ramped half and half
Maximum depth for population tree	7
Crossover probability	0.8
Mutation probability	0.1
Recreation probability	0.1

Regarding the dispatching rules, the following are the SPRs that we examined and compared with the *BACAP_GP*.

- *R1: FCFS*: first-come-first-serve rule; the vessel list will be ordered by *ETA*.
- *R2: Max. QC needed low priority*: the vessel list will be ordered by q_i^{max} ascending.

- *R3: Max. QC needed high priority:* in this dispatching rule, the vessel list will be ordered by q_i^{max} descending.
- *R4: Vessel length high priority:* gives larger vessels higher priority in the sorting order.
- *R5: Vessel length low priority:* gives smaller vessels higher priority in the sorting order.
- *R6: Movement low priority:* here we give the lesser numbers of containers for the vessel that need to be loaded/unloaded higher priority.
- *R7: Movement high priority:* same as above, but the vessels with a higher number of movements take priority.
- *R8: Min. Handling time low priority:* in this dispatching rule, the vessel list will be ordered by $v_i MinHT$ ascending.
- *R9: Min. Handling time high priority:* same as above, but gives priority to the vessels that need a long handling time.

4.6.2 Results of SPRs and *BACAP_GP*

We presented the results of using the nine SPRs explained above and compared the best results with the CDRs generated by the proposed algorithm (*BACAP_GP*). All of the tests were run for 10 hours or until they hit the stopping condition. We used a core i3 Intel processor at 1.8GHz and 4GB of RAM.

A) SPRs Results:

In this approach, we solved the BACAP using the *BACAP_Scheduler* with each of the SPRs proposed for ordering, giving priority to the vessels in the queue. This approach was examined without applying the GP algorithm.

Table 4-5 shows the results of the SPRs of the benchmark. We denoted the total waiting time for the vessels in the instance by (W), while the (T_s) is the total service time for the vessels in the same instance.

From the results of all 100 instances used, we noted that the highest number of values for the SPRs for the T_s comes from the rule (*R2*), which is gives priority to the vessels according to their q_i^{max} values ordered ascendingly. Rule (*R1*) is the next best rule, which gives priority to first-come-first-serve. This is because we restrict the vessels from being moored before its *ETA*. We found that the worst

results come from $R7$, which gives priority to the vessels that have a larger number of containers to be moved.

Previous results show that the strategy of giving priority to incoming vessels is a decision for terminal managers and ship owners. They often prefer different service strategies. However, the results reflect the advice concluded by Ji et al. (2015), which says that “when the large vessel arrival probability is small, the terminal manager may adopt the service strategy of giving priority to large vessels; and when the arrival probability of large vessels becomes larger, they may adopt first-come-first-served”.

Table 4-5: Results of SPRs to solve BACAP.

no.	Results	$R1$	$R2$	$R3$	$R4$	$R5$	$R6$	$R7$	$R8$	$R9$	Best	Worst
1	W	4	2	14	29	0	30	0	35	0	0	35
	T_s	598	598	610	627	598	626	599	631	598	598	631
2	W	13	25	28	39	34	21	45	44	32	13	45
	T_s	623	633	641	649	643	630	653	651	639	623	653
3	W	3	7	25	27	7	0	3	24	7	0	27
	T_s	582	585	606	610	583	578	587	601	589	578	610
4	W	30	34	35	50	81	48	90	52	90	30	90
	T_s	618	622	626	633	666	631	677	637	676	618	677
5	W	20	20	34	36	34	49	19	21	19	19	49
	T_s	614	613	635	638	626	642	615	622	613	613	642
6	W	13	20	13	48	45	53	59	41	59	13	59
	T_s	732	732	732	770	753	761	777	757	776	732	777
7	W	13	13	13	15	22	18	19	16	19	13	22
	T_s	592	591	592	596	600	591	604	588	604	588	604
8	W	22	59	40	43	68	62	43	48	31	22	68
	T_s	671	712	691	696	726	708	698	694	691	671	726
9	W	9	23	30	37	27	45	27	41	16	9	45
	T_s	630	640	652	661	644	664	648	662	635	630	664
10	W	11	11	11	9	40	14	12	9	11	9	40
	T_s	615	613	616	615	643	615	618	612	616	612	643
11	W	10	16	12	26	16	16	29	41	21	10	41
	T_s	651	662	653	675	661	665	676	683	667	651	683
12	W	16	28	30	26	41	55	31	49	34	16	55
	T_s	612	618	623	632	635	647	636	642	637	612	647
13	W	7	5	12	15	22	8	24	15	22	5	24
	T_s	694	694	701	705	712	697	713	700	714	694	714
14	W	0	0	0	0	12	0	12	0	12	0	12
	T_s	628	628	629	627	644	626	645	626	645	626	645
15	W	12	15	58	59	26	59	36	55	12	12	59
	T_s	623	623	675	678	637	672	645	664	624	623	678
16	W	29	33	40	48	28	35	40	40	40	28	48
	T_s	656	659	670	678	657	662	672	664	670	656	678
17	W	5	6	24	25	24	45	9	45	8	5	45
	T_s	599	600	621	622	616	637	609	639	607	599	639
18	W	15	15	14	21	34	37	18	53	16	14	53
	T_s	596	596	598	608	615	619	603	642	600	596	642
19	W	14	44	14	42	53	96	32	86	42	14	96
	T_s	595	622	598	627	629	671	614	665	619	595	671
20	W	3	3	31	31	42	16	21	23	21	3	42
	T_s	630	625	661	661	663	642	648	649	647	625	663
21	W	11	15	11	29	68	60	39	46	32	11	68
	T_s	640	641	640	657	694	685	664	674	658	640	694
22	W	9	10	18	19	10	10	26	10	18	9	26
	T_s	622	620	632	635	623	614	648	615	640	614	648
23	W	14	18	40	53	18	62	28	63	26	14	63
	T_s	621	624	644	665	621	668	641	670	638	621	670
24	W	15	15	17	18	48	40	34	42	34	15	48
	T_s	604	604	609	611	638	628	620	633	619	604	638
25	W	17	31	34	40	51	51	32	41	34	17	51
	T_s	592	604	611	618	617	628	602	613	603	592	628
26	W	4	4	4	6	11	16	8	7	8	4	16
	T_s	626	626	626	627	630	634	628	627	628	626	634
27	W	28	21	16	20	42	38	16	38	16	16	42
	T_s	668	660	660	673	685	683	664	685	663	660	685
28	W	12	7	12	10	15	32	31	18	29	7	32
	T_s	577	575	580	580	585	600	602	585	601	575	602
29	W	55	55	85	107	54	94	59	104	59	54	107
	T_s	690	690	720	739	682	720	698	728	697	682	739
30	W	27	17	76	77	60	88	34	76	34	17	88
	T_s	725	715	780	783	754	786	733	780	733	715	786

4.6.Computational Experiments and Analysis

no.	Results	<i>R1</i>	<i>R2</i>	<i>R3</i>	<i>R4</i>	<i>R5</i>	<i>R6</i>	<i>R7</i>	<i>R8</i>	<i>R9</i>	Best	Worst
31	<i>W</i>	24	31	38	47	38	58	30	46	34	24	58
	<i>T_s</i>	597	601	618	629	613	628	608	622	611	597	629
32	<i>W</i>	7	7	7	21	26	27	16	27	16	7	27
	<i>T_s</i>	638	637	639	650	653	657	646	657	646	637	657
33	<i>W</i>	18	18	21	19	23	19	19	19	12	12	23
	<i>T_s</i>	624	624	632	628	628	620	629	622	621	620	632
34	<i>W</i>	6	8	5	6	28	24	15	12	15	5	28
	<i>T_s</i>	610	612	611	612	630	625	621	615	620	610	630
35	<i>W</i>	8	6	13	12	10	7	12	12	12	6	13
	<i>T_s</i>	629	629	638	637	632	623	636	633	636	623	638
36	<i>W</i>	8	13	8	22	29	21	30	23	25	8	30
	<i>T_s</i>	682	683	686	698	702	695	702	695	698	682	702
37	<i>W</i>	14	20	25	32	45	21	26	21	15	14	45
	<i>T_s</i>	647	649	663	665	670	648	663	649	650	647	670
38	<i>W</i>	18	23	40	46	35	79	36	43	36	18	79
	<i>T_s</i>	697	695	712	718	704	746	707	713	706	695	746
39	<i>W</i>	16	11	19	20	37	19	19	19	13	11	37
	<i>T_s</i>	635	632	645	645	655	639	641	641	637	632	655
40	<i>W</i>	26	23	45	63	28	47	43	66	43	23	66
	<i>T_s</i>	628	628	652	670	632	654	648	671	648	628	671
41	<i>W</i>	10	10	10	38	31	15	25	15	25	10	38
	<i>T_s</i>	647	647	647	676	660	648	663	648	661	647	676
42	<i>W</i>	3	4	12	13	9	16	3	13	4	3	16
	<i>T_s</i>	624	623	634	637	630	639	626	637	624	623	639
43	<i>W</i>	32	40	41	36	78	61	47	57	49	32	78
	<i>T_s</i>	680	685	693	684	721	708	696	707	696	680	721
44	<i>W</i>	16	16	14	40	39	33	23	35	25	14	40
	<i>T_s</i>	675	673	675	702	694	693	683	692	683	673	702
45	<i>W</i>	28	28	30	41	54	33	47	39	55	28	55
	<i>T_s</i>	681	682	686	700	708	697	704	702	712	681	712
46	<i>W</i>	19	18	17	36	21	16	26	16	31	16	36
	<i>T_s</i>	645	643	645	665	644	642	657	641	660	641	665
47	<i>W</i>	46	54	66	75	60	53	72	75	80	46	80
	<i>T_s</i>	633	638	656	669	651	649	670	666	672	633	672
48	<i>W</i>	18	13	38	48	23	57	55	29	30	13	57
	<i>T_s</i>	628	620	644	655	629	662	664	635	638	620	664
49	<i>W</i>	7	5	7	21	11	9	16	10	16	5	21
	<i>T_s</i>	661	665	665	683	673	669	674	665	673	661	683
50	<i>W</i>	9	15	17	17	32	17	17	17	17	9	32
	<i>T_s</i>	606	608	613	617	630	612	620	611	621	606	630
51	<i>W</i>	3	3	54	78	6	35	17	35	17	3	78
	<i>T_s</i>	614	615	663	685	618	641	626	640	627	614	685
52	<i>W</i>	5	5	8	22	39	42	9	9	9	5	42
	<i>T_s</i>	616	615	623	636	650	655	623	625	623	615	655
53	<i>W</i>	16	19	18	21	47	72	29	37	30	16	72
	<i>T_s</i>	630	630	634	641	658	688	641	656	643	630	688
54	<i>W</i>	3	3	3	11	3	9	0	9	0	0	11
	<i>T_s</i>	585	585	587	597	587	594	589	595	588	585	597
55	<i>W</i>	9	11	16	21	40	42	25	38	25	9	42
	<i>T_s</i>	620	620	628	628	646	649	635	646	635	620	649
56	<i>W</i>	4	5	4	10	33	22	6	10	22	4	33
	<i>T_s</i>	614	613	616	621	642	630	620	621	630	613	642
57	<i>W</i>	4	4	4	12	0	11	20	0	4	0	20
	<i>T_s</i>	649	647	652	662	647	660	663	650	645	645	663
58	<i>W</i>	10	7	18	32	21	45	29	21	8	7	45
	<i>T_s</i>	643	643	648	663	656	673	667	646	648	643	673
59	<i>W</i>	30	37	115	138	38	153	84	106	60	30	153
	<i>T_s</i>	585	584	672	697	588	706	641	656	615	584	706
60	<i>W</i>	34	41	54	78	62	76	44	75	44	34	78
	<i>T_s</i>	684	693	706	736	706	726	707	726	706	684	736
61	<i>W</i>	19	18	55	81	42	65	48	71	44	18	81
	<i>T_s</i>	572	573	607	638	602	615	610	623	604	572	638
62	<i>W</i>	8	4	17	21	43	19	58	20	56	4	58
	<i>T_s</i>	661	658	670	677	693	668	715	669	712	658	715
63	<i>W</i>	14	14	19	15	28	7	37	7	34	7	37
	<i>T_s</i>	589	588	594	589	606	578	615	579	612	578	615
64	<i>W</i>	16	16	37	21	50	53	51	53	31	16	53
	<i>T_s</i>	628	628	651	636	658	665	666	666	646	628	666
65	<i>W</i>	14	15	74	64	52	99	23	79	28	14	99
	<i>T_s</i>	705	710	768	757	748	785	715	768	724	705	785
66	<i>W</i>	41	37	55	56	49	72	79	52	50	37	79
	<i>T_s</i>	680	674	694	703	688	717	727	701	695	674	727
67	<i>W</i>	8	7	13	33	6	55	7	33	7	6	55
	<i>T_s</i>	710	708	717	739	706	753	713	734	710	706	753
68	<i>W</i>	2	0	3	22	0	30	19	0	19	0	30
	<i>T_s</i>	609	605	614	631	609	637	630	608	625	605	637
69	<i>W</i>	15	10	28	49	10	40	33	38	33	10	49
	<i>T_s</i>	660	658	672	695	661	676	680	678	680	658	695
70	<i>W</i>	10	10	28	33	21	36	15	36	15	10	36
	<i>T_s</i>	587	586	605	609	599	614	593	614	593	586	614
71	<i>W</i>	13	13	19	42	11	13	16	13	16	11	42
	<i>T_s</i>	596	597	601	636	600	601	615	600	611	596	636
72	<i>W</i>	9	13	13	45	13	13	23	13	22	9	45
	<i>T_s</i>	607	607	614	649	613	618	624	616	622	607	649
73	<i>W</i>	4	4	20	20	28	13	53	25	45	4	53
	<i>T_s</i>	588	587	604	605	604	594	635	605	627	587	635
74	<i>W</i>	6	6	33	34	6	29	29	34	34	6	34
	<i>T_s</i>	650	650	678	682	654	675	676	674	677	650	682
75	<i>W</i>	11	8	18	23	19	23	36	15	21	8	36
	<i>T_s</i>	699	695	704	711	707	714	727	699	714	695	727
76	<i>W</i>	34	30	66	84	61	62	65	80	80	30	84
	<i>T_s</i>	697	690	732	746	720	722	733	740	742	690	746
77	<i>W</i>	23	13	19	18	39	68	35	27	13	13	68
	<i>T_s</i>	696	679	693	693	702	740	711	692	686	679	740
78	<i>W</i>	16	10	16	16	10	12	20	12	20	10	20
	<i>T_s</i>	640	634	650	655	634	642	652	642	650	634	655
79	<i>W</i>	8	6	46	41	9	18	25	23	6	6	46
	<i>T_s</i>	650	646	685	685	645	655	666	666	646	645	685
80	<i>W</i>	11	11	16	16	14	22	13	16	16	11	22

no.	Results	$R1$	$R2$	$R3$	$R4$	$R5$	$R6$	$R7$	$R8$	$R9$	Best	Worst
81	T_s	617	617	623	621	621	632	619	623	619	617	632
	W	26	32	50	51	55	20	89	41	54	20	89
	T_s	644	651	672	673	676	637	712	661	676	637	712
82	W	5	5	31	60	36	46	20	51	21	5	60
	T_s	663	662	693	722	691	703	683	709	682	662	722
	W	19	27	69	88	75	68	73	73	42	19	88
83	T_s	622	626	675	692	674	666	671	677	639	622	692
	W	11	11	33	38	23	20	43	16	39	11	43
	T_s	592	590	617	624	603	600	625	595	622	590	625
85	W	6	6	33	33	7	7	15	35	15	6	35
	T_s	636	635	667	668	635	642	641	668	641	635	668
	W	18	22	30	33	62	26	76	40	28	18	76
86	T_s	694	699	713	716	734	705	752	719	704	694	752
	W	23	11	28	33	30	57	36	37	11	11	57
	T_s	671	658	675	678	681	696	686	682	664	658	696
88	W	12	12	12	4	9	42	5	21	5	4	42
	T_s	628	628	627	622	627	651	626	633	626	622	651
	W	38	41	62	77	44	79	54	83	61	38	83
89	T_s	660	657	684	699	659	693	673	692	676	657	699
	W	11	2	9	12	36	47	21	41	13	2	47
	T_s	592	582	592	600	617	630	600	625	594	582	630
91	W	19	19	20	31	24	40	41	40	42	19	42
	T_s	643	640	645	661	646	665	668	664	671	640	671
	W	1	1	24	24	39	49	9	59	8	1	59
92	T_s	584	583	605	607	623	631	591	641	594	583	641
	W	18	22	32	55	65	67	31	36	30	18	67
	T_s	669	669	684	708	714	718	681	687	678	669	718
94	W	21	21	21	48	38	54	32	54	26	21	54
	T_s	650	649	651	683	669	682	665	682	659	649	683
	W	16	15	38	54	40	67	45	48	32	15	67
95	T_s	651	651	677	692	675	705	686	684	669	651	705
	W	4	4	30	28	29	43	4	39	4	4	43
	T_s	657	653	683	680	677	691	654	689	655	653	691
97	W	33	22	45	51	55	45	27	29	32	22	55
	T_s	664	646	675	676	681	667	657	655	659	646	681
	W	14	19	15	15	34	23	12	17	10	10	34
98	T_s	638	644	640	641	652	645	637	636	636	636	652
	W	67	88	109	128	97	129	121	116	98	67	129
	T_s	726	742	767	786	756	780	772	763	757	726	786
100	W	11	11	11	37	14	25	14	25	20	11	37
	T_s	637	637	642	672	643	656	648	656	647	637	672

B) BACAP_GP Results

In this approach, we used CDRs to solve the BACAP through the GP algorithm as described before. The experiment tested over 10 runs of the 100 instances benchmark.

CDR is the combination between the SPRs that was formulated in a mathematical equation, e.g. Equation (4.9). By applying the GP approach to the vessel list, we obtained the results shown in Table 4-6. The table shows the worst and the best total service values for all instances compared with the GP.

Table 4-6: Comparing the best T_s results from SPR with BACAP_GP

Instance no.	Worst SPR	Best SPR	GP	Instance no.	Worst SPR	Best SPR	GP
1	631	598	595	51	685	614	608
2	653	623	617	52	655	615	615
3	610	578	577	53	688	630	623
4	677	618	612	54	597	585	582
5	642	613	612	55	649	620	618
6	777	732	724	56	642	613	610
7	604	588	588	57	663	645	645
8	726	671	669	58	673	643	640

Instance no.	Worst SPR	Best SPR	GP	Instance no.	Worst SPR	Best SPR	GP
9	664	630	630	59	706	584	578
10	643	612	610	60	736	684	674
11	683	651	650	61	638	572	570
12	647	612	603	62	715	658	656
13	714	694	687	63	615	578	577
14	645	626	622	64	666	628	624
15	678	623	622	65	785	705	705
16	678	656	650	66	727	674	668
17	639	599	597	67	753	706	702
18	642	596	592	68	637	605	605
19	671	595	595	69	695	658	655
20	663	625	624	70	614	586	583
21	694	640	640	71	636	596	593
22	648	614	613	72	649	607	603
23	670	621	616	73	635	587	583
24	638	604	597	74	682	650	650
25	628	592	587	75	727	695	690
26	634	626	625	76	746	690	674
27	685	660	659	77	740	679	677
28	602	575	572	78	655	634	628
29	739	682	671	79	685	645	644
30	786	715	707	80	632	617	608
31	629	597	596	81	712	637	628
32	657	637	633	82	722	662	660
33	632	620	614	83	692	622	622
34	630	610	608	84	625	590	588
35	638	623	622	85	668	635	632
36	702	682	679	86	752	694	690
37	670	647	643	87	696	658	655
38	746	695	693	88	651	622	619
39	655	632	630	89	699	657	646
40	671	628	620	90	630	582	582
41	676	647	640	91	671	640	637
42	639	623	622	92	641	583	581
43	721	680	671	93	718	669	660
44	702	673	664	94	683	649	645
45	712	681	680	95	705	651	645
46	665	641	636	96	691	653	651
47	672	633	619	97	681	646	639
48	664	620	615	98	652	636	628
49	683	661	661	99	786	726	716
50	630	606	600	100	672	637	635

Figure 4-11 shows the overall observation of the best SPRs for 100 instances benchmarked with the *BACAP_GP* approach. As can be seen, the *BACAP_GP* has outperformed all of the other SPRs in almost all instances in the benchmark. Moreover, the flexibility of the model to cope with the needs of the terminal and

vessel operators by balancing between the dispatching rules was used to order all incoming vessels and to finish all services with the minimum time possible.

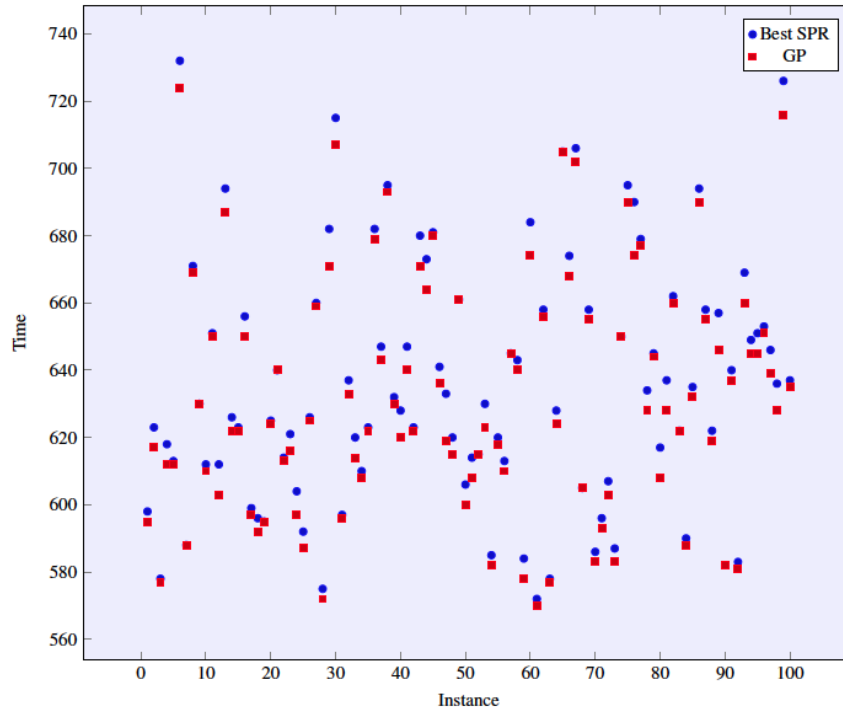


Figure 4-11: Comparing the results of best SPRs with the *BACAP_GP* for 100 instances

To study and examine the GP performance, Figure 4-12 shows the GP performance of three random runs out of ten for one instance over 50 generations. We noticed the distribution of the results over the 50 generations; all runs try to minimise the fitness by evolving the CDRs. The first generations are fast trying to get better results up to generation 25. After that, the GP hardly improves the results until the end of the run. As we know the GP method is an approximate method that cannot guarantee optimality, it still tries to get near to optimal. The more runs and trials where the characteristics of the runs are changed, (e.g. change the number of generations, population, crossover and mutation) the more it may lead to better results.

The distribution of the solutions for six randomly selected instances over ten runs has been shown in the box-plots in Figure 4-13. Each box-plot represents the distribution of the objective values where the horizontal line within the box encodes the median, and the upper and lower ends of the box are the third and first quarter respectively. The upper and lower vertical lines encode the maximum and minimum values. The box represents the spread of the data.

It appears that the minimum and maximum values are close to each other and since the objective function of the problem is to minimise the value of the objective function, we noticed that the spread of the data is close to the minimum in the box plot. These results show the efficiency of the GP in solving the problem over the different instances.

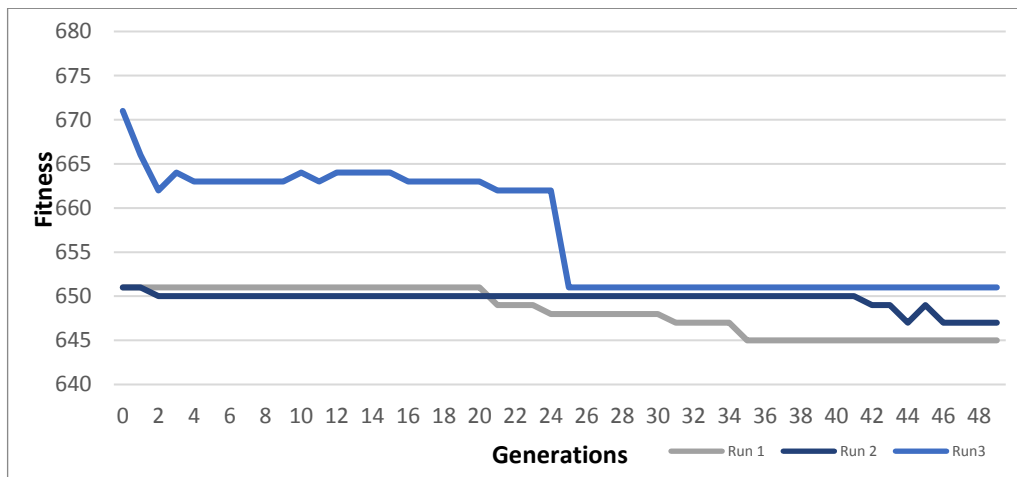


Figure 4-12: The GP performance of one instance and three runs over 50 generations.

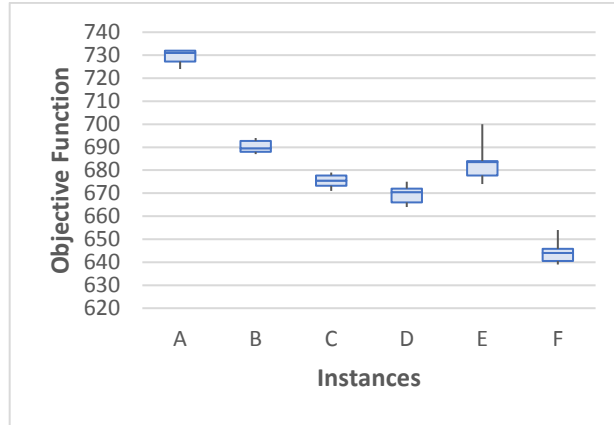


Figure 4-13: Box-plots of the distribution of the solutions for six randomly selected instances over ten runs.

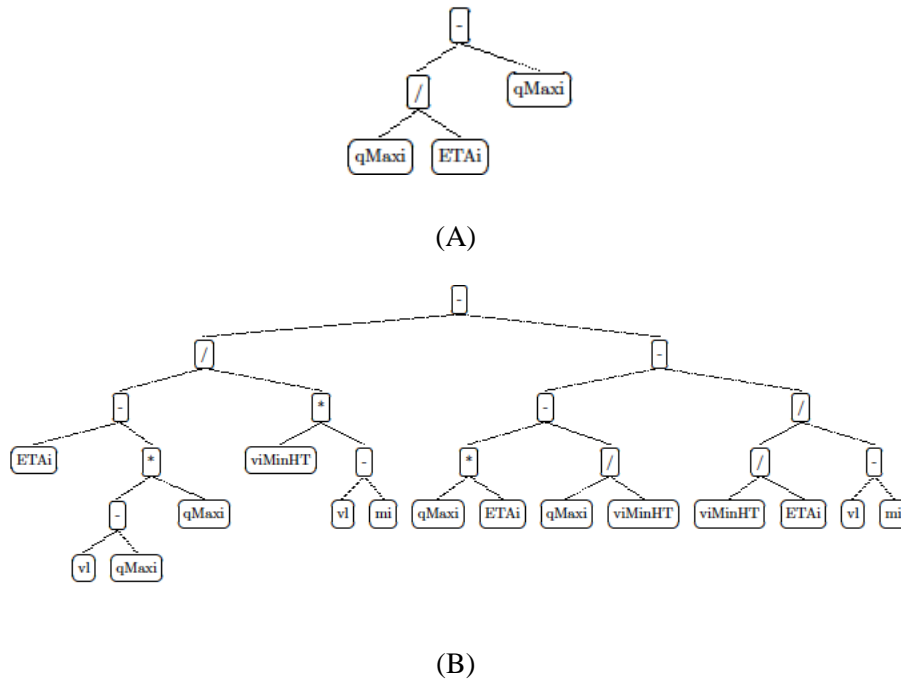


Figure 4-14: The GP trees, representing the worst (A) and best (B) CDRs evolved for a random instance.

Regarding the CDRs evolved by the GP, Figure 4-14 shows the worst CDR (A) and the best CDR (B) created by the GP to solve a random instance. The trees were generated automatically by the ECJ at the end of each run. We can observe that the CDR was used to get the best result is more complex than the other one. Almost all terminal sets used in this CDR were part of a complex combination with the function set. This means that it is hard to solve the instance in this experiment using a simple SDR or a straightforward CDR.

4.7 Conclusion

This chapter aims to contribute to the development of efficient optimisation methods to improve the performance of seaside container terminals. It presents a new optimisation method for the integrated berth allocation and quay crane assignment problem BACAP with no restriction of the vessels berthing position. The aim was to minimise the total service time for all incoming vessels.

The study presented a *BACAP_GP* framework based on two developed approaches to solve the BACAP. First, we introduced a genetic programming approach *BACAP_GP* which evolves vessel dispatching rules and determines an efficient and customised CDRs in order to provide the best priorities to vessels. The *BACAP_GP* combines different sets of attributes (vessel related attribute, berth related attribute and quay crane-related attribute). This process allows the *BACAP_GP* to generate a tailored CDRs based on the attributes given in each set, as the attributes may vary from one port to another.

Another advantage of the proposed *BACAP_GP* approach is that the evolved CDRs (construction heuristics) are reusable. In other words, the evolved DR heuristics are made up of mathematical equations that define the order of scheduling the vessels, as the dispatching rules play a key role in the overall performance of most optimisation methods based on performance-improving heuristics.

Second, we developed an independent efficient algorithm, *BACAP_Scheduler*, which is responsible for scheduling the vessel list ordered by given dispatching rules without violating given constraints. The novelty of the proposed scheduler can work independently with any method of optimisation techniques. We tested the *BACAP_Scheduler* using nine different SPRs. We concluded that the terminal manager may adopt the terminal priority strategy and use *R2 (Max. QC needed low priority)* or *R1 (FCFS)* if the large vessels' arrival probability is high. The study also provides a review of the previous research studies which solved the BACAP that considered dispatching rules in their approaches.

In terms of performance, the *BACAP_GP* approach has been evaluated against a wide range of SPRs available in the literature and have been tested on a benchmark

that contains a wide range of a large-scale instances. BACAP_GP outperformed all of the other SPRs in almost all instances in the benchmark.

In the next chapter, we will focus on improving the proposed approach after verified its performance to tackle the BACAP with desired berthing position and make it more generic to solve the problem with a technique to predict the possibility of vessels overlapping during the scheduling processes.

Chapter 5 : Enriched meta-heuristic approach and cost-based model for the BACAP with desired berthing position

5.1 Introduction

Water transportation is the cheapest transportation mode which allows for the transfer of massive volumes of cargo between continents. Containerised trade volume is rapidly increasing year after year, and seaport container terminal competition has also increased considerably. For large container terminals, terminal operators have to decide which berthing position suits which incoming vessel. This depends on many factors such as the number of quay cranes available to serve the vessel, the nearest berth to the dedicated space on the yard to store the containers in, contract issues, and the vessel's characteristics (size and draft). The main goal of the terminal operators is to minimise the total service cost for all incoming vessels, which in return will minimise the container terminal charging fee.

This chapter is to adapt and apply our framework to an extended version of the integrated berth allocation and quay crane assignment problem (BACAP). In this chapter, we will look at BACAP where the desired berthing position for incoming vessel is known in advance. Moreover, the vessel can be moored earlier than its ETA with an extra involved cost. The objective is to minimise the total service cost of all vessels. We consider the BACAP as presented in Meisel & Bierwirth (2009) which is an improved model formulation that was introduced by Park & Kim (2003). It is assumed that the vessel information provided in advance by the vessel operators to the terminal operator includes the berthing position. It is also assumed that the handling time for each vessel depends on the QC capacity demand, which is given as a number of QC-hours. The QC-hours required to serve a vessel will increase if it is not berthed at its desired position.

In this chapter, we improved the proposed algorithm that we introduced in the previous chapter, *BACAP_GP*, to solve the problem. Moreover, we provided a more terminal set to the genetic programming approach which concluded with better results.

The contributions of this chapter are (1) a generic genetic programming-based approach for evolving the dispatching rules for use related to the BACAP problem within the desired berthing position and time-variant QC assignment, (2) the development of a new technique to improve quay space utilisation while scheduling, (3) to provide a study and analysis of the real problems that are facing current container terminals and (4) to compare the proposed approach results with the results in the literature using the benchmark provided by Meisel & Bierwirth (2009).

The chapter is organised as follows. In the next Section, the real BACAP problem and a survey-based study has discussed. In Section 5.3, the problem description has been defined. In Section 5.4, the mathematical model with its notations and assumptions has been presented. The proposed algorithm, *BACAP_GP_DP*, has been presented in Section 5.5. In sub-section 5.5.3, a complete numerical example has been presented. The extensive computational results have been presented and analysed in Section 5.6. In Section 5.7, we have concluded the chapter and presented the suggested future research directions.

5.2 Real BACAP problem discussion

In this section, we have tried to investigate the real BACAP problem encountered in container terminals (problems faced, challenges, the current system for scheduling and any new technology used) and compared it with the existing literature. We visited one of the most important container ports in Egypt, which is DP World Sokhna.

5.2.1 DP World Sokhna General Information

DP World Sokhna is one of DP World's international networks that manages 60 terminals across 31 countries. Therefore, it is considered to be one of the largest

marine terminal operators in the world. DP World Sokhna has a strategic location at the heart of the vitally important East-West trade route. The location is below the southern entrance to the SUEZ Canal, on the Red Sea in Egypt as shown in Figure 5-1(A). The port has a West quay and East quay with a length of 750 meters each, with a water depth of 17 meters (Figure 5-1(B); website (<https://www.dpworldsokhna.com>)).

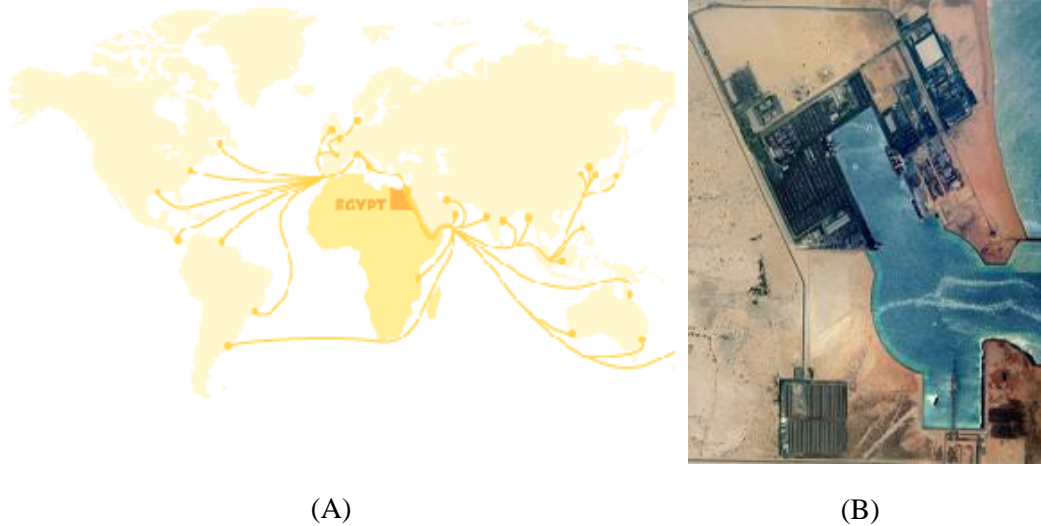


Figure 5-1: DP World Sokhna. (A) Location. (B) Masterplan

Regarding the quay cranes, the port has 4 Super Post PANAMAX twin lifts which can handle twin 60 ton containers under a spreader with 22 rows of width. There is also a 2 Post PANAMAX which can handle twin 60 ton containers under a spreader with an 18-row width. Figure 5-2 shows the difference between the Post PANAMAX cranes. Moreover, it also has 4 mobile quay cranes which can be used with no restriction in movement.



Figure 5-2: Post Panamax Crane

Regarding technology, DP World Sokhna uses a fully automated processing system which integrates all information into one system. The port uses a system called NAVIS & SPARCS for berth scheduling, managing and tracking terminals, yard operations and cargo movement. For more information on the features of NAVIS, we direct the reader to the website (<http://www.navis.com>).

5.2.2 DP World Sokhna Study

This study was done by visiting the DP World Sokhna port in December 2014. The main focus was to collect the real information regarding the BACAP problem and how the port handles such uncertainties. The summary of the questions that we requested being answered are as follows:

- What was the process that the seaport operators applied to handle incoming vessels?
- How to determine the number of cranes required to serve the vessels?
- What are the uncertain situations that can happen and how can the port operators handle it?
- Is there a software system used to schedule incoming vessels?
- What are the problem(s) facing this software?
- What is the average waiting time for vessels outside of the port?

- How long does it take the cranes to move (load/unload) containers?
- What is the type of contract between the port and ship owners? (constraints related to assigning a number of quay cranes and the overall service time)
- How does the port assign the priority of incoming vessels?

The summary of the outcomes are as follows:

- The ship sends the Baplie file (Appendix B.1), which contains information about the ship, the design, the number of containers that need moving, the placement of the containers on the ship and the ETA. The port confirms the ETA and assigns a berth to the ship, before calculating the handling time. This is the number of containers moved divided by the number of assigned QCs.
- The distance between two ships = 2 bollards = 27 meters.
- The distance between QCs = 1 Bay.
- QC time = from 17 to 25 movements per hour.
- The software system used is NAVIS & SPARCS in the beginning, before sending the information to Express to be saved.
- Regarding the contracts, there are many different types of contract. Usually the contract contains the average number of movements and the time window for arrivals. If the ship arrives in this time window, the port must give it priority and the contracted number of cranes to finish within the contracted handling time. If the ship comes later, the port gives it less priority and will have fewer QCs available for the ship. If the ship does not have a contract with the port in advance, the port gives it less priority.
- The port has new QCs that can handle two containers of 20 feet at the same time, but the containers must be beside one another.
- Sometimes, it is required to move some containers away from the desired one that needs to be unloaded from the ship. In this case, the port will assign one QC. Moreover, if the containers are all in one bay, then the port will assign one QC as it cannot assign two QCs to the same bay at the same time.

5.2.3 Sensitivity Analysis

CT operations in real ports comprise of more limitations and uncertainties than the constraints usually considered in most of the literature. Below is a description of some of the real-port challenges.

1. **Rescheduling:** In some cases, port operators have faced a limitation concerning the software that they are using to reschedule incoming vessels. For example, in the case of an unexpected last-minute change of a vessel's ETA due to bad weather or ship breakdown or an unforeseen situation that has occurred like the breakdown of a QC. Rescheduling the timetable has to be done manually, so they try to handle the situation using ad hoc rescheduling (Bruggeling et al., 2011; Du, Xu, & Chen, 2010; Mario Rodriguez-Molins, Salido, et al., 2014).
2. **Uncertainty:** there is a considerable amount of uncertainties associated with ETA. This is as ships may experience unexpected delays or be forced to change their direction. In addition to the ETA, there are several uncertainties related to the BAP. For example, it could be due to crane failure, sometimes the vessels are not exactly sure how many containers need to be loaded or unloaded, they need more services to sort out some of the containers onboard before they visit the next port or some of the containers to be unloaded needs more QC movement to reach the desired one. All of the above uncertainties may affect the schedule and delay the service time finish.
3. **Distribution of containers onboard:** Depending on the vessel length and the number of containers to be moved, a number of QCs can be assigned if there is a safe distance between cranes, and depending on the placement of the containers in the vessel's bay. For example, if we have a vessel that has 50 containers that need to be served on a large vessel length but all of them are in one bay, we cannot assign two QCs in this bay in order to keep the appropriate amount distance between the cranes. We will assign only one QC and will neglect the constraint of vessel length in our calculations.

4. Different constraints: The constraints are different from one port to another. Moreover, the constraints could change over time in the same port. For example, some ports may use mobility QCs, which are flexible cranes that can move from one berth location to another, thus they by-pass other cranes. This type of crane is more fixable but slower than a standard crane and therefore, they are governed by a different set of constraints (Carlo et al., 2015).
5. The contract between terminal operators (TO) and vessel operators (VO) includes the ETA, the average number of containers that need to be processed and the expected time to be taken to handle them. The TO might give priority to a vessel arriving within the time window (time between ETA and ETF) with the maximum number of QCs needed. If the handling time for serving this vessel exceeded the ETF, a fewer cranes may be assigned to complete its service. This mean, even the priorities are given to vessels with the maximum number of QCs contracted, the number of QCs might be change during the service time.
6. Number of quays: to the best of our knowledge, the models and algorithms used in the literature for solving the BACAP consider only one quay in the port. However, in practice, some ports have more than one quay. In such ports, ships with the same or close ETAs could be moored in different quays at the same time. To the best of our knowledge, most ports nowadays have started to expand their areas and add more container terminals in order to handle the increase of incoming vessels. The literature, however, often considers the port to have one quay for scheduled vessels. For this reason, we have directed our research to tackle the multi-port BACAP in the next chapter.

5.3 Problem Description

In this problem, the aim is to solve the BACAP in order to determine vessel starting time, the berthing position and to determine the number of QCs per hour that will serve each vessel in the planning horizon. The problem considers there to be a continuous single quay layout in a container terminal port with dynamic vessels arriving on time. The handling time for each vessel is evaluated by the number of assigned QCs for each one-hour period.

For the BACAP, the optimal scenario while scheduling incoming vessels is to find a space in the quay that satisfies the requirements of the vessel. For instance, if the vessel is scheduled to start being loaded/unloaded at the same time of its ETA, there is a space in the quay at the vessel's desired position and the maximum number of QCs requested are available, then in this case, the handling time and the total service cost for this vessel will be minimal.

Figure 5-3 illustrates the impact of changing a vessel's desired position while scheduling on its handling time and cost. The vessel shown in the blue rectangle is in the best position as it is on time (x-axis), as its starting time is similar to its ETA. It is also in the best position on the quay (y-axis), as it is moored similar to its desired berthing position. If this position during planning and scheduling process is occupied by another vessel, then the vessel should be moved either to be on time (x-axis) or in the quay (y-axis), or both. If the vessel was moved on time only, then the cost will increase. On the other hand, if the vessel was moved in the quay, then the handling time and cost will also increase.

Therefore, it is shown that if the vessel is not able to satisfy the requested desired berthing position in its ETA, it will be better for it to move to another available time that satisfies its desired berthing position and required number of QCs to minimise its handling time but the cost will be high.

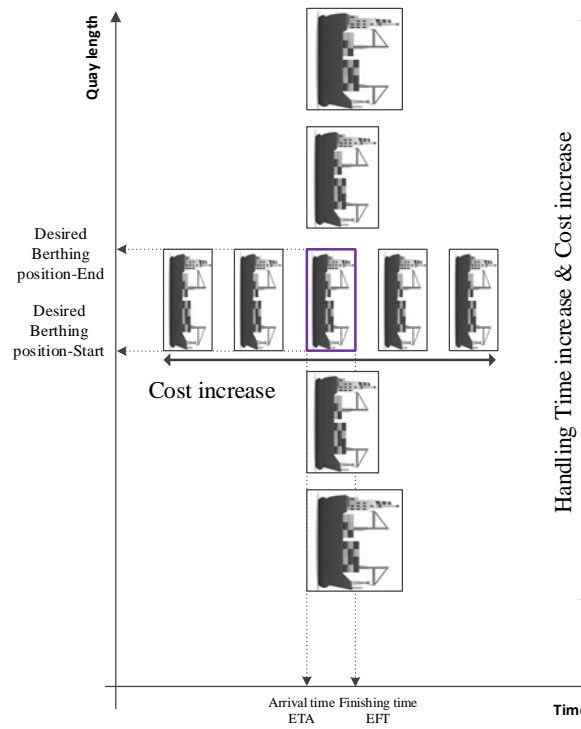


Figure 5-3: The impact of changing a vessel from its desired position on its handling time and cost.

This type of problem is more complicated than the one described in the previous chapter, where the BACAP did not involve a desired position requested in advance. Accordingly, the *BACAP_Scheduler* can allocate any berth in the quay without any additional cost, and it is manageable to relocate the vessel over the quay in the case of overlapping with other vessels with the same ETA. The method used for calculating the handling time of a vessel depends on the number of QCs, and their movement and assignment to a vessel only. On the contrary, the problem in this chapter calculates the handling time considering the previous in addition to the number of the vessel shifts from its desired position. This movement may lead to an increase of the handling time. The problem in this chapter also considers the possibility of a vessel arriving earlier than its ETA and berthing with an additional cost.

Figure 5-4, illustrates the structure of the quality service costs proposed by Meisel & Bierwirth (2009), which we have considered in our problem. If the vessel is berthed earlier than its ETA, a cost called (*speedup cost*) is added, and this cost

increases respectively to the time between the earlier starting time EST and the ETA. If the vessel exceeds its EFT, we start to add a cost called the (*delay cost*), which in return continuously increases with the increase of time between the expected finish time EFT until the late finish time LFT. If the vessel is delayed beyond the LFT, a cost called (*penalty cost*) is added to the delay cost until the actual finishing time when serving the vessel. The ideal scenario that can be shown is that if the vessel started at the ETA and finished before or at the EFT, then no cost is added.

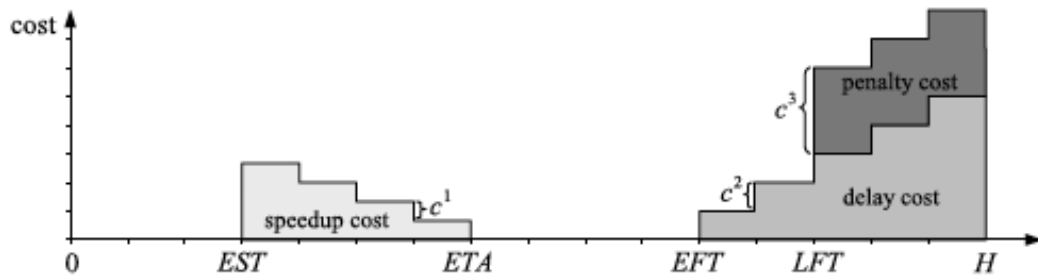


Figure 5-4: Structure of the service cost of a vessel.(Meisel & Bierwirth, 2009)

5.4 Mathematical Model Formulation

In this study, the BACAP is modelled as a single objective function with the aim of minimising the total service cost for all vessels. The problem's definition and the mathematical model was obtained from Meisel & Bierwirth (2009) with minor modifications.

According to the classification scheme in Bierwirth & Meisel (2010), our approach is represented by (BAP, QCAP (number)), which is the integration of the berth allocation problem and the quay crane assignment problem to decide on the berthing position, the berthing time, and the number of cranes to assign to each

vessel. Moreover, the problem is defined as *cont|dyn|pos*, *QCAP* | Σ ($w_1speed + w_2tard + w_3res$), which is described as follows:

- *Spatial attribute: Continuous layout:* The quay has a continuous layout and no partitioning, so the vessel can berth depending on its length within the boundaries of the quay's length.
- *Temporal attribute: Dynamic arrival:* All fixed arrival times are known in advance for all incoming vessels, so then the arrival times restrict the earliest berthing times by adding more costs.
- *Handling time attribute: unknown in advance:* The handling time of a vessel is unknown in advance, and it depends on its berthing position (*pos*) and the number of assigned QCs (*QCAP*).
- *Performance measure: Σ ($w_1speed + w_2tard + w_3res$):* The objective function used to measure the performance is intended to minimise the total sum of the combined weight of the following:
 - *speed:* Speedup of a vessel to reach the terminal before the expected arrival time.
 - *tard:* Tardiness of a vessel against the given due date.
 - *res:* Resource utilisation affected by the service of a vessel.

5.4.1 Model Assumption

The proposed model was established based on the following assumptions:

- The vessel can be moored in the quay if there is a space greater than or equal to the vessel length, and if there are the minimum number of QCs required available.
- The number of QCs assigned to vessels is greater than or equal to the minimum number of allowed quay cranes and less than or equal to the maximum number of allowed quay cranes.

- The QCs can be assigned to vessels using a dynamic approach (time-variant) so then the QCs can be assigned to another vessel before the original vessel departs.
- Information related to incoming vessels known in advance includes the length of the vessel, desired position, crane capacity demand, the minimum and maximum QC range, earliest starting time, the expected time of arrival, the expected finishing time, and the late finishing time.
- Every vessel has a draft that is lower than the draft of the quay.
- The vessel's length includes the required safety margin, which is the safe distance between two moored vessels.
- The handling time of a vessel depends on the number of assigned QCs, the QCs capacity demand and the vessel berthing position, details will be explained in section 5.4.3.

5.4.2 Notations

The following are the notations used in this approach:

Input data:

V	set of vessels to be served, $V = \{1, 2, \dots, n\}$.
Q	number of available QCs.
L	number of 10-meter berth segments (length of the quay).
T	set of 1-hour periods, $T = \{0, 1, \dots, H - 1\}$, H is the planning horizon.
l_i	length of vessel $i \in V$ given as a number of 10-m segments.
b_i^0	desired berthing position of vessel i .
m_i	crane capacity demand of vessel i given as a number of QC-hours.
r_i^{\min}	minimum number of QCs agreed to serve vessel i simultaneously.
r_i^{\max}	maximum number of QCs allowed to serve vessel i simultaneously.
R_i	feasible range of QCs assignable to vessel i , $R_i = [r_i^{\min}, r_i^{\max}]$.
ETA_i	expected time of arrival of vessel i .
EST_i	earliest starting time if journey of vessel i is speeded up, $EST_i \leq ETA_i$
EFT_i	expected finishing time of vessel i .
LFT_i	latest finishing time of vessel i without penalty cost arising.
c_i^1, c_i^2, c_i^3	service cost rates for vessel i given in units of 1000 USD per hour.
c^4	operation cost rate given in units of 1000 USD per QC-hour.
c_i^5	Overlap cost rate which is the possibility of overlap between vessel i and the rest of vessels, given in units of 1000 USD per space 10-m segment.
α	interference exponent.
β	berth deviation factor.
M	a large positive number.
w	Weight of overlap

Decision and calculated variables:

b_i	integer, berthing position of vessel i .
s_i	integer, time of starting the handling of vessel i (berthing time).
e_i	integer, time of ending the handling of vessel i (finishing time).
r_{it}	binary, set to 1 if at least one QC is assigned to vessel i at time t , 0 otherwise.
r_{itq}	binary, set to 1 if exactly q QCs are assigned to vessel i at time t , $q \in R_i$, 0 otherwise.
Δb_i	integer, deviation between the desired and the actually chosen berthing position of vessel i , $\Delta b_i = b_i^0 - b_i $
ΔETA_i	integer, required speed up of vessel i to reach its berthing time, $\Delta ETA_i = (ETA_i - s_i)^+$
ΔEFT_i	integer, tardiness of vessel i , $\Delta EFT_i = (e_i - EFT_i)^+$
u_i	binary, set to 1 if the finishing time of vessel i exceeds LFT_i , 0 otherwise
y_{ij}	binary, set to 1 if vessel i is berthed below of vessel j , i.e. $b_i + l_i \leq b_j$, 0 otherwise
z_{ij}	binary, set to 1 if handling of vessel i ends not later than handling of vessel j starts, 0 otherwise
d_i^{min}	Minimum duration needed to serve vessel i (minimum handling time).
ox_i	Number of time slot (hours) overlap vessel i with the rest of vessels if any.
oy_i	Number of spaces (10m-segment) overlap vessel i with the rest of vessels if any.
v_iOrder	The order of vessel i in the V .

5.4.3 Mathematical Model

The mathematical model obtained from Meisel & Bierwirth (2009) with minor modifications.

$$\text{minimize } Z = \sum_{i \in V} \left(c_i^1 \cdot \Delta ETA_i + c_i^2 \cdot \Delta EFT_i + c_i^3 \cdot u_i + c^4 \cdot \sum_{t \in T} \sum_{q \in R_i} q \cdot r_{itq} + c_i^5 \right) \quad (5-1)$$

$$d_i^{min} = \left\lceil \frac{(1 + \beta \cdot \Delta b_i) \cdot m_i}{(r_i^{max})^\alpha} \right\rceil \quad (5-2)$$

$$c_i^5 = \begin{cases} w \cdot \sum_{j \in V} oy_{ij}, & \text{if } (ox_{ij} \text{ and } oy_{ij}) > 0 \\ 0, & \text{Otherwise} \end{cases} \quad \forall i \in V \quad (5-3)$$

$$\sum_{t \in T} \sum_{q \in R_i} q^\alpha \cdot r_{itq} \geq (1 + \beta \cdot \Delta b_i) \cdot m_i \quad \forall i \in V \quad (5-4)$$

$$\sum_{i \in V} \sum_{q \in R_i} q \cdot r_{itq} \leq Q \quad \forall t \in T \quad (5-5)$$

$$\sum_{q \in R_i} r_{itq} = r_{it} \quad \forall i \in V \quad \forall t \in T \quad (5-6)$$

$$\sum_{t \in T} r_{it} = e_i - s_i \quad \forall i \in V \quad (5-7)$$

$$(t + 1) \cdot r_{it} \leq e_i \quad \forall i \in V \quad \forall t \in T \quad (5-8)$$

$$t \cdot r_{it} + H \cdot (1 - r_{it}) \geq s_i \quad \forall i \in V \quad \forall t \in T \quad (5-9)$$

$$\Delta b_i \geq b_i - b_i^0 \quad \forall i \in V \quad (5-10)$$

$$\Delta b_i \geq b_i^0 - b_i \quad \forall i \in V \quad (5-11)$$

$$\Delta ETA_i \geq ETA_i - s_i \quad \forall i \in V \quad (5-12)$$

$$\Delta EFT_i \geq e_i - EFT_i \quad \forall i \in V \quad (5-13)$$

$$M \cdot u_i \geq e_i - LFT_i \quad \forall i \in V \quad (5-14)$$

$$b_j + M \cdot (1 - y_{ij}) \geq b_i + l_i \quad \forall i, j \in V, \quad i \neq j \quad (5-15)$$

$$s_j + M \cdot (1 - z_{ij}) \geq e_i \quad \forall i, j \in V, \quad i \neq j \quad (5-16)$$

$$y_{ij} + y_{ji} + z_{ij} + z_{ji} \geq 1 \quad \forall i, j \in V, \quad i \neq j \quad (5-17)$$

$$s_i, e_i \in \{EST_i, \dots, H\} \quad \forall i \in V \quad (5-18)$$

$$b_i \in \{0, 1, \dots, L - l_i\} \quad \forall i \in V \quad (5-19)$$

$$\Delta ETA_i, \Delta EFT_i \geq 0 \quad \forall i \in V \quad (5-20)$$

$$r_{itq}, r_{it}, u_i, y_{ij}, z_{ij} \in \{0, 1\} \quad \forall i, j \in V \quad \forall t \in T \quad \forall q \in R_i \quad (5-21)$$

The objective function in this model is shown in equation (5-1), which is designed to solve the BACAP with the minimum total service costs for scheduling all incoming vessels to the quay. Equation (5-2) calculates the minimum handling time needed to serve vessel i . From this equation, we can note that the minimum handling time is obtained if the desired position and maximum QCs required are both satisfied. Equation (5-3) predicts the possibility of overlap between vessels and calculates the cost. The cost of this overlap is the total length of the overlapped vessel i with the rest of the unscheduled further vessels multiplied by the weight of the overlap. This will be described in the next sections. Constraints (5-4)-(5-6) are

responsible for the QCs assigned to each vessel that satisfy the required QC capacity needed concerning productivity losses caused by QC interference and the chosen berthing position. Constraints (5-7)- (5-9) set the start and end times for serving vessels while considering the QC assignment to a vessel. Constraints (5-10)- (5-13) are used to find the deviations from the desired berthing position and to determine the expected arrival and finish time for each vessel. Constraint (5-14) defines if the vessel finishes later than the EFT (delay). Constraints (5-15)-(5-17) prevent an overlap in the time and space of the vessels assigned to the same position. Constraint (5-18) sets the boundaries for the start and end times between EST and the planning horizon. Constraint (5-19) ensures that each vessel is positioned within the quay boundaries. Constraints (5-20)-(5-21) define the domains of the remaining decision variables.

5.5 Genetic Programming for the BACAP with desired berthing position (*BACAP_GP_DP*)

The proposed algorithm *BACAP_GP_DP* is the improved algorithm of *BACAP_GP* as described in the previous chapter, which can solve the BACAP with more realistic constraints such as, vessel's desired berthing position and dynamic arrival time. The improvements consist of the two stages of solving the BACAP problem, which are the *BACAP_Scheduler* and the GP approach respectively. More detail will be explained in the following sections.

5.5.1 *BACAP_GP_DP* Framework

In this approach, we added more terminal sets to the GP to improve the CDRs generated by the GP. Moreover, we tuned the GP Tree parameters in order to enhance the solution and make the algorithm run faster.

GP is an evolutionary computation technique that automatically finds computer programs used to solve a specific task; it is a domain-independent method for getting a solver rather than for finding a solution (Langdon, Poli, McPhee, & Koza,

2008). Comparing the GP with the genetic algorithm GA indicates that the GP population is not represented by a fixed string length of genes. Individuals in the GP are usually represented by different lengths of tree program, which makes the GP a powerful tool to solve NP-hard problems such as the BACAP.

After introducing the *BACAP_GP* in the previous chapter and approving its performance when solving the BACAP, we concluded the importance of the dispatching rules and its impact on solving the BACAP. We applied the GP in this chapter, *BACAP_GP_DP*, in order to discover the superior construction of the CDRs and to compare their performance with the state of the art results.

The main relationship between the GP engine and the scheduler has been shown in Figure 4-5. The *BACAP_GP_DP* is responsible for evolving the ship's dispatching rules and ordering the vessel list accordingly, before sending the vessel list to the *BACAP_Scheduler_DP*. The *BACAP_Scheduler_DP* is responsible for handling the scheduling process using the order of vessels obtained by the *BACAP_GP_DP*, and then returning the best possible schedule for evaluation using the *BACAP_GP_DP* and vice versa.

The following is the *BACAP_GP_DP* representation, which includes the terminal set and function set that was used in our algorithm.

5.5.1.1 BACAP_GP_DP Representation

In this section, we will describe the GP representation proposed for generating individuals using the terminal set and function set.

3) Terminal set:

It is crucial for the results' quality and efficiency to choose a good terminal set and function set. Moreover, reducing these sets will reduce the search space and make the GP algorithm run faster. Therefore, we proposed highly effective sets which can be used to find a better solution for the BACAP. The proposed terminal set in this study is shown in Table 5-1; these values can be found and calculated in the initialisation stage or during the scheduling process for each vessel.

Table 5-1: Terminal set for the *BACAP_GP_DP* tree representation.

Notation	Description
vl	Vessel length
ETA_i	Vessel expected time of arrival for the given vessel i
EFT_i	Vessel expected finishing time for the given vessel i
EST_i	Earliest arrival time for the given vessel i
LFT_i	Latest finish time for the given vessel i
q_i^{Max}	The maximum number of QCs that can be assigned to a vessel
m_i	Number of movements (loading/unloading)
v_iMinHT	Time for handling a vessel while it is working with q_i^{Max}
v_{pos}	vessel's preferred mooring position
pos	Terminal preferred position
<i>Overlap</i>	The amount of overlap with other vessels
<i>Tolerance</i>	Reflects the amount of time a vessel can wait and still finishes
<i>Density</i>	Amount of movement need per hour, so the ship service can finish before the vETA
<i>Gap</i>	The length of unusable of the berth if allocated in its preferred position.
<i>Allowance</i>	The time between estimated time of finish and latest finish time
<i>SpeedUp</i>	Duration between earliest time of arrival and expected time of finish
<i>Space</i>	The minimum required slots to serve the ship which is length of the ship multiplied by the minimum service time
<i>Volume</i>	The space multiplied by maximum number of QC

In Table 5-1, v_{pos} represents the given vessel's preferred position as stated by the vessels' operators, while pos represents the preferred position from the terminal operator's perspective during the scheduling process. *Overlap* represents the number of predicted *Overlap Times* and *Overlap Spaces* that are identified between a vessel and others (described in the next Section). The rest of the terminal set proposed has been explained in the above table.

4) *Function Set:*

The function set will consist of standard mathematical operators as shown in Table 5-2.

Table 5-2: Function set for the *BACAP_GP_DP* tree representation.

Notation	Description
<i>Add</i> (x, y)	Addition (+)
<i>Sub</i> (x, y)	Subtraction (-)
<i>Mul</i> (x, y)	Multiplication (*)
<i>Div</i> (x, y)	Safe division (/), returns 1 if the denominator equals to 0
<i>Avg</i> (x, y)	Returns the average value of the input variables.
<i>Min</i> (x, y)	Returns the minimum of the two inputs.
<i>Max</i> (x, y)	Returns the maximum of the two inputs.
<i>Abs</i> (x)	Returns the absolute value of variable x .
<i>Ceiling</i> (x)	Returns the ceiling value of the input.
<i>Floor</i> (x)	Returns the floor value of variable x .

5.5.1.2 Generate a set of initial solutions

The *BACAP_GP_DP* generates an initial population randomly by creating a number of individuals as illustrated before. Each individual represents ordering vessels, and one possible solution with its fitness function. The *BACAP_GP_DP* determines the best individual out of the initial population and evolves it to the next generation.

5.5.1.3 Fitness function

In this study, our objective is to find effective CDRs for solving the BACAP with a minimum total cost of Z for all vessels, as shown in Equation (5-1). Therefore, we have proposed a method to form a CDR from the tree-based result of GP. This CDR is then, with the minimum service cost, used to evaluate the fitness value of the BACAP.

5.5.1.4 Crossover & Mutation Operation

The crossover and mutation operation in the proposed *BACAP_GP_DP* system is implemented as the same “Koza-style” (Koza, 1994). This is explained in detail in the previous chapter.

5.5.2 BACAP_Scheduler_DP

This approach is the extended approach for the *BACAP_Scheduler* that was described in the previous chapter. After investigating the literature, we found that this approach is more generic and can handle scheduling all incoming vessels in

order to solve the BACAP in most scenarios. The significant improvements of this approach include: (1) considering the desired berthing position of a vessel and providing a cost for any shifts; (2) accepting the earliest starting time EST if the journey of a vessel is speeded up; (3) determining the different costs if a vessel is berthed at a different time/position than what is expected and (4) examining the possibility of overlap between each vessel.

The model starts by reading the vessel data and initialising the solution by assigning Z to infinity, resetting Q , filling in the schedule array with zeros, calculating the minimum handling time for each vessel using equation (5-2) and checking the possibility of overlap between incoming vessels while also assigning a cost c^5 for each vessel. We can calculate c^5 as it is the total area of intersection between two vessels that are overlapping. First, we check the availability of QCs in the time between vessels ETA and EFT . Second, we check to see if the desired position of the vessel is empty. If the QCs are not available or the vessel is overlapping with another vessel, then we start testing all of the times from EST to time horizon H and all the possibility berths in the quay. We then calculate the total cost of the vessel in this time and position using equation (5-1). From this, we can determine the best time and position data with the minimum cost to the vessel. These processes continue until the end of the vessel list V . Finally, we calculate the total cost of the schedule then send it to the GP engine for evaluation. To speed up the running time, we implemented stopping conditions if there is no improvement to be found in the solution or if there is no availability regarding the QCs.

After the *BACAP_Scheduler_DP* places a vessel in the array, it fills in the location with the correspondent Vessel ID. This shows the location of the vessel in the schedule. Zeros denote empty spaces. A rectangle demonstrates the vessel, with its position on the horizon starting from s and ending in e for the time horizon (x-axis), and the length of the rectangle showing the length of the vessel v_i in the quay (y-axis).

From Figure 5-3, we can determine that the best solution with the minimum cost involved can be obtained if all vessels can berth in their desired berthing positions

at the expected time of arrival. If this is not the case, two possible scenarios can be found, as follows:

1. If the available QCs between vessel ETA and vessel $EFT < r_i^{min}$ for the vessel leads to the vessel finding another suitable starting time between EST and the time horizon.
2. If the desired position for a vessel at its ETA is occupied, this means that the vessel will overlap with another vessel. This will lead the new vessel to shifting from the desired position to search for a free space in the quay.

From equation (5-1) which calculates the total cost, and from equation (5-2) which calculates the minimum handling time, we can notice that scenario (1) is worse than scenario (2) as above, because shifting the vessel from the desired position will cost less than changing its starting time. However, scenario (2) can be worse than scenario (1) if the distance of shifting exceeds the limit and the vessel handling time duration begins to increase, as it would then need more quay cranes/hours to finish loading or unloading. Therefore, the proposed approach to the *BACAP_Scheduler_DP* can manage the above scenarios by executing the following:

1. The *BACAP_Scheduler_DP* starts to test every vessel against the rest of incoming vessels in the list to see if there is *Overlap Time* with it. If yes, the *BACAP_Scheduler_DP* will decrease the number of maximum quay cranes needed for all vessels that *Overlap Time*, such that the total number of quay cranes for all vessels with an *Overlap Time* less or equal to the total number of QCs in the quay. This approach will solve scenario (1). We defined vessel 1 as *Overlap Time* vessel 2 if one of the following occurred:

- 1.a. If $((ETA_2 \geq s_1) \text{ and } (ETA_2 \leq e_1))$
- 1.b. If $((EFT_2 \geq s_1) \text{ and } (EFT_2 \leq e_1))$
- 1.c. If $((ETA_2 \leq s_1) \text{ and } (EFT_2 \geq e_1))$

From the above, we can determine the total time that vessel 1 overlapped vessel 2 by:

$$ox_{12} = \min ((e_1 - ETA_2), (EFT_2 - s_1)).$$

Figure 5-5 illustrates the *Overlap Time*. The *BACAP_Scheduler_DP* calculates the *ox* and predicts the future schedule for the current vessel and the vessels further along in the order, so it minimises the future overlap between vessels and utilises the QCs usage in each time slot.

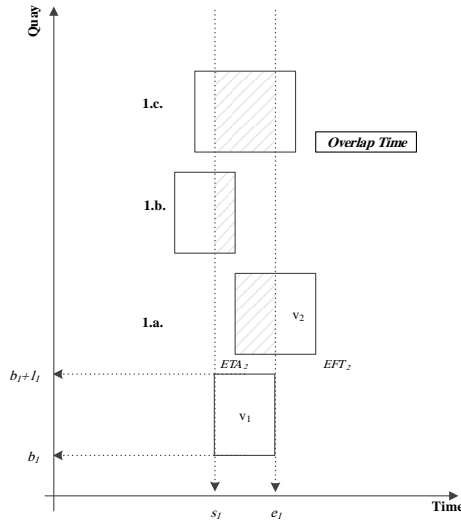
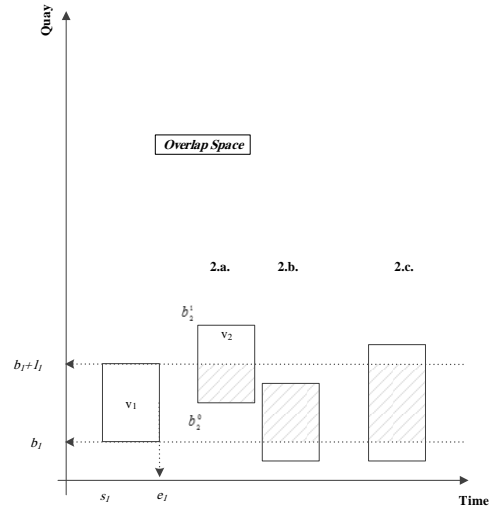
2. The *BACAP_Scheduler_DP* starts to test every vessel in the list to see if there is an *Overlap Space*. If yes, then the *BACAP_Scheduler_DP* will start to shift the current vessel to be scheduled by adding coast c^5 , which will allow for the shifting of the vessel in the quay from its desired position while not exceeding the limit as described before. This approach will manage scenario (2). We defined vessel 1 as *Overlap Space* vessel 2 if one of the following occurred:

- 2.a. If $((b_2^0 \geq b_1) \text{ and } (b_2^0 \leq b_1 + l_1))$
- 2.b. If $((b_2^1 \geq b_1) \text{ and } (b_2^1 \leq b_1 + l_1))$
- 2.c. If $((b_2^0 \leq b_1) \text{ and } (b_2^1 \geq b_1 + l_1))$

From the above, we can determine the total spaces that vessel 1 overlapped vessel 2 by: $oy_{12} = \min ((b_1 + l_1 - b_2^0), (b_2^1 - b_1))$.

Figure 5-6 illustrates the *Overlap Space*. The *BACAP_Scheduler_DP* calculates the *oy* and predicts the future schedule for the current vessel and the vessels further along in the order, so it minimises any future overlap between vessels and utilises the QCs in each space.

The *BACAP_Scheduler_DP* will continue testing each vessel in the list with the next vessels' order. If case 1 (*overlap Time*) and case 2 (*Overlap Space*) are satisfied between two vessels, then full overlap has occurred and we can start to calculate c^5 to the current vessel. The aim of using c^5 is to force the current vessel, that will overlap future vessels, from its desired position so then the future vessels that need this position in the quay can find a space to berth with minimum cost.

Figure 5-5: Definition of *Overlap Time*Figure 5-6: Definition of *Overlap Space*

5.5.3 Numerical Example

In this section, we demonstrate a complete example of solving the BACAP with desired berthing position in a real instance as provided by Meisel & Bierwirth (2009). We solved the problem using our proposed method, *BACAP_GP_DP*.

Table 5-3 shows a sample of the instance including 20 vessels and the data provided by the vessel's operators. The quay length and setting parameters used in this example are shown in section 5.6.1. In the BACAP problem with desired berthing position, we need to schedule all of the vessels by allocating a suitable berth on the quay and finding a starting time for each vessel. We then need to determine the number of QCs to serve the vessels in a time-variant manner. The objective is to find the minimum total service cost for this schedule.

Table 5-3: A sample of instance with 20 vessels produced by (Meisel & Bierwirth, 2009) and solved using *BACAP_GP_DP*.

<i>Vessel ID</i>	l	b^0	m	ETA	EST	EFT	LFT	r^{min}	r^{max}	c^1	c^2	c^3
1	15	60	6	3	3	7	8	1	2	1	1	3
2	8	75	7	9	8	13	15	1	2	1	1	3
3	19	69	6	9	8	13	14	1	2	1	1	3
4	10	2	13	13	12	20	23	1	2	1	1	3
5	21	17	50	29	26	44	51	2	4	2	2	6

6	39	14	57	41	37	53	58	4	6	3	3	9
7	20	59	14	43	39	51	54	1	2	1	1	3
8	25	31	28	57	52	66	69	2	4	2	2	6
9	18	42	8	64	58	69	70	1	2	1	1	3
10	21	39	15	69	62	74	75	2	4	2	2	6
11	15	21	5	86	78	89	90	1	2	1	1	3
12	34	16	53	88	80	99	104	4	6	3	3	9
13	14	66	12	96	87	103	106	1	2	1	1	3
14	10	73	9	102	92	107	109	1	2	1	1	3
15	19	74	13	108	98	115	118	1	2	1	1	3
16	15	79	15	110	99	119	122	1	2	1	1	3
17	26	55	20	130	117	136	139	2	4	2	2	6
18	19	43	10	137	124	143	145	1	2	1	1	3
19	24	23	20	141	127	147	150	2	4	2	2	6
20	25	73	34	150	135	160	165	2	4	2	2	6

First, the *BACAP_GP_DP* tries to evolve the CDRs using the proposed terminal and function sets. The CDRs are responsible for ordering the vessels as part of the dispatching rule while scheduling. Second, the *BACAP_GP_DP* sends the ordered vessels to *BACAP_Scheduler_DP* to schedule the vessels, to get the total costs of the schedule and to send it back to the *BACAP_GP_DP* to evaluate the fitness function of the schedule.

Figure 5-7 shows the best CDR evolved by the *BACAP_GP_DP* after running 50 populations and 50 generations. This CDR will generate a function (program) as shown in Equation (5-22) which can be simplified to $-1/(EFT_i * ETA_i)$ to orders the vessels shown in Table 5-3 to be as follows. Vessel ID: 20, 12, 2, 1, 3, 4, 8, 16, 15, 14, 13, 19, 17, 18, 11, 10, 9, 7, 6, 5.

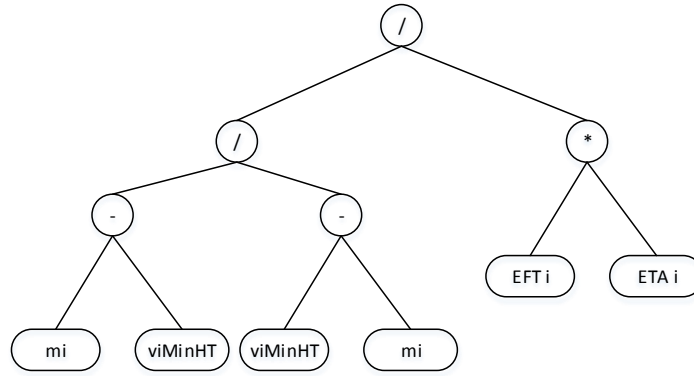


Figure 5-7: Example of the best CDR evolved by BACAP_GP_DP to solve the BACAP instance.

$$v_iOrder = \frac{m_i - v_iMinHT}{v_iMinHT - m_i} / EFT_i * ETA_i \quad (5-22)$$

The *BACAP_Scheduler_DP* uses the previous ordered vessels and applies the proposed model to determine the best schedule plan that minimises the total service cost (Z). Figure 5-8 shows the optimal solution plan for the 20 vessels with the value of $Z=53.9$. The x-axis is the time horizon (10-hour time segments) and the y-axis is the quay length (10-metre berth segments). The plan shows each vessel where and when it will berth. Moreover, it shows how many QCs will serve each vessel in each one-hour period of time. The small shaded square shows the QC.

For instance, the ETA and actual starting time (s) for V1 is 3, and the desired and actual berthing position is 60. The EFT and the actual finish time (e) is 7. The minimum QCs requested is 1 and the maximum is 2, so the scheduler assigned 2 QCs for the first 3 hours and then assigned 1 QC for the fourth hour (time-variant). The total cost for serving V1 using Equation (5-1) means that $Z=1*0+1*0+3*0+0.1*7+0.01*0=0.7$ (\$700). For vessel V11, the ETA is 86 but the actual is 85. The EFT is 89 and the actual is 88. The desired berthing position and the actual is 21. Calculating the total cost means that $Z=1*1+1*0+3*0+0.1*6+0.01*0=1.6$ (\$1600). By evaluating the rest of the vessels, we can determine that the total service cost for the 20 vessels is 53.9 (\$53,900) which is the optimal solution obtained using CPLEX as reported by Meisel & Bierwirth (2009).

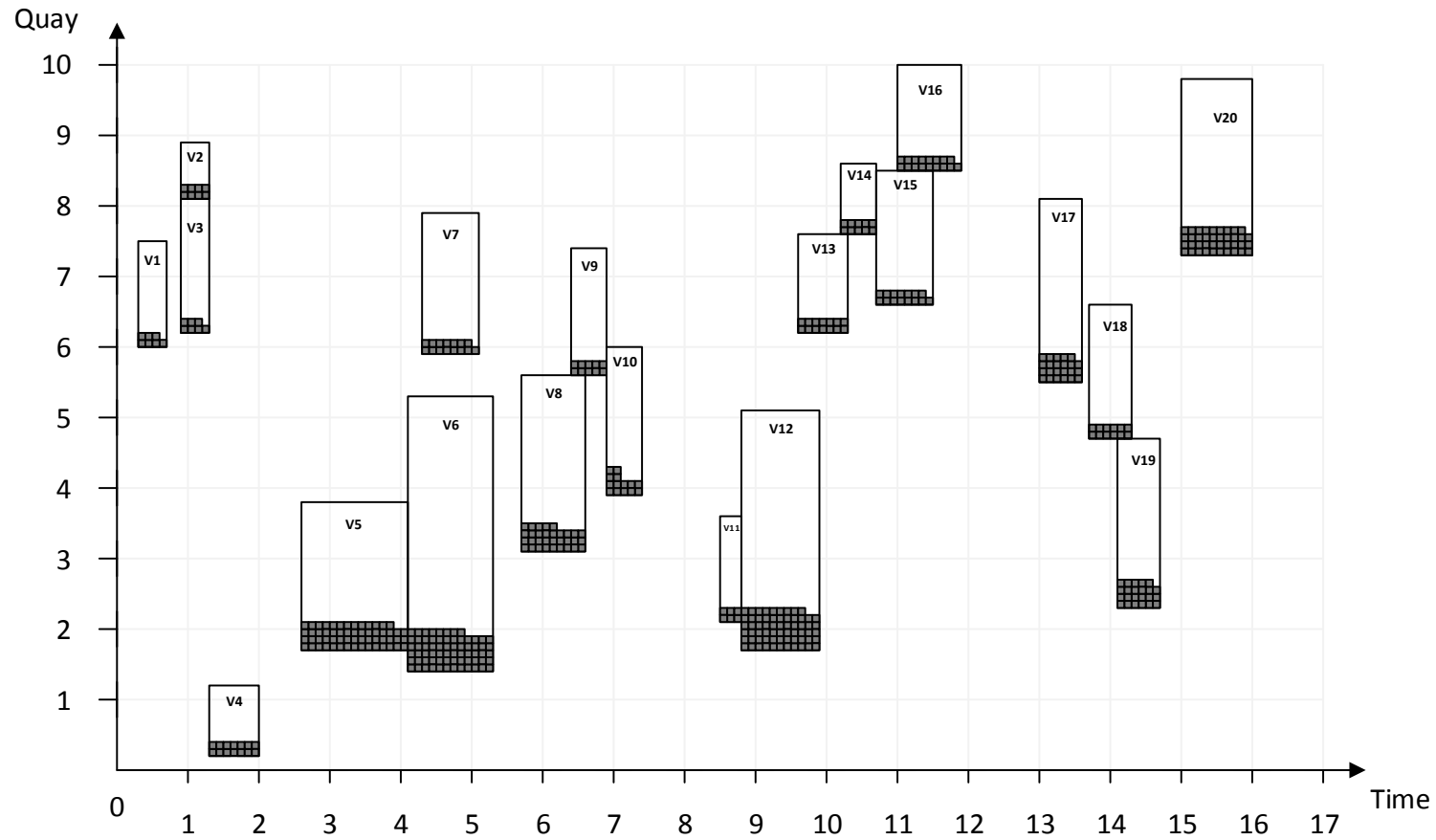


Figure 5-8: Example of complete berth Plan with QCs.

5.6 Computational Experiments and Analysis

In this section, computational experiments were conducted to evaluate the performance of the proposed algorithm. We defined the standard SPRs commonly used in the literature and then produced a comprehensive study and comparison of the SPRs along with the evolved CDRs obtained from the *BACAP_GP_DP*. The *BACAP_Scheduler_DP* and the *BACAP_GP_DP* were developed using Java and included the ECJ22 library (A Java-based Evolutionary Computation Research System, n.d.) in order to implement the GP. All tests were run for 10 hours or until the stopping conditions were reached on a core i3 intel processor with 1.8GHz and 4GB of RAM.

5.6.1 Datasets and parameter settings

The experiments were performed using the same benchmark introduced by Meisel & Bierwirth (2009) to compare our approach. The benchmark contains three sets of test instances including 20, 30 and 40 vessels with ten instances each. The vessels size is classified in each instance into three classes; 60% belong to class Feeder, 30% belong to class Medium, and 10% belongs to class Jumbo. Moreover, the minimum and maximum QCs needed for each vessel and the costs applied depending on the vessel class, which makes the benchmark closer to the real CT data.

The dataset for each vessel includes: vessel ID, length of vessel, desired berthing position, crane capacity demand of the vessel, expected time of arrival, earliest starting time, expected finishing time, latest finishing time, minimum and maximum number of cranes to assign and vessel service costs. A sample of the instances has been shown in Table 5-3.

In this study, the following assumptions were considered:

- Quay length $L = 100$ segment (1000 metres)
- Number of QCs on the quay $Q = 10QCs$
- Time Horizon $H = 168$ hour (one week)
- Interference exponent $\alpha = 0.9$

- Berth deviation factor $\beta = 0.01$
- Operation cost per QC-hour $c^4 = 0.1$
- Weight of overlap $w = 0.01$

From the above assumptions, the input data and using Equation (5-2), we can initially calculate the best values for the minimum handling time (duration time) for the vessel d_i^{min} .

In this experiment, the GP parameters were chosen as shown in Table 5-4, which is the common values used in the literature. We applied the ramped-half-and-half method (Koza, 1992). The following are the SPRs that we examined and compared with the *BACAP_GP_DP*.

- *R1: FCFS*: first-come-first-serve rule; the vessel list will be ordered by *ETA*.
- *R2: Max. QC needed low priority*: the vessel list will be ordered by r_i^{max} Ascending.
- *R3: Max. QC needed high priority*: in this dispatching rule, the vessel list will be ordered by r_i^{max} descending.
- *R4: Vessel length high priority*: gives the larger vessels high priority in the sorting order.
- *R5: Vessel length low priority*: gives the smaller vessels high priority in the sorting order.
- *R6: Min. Handling time low priority*: in this dispatching rule, the vessel list will be ordered by d_i^{min} ascending.
- *R7: Min. Handling time high priority*: same as above, but gives priority to vessels which need a long handling time.

Table 5-4: *BACAP_GP_DP* Parameters

Parameter	Value
Number of iteration	50
Population size	50
Number of generation	50
Maximum depth for population tree	7
Crossover probability	0.8
Mutation probability	0.1
Recreation probability	0.1

5.6.2 Results of SPRs and the *BACAP_GP_DP*

In this section, we present the results and the in-depth analysis of the *BACAP_GP_DP* framework. We have also provided a comparison between the

BACAP_GP_DP and the seven SPRs as discussed above. Extensive numerical experiments and analysis were performed and the outcome was compared with the results produced by Meisel & Bierwirth (2009).

A) SPRs Results:

In this approach, we solved the BACAP using the *BACAP_Scheduler_DP* with each of the SPRs proposed for ordering and giving priority to the vessels in the queue. The approach was examined without applying the GP algorithm.

Table 5-5, shows the results of the objective function (*Z*) using the proposed SPRs for all instances of the benchmark. It can be noticed that no dominating rule outperforms all of the other rules in all instances. Instead, the performance of the rules varies from one instance to another. For example, the *Best SPR* for instance number one was obtained by applying rule R4. Instance number 5 was obtained from applying rule R3 or rule R7. Instance number 7 was obtained by applying rule R3 or rule R4 or rule R7.

The last column in Table 5-5 (FCFS) shows the results provided by Meisel & Bierwirth (2009). If we compare these results with *RI*, which also uses the same rule of first-come-first-serve, then we find that the change made to our proposed algorithm, the *BACAP_Scheduler_DP*, improved performance in almost all of the results. This approves of the powerful technique used in our algorithm to solve the BACAP based on construction manner.

To study and understand the impact of using the proposed SPRs to solve the BACAP, we tried to count how many times each rule outperformed all of the other rules. Figure 5-9 shows the proportion of the *Best SPR* according to each dispatch rule. We discovered that 34% of the *Best SPR* results were obtained from rule R3, 29% were obtained from rule R4, 20% were obtained from rule R7, 17% were obtained from rule RI, and 0% were from the rest of the rules.

Table 5-5: Performance comparison between the dispatching rules (SPRs)

<i>ships</i>	<i>ins.</i>	<i>R1</i>	<i>R2</i>	<i>R3</i>	<i>R4</i>	<i>R5</i>	<i>R6</i>	<i>R7</i>	<i>Best SPR</i>	<i>FCFS</i>
20	1	118.5	132.7	124.6	90.2	132.7	94.1	106.8	90.2	118.5
	2	60.1	62.5	55.1	56.1	61.4	58.6	58	55.1	60.1
	3	94.7	119.9	84.3	134.7	108.2	161.8	93.1	84.3	97.6
	4	96.4	150.3	128.6	126.8	150.4	148.4	121.1	96.4	96.4
	5	65.2	77.2	57.7	57.9	69.4	72.4	57.7	57.7	73.1
	6	57.6	73.8	57.6	59.7	73.8	76.9	58.5	57.6	57.6
	7	88.3	99.1	77.1	77.1	104.8	101.1	77.1	77.1	93.3
	8	78.9	92.5	77.9	102.2	95.7	95.7	77.9	77.9	78.9
	9	86.5	104.7	90.1	84.9	105.6	124.4	93.2	84.9	96.4
	10	115.5	176.5	134.5	116.4	191.1	162.9	115.6	115.5	115.5
30	11	209	389.9	213.5	221.6	399.4	392.4	267.2	209	216.0
	12	96.5	145.6	92.3	100.9	145.3	146	104.9	92.3	96.7
	13	134.7	166.5	118.7	124.5	166.6	165.6	111.8	111.8	135.0
	14	128.9	284.3	137.6	136.4	285.1	159.8	141.2	128.9	144.5
	15	200.3	307.5	178.2	175.5	492.2	332.2	184.6	175.5	197.5
	16	135.9	155.6	129.4	130.8	216.9	203	129.6	129.4	137.7
	17	139.8	182.6	120	111.4	178.4	176.4	113.3	111.4	139.8
	18	167.8	223.6	166.8	157.1	222.7	242.3	160.3	157.1	167.8
	19	260.3	516.4	232	210.9	492.5	501.5	238.6	210.9	268.7
	20	153.9	269.4	146.5	150.4	313.3	442	161	146.5	184.7
40	21	322.4	551.7	393.1	321	637.7	754.1	430.1	321	317.0
	22	272.5	559.2	257.2	262.2	393	747.4	278.6	257.2	276.9
	23	476.8	926.3	451.2	452.6	1266.4	1083.9	363.9	363.9	550.4
	24	439.8	829.1	439.9	438.1	868.5	790.1	438.9	438.1	453.3
	25	221.2	578.7	228.6	258.8	867.6	622.7	191.8	191.8	239.1
	26	420.8	1110.8	486	430.2	1182	1011.4	414.4	414.4	398.9
	27	325.8	714.6	369	281.7	718	816	293.3	281.7	354.6
	28	423.1	1596.8	534.5	694.8	1281.3	1048.8	523.4	423.1	424.2
	29	293.5	622.5	259.8	414	569.5	747.5	413.6	259.8	334.2
	30	387.8	715.6	331.1	542.8	766.1	610.2	466.7	331.1	425.8

The above results show that the strategy of giving priority to incoming vessels is a crucial decision to minimise the total cost of scheduling, and it can be different from one port to another or even from one case to another. However, using the rule *R3*, which gives priority to vessels according to their r_i^{max} values in descending order, will obtain better solutions. The next rule is to use *R4*, which is giving priority to large vessels first, because the large vessel's arrival probability is small, and this again verifies the advice concluded by Ji et al. (2015), as described in the previous chapter.

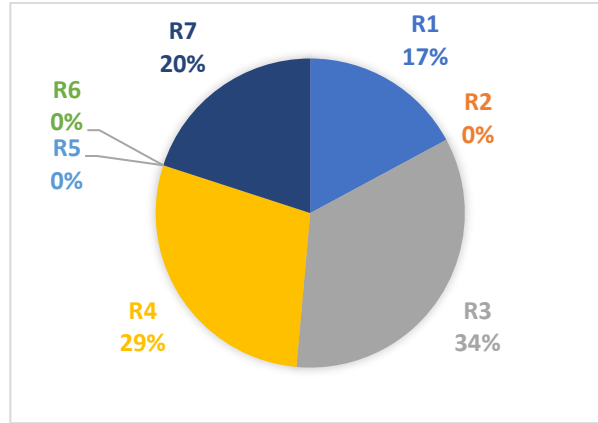


Figure 5-9: Proportion of the *best SPR* by dispatch rule performance.

B) *BACAP_GP_DP* Results

In this approach, we used composite dispatching rules (CDRs) evolved by the GP algorithm to solve the BACAP with the desired position.

We conducted numerous experiments using different parameters of the GP and the proposed algorithm *BACAP_GP_DP* to test the performance. We concluded the results into two main groups. The first one was using a population size equal to 50, and we denoted this as (GP1). The second one was using a population size equal to 100, denoted as (GP2). The rest of the parameters have been explained in Table 5-4.

In the first experiment, GP1 is compared to the SPRs explained above. Figure 5-10 illustrates the comparison between the results of GP1 and the *Best SPR* obtained from the previous approach, using a single dispatching rule. We noticed that the GP1 results outperformed the best results of SPR. We can conclude in this part that using CDRs is always better to solve the BACAP with rather than using a single dispatching rule SPR.

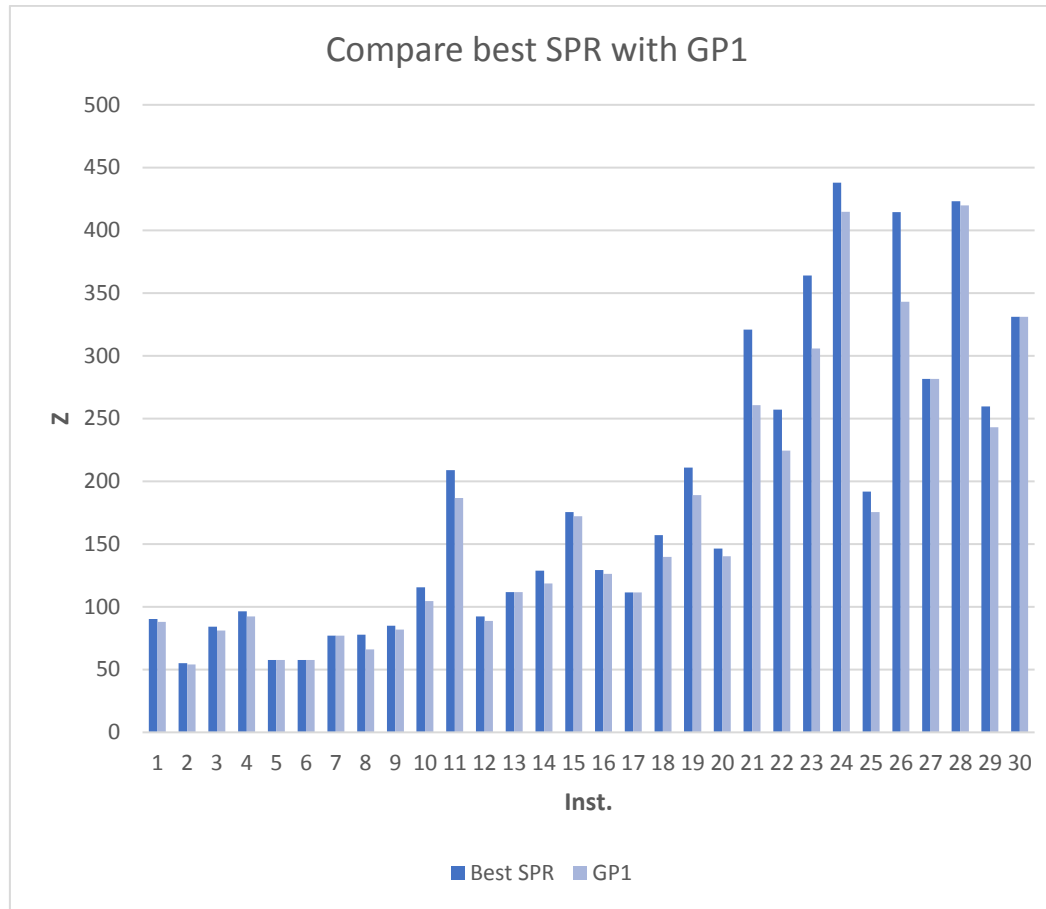


Figure 5-10: Compare GP1 performance with the Best SPR.

In the second experiment, GP2 was compared to GP1. Table 5-6 reports the obtained objective function value of Z , representing the total service cost of the berth plan. The *Gap* is calculated as $(Gap = \frac{Z-LB}{LB})$, and LB is the lower bound obtained by the CPLEX and reported in Meisel & Bierwirth (2009). We noticed that the GP2 results outperformed in almost all instances compared to the results of GP1, but the results took a longer time to obtain. A sample of the evolved CDR tree by the *BACAP_GP_DP* is shown in Figure 5-7 and Appendix B.2.

To the best of our knowledge, this is the only dataset available with existing results. Therefore, this dataset is used here as a benchmark to compare the GP results with other existing methods in the literature. Table 5-6 also shows the performance of the GP against the other improvement and construction methods reported in Meisel & Bierwirth (2009). FCFS_{LR} uses the DR with a local search, SWO uses the same

DR with Squeaky wheel optimisation, while TS is based on the Tabu Search optimisation method.

The GP outperformed the FCFS construction approach reported in Meisel & Bierwirth (2009). As expected, the methods that use both (construction and improvement) meta-heuristics/heuristics has outperformed the GP as a construction approach. However, the GP results are very competitive related to the improved methods. The aim in the future is to combine the GP evolved heuristics with an improvement optimisation layer.

Table 5-6: Performance comparison between the evolved GP scheduling and the other heuristics.

ships	ins.	Construction							Improvement		
		FCFS			GP1		GP2		FCFS _{LR}	SWO	TS
		Z	LB	Gap	Z	Gap	Z	Gap	Z	Z	Z
20	1	118.5	84.0	0.41	88.1	0.05	84.3	0.00	86.1	85.1	85.1
	2	60.1	53.9	0.12	54	0.00	54.29	0.01	53.9	53.9	53.9
	3	97.6	75.2	0.30	81.2	0.08	79.69	0.06	87.3	77.4	77.4
	4	96.4	75.8	0.27	92.4	0.22	90.4	0.19	79.7	79.7	77.9
	5	73.1	56.8	0.29	57.7	0.02	58.09	0.02	56.8	56.8	56.8
	6	57.6	57.6	0.00	57.6	0.00	57.6	0.00	57.6	57.6	57.6
	7	93.3	67.5	0.38	77.1	0.14	69.5	0.03	69.9	68.9	68.9
	8	78.9	56.1	0.41	66	0.18	61.9	0.10	69.6	57.0	56.1
	9	96.4	75.0	0.29	81.9	0.09	78.4	0.05	76.3	75.9	75.5
	10	115.5	88.2	0.31	104.5	0.18	98.59	0.12	101.1	94.6	93.0
30	11	216.0	137.7	0.57	186.7	0.36	178.3	0.29	152.6	147.8	149.5
	12	96.7	81.4	0.19	88.9	0.09	86.8	0.07	86.4	83.3	82.5
	13	135.0	100.9	0.34	111.8	0.11	112.3	0.11	107.6	105.7	104.5
	14	144.5	96.8	0.49	118.7	0.23	107.69	0.11	113.2	105.8	113.2
	15	197.5	136.9	0.44	172.1	0.26	145.3	0.06	173.8	159.0	157.4
	16	137.7	106.2	0.30	126.2	0.19	120.4	0.13	127.2	118.5	119.5
	17	139.8	99.6	0.40	111.4	0.12	110.5	0.11	110.2	104.5	104.2
	18	167.8	117.8	0.42	139.7	0.19	130.3	0.11	131.4	125.5	131.2
	19	268.7	156.4	0.72	189	0.21	185.7	0.19	185.0	173.8	173.8
	20	184.7	125.6	0.47	140.3	0.12	136.7	0.09	140.5	135.2	138.3
40	21	317.0	165.7	0.91	260.7	0.57	245.9	0.48	261.3	215.0	226.7
	22	276.9	159.6	0.73	224.5	0.41	222.4	0.39	189.0	178.8	183.4
	23	550.4	185.0	1.98	305.9	0.65	301.1	0.63	325.7	273.9	264.3
	24	453.3	224.1	1.02	414.8	0.85	410.5	0.83	360.2	326.6	342.2
	25	239.1	133.3	0.79	175.4	0.32	165.5	0.24	162.0	155.1	154.8
	26	398.9	201.3	0.98	343.1	0.70	304.3	0.51	273.1	260.4	259.6
	27	354.6	172.2	1.06	281.7	0.64	255.3	0.48	233.0	200.8	215.8
	28	424.2	211.7	1.00	419.8	0.98	390.5	0.84	408.5	286.2	294.3
	29	334.2	180.3	0.85	243.2	0.35	259.8	0.44	268.4	219.4	223.4
	30	425.8	170.1	1.50	331.1	0.95	326.2	0.92	280.8	240.9	254.7

5.7 Conclusion

This study presents a new optimisation method for the integrated berth allocation and quay crane assignment problem (BACAP) with predefined desired berthing positions for the incoming vessels. The objective function aims to minimise the total service costs of a berth plan. We investigated how real container terminal operations work by visiting the DP World Sokhna port, providing a sensitive study analysis to understand the practical BACAP compared with the literature.

First, we have improved the *BACAP_Scheduler* to *BACAP_Scheduler_DP*, which is capable of tackling the BACAP with the desired position. We improved the mathematical model to predict the overlap that might happen between vessels during scheduling. This technique gives the proposed algorithm a powerful tactic to use to minimise the total costs that will be added in the case of overlapping.

Second, we have introduced a genetic programming approach called *BACAP_GP_DP*, which is also an improved version of the *BACAP_GP*, which automatically evolves efficient and customised dispatching rules for the BACAP. The *BACAP_GP_DP* is capable of combining different sets of attributes (vessel-related attribute set, berth-related attribute set and quay crane-related attribute set). This allows the *BACAP_GP_DP* to generate tailored DRs based on the attributes given in each set, as the attributes may vary from port to another. The results show that the GP outperformed all SDRs in all instances in the (construction) method and that it is very competitive with the well-known literature (improvement) methods.

The performance of the proposed approach leads us to tackle the BACAP with multiple-ports which is multiple ports and terminals owned by one terminal operator. This will be explained in detail in the next chapter. For further research, the aim is to extend the *BACAP_Scheduler_DP* to test and evaluate the behaviour of the evolved heuristics when combined with other optimisation methods. Future work should combine the *BACAP_GP_DP* with an improved optimisation layer.

Chapter 6 : Berth and Quay Crane

Optimisation Model with Multiple Ports

6.1 Introduction

This chapter focuses on the multiple ports BACAP. Most of the previous studies tackled the seaside container terminal problems by solving the three main ones which are the berth allocation problem (BAP), the quay crane assignment problem (QCAP) and the quay crane scheduling problem (QCSP) in an independent or integrated manner, but these studies consider there to be one container terminal in one port. Only a few studies considered the multi-terminal scenario where several ports are managed by one operator, and the target is to schedule incoming vessels across the different available ports. This is known as multi-quay/multi-terminal or multi-port BAP. In this chapter, we extended the dynamic continuous BACAP to multi-port BACAP.

In our thesis, we give the following terms (multi-port, multiple ports, multiple terminals and multiple quays) to locations that have a set of terminals, where each terminal has a set of quays. These terminals are managed by one port operator regardless of the distance between them within the same country. In this case, the port operator needs to schedule all incoming vessels from the strategic level plan on these terminals and available quays in each terminal.

In this regard, this chapter aims to (1) develop a generic model to solve the BACAP with multiple ports, which includes multiple quays, (2) integrating the strategic planning level with the operational level to provide a guideline for container operator decision-making and (3) to develop a two-level heuristic algorithm to evolve the dispatching priority rules for scheduling vessels in the quays. The objective is to minimise the total cost of services for all vessels.

To the best of our knowledge, this is the first study that tackles the BACAP with multiple ports and multiple quays, in addition to there being integrated levels of planning. We, therefore, believe that our contribution to this problem is novel and generic. Finally, we generated a new dataset based on a well-known pre-existing one to handle a new problem type.

This chapter is structured as follows. Section 6.2 describes how multiple ports operate. Section 6.3 illustrates the problem. The proposed mathematical model with assumptions and used notations has been given in Section 6.4. Section 6.4.4 presents an illustration of the multiple ports BACAP plan framework. In Section 6.5, a solution approach, namely MultiP_BACAP_GA, has been presented. Section 6.6 describes the computational experiments of CPLEX and GA. Finally, we have provided our conclusions in Section 6.7.

6.2 Multiple Ports Operations

Nowadays, mega-ports in the world are usually multi-terminal port systems rather than single-terminal systems (Zhen et al., 2016). For instance, there are many container terminals (CTs) that have multiple quays in one place (port) or different places (ports). For example, Valencia (Spain) has 2 quays; Jebel Ali port (Dubai) has three terminals with five quays (Frojan, Correcher, Alvarez-Valdes, Koulouris, & Tamarit, 2015). The ports of Rotterdam, Los Angeles and Shanghai also has nine, nine, and seven container terminals, respectively (Zhen et al., 2016). In Singapore, PSA Singapore has four container terminals in Tanjong Pagar (3 quays), Keppel (4 quays), Brani (3 quays) and Pasir Panjang (9 quays). It operates them as one seamless and integrated facility; the port of Hong Kong has nine container terminals situated in Kwai Chung-Tsing Yi basin, and five different operators run it. The productivity of these ports depends mainly on the efficient berth allocation of the calling vessels. From the point of view of the ship operators, the timeline is a crucial factor as a delay at one port often results in a domino effect on ship routing and scheduling for the following ports.

To solve the BACAP for the above type of CTs, considering them as a single quay is unprofessional and this might lead to an increase in the total service cost. The novelty of our proposed model is that it integrates a scheduling plan for all of the ports/quays at once. The computational results approved that, if we solve the BACAP problem separately (each quay in each port), then this will decrease the utilisation of the QCs and increase the total service cost. For instance, if we have two vessels with the same ETA, and the quay was crowded to accept both of them, then to solve the BACAP for this scenario we will move one of the vessels to another starting time which may affect its EFT and add a cost to the terminal. However, if we have multiple quays and we can integrate them together, then we can allocate each vessel to a different quay so then both vessels can moor at the same ETA but in a different quay.

Moreover, in the case of an emergency or uncertainty that prevents one vessel from berthing in one quay, and we need to change its berth to another one. It will be more useful to use such an algorithm to tackle the problem as a multiple quays situation rather than as a single quay. We understand that to change a vessel berth from one port to another or from one quay to another will add costs. However, this cost might be less than if opting to change the vessel's starting and finishing time, as time is crucial for the vessel owner's plans. The proposed model will discover the optimal scheduling solution plan that suits the incoming vessels and the availability of the ports/quays.

For decision-making planning problems, there are three different levels of planning: strategic, tactical and operational planning. We can consider the BACAP at these distinct levels. For the strategic level, it is the most extended time plan, which extends from one year to several years. It is used to determine the most suitable schedule for terminal and vessel operators. This plan could be contracted between them for a prolonged period with no change until they come up for renewal or negotiate for a new contract. As for the tactical level, the plan will be for the short-term (one week up to several months). At this level, we will consider the timetable of arrivals and the departure of vessels at a studied terminal. Within this level, the decisions and schedules for seeking optimality have been made. Finally, regarding

the operational level, the time horizon goes from one day to one week. In this level, it aims to optimise the service time for the vessels by minimising the waiting and handling time, and minimising terminal costs. Moreover, the actual arrival time, tide, breakdown of equipment, uncertainty and real-time operational constraints should be tackled at this level (Hendriks, Armbruster, Laumanns, Lefebvre, & Udding, 2012).

The typical contract between a port operator and a shipping line process starts by the shipping line requesting a berth with a condition that suits the shipping schedule and the characteristics of the ship with the average number of containers that need to be moved stated (loading/unloading). The port operator evaluates the berthing requirement and its current operations. The negotiation usually takes several rounds before reaching an agreement. For a port operator that manages more than one port and with every port having multiple terminals, the decision might be quite complicated since they need to schedule all incoming vessels between their respective ports and terminals. Moreover, they need to consider the average costs of services in each container terminal in addition to the cost of changing the desired berthing position from one terminal to another, or from one port to another.

6.3 Problem Description

In the Multi-Port BACAP Figure 6-1, we have a set of incoming vessels that need to be served, and we have a set of ports. Each port has multiple terminals. The first part of the problem is to allocate the time, port, quay and a berth position in the terminal for all vessels to solve the BAP. Once a berth is occupied by a vessel, no other vessel can occupy the same berth at the same time. The second part of the joint problem is to assign a set of QCs in the quay to serve the vessel during staying in the berth to solve the QCAP, taking into consideration that the number of QCs should be within the minimum and maximum range of QCs required or contracted between the terminal operator and shipping line operator in advance. The decision made on the number of QCs may depend on the length of the vessel, the number of containers to be moved and how the containers are distributed on the vessel. In this

regard, the vessel's operators should send the vessel information to the CT operators in advance. This information includes the vessel length, expected time of arrival, the number of containers to be moved and its desired position in a specific quay within a particular port. The primary objective is to minimise the total assignment costs.

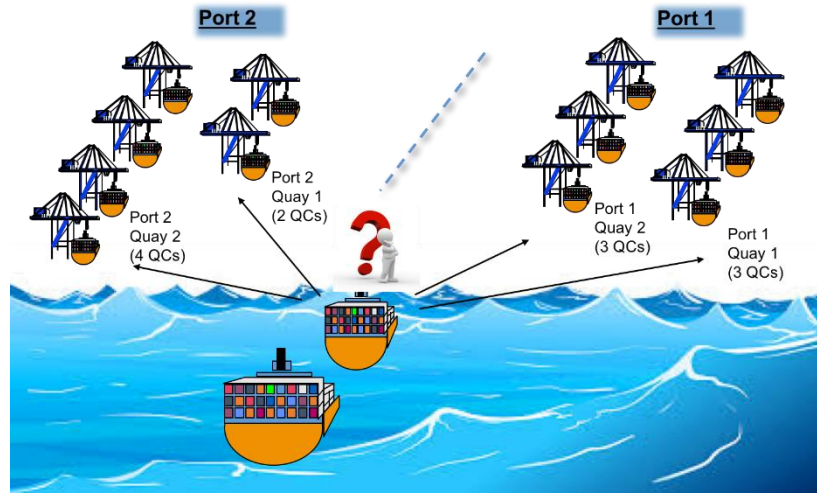


Figure 6-1: The Multi-Port BACAP problem description

Figure 6-2 illustrates the relationships between the multi-port BACAP components, which includes: BAP, QCAP, ship ordering priorities, Multi-Port, and the heuristic solver method that we chose to solve the problem. The intersection between these components defines the area of the proposed problem. We have integrated the problems of BAP and QCAP that we need to solve in the case of multi-port; the solver will use the genetic method with the priority of dispatching rules to obtain the solution.

We noticed that this type of problem is entirely different from that of a single quay, where each quay has a number of quay cranes and the specifications of quay length, depth and the distance of the desired vessel berthing position. Accordingly, all of these factors will affect the solution and objective function in return.

On the other hand, since the vessel's characteristics - such as the vessel's length, time of arrival and the working hours needed to load/unload the containers - can

be expected beforehand, it becomes possible to determine the usage of berths that can enable the vessels to be served as soon as they arrive.

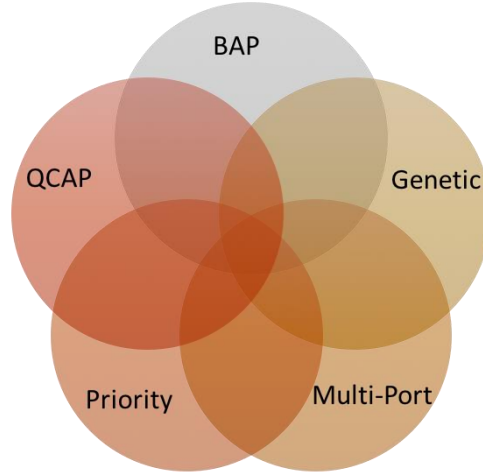


Figure 6-2: The Multi-Port BACAP Relationships

6.4 Mathematical Model Formulation

In this section, we have presented a novel model for the multi-port BACAP.

6.4.1 Model Assumptions

The proposed model was established based on the following assumptions:

- The vessel i can be moored on the quay if there is a space greater than or equal to the vessel length, and if there is at least a minimum number of QCs available to start with.
- The number of QCs assigned to vessel i is in the range between the minimum and maximum number of QCs contracted to serve the vessel. The optimal value in this range will be found by the model.
- The QCs can be assigned to vessels using a dynamic approach (time-variant) so then the QCs can be attributed to another vessel before the original one departs. For more details on the differences between static and dynamic assigning for QCs, we direct the reader to Rodriguez-Molins, Salido, & Barber (2014).

- All vessels have a draft that is lower than the depth of the quay that they are assigned to.
- The quay layout is continuous. The ETA is Dynamic arrival.
- The handling time of a vessel depends on the number of assigned QCs (QCAP), moves required and the vessel position which are calculated depending on the equation (6.2) as shown in the next section. The performance measure is the total cost of position, waiting time, tardiness and overlap, and the objective function is to minimise these costs.
- The vessel length includes the required safety margins, which is the safe distance between two moored vessels.

The following notations have been used to solve the Multi-Port BACAP.

6.4.2 Notations

Input data and calculated variables:

P	Set of ports. Each port $p \in P$ has a set of quays.
X	Set of quays in all ports.
V	Set of vessels to be served, $V = \{1, 2, \dots, n\}$
Q_x	Number of available QCs at quay x
L_x	Number of 10-m berth segments (length of the quay x)
T	Set of 1-hour periods, $T = \{0, 1, \dots, H - 1\}$, H is the planning horizon
l_i	Length of vessel $i \in V$ given as a number of 10-m segments
b_{ix}^0	Desired berthing position of vessel i at quay x
m_i	Crane capacity demand of vessel i given as a number of QC-hours.
r_i^{\min}	Minimum number of QCs agreed to serve vessel i simultaneously
r_i^{\max}	Maximum number of QCs allowed to serve vessel i simultaneously
R_i	Feasible range of QCs assignable to vessel i , $R_i = [r_i^{\min}, r_i^{\max}]$
ETA_i	Expected time of arrival of vessel i
EST_i	Earliest starting time if journey of vessel i is sped up, $EST_i < ETA_i$
EFT_i	Expected finishing time of vessel i
LFT_i	Latest finishing time of vessel i without penalty cost arising
c_i^1, c_i^2, c_i^3	Service cost rates for vessel i given in units of 1000 USD per hour
c^4	Operation cost rate given in units of 1000 USD per QC-hour
c_i^5	Overlap cost rate which is the possibility of overlap between vessel i and the rest of vessels given in units of 1000 USD per space 10-m segment.
c_{ix}^{ETA}	Assignment cost of vessel i to quay x given in 1000 USD.
α	Interference exponent of QC productivity
β	Berth deviation factor
M	A large positive number
d_i^{\min}	Minimum handling time needed to serve vessel i
w	Weight of overlap

Decision variables:

b_i	Integer, berthing position of vessel i at the quay to which it is assigned
s_i	Integer, time of starting the handling of vessel i (berthing time)
e_i	Integer, time of finishing the handling of vessel i (finishing time)
r_{it}	Binary, set to 1 if at least one QC is assigned to vessel i at time t , 0 otherwise
r_{itq}	Binary, set to 1 if exactly qQCs are assigned to vessel i at time t , $q \in R_i$, 0 otherwise
Δb_i	Integer, the deviation between the desired and the chosen berthing position of vessel i at the quay to which it is assigned. $\Delta b_i = b_{ix}^0 - b_i $
ΔETA_i	Integer, required speed up of vessel i to reach its berthing time, $\Delta ETA_i = (ETA_i - s_i)^+$
ΔEFT_i	Integer, tardiness of vessel i , $\Delta EFT_i = (e_i - EFT_i)^+$
u_i	Binary, set to 1 if the finishing time of vessel i exceeds LFT_i , 0 otherwise
y_{ij}	Binary, set to 1 if vessel i is berthed below of vessel j i.e. $b_i + l_i \leq b_j$, 0 otherwise
z_{ij}	Binary, set to 1 if handling of vessel i ends not later than handling of vessel j starts, 0 otherwise
md_{ix}	Binary, set to 1 if vessel i moored at quay x , 0 otherwise
ox_i	Number of time slot (hours) overlap vessel i with the rest of vessels if any.
oy_i	Number of spaces (10m-segment) overlap vessel i with the rest of vessels if any.

6.4.3 Mathematical Model

$$\begin{aligned} \text{minimize } Z = & \sum_{i \in V} (c_i^1 \cdot \Delta ETA_i + c_i^2 \cdot \Delta EFT_i + c_i^3 \cdot u_i + c^4 \cdot \sum_{t \in T} \sum_{q \in R_i} q \cdot r_{itq} + c_i^5) \\ & + \sum_{x \in X} \sum_{i \in V} (c_{ix}^{ETA} \cdot md_{ix}) \end{aligned} \quad (6.1)$$

$$d_i^{min} = \lceil \frac{(1 + \beta \cdot \Delta b_i) \cdot m_i}{(r_i^{max})^\alpha} \rceil \quad (6.2)$$

$$c_i^5 = \begin{cases} w \cdot \sum_{i \in V} oy_i, & \text{if } (ox_i \text{ and } oy_i) > 0 \\ 0, & \text{Otherwise} \end{cases} \quad \forall i \in V \quad (6.3)$$

$$\sum_{t \in T} \sum_{q \in R_i} q^\alpha \cdot r_{itq} \geq (1 + \beta \cdot \Delta b_i) \cdot m_i \quad \forall i \in V \quad (6.4)$$

$$\sum_{i \in V} \sum_{q \in R_i} (q \cdot r_{itq}) \cdot md_{ix} \leq Q_x \quad \forall t \in T, \forall x \in X \quad (6.5)$$

$$\sum_{q \in R_i} r_{itq} = r_{it} \quad \forall i \in V, \forall t \in T \quad (6.6)$$

$$\sum_{t \in T} r_{it} = e_i - s_i \quad \forall i \in V \quad (6.7)$$

$$(t + 1) \cdot r_{it} \leq e_i \quad \forall i \in V, \forall t \in T \quad (6.8)$$

$$t.r_{it} + H.(1 - r_{it}) \geq s_i \quad \forall i \in V, \forall t \in T \quad (6.9)$$

$$\Delta b_i \geq b_i - \sum_{x \in X} (b_{ix}^0 \cdot md_{ix}) \quad \forall i \in V \quad (6.10)$$

$$\Delta b_i \geq \sum_{x \in X} (b_{ix}^0 \cdot md_{ix}) - b_i \quad \forall i \in V \quad (6.11)$$

$$\Delta ETA_i \geq ETA_i - s_i \quad \forall i \in V \quad (6.12)$$

$$\Delta EFT_i \geq e_i - EFT_i \quad \forall i \in V \quad (6.13)$$

$$M.u_i \geq e_i - LFT_i \quad \forall i \in V \quad (6.14)$$

$$b_j + M.(1 - y_{ij}) \geq b_i + l_i \quad \forall i, j \in V, i \neq j \quad (6.15)$$

$$s_j + M.(1 - z_{ij}) \geq e_i \quad \forall i, j \in V, i \neq j \quad (6.16)$$

$$y_{ij} + y_{ji} + z_{ij} + z_{ji} \geq md_{ix} + md_{jx} - 1 \quad \forall i, j \in V, i \neq j, \forall x \in X \quad (6.17)$$

$$b_i + l_i \leq \sum_{x \in X} md_{ix} \cdot L_x \quad \forall i \in V \quad (6.18)$$

$$\sum_{x \in X} md_{ix} = 1 \quad \forall i \in V \quad (6.19)$$

$$s_i, e_i \in \{EST_i, \dots, H\} \quad \forall i \in V \quad (6.20)$$

$$b_i \in \{0, 1, \dots, L_x - l_i\} \quad \forall i \in V, x \in X \quad (6.21)$$

$$\Delta ETA_i, \Delta EFT_i \geq 0 \quad \forall i \in V \quad (6.22)$$

$$md_{ix} \in \{0, 1\} \quad i \in V, x \in X \quad (6.23)$$

$$r_{itq}, r_{it}, u_i, y_{ij}, z_{ij} \in \{0, 1\} \quad \forall i, j \in V, i \neq j, \forall t \in T, \forall q \in R_i \quad (6.24)$$

When solving the BACAP problem, the objective is to minimise the total service time, minimise the total service cost, or minimise the waiting time while increasing the utilisation of the terminals' QCs. The most widely used objective function in the literature was minimising the total service cost.

In our proposed mathematical model, the objective function is to minimise the total service cost as shown in equation (6.1). The vessel service costs include, the costs for assigning all incoming vessels to quays, their berthing position, the precise berthing time and the number of cranes needed to serve the vessel within the handling period.

From the mathematical model above, constraint (6.2), calculates the minimum handling time required to serve vessel i . Equation (6.3) predicts the possibility of overlap between vessels and calculates its cost. The cost of this overlap is the total length of the overlap vessel i with the rest of the unscheduled further vessels multiplied by the weight of overlap, as described in the previous chapter. Constraints (6.4)-(6.6) ensure that every vessel receives the required QC capacity with respect to productivity losses caused by QC interference and the chosen berthing position. Constraints (6.7)-(6.9) set the starting and ending times for serving vessels, considering the QCs assignment to a vessel. Constraints (6.10)-(6.13) find the deviations from the desired berthing position at quay x to which it is assigned and determines the expected arrival and finish time for each vessel. Constraint (6.14) defines if the vessel finishes later than the *EFT* (delay). Constraints (6.15)-(6.17) prevent overlapping in time and space for vessels assigned to the same quay. Constraint (6.18) ensures that the position at the quay is valid and constraint (6.19) ensures that every vessel is assigned to a quay. Constraints (6.20) and (6.21) ensure the boundaries of each vessel to the berth is inside quay space L_x and planning horizon H . The additional constraints define the domains for the remaining decision variables.

6.4.4 Multiple Ports BACAP Plan Framework

In this research, we integrated the main three levels of planning in order to solve the Multiple Ports BACAP. Figure 6-3 shows the proposed plan framework. On the strategic level, the port operator considers two types of information. The first one is related to the ports' information and its characteristics, and the second one is related to the vessel's information which provided by ship lines in advance according to the contract. At this level, many decisions should be undertaken such as the number of ports and the number of available quays in each port. This is in addition to the number of available QCs for each quay, the desired berthing quay for each vessel and how much it cost to change this quay to another one.

In this research, the strategic plane is responsible for dividing the incoming vessels to the available quays in the ports. The outcome will find the best quay for each vessel.

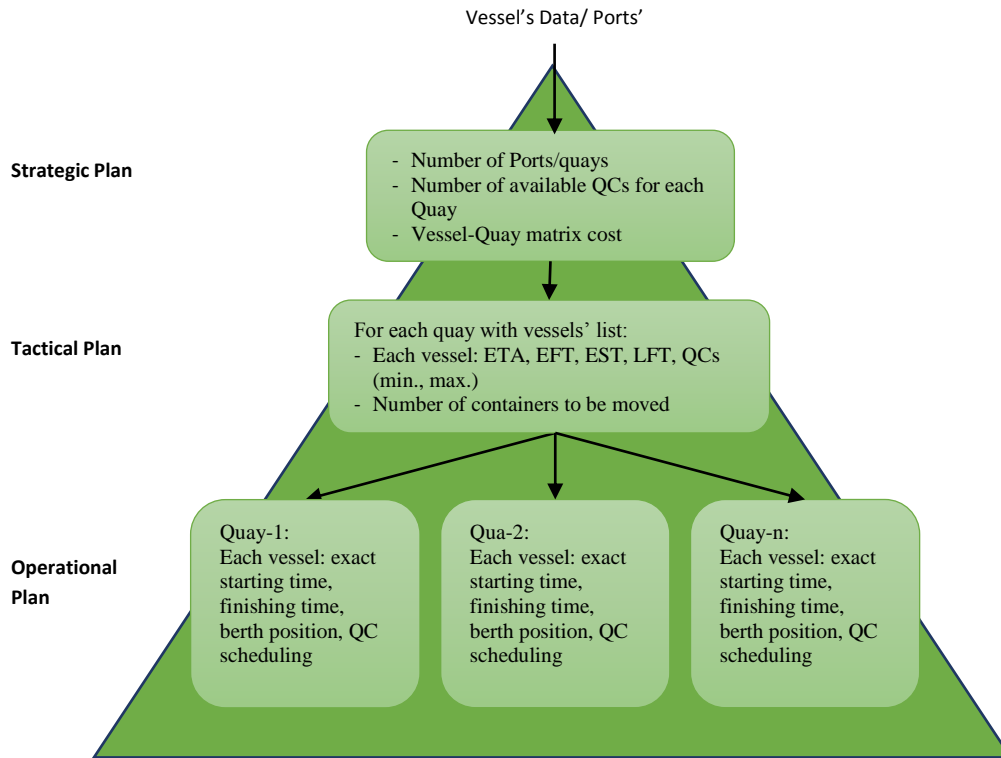


Figure 6-3: Multi-Port BACAP Plan Framework

There are four types of incoming vessel. The first one has its desired position in a specific port in a particular quay and cannot be in another port, but it can be shifted or await its desired position. The second can berth in any quay within the same port. The third case is flexible, and able to berth in any port or quay. Finally, is the case in which a vessel can berth in a specific quay in each port. In this research, we converted the set of ports and its quays into a number of quays, and managed the cost for berthing from one port to another. Therefore, each vessel encounters a cost when it comes to berthing in a different quay depending on the pre-defined cost to berth in a different port. Furthermore, we must bear in mind that we have encountered some constraints that prevent a vessel from berthing in a specific

port/quay. For instance, the vessel's draft with relevance to the quays depth or a contract restriction between the vessel operators and port operators.

From the previous illustration of the types of incoming vessels, we have proposed a Vessel-Quay Matrix costs list as shown in Table 6-1. This table shows the costs for each vessel v_i in each quay x_i . The power of this matrix is to control the type of problem in order to solve it or to direct a vessel to berth in a specific quay. This means that we can solve the problem as if it was in one port with one quay, one port with multiple quays or multiple ports with multiple quays. To do so, we can assign a high cost (M) to the quays where the vessel cannot berth. For instance, v_1 can berth in quays x_1 and x_2 with costs 3 and 4 respectively, but cannot berth in the quays x_3 and x_4 , while v_5 and v_6 can berth only in quay x_2 and x_3 with costs 3 and 5 respectively.

Table 6-1: Vessel-Quay Matrix cost list

	v_1	v_2	v_3	v_4	v_5	v_6
x_1	3	M	M	5	M	M
x_2	4	5	M	M	3	M
x_3	M	M	3	6	M	5
x_4	M	6	4	4	M	M

The second level is the tactical plan, which takes place after the first level. In this level, the port operator builds the schedule for each quay in each port for a period of one or two weeks. They can construct a schedule up to one month in advance. The input data will be the vessel list for each quay, which should be as defined earlier by the strategic plan. In addition, the port operator can use the information provided by the ship-lines for the vessels such as (ETA , EST , EFT , LFT , r_i^{min} and r_i^{max}).

The third level is the operational level. In this level, the schedule that was built from and in the tactical level is tested and evaluated for a short period (a few days). The exact starting time and finishing time for each vessel and the exact berthing position in each quay should be defined at this point. Moreover, the QCs scheduling to serve

each vessel can also be identified. The port operator should strive to handle the uncertainties that can happen in the terminals, such as QC breakdown in order to keep the operational and tactical plans aligned with any minor changes.

6.5 Genetic Algorithm for Multiple Ports BACAP (MultiP_BACAP_GA)

In this section, we proposed a solution to the Multiple ports BACAP that integrates the three main decision plans as described before. The primary goal of the GA, which is responsible for the strategic decisions, is to classify all incoming vessels (list of vessels) into sub-lists of vessels that will be sent to each quay in each port. For the tactical and operational decisions, we applied the *BACAP_Scheduler_DP* (described in the previous chapter) in order to schedule the vessels in each quay and to determine the minimum scheduling cost.

To illustrate the operations involved in classifying the vessels to ports/quays considering the constraints of the vessels and the port/quay, we have different methods:

First, we can calculate the cost of each vessel in each port/quay and try to find the best solution with our objective function, which is the minimum cost and assigns the vessel to the cheapest quay, as in Figure 6-4.

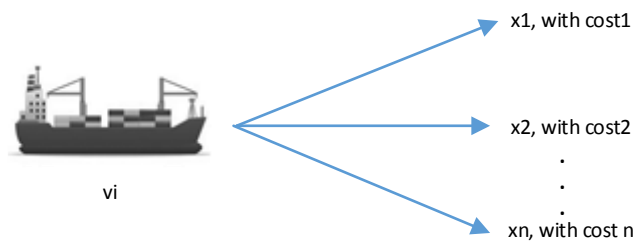


Figure 6-4: Distributing vessels to Ports/Quays (Case1)

Second, we can start to assign, vessel by vessel, to the first quay according to its ETA and the quay availability. We can then assign the minimum number of quay

cranes that the vessel can work with. If the solution is not feasible, then we will assign the vessel to the next nearest quay in the same port. If it is still not feasible, then we can assign it to another port as in Figure 6-5.

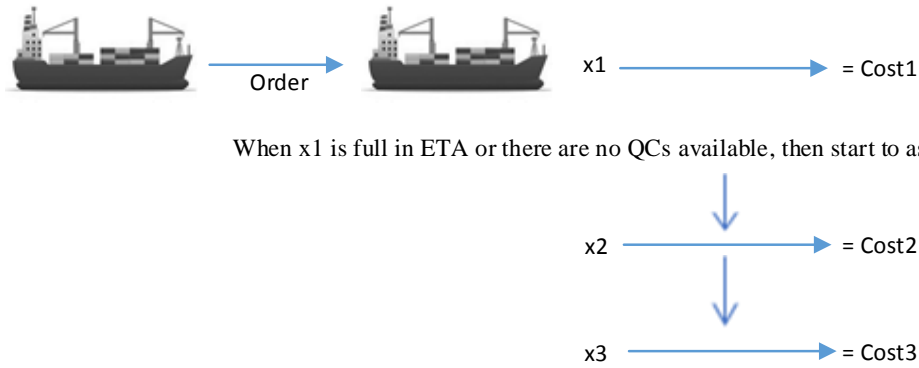


Figure 6-5: Distributing the vessels to the Ports/Quays (Case2)

Third, we can assign the vessels to their desired ports/quays and then calculate the fitness. We can then swap the vessels to a different location (ports/quays) and calculate the eligibility again for this combination and so on. We repeat this until we get the best total fitness, as shown in Figure 6-6. In this research, we used the third method of distributing the incoming vessels to the available ports.

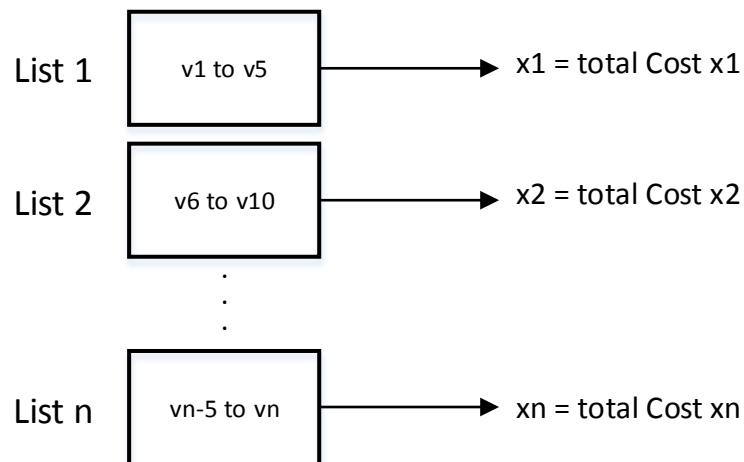


Figure 6-6: Distributing the vessels to the Ports/Quays (Case3)

6.5.1 MultiP_BACAP_GA Framework

The flowchart of the proposed MultiP_BACAP_GA has been illustrated in Figure 6-7. It starts by setting up the GA parameters needed, such as the number of population and generation, crossover, mutation, reproduction rate, seeds, etc. We set up the number of ports/quays and their characteristics, such as the length of the quay, depth, the number of quay cranes available in the quay etc. Next, we read the vessels' data, which included the vessel's costs to berth for each port/quay. From this, we created the vessel-quay matrix cost list as described before. The GA uses this matrix to generate the chromosomes and individuals randomly in order to initialise a population.

From the initial population, the GA starts to process the first generation for evaluation. The evaluation procedure occurs by calculating the total cost which includes two parts. The first one is the cost saved in the vessel-quay matrix cost list (cost1). For instance, from Table 6-1, the cost for berthing vessel 1 in the second quay is $cost1(v_{12}) = 4$.

The second part was obtained by creating a list of vessels for each port/quay and sending the list, one by one, to the *BACAP_Scheduler_DP* for evaluation to get the best schedule cost (cost2) for each quay. The GA tries to use different rules to order the vessel's list for each quay and sends it again to the *BACAP_Scheduler_DP* to find a better solution, which is the minimum of cost2. If all rules and all ports/quays are tested, then the fitness of the current solution can be calculated. Now, we can finish evaluating the first individual. If all individuals are checked, then the first generation is completed. GA can determine the chromosomes that obtained the best results for this generation and continue to generate the second generation using the crossover and mutation, sending them again for evaluation as before. The process will continue as above until it reaches the last generation. Finally, the GA can find the best quay for each vessel and the best berth schedule for each quay.

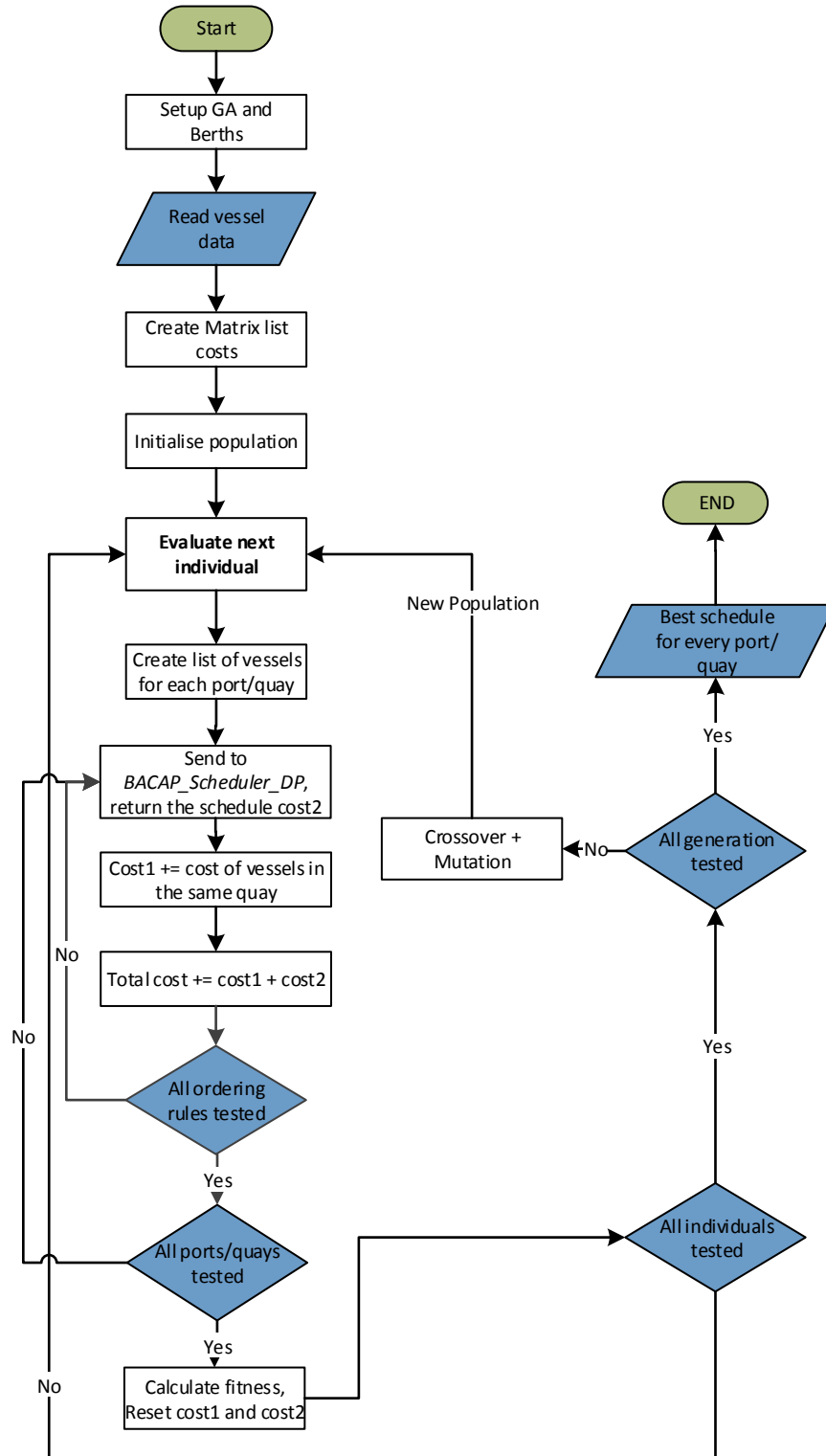


Figure 6-7: The MultiP_BACAP_GA flowchart

6.5.2 Chromosome Representation

We used an integer-coded chromosome representation to solve the strategic assigning of quays to vessels. We fixed the length of a chromosome to the number of vessels. The value of every gene indicates the id of the quay that a vessel should be berthed in. These values must be satisfied by the vessel-quay matrix cost list. Figure 6-8 shows an example of chromosome representation in which the length is six, and this equals a number of vessels. Vessel 1 can berth in quay 1 and quay 2, vessel 2 can berth in quay 2 and quay 4 and so on. The generated chromosome indicates that vessel 1 was chosen to berth in quay 1 and vessel 2 was chosen to berth in quay 4 and so on.

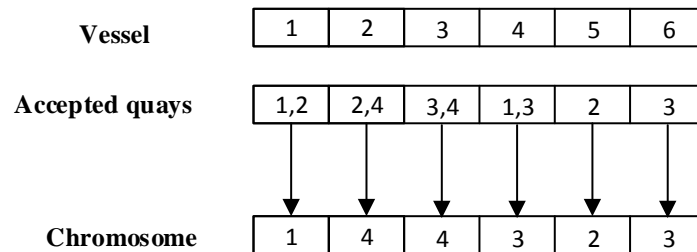


Figure 6-8: GA chromosome encoding

6.5.3 Generate a set of initial solutions

The GA generated the initial population randomly by creating a number of individuals (chromosomes) as illustrated above. Each gene inside the chromosome represents one possibility that the vessel can berth in one quay, before sending the individuals one by one to the *BACAP_Scheduler_DP* for evaluation. Finally, it calculates the total cost for each chromosome.

6.5.4 Fitness function

The objective function is to minimise the total cost of distributing the incoming vessels toward the ports/quays, taking into consideration minimising the cost of changing the vessels' desired position (cost1) and minimising the cost of scheduling the vessels in the quays (cost2). This was along with the availability of quay cranes that will serve the vessels. We calculated the fitness function using equation (6.1), simplified as follows:

$$\min. F = \sum_{x \in X} \sum_{i \in V} cost1(v_{ix}) + cost2(v_{ix}) \quad \forall i \in V \quad (6.25)$$

6.5.5 Crossover Operation

The crossover operator was applied to create the next generation by selecting two individuals from the previous generation to be the parents, thus creating the children. We used different methods for the crossover operator selection, such as a single-point crossover and two-point crossover, which is commonly used in permutation-based encodings to produce the offspring individuals (Rodriguez-Molins, Ingolotti, et al., 2014). Figure 6-9.a illustrates the single point crossover operator which randomly choose a point in each parents' chromosome. The genes before the point are inherited from parent 1 and the genes after the point are copied from parent 2.

6.5.6 Mutation Operation

In the mutation operation, high priority individuals were selected to reproduce the offspring ones. This happens by changing one gene randomly inside the chromosome to another acceptable quay which matches the vessel-quay list as described before. Figure 6-9.b illustrates the mutation operator.

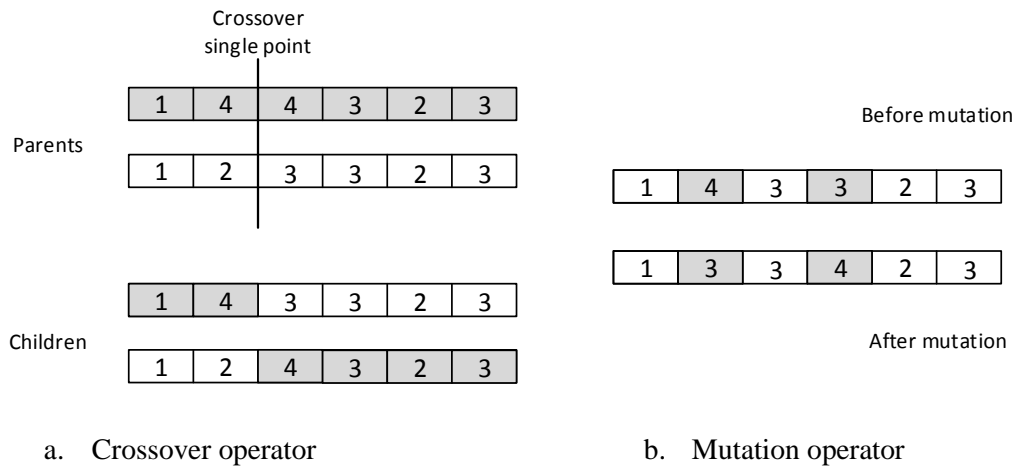


Figure 6-9: Crossover / Mutation operators.

6.6 Computational Experiments and Analysis

In this section, a comprehensive study of the computational experiments was conducted to test the performance of the proposed model. The model was developed to examine the minimum cost for scheduling a list of incoming container vessels to several ports managed by one port operator, and each port had many container terminals (quays). The problem considers the integrated BAP & QCAP. Therefore, the results should satisfy the decisions needed for the three planning determinations as described in section 6.4.4. To verify our model, we solved the initial problem using IBM ILOG CPLEX 12.6.1 (IBM ILOG, 2014) as an exact solution method, Appendix C.2. The GA heuristic algorithm and the *BACAP_Scheduler_DP* were developed in Java. All experiments were run parallel and independent on a server that had multiple cores; each core had an Intel Xeon processor 2.66GHz and 4.0 GB of RAM. The stopping condition was either the computational time limit of 10 hours being reached or running out of memory.

6.6.1 Datasets and parameter settings

6.6.1.1 Datasets and instances

There are three datasets used in this chapter. This is the extended data of the proposed set from Meisel & Bierwirth (2009) to accommodate the new problem with multiple ports/quays that we have presented. The datasets have three types of classes (Feeder, Medium, and Jumbo). These classifications are related to the vessels' length and the number of QCs required for loading and unloading the containers. We added the desired position for each vessel in each port/quay and the cost that might be incurred if the vessel is scheduled to berth in different quays.

The first and second benchmark consists of 30 instances; 10 instances of 20, 30 and 40 vessels to be berthed on two ports one quay each, and two ports with three quays respectively. The third benchmark consists of 10 instances of 100 vessels to be berthed in 5 quays. For each instance, 60% of the vessels belong to Feeder, 30% of the vessels belong to Medium, and 10% of the vessels belong to Jumbo.

6.6.1.2 Parameters

The common parameter values used in this chapter are as follows. The first five parameters were chosen as the previous experiment's suggestion, described in the previous chapter to solve the BACAP for a single quay. The GA parameters were chosen after some of the initial experiments were conducted and obtained excellent results.

- Number of available QCs at each quay x (Q_x): 10 quay cranes.
- Operation cost rate per QC-hour (c^4): 0.1
- Interference exponent of QC productivity (α): 0.9
- Berth deviation factor (β): 0.01
- Weight of overlap $w = 0.01$
- For the GA parameters: crossover rate (0.6), mutation rate (0.3) and reproduction rate (0.1)

The rest of the input parameters are different from one experiment to another.

6.6.2 Results of Multi-Port BACAP, CPLEX and GA comparison

In this section, we conducted many experiments to evaluate and measure how our model performed. The following is the results of the performance analysis, including the evolution, and the heuristic analysis.

6.6.2.1 Performance Analysis

Part 1 (Fixed assigning cost): In this experiment, we used the first two benchmarks that we created, which involve 30 instances with a maximum of 40 vessels. We assume that we have two ports, one quay for each, and we need to distribute the incoming vessels over the available quays, such that the final schedule cost of the total services for all ports is minimised. The quay length for each port is 1000 metres and the time horizon is one week (168 hours). We assumed here that there is a fixed cost for all vessels to berth in the first quay (x) equal to zero ($c_{i1}^{ETA} = 0$), and the cost to berth in the second quay is equal to two ($c_{i2}^{ETA} = 2$). In this scenario, the port operator's strategy decision tries to force all incoming vessels to berth in the first port, to reduce the load on the second port. However, in the case of uncertainty or

if the first port is busy, then the *BACAP_Scheduler_DP* will try to send some of the incoming vessels to the second port while keeping the total cost minimised.

In this experiment, we tried to use different configurations for the CPLEX to find the most optimal solution, which in turn will be the guide to prove how far our proposed model is from the optimal. The CPLEX was configured as *MIPEMPHASIS_HIDDENFEAS* (emphasise finding hidden feasible solutions), with five parallel mode threads, *fraccuts* (generate Gomory fractional cuts aggressively) and applying local branching heuristics to the new incumbent.

Table 6-2 shows the comparison between the results of the CPLEX and the proposed heuristic model (GA). We noticed that the CPLEX solved the first small group of vessels (20 vessels) and we obtained the optimal solution for instances 1, 3, 8 and 9. We also observed that the CPLEX solved three instances from the group of 30 vessels (30-1, 30-4, 30-10) and that it could not solve any of the 40 vessel instances.

The BACAP is also known as the NP-Hard problem, so the complexity of the problem increases with the number of vessels. This is the case shown by the results of the CPLEX. Accordingly, we proposed a meta-heuristic genetic algorithm to solve this type of problem, which can also be used to solve a vast array of issues but cannot prove the optimality. To run the GA in this experiment, we used 20 generations and 30 populations as an initial stage and ran them five times. Comparing the results of the proposed heuristic algorithm (GA) with the CPLEX, we found that our model performed well. The results that we got from GA were near to the results of the CPLEX for the 20 and 30 vessel groups with a small gap. The GA solves all instances even with a large number of vessels, finding the optimal solution for instance 20-8.

Table 6-2: Multi-Port BACAP, GA and CPLEX comparison (2 Quays)

n	#	2Q-CPLEX	2Q-GA	Gap
20	1	67*	67.6	0.01
	2	49.9	50.8	0.02
	3	60.4*	61.9	0.02
	4	54.8	55.5	0.01
	5	50.1	51.9	0.04
	6	57.6	57.6	0
	7	59.8	60.7	0.02
	8	51.1*	51.1*	0
	9	61.3*	62.3	0.02
	10	60.4	61.3	0.01
30	11	91.2	93.9	0.03
	12	-	77.2	-
	13	-	87.1	-
	14	83.2	85	0.02
	15	-	101.4	-
	16	-	84.9	-
	17	-	87.2	-
	18	-	89.5	-
	19	-	101.6	-
	20	92.1	92.9	0.01
40	21	-	131	-
	22	-	118.8	-
	23	-	121.5	-
	24	-	151.3	-
	25	-	112.2	-
	26	-	130.1	-
	27	-	131.2	-
	28	-	137	-
	29	-	120.4	-
	30	-	123.99	-

* Optimal solution. Gap: (GA- CPLEX)/CPLEX

Figure 6-10 shows the box-plot for the results of five runs. The chart displays the distribution of the maximum, first quartile, median, third quartile and the minimum values of the thirty instances above. The best values regarding our objective function were the minimum ones as displayed in Table 6-2.

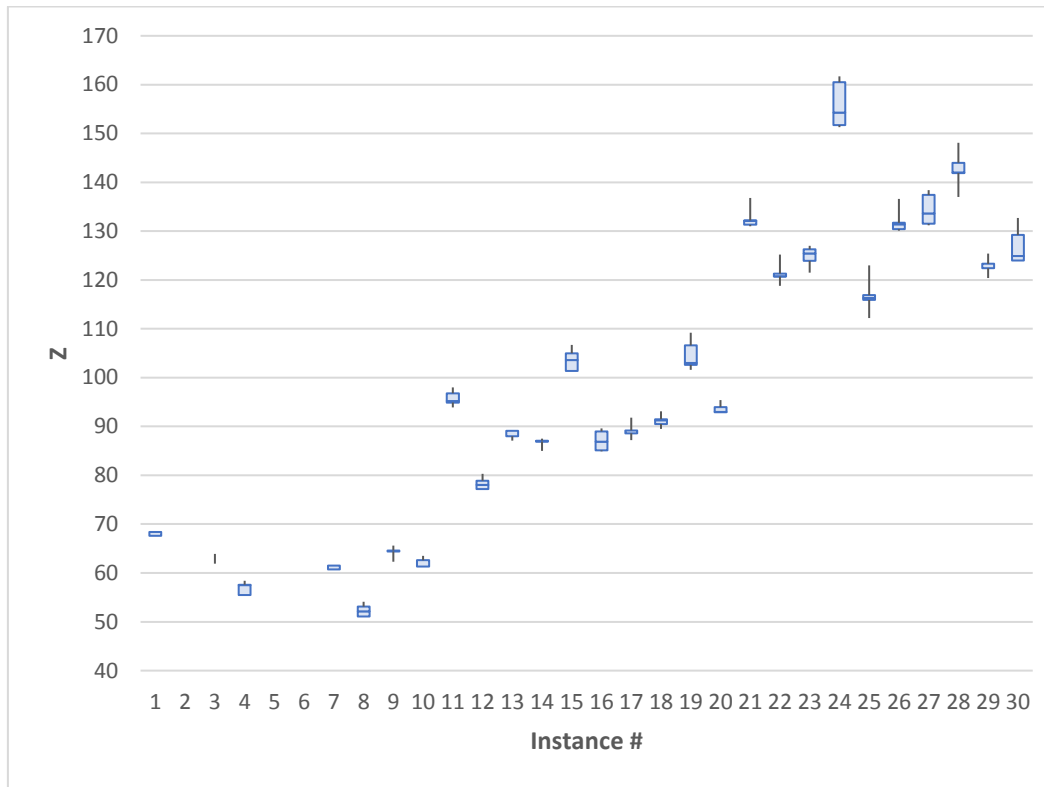


Figure 6-10: Box-plot for the objective function (z-values) of five runs, 2 Quays.

From the previous experiment, we extended the problem to two ports with three quays. The first port has two quays and the second one has one quay. Each quay is 1000 metres in length and the cost for all vessels to berth in the first port first quay (x) equals zero ($c_{i1}^{ETA} = 0$). The cost to berth in the first port second quay equals two ($c_{i2}^{ETA} = 2$) and the cost to berth in the second port equals three ($c_{i3}^{ETA} = 3$). The rest of the assumptions are similar those in the previous experiment.

Table 6-3 shows the comparison between the results of the CPLEX and the proposed model (GA) in the case of three quays. We noticed that the CPLEX also solved the first small group of vessels (20 vessels) and that it obtained the optimal solution for instances 8 and 9. We also observed that the CPLEX solved five instances from the group of 30 vessels and one of the 40 vessel instances (40-5), which shows that the number of feasible solutions is higher than the first experiment.

Table 6-3: Multi-Port BACAP, GA and CPLEX comparison (3 Quays)

n	#	3Q-CPLEX	3Q-GA	Gap
20	1	63.8	64.4	0.01
	2	49.9	50.8	0.02
	3	60.4	63.1	0.04
	4	54.8	55.5	0.01
	5	50.1	51.9	0.04
	6	57.6	57.6	0.00
	7	59.8	60.7	0.02
	8	51.1*	51.1*	0.00
	9	56.3*	57.3	0.02
	10	60.4	62.3	0.03
30	11	-	94.3	-
	12	-	79	-
	13	87.9	87.1	-0.01
	14	86.1	85	-0.01
	15	-	104.3	-
	16	-	85.6	-
	17	87.2	88.2	0.01
	18	87.1	92.9	0.07
	19	-	107.1	-
	20	91.2	96.7	0.06
40	21	-	138.3	-
	22	-	128.6	-
	23	-	130.6	-
	24	-	138.6	-
	25	122.1	122.5	0.00
	26	-	133.6	-
	27	-	133.8	-
	28	-	139.2	-
	29	-	126.2	-
	30	-	134.2	-

* Optimal solution. Gap: (GA- CPLEX)/CPLEX

Figure 6-11 shows the box-plot of the results of five runs. The chart displays the distribution of the maximum, first quartile, median, third quartile and the minimum values of the thirty instances above. The best values regarding our objective function are the minimum ones, as displayed in Table 6-3. When comparing Figure 6-10 and Figure 6-11, we can observe that the difference between the results in the case of 3 quays was more than in the case of 2 quays, while the solution space increased and more obvious solutions were found.

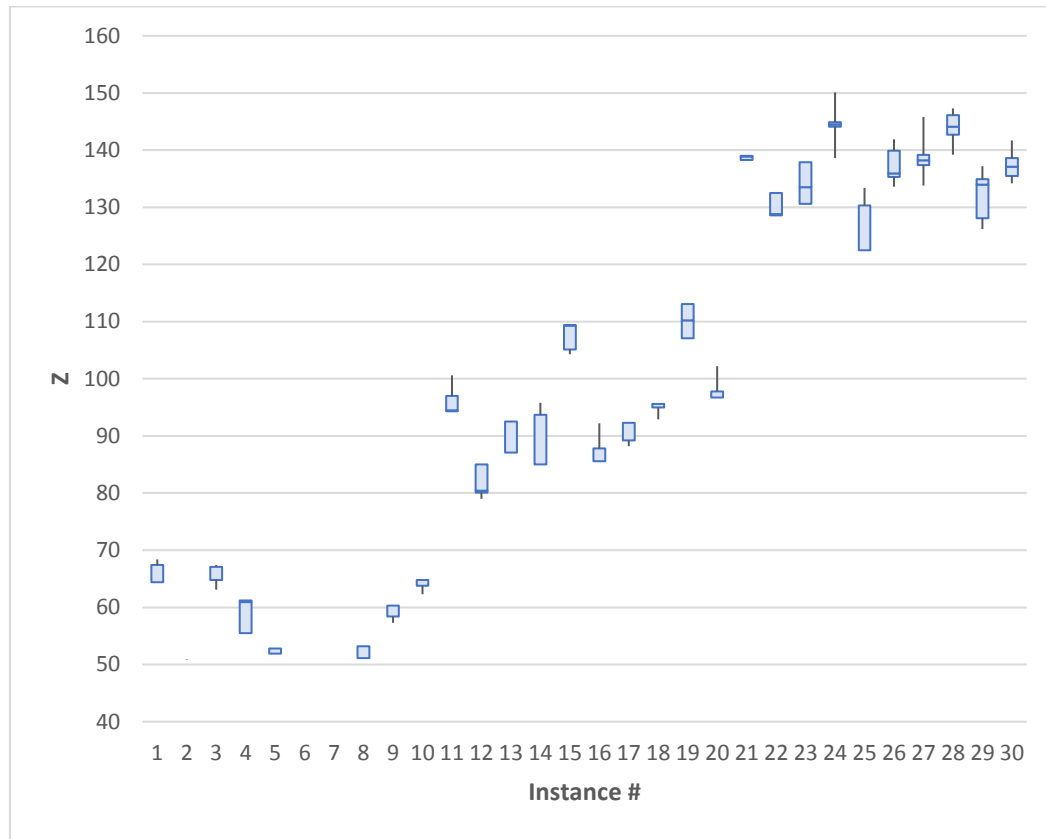


Figure 6-11: Box-plot for the objective function (z-values) of five runs, 3 Quays.

Accordingly, we can determine that the CPLEX can solve a number of instances for a higher number of vessels while the number of ports increases. This is because of the increase in the solution space. Returning to the performance of our proposed GA, we identified that the model can also solve all large instances with a small gap from CPLEX, which means that we can use our proposed algorithm for the multiple ports BACAP for a large number of vessels in an acceptable time.

Part 2 (Random assigning cost): in this experiment, we further studied the impact of changing the assigning cost of vessels to ports. We re-evaluated the previous experiments but with a random cost of assigning vessels to quays (vessel-quay matrix cost). This scenario is practical in many ports, as we have multiple ports and multiple shipping lines. Each vessel can berth in any port with a different cost to berth in each port. In this case, the port operators have the chance to choose which port is better for which vessel to berth in, to speed up the turnaround time of the

vessel's operations and to minimise the total cost for all ports that the port operator manages.

In this experiment, the CPLEX was ran twice. First, we configured it as MIPEMPHASIS_HIDDENFEAS (emphasise finding hidden feasible solutions), with five parallel mode threads, fraccuts (Generate Gomory fractional cuts aggressively) and a local branching heuristic applied to the new incumbent. Second, we configured it as previous to the change to MIPEMPHASIS_BALANCED (balance optimality and feasibility), with twelve parallel mode threads, which might get the results faster than five threads.

Table 6-4 shows the comparison between the results of the CPLEX and the proposed model GA for two quays and three quays in the case of assigning random costs. We noticed that the results were completely different than the results shown in Table 6-2 and Table 6-3. In the case of two quays, we found, that the CPLEX solved almost all 20 vessels and 30 vessels, and obtained the optimal solution for instances 1, 6, 8 and 9. Moreover, it also solved one of the 40 vessels instances (40-8). For the case of three quays, the CPLEX solved all 20 vessels, 50% of the 30 vessels and 40% of 40 vessels. This again indicates the high density of solutions when the number of quays increases. However, the total cost might also be increased. Appendix C.1 shows an example of multiple ports BACAP.

Comparing the results of the GA with the CPLEX, we found that the GA also performs well with a small gap and that it can solve all instances from small group up to large groups of vessels. Moreover, the GA obtained the optimal solution for some instances like 20-1 and 20-6 for both cases of two and three quays respectively. From Table 6-2, Table 6-3 and Table 6-4, we noticed from the results that using three quays performs better than using two quays regarding the total cost.

Table 6-4: Proposed model results, assuming random costs

n	#	2-Quay				3-Quay			
		CPLEX (5 threads)	CPLEX (12 threads)	(G=20, P=30)	(G=50, P=40)	CPLEX (5 threads)	CPLEX (12 threads)	(G=20, P=30)	(G=50, P=40)
20	1	109.4	109.4*	109.4*	109.4*	88.4*	88.4*	88.4*	88.4*
	2	86.4	86.4	86.4	86.4	80.4	80.4	80.4	80.4
	3	98.7	98.7	98.7	98.7	96.6	96.6	96.6	96.6
	4	90.8	90.8	90.8	90.8	84.8	84.8	84.8	84.8
	5	92	92	94	94	77.7	77.7	78.4	78.4
	6	103.7	103.7*	103.7*	103.7*	92.7*	92.7*	92.7*	92.7*
	7	89.3	89.3	89.3	89.3	86.2	86.2	86.2	86.2
	8	93.1*	93.1*	94.1	94.1	82.9*	82.9*	83.9	83.9
	9	93.4*	93.4*	96.6	97.5	90.2*	90.2	91	91
	10	99.1	-	100.9	99.9	84.6	84.6	84.8	84.8
30	11	143.9	-	147.1	147.1	135.3	-	135.3	135.3
	12	133.9	133.9	135.1	135.1	122.6	-	123.6	122.6
	13	138.8	138.8	138.8	139.1	-	-	132.8	132.8
	14	127.6	127.6	128.6	128.6	125.8	-	126.5	127.5
	15	150.9	-	153.9	153	-	-	141.3	141.3
	16	132.8	-	132.8	132.8	134.4	134.4	135.9	134.4
	17	139.5	139.5	139.6	139.6	-	-	132.9	132.9
	18	142.6	142.6	144.6	146.5	-	-	134.8	134.8
	19	151.6	139.5	156.7	153.7	-	-	143	140
	20	-	146.3	150.2	149.7	134.4	134.4	135.9	135.3
40	21	-	-	191.7	188.9	-	-	176.9	174.7
	22	-	-	191.4	188.4	170.3	-	170.5	170.5
	23	-	-	195.4	196.3	168.4	168.4	171.1	171.1
	24	-	-	213.9	214.2	-	-	186.4	185.6
	25	-	-	179.8	180.8	164.8	164.8	165.8	166.8
	26	-	-	199.1	197.1	-	-	178.8	180.7
	27	-	-	198.3	194	-	-	171	170
	28	187.6	-	202.4	209.1	-	179.7	180.5	180.5
	29	-	-	185.7	187.7	-	-	165.5	165.5
	30	-	-	193.2	194.4	178.8	178.8	179.9	179

* Optimal solution.

In the following experiment, we expanded the dataset to 100 vessels and 5 quays to test the performance of our GA model, Table 6-5 shows the results. The CPLEX could not handle any of the instances in the large dataset, but the GA found a

feasible solution. Moreover, the GA stopped after the stopping condition of 10 hours was reached and thus could not reach the maximum number of generations due to the complexity of the problem.

Table 6-5: Results of 100 vessels with 5 Quays.

n	#	5Q-CPLEX	5Q-GA (G=20, P=30)	5Q-GA (G=50, P=40)
100	1	-	407.1	408.1
	2	-	370.2	372.2
	3	-	417.8	428.3
	4	-	394.8	390.2
	5	-	447.1	434.5
	6	-	419.8	414
	7	-	486.4	466.3
	8	-	404.2	404.6
	9	-	453.8	451.8
	10	-	475.3	478.5

6.6.2.2 Evolution Analysis

In this experiment, we studied the impact of changing the GA parameters on the results. We changed the number of generations from 20 to 50 and changed the population from 30 to 40. Figure 6-12 and Figure 6-13 show the results in the case of 2 quays and 3 quays with a fixed cost of assignment as mentioned earlier. We noticed that there is not much difference between changing the generation and population on the results. However, increasing the number of generations and the population ended with better results in the case of a large number of vessels in the 3 quays problem as shown in Figure 6-13.

In the case of 2 quays and 3 quays with a random cost of assignment, as in Figure 6-14 and Figure 6-15, we observed that increasing the number of generations and the population ended with almost similar results with slight improvement.

Accordingly, we can conclude that changing the number of population and generation in the GA has a better effect on the results while the complexity of the problem increased.

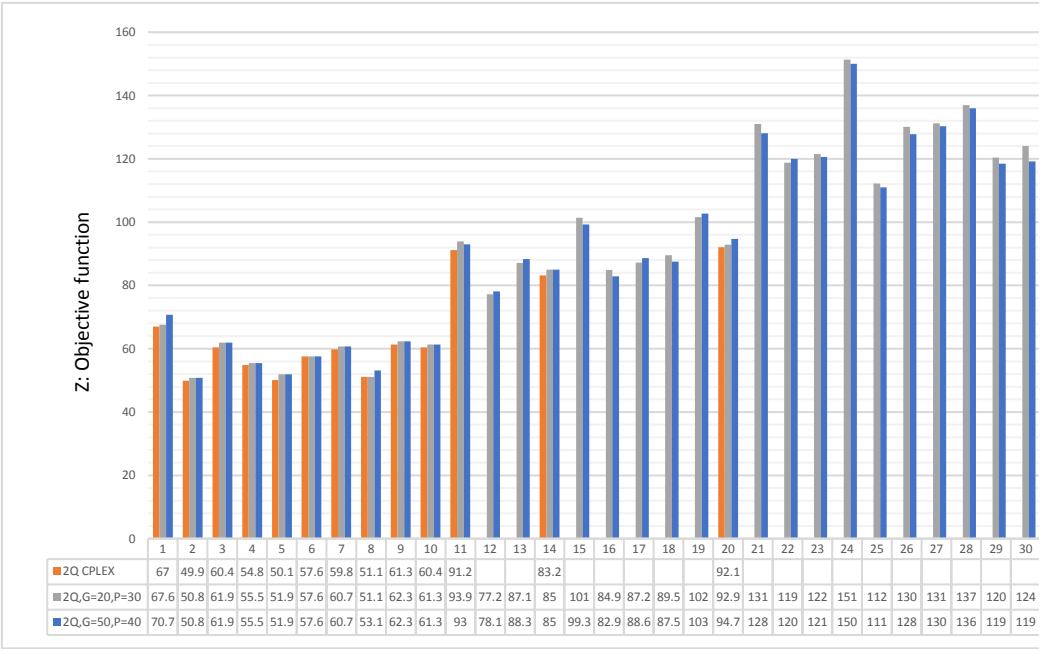


Figure 6-12: Evolution Analysis, 2 Quays, fixed cost

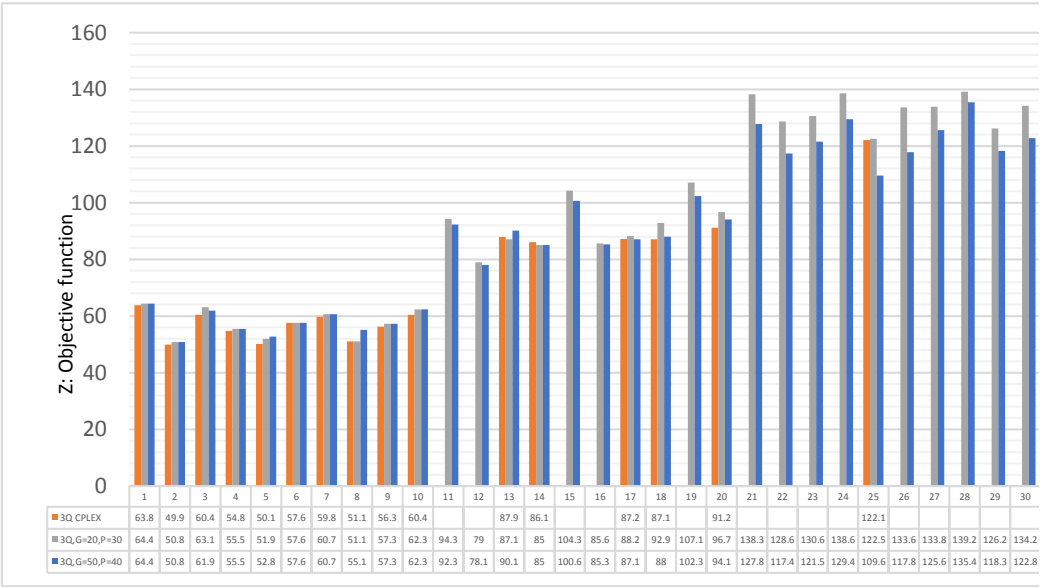


Figure 6-13: Evolution Analysis, 3 Quays, fixed cost

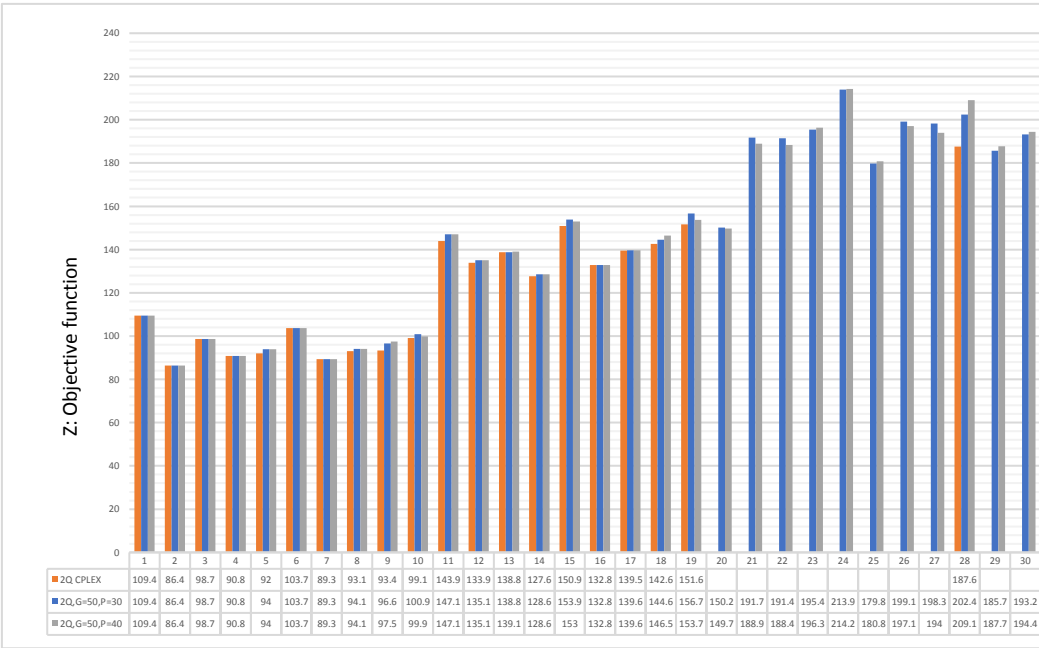


Figure 6-14: Evolution Analysis, 2 Quays, random cost

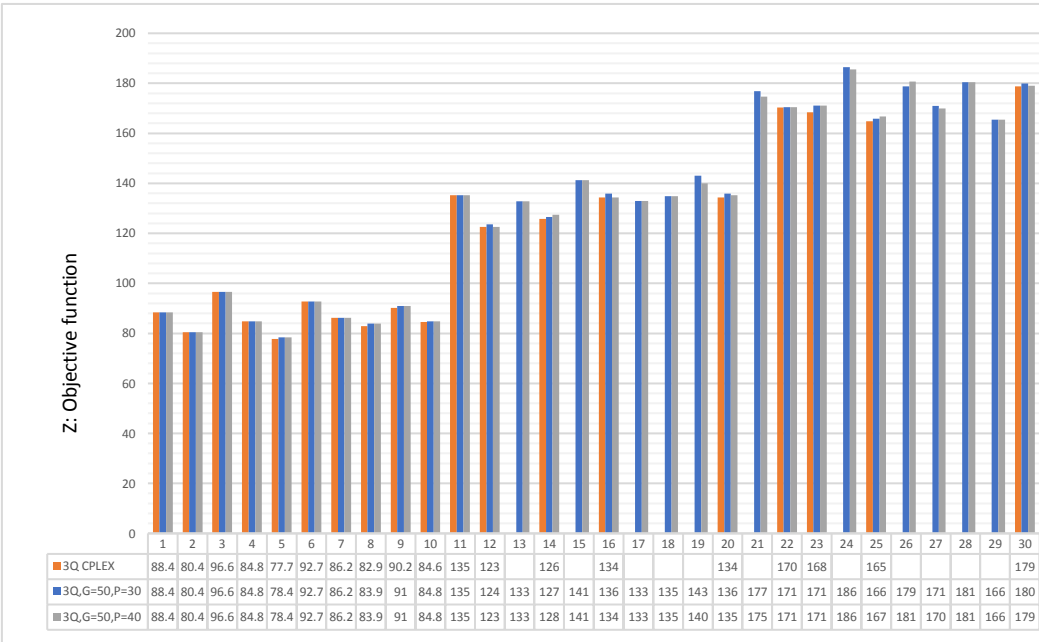


Figure 6-15: Evolution Analysis, 3 Quays, random cost

6.6.2.3 Heuristic Analysis

In this sub-section, we studied the impact of changing the dispatching rules on the results. The rules considered have been shown in the following table.

ID	Rules	Description
1.	FCFS	First-Come-First-Serve
2.	maxOCNeededASC	Maximum Quay Crane needed ascending
3.	maxOCNeededDES	Maximum Quay Crane needed descending
4.	vLengthBigToSmall	Vessel length sorted by descending
5.	vLengthSmallToBig	Vessel length sorted by ascending
6.	movementLowToHigh	Number of container needs to be loaded or unloaded ascending
7.	movementHighToLow	Number of container needs to be loaded or unloaded descending
8.	minTimeHandlingASC	Minimum time needed to handle a vessel ascending
9.	minTimeHandlingDES	Minimum time needed to handle a vessel descending
10.	EFT	Expected vessel finish time ordered ascending

Figure 6-16 shows the efficient rules used in all of the previous GA experiments. The x-axis is the rule ID, and the y-axis is the count of how many times the rules succeed or were chosen during the GA generations. We noticed that the best five rules in descending order used in the process of optimisation for the case of 2 quays was the following rules ID (1, 3, 9, 4 and 2); FIFS, maxOCNeededDES, minTimeHandlingDES, vLengthBigToSmall, and maxOCNeededASC.

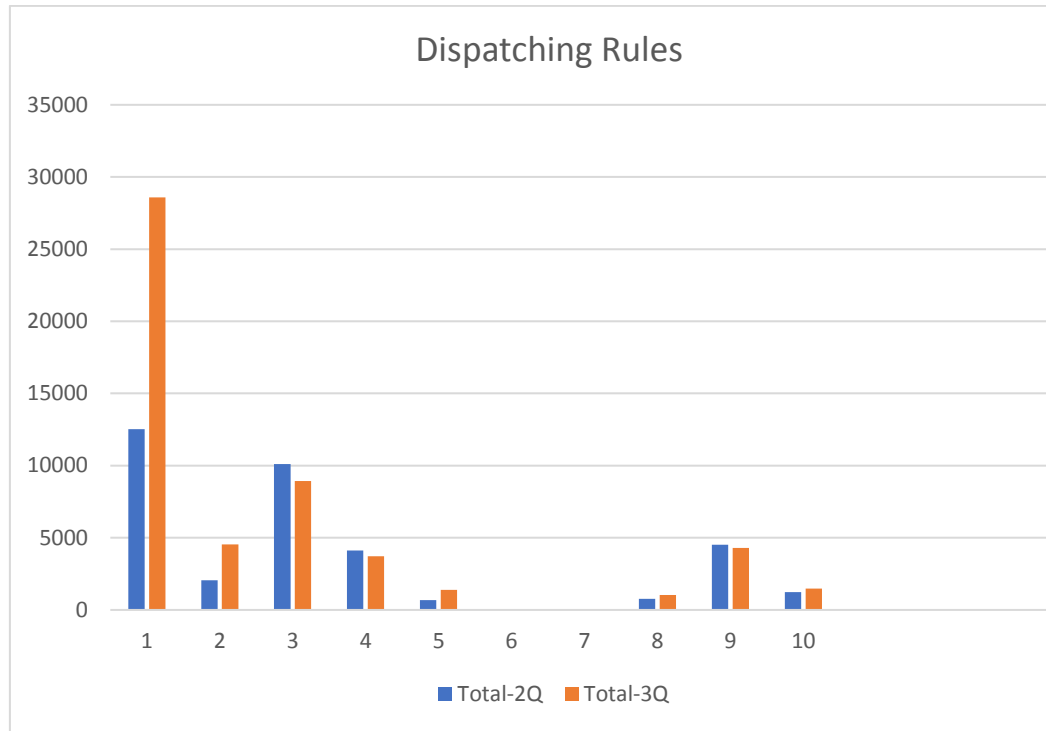


Figure 6-16: Counts of dispatching rules used in the GA generations

On the other hand, in the case of 3 quays, we recognised that the best five rules are the same rules as in the case of 2 quays, but in a different order. The rules are 1, 3, 9, 2 and 4 respectively, which are FIFS, maxOCNeededDES, maxOCNeededASC, minTimeHandlingDES and vLengthBigToSmall. Moreover, we found that there are a few rules that have not been counted in the process of optimisation, for instance, rules 6 and rule 7. This means that these rules have less of an impact on the final solution. There is a crucial relationship between the rules used for dispatching the list of incoming vessels and the performance of the solution.

6.7 Conclusions

In this chapter, we studied the berth allocation problem integrated with the quay crane assignment problem on an incoming set of container vessels related to a set of container ports. These ports were managed by one port operator. The objective function is to minimise the total cost of services for all vessels and to minimise their turnaround time. To the best of our knowledge, this is the first study that has tackled the BACAP with multiple ports and multiple quays, in addition to the integrated levels of planning.

The port operator at the strategic level can identify the number of ports needed to work for an extended period (e.g. a year). We can do this by solving the problem by assigning a fixed cost or a high cost to a vessel-quay matrix cost list, as shown before. Consequently, we can solve it again with different cost values. The results should identify how many ports/quays are used and how the solution performs regarding the fitness function. From this end, the port operator can decide if it is better to use all of the ports/quays or just some of them. They will get useful information in order to conduct an analysis of how the quays will perform for the next year. For the benefit of this model, the port operator can pre-plan how many new contracts can be accepted or refuse them in the following year.

The proposed model that has integrated the three levels of plans attempted to apply the high-level plan's information to the low-level plan, and we found out how this performed before returning this to the high-level plan again. This model continued until it found a better solution that satisfied the ship lines and port operator.

The problem was solved using an exact method, using the CPLEX. The results indicate that this approach can solve a small group of vessels only. Therefore, a proposed a meta-heuristic GA algorithm was developed which is capable of solving this type of problem with a large group of vessels.

The results of the proposed GA algorithm prove that our model can perform well with both a small and high number of vessels, and in the case where we increased the complexity of the problem.

Chapter 7 : Conclusions and Future Work

This chapter covers the conclusions of the research. It reviews the overall research findings, and summaries the contributions. It demonstrates the success related to achieving the research aim and objectives. Finally, it identifies future work and recommendations.

7.1 Research Summary and Conclusions

In this thesis, we focused on improving container terminal efficiency in the seaside operations. Two crucial container terminal planning problems have been investigated; the berth allocation problem (BAP) and the quay crane assignment problem (QCAP). This has covered both single and multiple ports scenarios. We developed a new method to solve the integrated berth allocation and quay crane assignment problem (BACAP), aiming to optimise the total service time for all incoming vessels and to minimise the terminal's service costs. Applying the integration concept enables container terminal planners to obtain feasible plans for all incoming vessels. This is where berthing positions, berthing times and assigning quay cranes are dependently determined for the vessels.

While the BACAP is a NP-hard problem and it has been solved by exact methods in the literature for small instances only, most of researchers use heuristic-based methods to solve the problem in a large number of instances. Genetic programming (GP) is a meta-heuristic method which is considered to be a sub of genetic algorithm (GA). It can search in the solutions' problem space of and finds a method/solver rather than finding only one specific solution to the specific problem. In this regard, we were interested in using the GP method to solve the BACAP. The GP framework in this research plays an important role in optimising the vessels' dispatching rules (DRs) and combining them into composite dispatching rules (CDRs) used for ordering any given vessel list. The obtained plans from the proposed evolved

solvers enable a highly productive utilisation of quay space and quay cranes as well as minimising the makespan for vessels' berthing schedule. We presented the state-of-the-art solution to the problem, and discussed the gaps and new trends that the literature does not cover.

Firstly, we solved the BACAP using nine different common priority/dispatching rules by developing an independent scheduler (*BACAP_Scheduler*). The aim was to schedule all incoming vessels in a two-dimensional array while considering the constraints of the problem. The novelty of the proposed *BACAP_Scheduler* is can work independently with any optimisation techniques. Secondly, we introduced a GP approach (*BACAP_GP*) to dynamically evolve efficient DRs and to obtain CDRs to solve the problem. The presented study was developed in order to determine the most effective CDRs and their impact on the total vessel service time. We provided a literature review of the research studies that solved the BACAP, considering DRs. We discovered the importance of the terminal operators concerning adopting the vessel's priority strategy used to solve the BACAP with relevance to the different input constraint.

The *BACAP_GP* searches the heuristic in order to find a better solution. As soon as a berth is available with QCs, the generated rule is applied directly to the vessel list. The vessel with the highest order is selected to be processed. The benchmark that we used was well-known large-scale instances drawn from the literature, which contains 100 instances generated randomly, following the suggestions of the container terminal operators. Each instance is composed of a queue with 100 vessels. Computational results and comparisons showed that the proposed *BACAP_GP* outperforms the standard priority rules in all instances and that it is more flexible when it comes to handling different scenarios and different port layouts. Moreover, solving the BACAP using the *BACAP_GP* while taking into consideration dynamic quay cranes (time-variant) also helps to decrease the waiting time for the vessels and speeds up the work in the terminals by minimising the total service time for all incoming vessels. It improves the performance of the container terminal operations, which was the major aim of this thesis.

From the above efficient outcome of the proposed algorithms, we used the improved *BACAP_Scheduler_DP* and *BACAP_GP_DP* to tackle the BACAP with desired berthing position by innovating a mathematical model to predict any overlap that might happen between the vessels during scheduling. This approach predicts the total costs that might be incurred in case of overlapping, while finding the optimal berthing position with minimum costs. The *BACAP_GP_DP* is a construction method. The computational results show the efficient performance of the developed algorithms for *BACAP_GP_DP* that solve the above problem. It outperformed all DRs in all instances using a construction method, and is very competitive when compared to well-known literature improvement methods.

To the best of our knowledge, there are insufficient research studies available in the literature that tackle the BACAP in the case of multiple ports in which the incoming vessels have the opportunity to berth in different ports where multiple terminals are owned by one port operator. This encouraged us to apply the above approaches to this type of problem. Moreover, we studied the problem while considering the integrated levels of planning, strategic, tactical and operational levels. The aim was to achieve the satisfaction of the shipping lines and port operator's aims. We developed a mathematical model to solve the problem *MultiP_BACAP_GA* and generated a suitable benchmark for testing. We solved the problem with two methods; an exact method using the commercial solver CPLEX, which can solve only a small number of instances, and a meta-heuristics method using GA which can solve any number of instances and find a near-optimal solution in an acceptable computational time. The computational results proved that our approach can perform well when faced with both a small and large number of instances.

7.2 Research Contributions

This research has the following major contributions, which have been summarised from each chapter and that demonstrate the success of the research aim and objectives.

1) Chapter 4:

- A. Develops an optimisation approach for solving the integrated BAP and QCAP.
- B. Applies composite dispatching rules rather than simple dispatching rules, and use a new priority-based schedule construction procedure.
- C. Determines that genetic programming meta-heuristic is not a conventional method to solve the BACAP in the literature from the optimisation viewpoint. We have presented a GP based approach to evolve the dispatching rules for the BACAP problem. In addition to the high performance of the evolved DRs, the main advantage of the approach is the self-adaptability of the proposed methods, which automatically discover/evolve high performing DR using different sets of variables based on what is available in each berth.
- D. Presents an independent scheduler for the BACAP that could be extended to deal with different objectives or combined with any appropriate optimisation method to find a better solution.
- E. Provides an analysis of a wide range of DRs and solves the BACAP for a large benchmark.

2) Chapter 5:

- A. Solves the continuous dynamic BACAP with desired berthing position for each vessel.

- B. Compares the results with well-known BACAP benchmark with an objective function to minimise the total cost of serving ships in a given terminal.
 - C. Develops a new technique to improve quay space utilisation while scheduling.
 - D. Demonstrates that the model is flexible enough to solve the problem of QC assignment using both Time-variant, and QC Time-invariant approaches.
 - E. Integrates both the terminal operator's and ship owner's goals.
- 3) Chapter 6:
- A. Develops a novel mathematical model to solve the BACAP in the case of multiple ports being available.
 - B. Produces a new generated dataset to test the model.
 - C. Solves the model with an exact method (CPLEX) for small instances.
 - D. Solves the model with GA and compares the solution with that of CPLEX.
 - E. Demonstrates a comparative study and analysis using CDR and SPR.
 - F. Applies a new concept to solve the BACAP using an integrated method at the strategic and operational planning levels for decision makers.

7.3 Future Work

Future work can go in three main directions. The first direction is related to expanding the framework in terms of the scope and functionality. The second direction is related to the literature benchmarks. The third direction is related to single and multiple ports datasets.

7.3.1 Framework Scope and Functionality

Further research in this direction aims to extend the proposed approaches in order to handle new features such as dual cycle cranes, which allow for QCs to discharge a container in the same cycle as a loading operation. Double spreader cranes which can transfer two containers at the same time. Moreover, the aim is to test and evaluate the behaviour of the evolved heuristics when combined with other optimisation methods. The aim in the future is to combine the GP-evolved heuristics with an improvement optimisation layer. In addition, it will be interesting to integrate the third problem present in seaside operational planning, which is the quay crane scheduling problem (QCSP) with the presented approaches. Also, more realistic constraints could be considered such as, varying depth of quay, other types or mixed types of quay layouts.

Last but not least, is to solve the BACAP when considering fuel consumption and vessel emissions. In recent years with the increasing number of visiting vessels to port bringing in a large volume of vessel emissions, this has attracted the extensive attention of society. The legislation on vessel emissions was brought into force by the International Maritime Organisation (IMO) in 2005 (MARPOL Annex VI “prevention of air pollution from ships”, IMO, 2005) to reduce the negative environmental impacts and to pursue green ports and clean air at sea.

7.3.2 Related to Literature Benchmarks

Most of the literature solves the BACAP when considering deterministic parameters. However, uncertainty is more realistic. The uncertainties might include changes in the ETA, a weather forecast that affects the vessel depth, QCs breakdowns etc. Future research might examine the problem considering uncertainty in order to find a robust optimisation model.

The literature review includes different benchmarks that cannot be used to compare the researchers’ work and most of them are not always available. It is crucial to generate a generic benchmark that accepts and meets all of the required criteria so then future researchers can use it to compare their work and obtain common findings to solve the BACAP. To cope with the benchmark redundancy, the target

of future work in this direction can create a website that includes the current literature benchmarks and develops a benchmark generator for container terminal operation problems. The following is our approach regarding the website content and facilities which may help future researchers to test and evaluate their work and compare them to other researchers' models:

- Collect the current literature benchmarks and upload them to the website with a full description from the authors related to the following:
 - o The problem to solve.
 - o Number of instances.
 - o Input data.
 - o Output data.
 - o Problem assumptions.
 - o Other useful information/descriptions.

These benchmarks will be collected by communicating directly with the authors to retrieve their data or creating an account for them on the website, so then the authors will find it more flexible when it comes to adding, editing or deleting their information.

7.3.3 Single/Multi-port Dataset Generator

The aim of future work in this direction is to develop a generic benchmark generator accessible to future researchers. This generator can be used to create different benchmarks for different types of seaside problems. The generator might be flexible enough to suit the researcher's need, taking into consideration the problem constraints, quay layout and if the problem is in a single or multiple ports setting. The following is our approach regarding this direction:

- Define the standard ranges for each data-set of the benchmark that is close to real terminal data.
- Define a benchmark that covers different types of classified operational problems (BAP, QCAP, QCSP) and the integrated ones.
- Develop a flexible benchmark generator that helps the user to generate a benchmark with various options to cover his assumptions and then to

provide an ID for the generated benchmark. This benchmark will be available on the website for any further researchers to use it.

- The researchers can create an account on the website to upload their updated experimental results concerning a specific benchmark and to provide the method that they used to solve the problem.
- Obtain real data from specific ports and upload it to the website. Moreover, to generate different benchmarks related to the ports in different scenarios.

References

- A Java-based Evolutionary Computation Research System. (n.d.). ECJ. Retrieved January 1, 2014, from <http://cs.gmu.edu/~eclab/projects/ecj/>
- Abbas, A., Al-Bazi, A., & Palade, V. (2018). A Constrained Fuzzy Knowledge-Based System for the Management of Container Yard Operations. *International Journal of Fuzzy Systems*, 20(4), 1205–1223. <http://doi.org/10.1007/s40815-018-0448-9>
- Agra, A., & Oliveira, M. (2018). MIP approaches for the integrated berth allocation and quay crane assignment and scheduling problem. *European Journal of Operational Research*, 264(1), 138–148. <http://doi.org/10.1016/j.ejor.2017.05.040>
- Ali, I., Abouelseoud, Y., & Elwany, M. H. (2011). Container terminal berth allocation and quay crane assignment using IP and simulated annealing. *41st International Conference on Computers & Industrial Engineering*, 31–37.
- Alsoufi, G., Yang, X., & Salhi, A. (2015). A combined Mixed Integer Programming model of seaside operations arising in container ports, (May), 1–16.
- Ambrosino, D., Caballini, C., & Siri, S. (2013). A mathematical model to evaluate different train loading and stacking policies in a container terminal. *Maritime Economics & Logistics*, 15(3), 292–308. <http://doi.org/10.1057/mel.2013.7>
- Ambrosino, D., Sciomachen, A., & Tanfani, E. (2004). Stowing a containership: the master bay plan problem. *Transportation Research Part A: Policy and Practice*, 38(2), 81–99. <http://doi.org/10.1016/j.tra.2003.09.002>
- Ambrosino, D., Sciomachen, A., & Tanfani, E. (2006). A decomposition heuristics for the container ship stowage problem. *Journal of Heuristics*, 12(3), 211–233. <http://doi.org/10.1007/s10732-006-5905-1>
- Angeloudis, P., & Bell, M. G. H. (2011). A review of container terminal simulation models. *Maritime Policy & Management*, 38(5), 523–540. <http://doi.org/10.1080/03088839.2011.597448>
- Aras, N., Türkoğulları, Y., Taşkın, Z. C., & Altınel, K. (2014). Simultaneous Optimization of Berth Allocation, Quay Crane Assignment and Quay Crane Scheduling Problems in Container Terminals. In *Research Proceedings 2012* (pp. 101–107). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-00795-3_15
- Basri, A. H., & Zainuddin, Z. M. (2014). Development of integrated model for continuous berth allocation problem and quay crane scheduling with non crossing constraint. In *AIP Conference Proceedings* (Vol. 1613, pp. 113–121). <http://doi.org/10.1063/1.4894337>
- Bierwirth, C., & Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3), 615–627. <http://doi.org/10.1016/j.ejor.2009.05.031>
- Bierwirth, C., & Meisel, F. (2015). A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 244(3), 675–689. <http://doi.org/10.1016/j.ejor.2014.12.030>
- Bish, E. K., Chen, F. Y., Leong, Y. T., Nelson, B. L., Ng, J. W. C., & Simchi-Levi, D. (2005). Dispatching vehicles in a mega container terminal. *OR Spectrum*, 27(4), 491–506. <http://doi.org/10.1007/s00291-004-0194-2>
- Blazewicz, J., Cheng, T. C. E., Machowiak, M., & Oguz, C. (2011). Berth and quay crane allocation: a moldable task scheduling model. *Journal of the Operational Research Society*, 62(7), 1189–1197. <http://doi.org/10.1057/jors.2010.54>
- Bockstael-Blok, W., Mayer, I., & Valentin, E. (2003). Supporting the design of an inland container terminal through visualization, simulation and gaming. *Proceedings of the 36th Annual*

- Hawaii International Conference on System Sciences, HICSS 2003*. <http://doi.org/10.1109/HICSS.2003.1173652>
- Bose, J., Reinert, T., Steenken, D., & Voss, S. (2000). Vehicle dispatching at seaport container terminals using evolutionary algorithms. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 00(c), 1–10. <http://doi.org/10.1109/HICSS.2000.926669>
- Böse, J. W. (2011). Simulation of Container Ship Arrivals and Quay Occupation. In *Operations Research/ Computer Science Interfaces Series* (Vol. 49). <http://doi.org/10.1007/978-1-4419-8408-1>
- Bruggeling, M., Verbraeck, A., & Honig, H. J. (2011). Decision Support for Container Terminal Berth Planning: Integration and Visualization of Terminal Information. *Proceedings van de Vervoerslogistieke Werkdagen 2011 (VLW2011)*.
- Carlo, H. J., Vis, I. F. A., & Roodbergen, K. J. (2014a). Storage yard operations in container terminals: Literature overview, trends, and research directions. *European Journal of Operational Research*, 235(2), 412–430. <http://doi.org/10.1016/j.ejor.2013.10.054>
- Carlo, H. J., Vis, I. F. A., & Roodbergen, K. J. (2014b). Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. *European Journal of Operational Research*, 236(1), 1–13. <http://doi.org/10.1016/j.ejor.2013.11.023>
- Carlo, H. J., Vis, I. F. A., & Roodbergen, K. J. (2015). Seaside operations in container terminals: literature overview, trends, and research directions. *Flexible Services and Manufacturing Journal*, 27(2–3), 224–262. <http://doi.org/10.1007/s10696-013-9178-3>
- Carteni, A., & Luca, S. De. (2012). Tactical and strategic planning for a container terminal: Modelling issues within a discrete event simulation approach. *Simulation Modelling Practice and Theory*, 21(1), 123–145. <http://doi.org/10.1016/j.simpat.2011.10.005>
- Cavalcante, E., Oppen, J., Samer, P., & Urrutia, S. (2016a). Combinatorial Relaxation Bounds and Preprocessing for Berth Allocation Problems. *Electronic Notes in Discrete Mathematics*, 55, 85–88. <http://doi.org/10.1016/j.endm.2016.10.022>
- Cavalcante, E., Oppen, J., Samer, P., & Urrutia, S. (2016b). The matching relaxation for a class of generalized set partitioning problems. *ArXiv Preprint ArXiv:1606.09279*, (227084), 1–14.
- Chang, D., He, J., & Zhang, H. (2010). A rule-based joint berth allocation and quay crane assignment. In *2010 2nd International Conference on Industrial and Information Systems, IIS 2010* (Vol. 1, pp. 464–467). <http://doi.org/10.1109/INDUSIS.2010.5565809>
- Chang, D., Jiang, Z., Yan, W., & He, J. (2010). Integrating berth allocation and quay crane assignments. *Transportation Research Part E: Logistics and Transportation Review*, 46(6), 975–990. <http://doi.org/10.1016/j.tre.2010.05.008>
- Changchun, L., Canrong, Z., & Li, Z. (2016). Behavior perception-based disruption models for berth allocation and quay crane assignment problems. *Computers & Industrial Engineering*. <http://doi.org/10.1016/j.cie.2016.04.008>
- Chen, G., Govindan, K., & Yang, Z. (2013). Managing truck arrivals with time windows to alleviate gate congestion at container terminals. *International Journal of Production Economics*, 141(1), 179–188. <http://doi.org/10.1016/j.ijpe.2012.03.033>
- Chen, J. H., Lee, D. H., & Cao, J. X. (2012). A combinatorial benders' cuts algorithm for the quayside operation problem at container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 48(1), 266–275. <http://doi.org/10.1016/j.tre.2011.06.004>
- Chen, X., & Yang, Z. (2012). An algorithm for continuous berth allocation with quay crane dynamic allocation. In *Proceedings - 2012 5th International Conference on Intelligent Computation Technology and Automation, ICICTA 2012* (pp. 541–544). <http://doi.org/10.1109/ICICTA.2012.142>
- Cheong, C. Y., Habibullah, M. S., Goh, R. S. M., & Fu, X. (2010). Multi-objective optimization of large scale berth allocation and quay crane assignment problems. *Conference Proceedings -*

- IEEE International Conference on Systems, Man and Cybernetics*, 669–676. <http://doi.org/10.1109/ICSMC.2010.5641824>
- Cheong, C. Y., Tan, K. C., Liu, D. K., & Lin, C. J. (2010). Multi-objective and prioritized berth allocation in container ports. *Annals of Operations Research*, 180(1), 63–103. <http://doi.org/10.1007/s10479-008-0493-0>
- Cheung, R. K., Li, C.-L., & Lin, W. (2002). Interblock Crane Deployment in Container Terminals. *Transportation Science*, 36(1), 79–93. <http://doi.org/10.1287/trsc.36.1.79.568>
- Cordeau, J.-F., Gaudioso, M., Laporte, G., & Moccia, L. (2007). The service allocation problem at the Gioia Tauro Maritime Terminal. *European Journal of Operational Research*, 176(2), 1167–1184. <http://doi.org/10.1016/j.ejor.2005.09.004>
- Cordeau, J.-F., Laporte, G., Legato, P., & Moccia, L. (2005). Models and Tabu Search Heuristics for the Berth-Allocation Problem. *Transportation Science*, 39(4), 526–538. <http://doi.org/10.1287/trsc.1050.0120>
- Correcher, J. F., & Alvarez-Valdes, R. (2017). A biased random-Key genetic algorithm for the time-invariant berth allocation and quay crane assignment problem. *Expert Systems with Applications*, 89, 112–128. <http://doi.org/10.1016/j.eswa.2017.07.028>
- Dadashi, A., Dulebenets, M. A., Golias, M., & Sheikholeslami, A. (2017). A novel continuous berth scheduling model at multiple marine container terminals with tidal considerations. *Maritime Business Review*, 2(2), 142–157. <http://doi.org/10.1108/MABR-02-2017-0010>
- De Castillo, B., & Daganzo, C. F. (1993). Handling strategies for import containers at marine terminals. *Transportation Research Part B: Methodological*, 27(2), 151–166. [http://doi.org/10.1016/0191-2615\(93\)90005-U](http://doi.org/10.1016/0191-2615(93)90005-U)
- De León, A. D., Lalla-Ruiz, E., Melián-Batista, B., & Marcos Moreno-Vega, J. (2017). A Machine Learning-based system for berth scheduling at bulk terminals. *Expert Systems with Applications*, 87, 170–182. <http://doi.org/10.1016/j.eswa.2017.06.010>
- Diabat, A., & Theodorou, E. (2014). An Integrated Quay Crane Assignment and Scheduling Problem. *Computers and Industrial Engineering*, 73(December 2013), 115–123. <http://doi.org/10.1016/j.cie.2013.12.012>
- Dong, P., Hu, Z. H., & Tao, S. (2013). Berth and crane allocation problem based on cost analysis of quay cranes for container terminal. *Dalian Haishi Daxue Xuebao/Journal of Dalian Maritime University*, 39(2), 60–64.
- Du, Y., Xu, Y., & Chen, Q. (2010). A feedback procedure for robust berth allocation with stochastic vessel delays. In *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)* (pp. 2210–2215). <http://doi.org/10.1109/WCICA.2010.5554316>
- El-boghdady, T., Bader-El-Den, M., & Jones, D. (2016a). A Genetic Programming Algorithm for the Berth and Quay Crane Allocation Problem. In *2016 5th International Conference on Advanced Logistics and Transport (ICALT)*. IEEE.
- El-boghdady, T., Bader-El-Den, M., & Jones, D. (2016b). Evolving local search heuristics for the integrated berth allocation and quay crane assignment problem. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2880–2887). IEEE. <http://doi.org/10.1109/CEC.2016.7744153>
- Elwany, M. H., Ali, I., & Abouelseoud, Y. (2013). A heuristics-based solution to the continuous berth allocation and crane assignment problem. *Alexandria Engineering Journal*, 52(4), 671–677. <http://doi.org/10.1016/j.aej.2013.09.001>
- Emde, S., Boysen, N., & Briskorn, D. (2014). The berth allocation problem with mobile quay walls: Problem definition, solution procedures, and extensions. *Journal of Scheduling*, 17(3), 289–303. <http://doi.org/10.1007/s10951-013-0358-5>
- European Commission. (2013). Europe's Seaports 2030: Challenges Ahead. European Commission - MEMO/13/448 23/05/2013.

- Expósito-Izquierdo, C., Lalla-Ruiz, E., de Armas, J., Melián-Batista, B., & Moreno-Vega, J. M. (2017). Maritime Container Terminal Problems. In *Handbook of Heuristics* (pp. 1–27). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-07153-4_57-1
- Expósito-Izquierdo, C., Melián-Batista, B., & Moreno-Vega, M. (2012). Pre-Marshalling Problem: Heuristic solution method and instances generator. *Expert Systems with Applications*, 39(9), 8337–8349. <http://doi.org/10.1016/j.eswa.2012.01.187>
- Festa, P. (2014). A brief introduction to exact, approximation, and heuristic algorithms for solving hard combinatorial optimization problems. In *2014 16th International Conference on Transparent Optical Networks (ICTON)* (pp. 1–20). IEEE. <http://doi.org/10.1109/ICTON.2014.6876285>
- Fonseca, J. (2016). APM Terminals Broaden Portfolio, Business Model. Retrieved June 8, 2018, from <https://www.marinelink.com/news/terminals-portfolio404998>
- Frojan, P., Correcher, J. F., Alvarez-Valdes, R., Koulouris, G., & Tamarit, J. M. (2015). The continuous Berth Allocation Problem in a container terminal with multiple quays. *Expert Systems with Applications*, 42(21), 7356–7366. <http://doi.org/10.1016/j.eswa.2015.05.018>
- Ganji, S. R. S., Babazadeh, a., & Arabshahi, N. (2010). Analysis of the continuous berth allocation problem in container ports using a genetic algorithm. *Journal of Marine Science and Technology*, 15(4), 408–416. <http://doi.org/10.1007/s00773-010-0095-9>
- Gao, Z. J., Cao, J. X., & Zhao, Q. Y. (2014). Optimization Research of Berth Allocation and Quay Crane Assignment at Container Terminal Based on the Genetic Algorithm. *Applied Mechanics and Materials*, 505–506, 931–934. <http://doi.org/10.4028/www.scientific.net/AMM.505-506.931>
- Giallombardo, G. (2008). The Tactical Berth Allocation Problem With Quay Crane Assignment and Transshipment - Related Quadratic Yard. *Aet*.
- Giallombardo, G., Moccia, L., Salani, M., & Vacca, I. (2010). Modeling and solving the Tactical Berth Allocation Problem. *Transportation Research Part B: Methodological*, 44(2), 232–245. <http://doi.org/10.1016/j.trb.2009.07.003>
- Golias, M., Boile, M., & Theofanis, S. (2006). The berth allocation problem: a formulation reflecting time window service deadlines. In *Proceedings of the 48th Transportation Research*.
- Golias, M., Boile, M., & Theofanis, S. (2009). Berth scheduling by customer service differentiation: A multi-objective approach. *Transportation Research Part E: Logistics and Transportation Review*, 45(6), 878–892. <http://doi.org/10.1016/j.tre.2009.05.006>
- Golias, M., Boile, M., & Theofanis, S. (2010). A lamda-optimal based heuristic for the berth scheduling problem. *Transportation Research Part C: Emerging Technologies*, 18(5), 794–806. <http://doi.org/10.1016/j.trc.2009.07.001>
- Golias, M., Portal, I., Konur, D., Kaisar, E., & Kolomvos, G. (2014). Robust berth scheduling at marine container terminals via hierarchical optimization. *Computers and Operations Research*, 41(2014), 412–422. <http://doi.org/10.1016/j.cor.2013.07.018>
- Guan, Y., & Cheung, R. K. (2004). The berth allocation problem: models and solution methods. *OR Spectrum*, 26(1), 75–92. <http://doi.org/10.1007/s00291-003-0140-8>
- 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. [http://doi.org/10.1016/S0167-6377\(02\)00147-5](http://doi.org/10.1016/S0167-6377(02)00147-5)
- Guldogan, E. U., Bulut, O., & Tasgetiren, M. F. (2011). A dynamic berth allocation problem with priority considerations under stochastic nature. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6839 LNAI, 74–82. http://doi.org/10.1007/978-3-642-25944-9_10
- Han, X. Le, Lu, Z. Q., & Xi, L. F. (2010). A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. *European Journal of Operational Research*, 207(3), 1327–1340. <http://doi.org/10.1016/j.ejor.2010.07.018>

- He, J. (2016). Berth allocation and quay crane assignment in a container terminal for the trade-off between time-saving and energy-saving. *Advanced Engineering Informatics*, 30(3), 390–405. <http://doi.org/10.1016/j.aei.2016.04.006>
- He, J., Huang, Y., Chang, D., & Zhang, W. (2013). A Knowledge-based System for Berth Allocation in a Container Terminal. *Indonesian Journal of Electrical Engineering and Computer Science*, 11(5), 2291–2300.
- He, J., Mi, W., Chang, D., & Yan, W. (2009). An investigation into berth allocation and quay crane assignment based on hybrid parallel genetic algorithm. *2009 International Conference on Artificial Intelligence and Computational Intelligence, AICI 2009*, 2, 48–53. <http://doi.org/10.1109/AICI.2009.41>
- Hendriks, M., Laumanns, M., Lefeber, E., & Udding, J. T. (2010). Robust cyclic berth planning of container vessels. *OR Spectrum*, 32(3), 501–517. <http://doi.org/10.1007/s00291-010-0198-z>
- Hendriks, M. P. M., Armbruster, D., Laumanns, M., Lefeber, E., & Udding, J. T. (2012). Strategic allocation of cyclically calling vessels for multi-terminal container operators. *Flexible Services and Manufacturing Journal*, 24(3), 248–273. <http://doi.org/10.1007/s10696-011-9120-5>
- Hendriks, M. P. M., Lefeber, E., & Udding, J. T. (2013). Simultaneous berth allocation and yard planning at tactical level. *OR Spectrum*, 35(2), 441–456. <http://doi.org/10.1007/s00291-012-0305-4>
- Henesey, L. E. (2006). *Multi-Agent Systems for Container Terminal Management*. Blekinge Institute of Technology.
- Hsu, H.-P. (2015). A HPSO for solving dynamic and discrete berth allocation problem and dynamic quay crane assignment problem simultaneously. *Swarm and Evolutionary Computation*, 1–13. <http://doi.org/10.1016/j.swevo.2015.11.002>
- Hsu, H.-P., Wang, C.-N., Chou, C.-C., Lee, Y., & Wen, Y.-F. (2017). Modeling and Solving the Three Seaside Operational Problems Using an Object-Oriented and Timed Predicate/Transition Net. *Applied Sciences*, 7(3), 218. <http://doi.org/10.3390/app7030218>
- Hu, Q.-M., Hu, Z.-H., & Du, Y. (2014). Berth and quay-crane allocation problem considering fuel consumption and emissions from vessels. *Computers & Industrial Engineering*, 70(2014), 1–10. <http://doi.org/10.1016/j.cie.2014.01.003>
- Hu, Z. (2015). Heuristics for solving continuous berth allocation problem considering periodic balancing utilization of cranes. *Computers & Industrial Engineering*, 85, 216–226. <http://doi.org/10.1016/j.cie.2015.03.017>
- Hu, Z. a. (2010). A quay crane assignment approach for berth allocation problem in container terminal. *3rd International Symposium on Intelligent Information Technology and Security Informatics, IITSI 2010*, 17–21. <http://doi.org/10.1109/IITSI.2010.97>
- Hu, Z. H., Han, X. L., & Ding, Y. Z. (2009). Optimizing continuous berth allocation by immune algorithm. In *Proceedings - 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2009* (Vol. 3, pp. 631–635). <http://doi.org/10.1109/ICICISYS.2009.5358083>
- Huang, K., Suprayogi, & Ariantini. (2016). A continuous berth template design model with multiple wharfs. *Maritime Policy and Management*, 43(6), 763–775. <http://doi.org/10.1080/03088839.2016.1169449>
- Hwan Kim, K., & Bae Kim, H. (1999). Segregating space allocation models for container inventories in port container terminals. *International Journal of Production Economics*, 59(1), 415–423. [http://doi.org/10.1016/S0925-5273\(98\)00028-0](http://doi.org/10.1016/S0925-5273(98)00028-0)
- IBM ILOG. (2014). CPLEX User's Manual, Version 12 Release 6. *International Business Machines Corporation*.
- Imai, A., Chen, H. C., Nishimura, E., & Papadimitriou, S. (2008). The simultaneous berth and quay crane allocation problem. *Transportation Research Part E: Logistics and Transportation*

- Review*, 44(5), 900–920. <http://doi.org/10.1016/j.tre.2007.03.003>
- Imai, A., Nagaiwa, K., & Tat, C. W. (1997). Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation*, 31(1), 75–94. <http://doi.org/10.1002/atr.5670310107>
- Imai, A., Nishimura, E., & Papadimitriou, S. (2001). The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, 35(4), 401–417. [http://doi.org/10.1016/S0191-2615\(99\)00057-0](http://doi.org/10.1016/S0191-2615(99)00057-0)
- Imai, A., Nishimura, E., & Papadimitriou, S. (2003). Berth allocation with service priority. *Transportation Research Part B: Methodological*, 37(5), 437–457. [http://doi.org/10.1016/S0191-2615\(02\)00023-1](http://doi.org/10.1016/S0191-2615(02)00023-1)
- Imai, A., Nishimura, E., & Papadimitriou, S. (2008). Berthing ships at a multi-user container terminal with a limited quay capacity. *Transportation Research Part E: Logistics and Transportation Review*, 44(1), 136–151. <http://doi.org/10.1016/j.tre.2006.05.002>
- Imai, A., Nishimura, E., & Papadimitriou, S. (2013). Marine container terminal configurations for efficient handling of mega-containerships. *Transportation Research Part E: Logistics and Transportation Review*, 49(1), 141–158. <http://doi.org/10.1016/j.tre.2012.07.006>
- Imai, A., Sasaki, K., Nishimura, E., & Papadimitriou, S. (2006). Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks. *European Journal of Operational Research*, 171(2), 373–389. <http://doi.org/10.1016/j.ejor.2004.07.066>
- Imai, A., Sun, X., Nishimura, E., & Papadimitriou, S. (2005). Berth allocation in a container port: Using a continuous location space approach. *Transportation Research Part B: Methodological*, 39(3), 199–221. <http://doi.org/10.1016/j.trb.2004.04.004>
- Imai, A., Yamakawa, Y., & Huang, K. (2014). The strategic berth template problem. *Transportation Research Part E: Logistics and Transportation Review*, 72, 77–100. <http://doi.org/10.1016/j.tre.2014.09.013>
- Iris, C., & Lam, J. S. L. (2017). Models for continuous berth allocation and quay crane assignment: Computational comparison. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 374–378). IEEE. <http://doi.org/10.1109/IEEM.2017.8289915>
- Iris, C., Larsen, A., Røpke, S., & Pacino, D. (2016). *Exact and Heuristic Methods for Integrated Container Terminal Problems*. DTU Management Engineering.
- Iris, C., & Pacino, D. (2015). A Survey on the Ship Loading Problem. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9335, pp. 238–251). http://doi.org/10.1007/978-3-319-24264-4_17
- Iris, Ç., Pacino, D., & Ropke, S. (2017). Improved formulations and an Adaptive Large Neighborhood Search heuristic for the integrated berth allocation and quay crane assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, 105, 123–147. <http://doi.org/10.1016/j.tre.2017.06.013>
- Ji, M., Zhu, H., Wang, Q., Zhao, R., & Yang, Y. (2015). Integrated strategy for berth allocation and crane assignment on a continuous berth using Monte Carlo simulation. *Simulation*, 91(1), 26–42. <http://doi.org/10.1177/0037549714558128>
- Jin, J. G., Lee, D.-H., & Hu, H. (2015). Tactical berth and yard template design at container transshipment terminals: A column generation based approach. *Transportation Research Part E: Logistics and Transportation Review*, 73, 168–184. <http://doi.org/10.1016/j.tre.2014.11.009>
- Kalmar global. (2018). Kalmar Equipment. Retrieved June 8, 2018, from <https://www.kalmarglobal.com/>
- Karam, A., & Eltawil, A. B. (2015). A new method for allocating berths, quay cranes and internal trucks in container terminals. *2015 International Conference on Logistics, Informatics and*

- Service Science, LISS 2015*, 31–36. <http://doi.org/10.1109/LISS.2015.7369644>
- Karam, A., ElTawil, A. B., & Harraz, N. A. (2014). An improved solution for integrated berth allocation and quay crane assignment problem in container terminals. *CIE44 & IMSS'14 Proceedings, 14-16 October 2014, Istanbul / Turkey*, (1), 2057–2070. <http://doi.org/10.13140/RG.2.1.5041.6164>
- Kim, K. H., Kang, J. S., & Ryu, K. R. (2004). A beam search algorithm for the load sequencing of outbound containers in port container terminals. *OR Spectrum*, 26(1), 93–116. <http://doi.org/10.1007/s00291-003-0148-0>
- Kim, K. H., & Kim, K. Y. (1999a). An Optimal Routing Algorithm for a Transfer Crane in Port Container Terminals. *Transportation Science*, 33(1), 17–33. <http://doi.org/10.1287/trsc.33.1.17>
- Kim, K. H., & Kim, K. Y. (1999b). Routing straddle carriers for the loading operation of containers using a beam search algorithm. *Computers and Industrial Engineering*, 36(1), 109–136. <http://doi.org/10.1007/s00291-004-0194-2>
- Kim, K. H., & Moon, K. C. (2003). Berth scheduling by simulated annealing. *Transportation Research Part B: Methodological*, 37(6), 541–560. [http://doi.org/10.1016/S0191-2615\(02\)00027-9](http://doi.org/10.1016/S0191-2615(02)00027-9)
- Kim, K. H., & Park, Y. M. (2004). A crane scheduling method for port container terminals. *European Journal of Operational Research*, 156(3), 752–768. [http://doi.org/10.1016/S0377-2217\(03\)00133-4](http://doi.org/10.1016/S0377-2217(03)00133-4)
- Kim, K. H., Park, Y. M., & Ryu, K.-R. (2000). Deriving decision rules to locate export containers in container yards. *European Journal of Operational Research*, 124(1), 89–101. [http://doi.org/10.1016/S0377-2217\(99\)00116-2](http://doi.org/10.1016/S0377-2217(99)00116-2)
- Kordić, S., Davidović, T., Kovač, N., & Dragović, B. (2016). Combinatorial approach to exactly solving discrete and hybrid berth allocation problem. *Applied Mathematical Modelling*, 0, 1–22. <http://doi.org/10.1016/j.apm.2016.05.004>
- Koza, J. R. (1992). *Genetic programming : on the programming of computers by means of natural selection*. Massachusetts Institute of Technology. Cambridge, Mass. ; London: MIT Press.
- Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4(2), 87–112. <http://doi.org/10.1007/BF00175355>
- Krimi, I., Benmansour, R., el Cadi, A. A., Duvivier, D., Elhachemi, N., Deshayes, L., & Ouarrasse, A. A. (2018). The integrated multi-quay Berth Allocation and Crane Assignment Problem: Formulation and case study. In *2018 7th International Conference on Industrial Technology and Management (ICITM)* (pp. 157–161). IEEE. <http://doi.org/10.1109/ICITM.2018.8333938>
- Lalla-Ruiz, E., González-Velarde, J. L., Melián-Batista, B., & Moreno-Vega, J. M. (2014). Biased random key genetic algorithm for the Tactical Berth Allocation Problem. *Applied Soft Computing*, 22, 60–76. <http://doi.org/10.1016/j.asoc.2014.04.035>
- Lalla-Ruiz, E., Melián-Batista, B., & Marcos Moreno-Vega, J. (2012). Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. *Engineering Applications of Artificial Intelligence*, 25(6), 1132–1141. <http://doi.org/10.1016/j.engappai.2012.06.001>
- Langdon, W. B., Poli, R., McPhee, N. F., & Koza, J. R. (2008). Genetic programming: An introduction and tutorial, with a survey of techniques and applications. *Studies in Computational Intelligence*, 115, 927–1028. http://doi.org/10.1007/978-3-540-78293-3_22
- Le, M., Wu, C., & Zhang, H. (2012). An integrated optimization method to solve the Berth-QC Allocation Problem. In *Proceedings - International Conference on Natural Computation* (pp. 753–757). <http://doi.org/10.1109/ICNC.2012.6234544>
- Lee, D. H., Chen, J. H., & Cao, J. X. (2010). The continuous Berth Allocation Problem: A Greedy Randomized Adaptive Search Solution. *Transportation Research Part E: Logistics and Transportation Review*, 46(6), 1017–1029. <http://doi.org/10.1016/j.tre.2010.01.009>

- Lee, D. H., & Jin, J. G. (2013). Feeder vessel management at container transshipment terminals. *Transportation Research Part E: Logistics and Transportation Review*, 49(1), 201–216. <http://doi.org/10.1016/j.tre.2012.08.006>
- Lee, D. H., Jin, J. G., & Chen, J. H. (2012). Terminal and yard allocation problem for a container transshipment hub with multiple terminals. *Transportation Research Part E: Logistics and Transportation Review*, 48(2), 516–528. <http://doi.org/10.1016/j.tre.2011.09.004>
- Lee, D. H., Wang, H. Q., & Miao, L. (2008). Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 44(1), 124–135. <http://doi.org/10.1016/j.tre.2006.08.001>
- Lee, L. H., Chew, E. P., Tan, K. C., & Han, Y. (2006). An optimization model for storage yard management in transshipment hubs. *OR Spectrum*, 28(4), 539–561. <http://doi.org/10.1007/s00291-006-0045-4>
- Lee, Y., & Chen, C. Y. (2009). An optimization heuristic for the berth scheduling problem. *European Journal of Operational Research*, 196(2), 500–508. <http://doi.org/10.1016/j.ejor.2008.03.021>
- Lee, Y., & Hsu, N.-Y. (2007). An optimization model for the container pre-marshalling problem. *Computers & Operations Research*, 34(11), 3295–3313. <http://doi.org/10.1016/j.cor.2005.12.006>
- Legato, P., Gulli, D., & Trunfio, R. (2008). The quay crane deployment problem at a maritime container terminal. In *Proceedings 22nd European Conference on Modelling and Simulation ©ECMS Loucas* (pp. 53–63).
- Legato, P., & Mazza, R. M. (2001). Berth planning and resources optimisation at a container terminal via discrete event simulation. *European Journal of Operational Research*, 133(3), 537–547. [http://doi.org/10.1016/S0377-2217\(00\)00200-9](http://doi.org/10.1016/S0377-2217(00)00200-9)
- Li, F., Sheu, J.-B., & Gao, Z.-Y. (2015). Solving the Continuous Berth Allocation and Specific Quay Crane Assignment Problems with Quay Crane Coverage Range. *Transportation Science*, 49(4), 968–989. <http://doi.org/10.1287/trsc.2015.0619>
- Li, M. Z., Jin, J. G., & Lu, C. X. (2015). Real-Time Disruption Recovery for Integrated Berth Allocation and Crane Assignment in Container Terminals. *Transportation Research Record: Journal of the Transportation Research Board*, 2479, 49–59. <http://doi.org/10.3141/2479-07>
- Li, W., Wu, Y., & Goh, M. (2015). *Planning and Scheduling for Maritime Container Yards*. Springer. Cham: Springer International Publishing. <http://doi.org/10.1007/978-3-319-17025-1>
- Liang, C., Guo, J., & Yang, Y. (2011). Multi-objective hybrid genetic algorithm for quay crane dynamic assignment in berth allocation planning. *Journal of Intelligent Manufacturing*, 22(3), 471–479. <http://doi.org/10.1007/s10845-009-0304-8>
- Liang, C., Huang, Y., & Yang, Y. (2009). A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Computers & Industrial Engineering*, 56(3), 1021–1028. <http://doi.org/10.1016/j.cie.2008.09.024>
- Liang, C., Hwang, H., & Gen, M. (2012). A berth allocation planning problem with direct transshipment consideration. *Journal of Intelligent Manufacturing*, 23(6), 2207–2214. <http://doi.org/10.1007/s10845-011-0566-9>
- Liang, C., Lin, N. A., & Jo, J. (2009). Multiobjective hybrid genetic algorithm for quay crane scheduling in berth allocation planning. *International Journal of Manufacturing Technology and Management*, 16(1/2), 127. <http://doi.org/10.1504/IJMTM.2009.021506>
- Liang, X., Li, W., Zhao, W., & Li, B. (2012). Multistage collaborative scheduling of berth and quay crane based on heuristic strategies and particle swarm optimization. In *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 913–918). IEEE. <http://doi.org/10.1109/CSCWD.2012.6221930>
- Lim, A. (1998). The berth planning problem. *Operations Research Letters*, 22(April 1996), 105–

110. [http://doi.org/10.1016/S0167-6377\(98\)00010-8](http://doi.org/10.1016/S0167-6377(98)00010-8)
- Lin, S.-W., & Ting, C.-J. (2014). Solving the dynamic berth allocation problem by simulated annealing. *Engineering Optimization*, 46(3), 308–327. <http://doi.org/10.1080/0305215X.2013.768241>
- Lin, S.-W., Ting, C.-J., & Wu, K.-C. (2017). Simulated annealing with different vessel assignment strategies for the continuous berth allocation problem. *Flexible Services and Manufacturing Journal*, 1–24. <http://doi.org/10.1007/s10696-017-9298-2>
- Liu, J., Wan, Y., & Wang, L. (2006). Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. *Naval Research Logistics*, 53(1), 60–74. <http://doi.org/10.1002/nav.20108>
- Liu, Y., Wang, J., & Shahbazzade, S. (2018). The improved AFSA algorithm for the berth allocation and quay crane assignment problem. *Cluster Computing*, 3456789. <http://doi.org/10.1007/s10586-018-2216-x>
- Lokuge, P., & Alahakoon, D. (2007). Improving the adaptability in automated vessel scheduling in container ports using intelligent software agents. *European Journal of Operational Research*, 177(3), 1985–2015. <http://doi.org/10.1016/j.ejor.2005.12.016>
- Lu, Z., Han, X., & Xi, L. (2011). Simultaneous berth and quay crane allocation problem in container terminal. *Advanced Science Letters*, 4(6–7), 2113–2118. <http://doi.org/10.1166/asl.2011.1533>
- Ma, H., Chan, F. T., & Chung, S. (2014). A fast approach for the integrated berth allocation and quay crane assignment problem. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. <http://doi.org/10.1177/0954405414544555>
- Ma, H. L., Chan, F. T. S., Chung, S. H., & Wong, C. S. (2011). Maximizing the reliability of terminal service by vessel scheduling and quay crane assignment. In *2011 IEEE International Conference on Quality and Reliability* (pp. 85–89). IEEE. <http://doi.org/10.1109/ICQR.2011.6031686>
- Maarten Hendriks, Marco Laumanns, Erjen Lefebber, J. T. U. (2008). Robust periodic berth planning of container vessels. *Proceedings of the 3rd German-Korean Workshop on Container Terminal Management : IT-Based Planning and Control of Seaport Container Terminals and Transportation Systems*, 1–13.
- Meisel, F. (2009a). Maritime Container Transport (pp. 5–15). Heidelberg: Physica-Verlag HD. http://doi.org/10.1007/978-3-7908-2191-8_2
- Meisel, F. (2009b). *Seaside Operations Planning in Container Terminals*. Heidelberg: Physica-Verlag HD. <http://doi.org/10.1007/978-3-7908-2191-8>
- Meisel, F. (2011). Scheduling seaside resources at container ports. *Wiley Encyclopedia of Operations Research and Management Science*. <http://doi.org/10.1002/9780470400531.eorms0189>
- Meisel, F., & Bierwirth, C. (2006). Integration of berth allocation and crane assignment to improve the resource utilization at a seaport container terminal. *Operations Research Proceedings 2005*.
- Meisel, F., & Bierwirth, C. (2009). Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, 45(1), 196–209. <http://doi.org/10.1016/j.tre.2008.03.001>
- Meisel, F., & Bierwirth, C. (2013). A Framework for Integrated Berth Allocation and Crane Operations Planning in Seaport Container Terminals. *Transportation Science*, 47(2), 131–147. <http://doi.org/10.1287/trsc.1120.0419>
- Melián-Batista, B., Expósito-Izquierdo, C., Lalla-Ruiz, E., Lamata, M. T., & Moreno-Vega, J. M. (2013). Fuzzy optimization models for seaside port logistics. *IJCCI 2013 - Proceedings of the 5th International Joint Conference on Computational Intelligence*, 289–299.
- Moccia, L., & Astorino, A. (2007). The group allocation problem in a transshipment container

- terminal. *Proceedings of World Conference on Transport*, 1–13.
- Monaco, M. F., & Sammarra, M. (2007). The Berth Allocation Problem: A Strong Formulation Solved by a Lagrangean Approach. *Transportation Science*, 41(2), 265–280. <http://doi.org/10.1287/trsc.1060.0171>
- Moon, K. (2000). A mathematical model and a heuristic algorithm for berth planning. *Brain Korea 21 Logistics Team*.
- Moorthy, R., & Teo, C. P. (2007). Berth management in container terminal: The template design problem. *Container Terminals and Cargo Systems: Design, Operations Management, and Logistics Control Issues*, 28(4), 63–86. http://doi.org/10.1007/978-3-540-49550-5_4
- Murty, K. G., Liu, J., Wan, Y. W., & Linn, R. (2005). A decision support system for operations in a container terminal. *Decision Support Systems*, 39(3), 309–332. <http://doi.org/10.1016/j.dss.2003.11.002>
- Na, L., & Zhihong, J. (2009). Optimization of continuous berth and quay crane allocation problem in seaport container terminal. *2009 2nd International Conference on Intelligent Computing Technology and Automation, ICICTA 2009*, 3, 229–233. <http://doi.org/10.1109/ICICTA.2009.522>
- Nguyen, S., Zhang, M., Johnston, M., & Chen Tan, K. (2013). Hybrid evolutionary computation methods for quay crane scheduling problems. *Computers and Operations Research*, 40(8), 2083–2093. <http://doi.org/10.1016/j.cor.2013.03.007>
- Pan, J., & Xu, Y. (2015). Online Integrated Allocation of Berths and Quay Cranes in Container Terminals with 1-Lookahead. *Computing and Combinatorics*.
- Park, K. T., & Kim, K. H. (2002). Berth scheduling for container terminals by using a sub-gradient optimization technique. *Journal of the Operational Research Society*, 53(9), 1054–1062. <http://doi.org/10.1057/palgrave.jors.2601412>
- Park, Y.-M., & Kim, K. H. (2003). A scheduling method for Berth and Quay cranes. *OR Spectrum*, 25(1), 1–23. <http://doi.org/10.1007/s00291-002-0109-z>
- Pinedo, M. L. (2012). *Scheduling: Theory, Algorithms, and Systems*. Springer (Vol. 4). <http://doi.org/10.1007/s13398-014-0173-7.2>
- Raa, B., Dullaert, W., & Schaeren, R. Van. (2011). An enriched model for the integrated berth allocation and quay crane assignment problem. *Expert Systems with Applications*, 38(11), 14136–14147. <http://doi.org/10.1016/j.eswa.2011.04.224>
- Rashidi, H., & Tsang, E. P. K. (2013). Novel constraints satisfaction models for optimization problems in container terminals. *Applied Mathematical Modelling*, 37(6), 3601–3634. <http://doi.org/10.1016/j.apm.2012.07.042>
- Rodriguez-Molins, M., Barber, F., Sierra, M. R., Puente, J., & Salido, M. A. (2012). A Genetic Algorithm for Berth Allocation and Quay Crane Assignment. In *Advances in Artificial Intelligence – IBERAMIA 2012* (pp. 601–610). Springer. http://doi.org/10.1007/978-3-642-34654-5_61
- Rodriguez-Molins, M., Ingolotti, L., Barber, F., Salido, M. a., Sierra, M. R., & Puente, J. (2014). A genetic algorithm for robust berth allocation and quay crane assignment. *Progress in Artificial Intelligence*. <http://doi.org/10.1007/s13748-014-0056-3>
- Rodriguez-Molins, M., Salido, M. a., & Barber, F. (2014). A GRASP-based metaheuristic for the Berth Allocation Problem and the Quay Crane Assignment Problem by managing vessel cargo holds. *Applied Intelligence*, 40(2), 273–290. <http://doi.org/10.1007/s10489-013-0462-4>
- Rodriguez-Molins, M., Salido, M. A., & Barber, F. (2014). Robust scheduling for Berth Allocation and Quay Crane Assignment Problem. *Mathematical Problems in Engineering*, (2014).
- Ruiz, E., Batista, B., & Vega, J. (2013). Adaptive Variable Neighbourhood Search for Berth Planning in Maritime Container Terminals. In *GREEN-COPLAS 2013: IJCAI 2013 Workshop on Constraint Reasoning, Planning and Scheduling Problems for a Sustainable Future*.

- Ruiz, E. L., Izquierdo, C. E., Batista, B. M., & Moreno-Vega, J. M. (2013). A Metaheuristic Approach for the Seaside Operations in Maritime Container Terminals. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 7903 LNCS, pp. 28–35). http://doi.org/10.1007/978-3-642-38682-4_4
- Rushton, A., Croucher, P., & Baker, P. (2010). *The Handbook of Logistics and Distribution Management* (4th Editio). Kogan Page.
- Saharidis, G. K. D., Golias, M., Boile, M., Theofanis, S., & Ierapetritou, M. G. (2010). The berth scheduling problem with customer differentiation: a new methodological approach based on hierarchical optimization. *The International Journal of Advanced Manufacturing Technology*, 46(1–4), 377–393. <http://doi.org/10.1007/s00170-009-2068-x>
- Said, G. A. E.-N. a., & El-Horbaty, E.-S. M. (2015). An Optimization Methodology for Container Handling Using Genetic Algorithm. *Procedia Computer Science*, 65(Iccmit), 662–671. <http://doi.org/10.1016/j.procs.2015.09.010>
- Salhi, A., Alsoufi, G., & Yang, X. (2017). An evolutionary approach to a combined mixed integer programming model of seaside operations as arise in container ports. *Annals of Operations Research*. <http://doi.org/10.1007/s10479-017-2539-7>
- Salido, M. a., Rodriguez-Molins, M., & Barber, F. (2011). Integrated intelligent techniques for remarshaling and berthing in maritime terminals. *Advanced Engineering Informatics*, 25(3), 435–451. <http://doi.org/10.1016/j.aei.2010.10.001>
- Salido, M. a., Rodriguez-Molins, M., & Barber, F. (2012). A decision support system for managing combinatorial problems in container terminals. *Knowledge-Based Systems*, 29(2012), 63–74. <http://doi.org/10.1016/j.knosys.2011.06.021>
- Saxon, S., & Stone, M. (2017). *Container shipping : The next 50 years*. McKinsey and Company.
- Sciomachen, A., & Tanfani, E. (2007). A 3D-BPP approach for optimising stowage plans and terminal productivity. *European Journal of Operational Research*, 183(3), 1433–1446. <http://doi.org/10.1016/j.ejor.2005.11.067>
- Shang, X. T., Cao, J. X., & Ren, J. (2016). A robust optimization approach to the integrated berth allocation and quay crane assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, 94, 44–65. <http://doi.org/10.1016/j.tre.2016.06.011>
- Sheikholeslami, A., Ilati, G., & Hassannayebi, E. (2013). A Simulation Model for the Problem in Integrated Berth Allocation and Quay Crane Assignment. *Journal of Basic and Applied Scientific Research*, 3(2013), 343–354.
- Shen, S. Y., & Ko, F. S. (2013). A study of modeling the simultaneous berth and crane allocation problem. In *Lecture Notes in Engineering and Computer Science* (Vol. 2, pp. 1100–1105).
- Song, L., Cherrett, T., & Guan, W. (2012). Study on berth planning problem in a container seaport: Using an integrated programming approach. *Computers and Industrial Engineering*, 62(1), 119–128. <http://doi.org/10.1016/j.cie.2011.08.024>
- Stahlbock, R., & Voß, S. (2008). Operations research at container terminals: A literature update. *OR Spectrum*, 30(1), 1–52. <http://doi.org/10.1007/s00291-007-0100-9>
- Tang, L., & Dai, L. (2008). Berth allocation with service priority for container terminal of hub port. In *2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008*. <http://doi.org/10.1109/WiCom.2008.1988>
- The Journal of Commerce. (2016). Top 50 Container Ports in 2016. Retrieved June 4, 2018, from https://www.joc.com/port-news/top-50-container-ports-2016-shanghai-tightens-grip-crown_20170810.html
- Theodorou, E., & Diabat, A. (2015). A joint quay crane assignment and scheduling problem: formulation, solution algorithm and computational results. *Optimization Letters*, 9(4), 799–817. <http://doi.org/10.1007/s11590-014-0787-x>

- Theofanis, S., Boile, M., & Golias, M. (2009). Container Terminal Berth Planning. *Transportation Research Record: Journal of the Transportation Research Board*, 2100(1), 22–28. <http://doi.org/10.3141/2100-03>
- Ting, C., Lin, S., & Wu, K. (2013). The Continuous Berth Allocation Problem by Simulated Annealing. In *APIEMS Conference*.
- Türkoğulları, Y. B., Taşkın, Z. C., Aras, N., & Altınel, İ. K. (2014). Optimal Berth Allocation, Time-variant Quay Crane Assignment and Scheduling in Container Terminals. *European Journal of Operational Research*, 235(1), 88–101. <http://doi.org/10.1016/j.ejor.2013.10.015>
- Türkoğulları, Y. B., Taşkın, Z. C., Aras, N., & Altınel, İ. K. (2014). Optimal berth allocation and time-invariant quay crane assignment in container terminals. *European Journal of Operational Research*, 235(1), 88–101. <http://doi.org/10.1016/j.ejor.2013.10.015>
- Türkoğulları, Y. B., Taşkın, Z. C., Aras, N., & Altınel, İ. K. (2016). Optimal berth allocation, time-variant quay crane assignment and scheduling with crane setups in container terminals. *European Journal of Operational Research*, 254(3), 985–1001. <http://doi.org/10.1016/j.ejor.2016.04.022>
- UNCTAD. (2013). Review of Maritime Transport 2013. *United Nations*.
- UNCTAD. (2016). Review of Maritime Transport 2016. *United Nations*.
- Ursavas, E. (2014). A decision support system for quayside operations in a container terminal. *Decision Support Systems*, 59(1), 312–324. <http://doi.org/10.1016/j.dss.2014.01.003>
- Ursavas, E. (2015). Priority control of berth allocation problem in container terminals. *Annals of Operations Research*. <http://doi.org/10.1007/s10479-015-1912-7>
- Ursavas, E., & Zhu, S. X. (2016). Optimal Policies for the Berth Allocation Problem under Stochastic Nature. *European Journal of Operational Research*. <http://doi.org/10.1016/j.ejor.2016.04.029>
- Vacca, I., Salani, M., & Bierlaire, M. (2010). The Tactical Berth Allocation Problem Integrated optimization in container terminals. *Seminar Series*, 1–26.
- Vacca, I., Salani, M., & Bierlaire, M. (2013). An Exact Algorithm for the Integrated Planning of Berth Allocation and Quay Crane Assignment. *Transportation Science*, 47(2), 148–161. <http://doi.org/10.1287/trsc.1120.0428>
- Vis, I. F. A. (2006a). A comparative analysis of storage and retrieval equipment at a container terminal. *International Journal of Production Economics*, 103(2), 680–693. <http://doi.org/10.1016/j.ijpe.2006.01.002>
- Vis, I. F. A. (2006b). Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, 170(3), 677–709. <http://doi.org/10.1016/j.ejor.2004.09.020>
- Vis, I. F. A. (2009). Decision problems at container terminals. Retrieved June 10, 2018, from <http://www.irisvis.nl/container/>
- Vis, I. F. A., & de Koster, R. (2003). Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147(1), 1–16. [http://doi.org/10.1016/S0377-2217\(02\)00293-X](http://doi.org/10.1016/S0377-2217(02)00293-X)
- Vis, I. F. A., de Koster, R. (M. . B. M., & Savelsbergh, M. W. P. (2005). Minimum Vehicle Fleet Size Under Time-Window Constraints at a Container Terminal. *Transportation Science*, 39(2), 249–260. <http://doi.org/10.1287/trsc.1030.0063>
- Vis, I. F. A., de Koster, R., Roodbergen, K. J., & Peeters, L. W. P. (2001). Determination of the number of AGVs required at a semi-automated container terminal. *Journal of the Operational Research Society*, 52, 409–417.
- Vis, I. F. A., & Harika, I. (2004). Comparison of vehicle types at an automated container terminal. *OR Spectrum*, 26(1), 117–143. <http://doi.org/10.1007/s00291-003-0146-2>

- Voß, S., Stahlbock, R., & Steenken, D. (2004). Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, 26(1), 3–49. <http://doi.org/10.1007/s00291-003-0157-z>
- Wang, F., & Lim, A. (2007). A stochastic beam search for the berth allocation problem. *Decision Support Systems*, 42(4), 2186–2196. <http://doi.org/10.1016/j.dss.2006.06.008>
- Wang, K., Zhen, L., Wang, S., & Laporte, G. (2018). Column Generation for the Integrated Berth Allocation, Quay Crane Assignment, and Yard Assignment Problem. *Transportation Science*, (June), trsc.2018.0822. <http://doi.org/10.1287/trsc.2018.0822>
- Wiese, J., Kliewer, N., Suhl, L., & Str, W. (2009). *A Survey of Container Terminal Characteristics and Equipment Types*. *Decision Support & Operations Research Lab*.
- Worldshipping. (2014). CONTAINERS. Retrieved August 1, 2014, from <http://www.worldshipping.org/about-the-industry/containers>
- Wu, Y., Li, W., Petering, M. E. H., Goh, M., & Souza, R. de. (2015). Scheduling Multiple Yard Cranes with Crane Interference and Safety Distance Requirement. *Transportation Science*, 49(4), 990–1005. <http://doi.org/10.1287/trsc.2015.0641>
- Xiao, L., & Hu, Z. H. (2014). Berth allocation problem with quay crane assignment for container terminals based on rolling-horizon strategy. *Mathematical Problems in Engineering*, 2014, 1–11. <http://doi.org/10.1155/2014/845752>
- Xiaotao, J., Yuquan, D., & Qiushuang, C. (2011). A multi-objective genetic algorithm for berth allocation and quay crane assignment problem. In *2011 Chinese Control and Decision Conference (CCDC)* (pp. 891–896). IEEE. <http://doi.org/10.1109/CCDC.2011.5968309>
- Xu, D., Li, C. L., & Leung, J. Y. T. (2012). Berth allocation with time-dependent physical limitations on vessels. *European Journal of Operational Research*, 216(1), 47–56. <http://doi.org/10.1016/j.ejor.2011.07.012>
- Xu, Y., Chen, Q., & Quan, X. (2012). Robust berth scheduling with uncertain vessel delay and handling time. *Annals of Operations Research*, 192(1), 123–140. <http://doi.org/10.1007/s10479-010-0820-0>
- Xu, Z., & Lee, C. (2018). New Lower Bound and Exact Method for the Continuous Berth Allocation Problem. *Operations Research*, (March), opre.2017.1687. <http://doi.org/10.1287/opre.2017.1687>
- Yang, C., Wang, S., & Zheng, J. (2013). The allocation of berth and quay crane by using a particle swarm optimization technique. In *Lecture Notes in Electrical Engineering* (Vol. 254 LNEE, pp. 779–787). http://doi.org/10.1007/978-3-642-38524-7_86
- Yang, C., Wang, X., & Li, Z. (2012). An optimization approach for coupling problem of berth allocation and quay crane assignment in container terminal. *Computers and Industrial Engineering*, 63(1), 243–253. <http://doi.org/10.1016/j.cie.2012.03.004>
- Yang, C. X., Wang, N., & Yang, H. L. (2011). Coupling optimization for berth allocation and quay crane assignment problem in container terminals. *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems, CIMS*, 17, 2270–2277.
- Yang, J., Gao, H., Liu, W., & Liu, T. (2016). A NOVEL STRATEGY FOR BERTH AND QUAY CRANE ALLOCATION UNDER DISRUPTION IN CONTAINER TERMINAL. *International Journal of Innovative Computing*, 12(5), 1535–1551.
- Yuping, W., Yangyang, H., Yuanhui, Z., & Tianyi, G. (2018). Berth Allocation Optimization with Priority based on Simulated Annealing Algorithm. *Journal of Engineering Science and Technology Review*, 11(1), 74–83. <http://doi.org/10.25103/jestr.111.09>
- Zampelli, S., Vergados, Y., Van Schaeren, R., Dullaert, W., & Raa, B. (2013). The berth allocation and quay crane assignment problem using a CP approach. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8124 LNCS, 880–896. http://doi.org/10.1007/978-3-642-40627-0_64

- Zeng, Q., Hu, X., Wang, W., & Fang, Y. (2011). Disruption management model and its algorithms for berth allocation problem in container terminals. *International Journal of Innovative Computing, Information and Control*, 7(May 2011), 2763–2773.
- Zeng, Q., Yang, Z., & Hu, X. (2011). Disruption recovery model for berth and quay crane scheduling in container terminals. *Engineering Optimization*, 43(9), 967–983. <http://doi.org/10.1080/0305215X.2010.528411>
- Zhang, C., Wan, Y., Liu, J., & Linn, R. J. (2002). Dynamic crane deployment in container storage yards. *Transportation Research Part B: Methodological*, 36(6), 537–555. [http://doi.org/10.1016/S0191-2615\(01\)00017-0](http://doi.org/10.1016/S0191-2615(01)00017-0)
- Zhang, C., Wu, T., Qi, M., & Miao, L. (2018). Simultaneous Allocation of Berths and Quay Cranes under Discrete Berth Situation. *Asia-Pacific Journal of Operational Research*, 1850011, 1850011. <http://doi.org/10.1142/S0217595918500112>
- Zhang, C., Zheng, L., Zhang, Z., Shi, L., & Armstrong, A. J. (2010). The allocation of berths and quay cranes by using a sub-gradient optimization technique. *Computers and Industrial Engineering*, 58, 40–50. <http://doi.org/10.1016/j.cie.2009.08.002>
- Zhen, L., & Chang, D. F. (2012). A bi-objective model for robust berth allocation scheduling. *Computers and Industrial Engineering*, 63(1), 262–273. <http://doi.org/10.1016/j.cie.2012.03.003>
- Zhen, L., Chew, E. P., & Lee, L. H. (2011). An Integrated Model for Berth Template and Yard Template Planning in Transshipment Hubs. *Transportation Science*, 43(45(5)), 483–504. <http://doi.org/10.1287/trsc.1090.0306>
- Zhen, L., Liang, Z., Zhuge, D., Lee, L. H., & Chew, E. P. (2017). Daily berth planning in a tidal port with channel flow control. *Transportation Research Part B: Methodological*, 0, 1–25. <http://doi.org/10.1016/j.trb.2017.10.008>
- Zhen, L., Wang, S., & Wang, K. (2016). Terminal allocation problem in a transshipment hub considering bunker consumption. *Naval Research Logistics (NRL)*. <http://doi.org/10.1002/nav.21717>
- Zhou, P., & Kang, H. (2008). Study on Berth and Quay-crane Allocation under Stochastic Environments in Container Terminal. *Systems Engineering - Theory & Practice*, 28(1), 161–169. [http://doi.org/10.1016/S1874-8651\(09\)60001-6](http://doi.org/10.1016/S1874-8651(09)60001-6)
- Zhou, P., Kang, H., & Lin, L. (2006). A Dynamic Berth Allocation Model Based on Stochastic Consideration. In *2006 6th World Congress on Intelligent Control and Automation* (Vol. 2, pp. 7297–7301). IEEE. <http://doi.org/10.1109/WCICA.2006.1714503>

Appendix A

A.1 Sample of the instance produced by Mario Rodriguez-Molins, Ingolotti, et al., 2014 to solve BACAP using BACAP_GP

ID	Length	ETA	m	p	ID	Length	ETA	m	p
1	260	16	416	9	51	102	1030	606	5
2	232	31	968	3	52	369	1058	583	7
3	139	68	364	6	53	183	1145	667	2
4	193	82	761	6	54	136	1161	945	1
5	287	105	686	3	55	235	1175	715	3
6	318	116	630	7	56	117	1208	400	7
7	366	138	811	2	57	319	1241	105	6
8	166	157	156	2	58	210	1246	932	1
9	109	163	783	5	59	374	1253	467	8
10	251	179	222	1	60	292	1256	634	10
11	292	180	740	4	61	285	1263	917	6
12	147	208	921	1	62	352	1299	642	7
13	179	228	913	10	63	72	1312	249	1
14	177	241	392	2	64	150	1341	231	6
15	184	244	475	10	65	235	1352	340	7
16	391	259	407	9	66	93	1372	352	8
17	114	264	468	8	67	267	1423	255	10
18	143	265	924	10	68	283	1432	649	10
19	159	312	633	3	69	206	1462	849	10
20	153	312	444	2	70	75	1499	909	4
21	199	321	846	1	71	273	1513	555	4
22	244	332	282	9	72	272	1518	713	1

ID	Length	ETA	m	p	ID	Length	ETA	m	p
23	304	335	552	1	73	150	1527	614	2
24	87	392	399	8	74	212	1575	534	4
25	337	395	438	10	75	308	1587	616	4
26	304	438	189	5	76	79	1607	743	9
27	283	484	338	1	77	281	1616	297	8
28	180	508	278	4	78	134	1617	902	2
29	233	555	229	3	79	94	1662	487	9
30	89	589	636	1	80	100	1736	704	3
31	94	621	230	2	81	361	1741	254	7
32	137	667	351	2	82	257	1776	419	1
33	252	715	119	8	83	380	1777	380	2
34	107	716	392	9	84	246	1786	326	9
35	285	739	845	2	85	386	1790	358	4
36	349	754	470	5	86	178	1803	851	3
37	117	800	674	1	87	269	1820	125	10
38	77	800	211	6	88	149	1821	970	6
39	333	806	123	9	89	230	1831	415	6
40	352	810	956	5	90	339	1879	441	10
41	270	837	390	3	91	344	1883	274	2
42	78	881	148	8	92	123	1915	670	10
43	398	894	560	6	93	157	1938	671	2
44	108	941	569	9	94	262	1944	109	4
45	253	942	946	1	95	372	1957	293	2
46	74	983	215	10	96	353	1975	320	4
47	295	996	900	3	97	134	1992	544	4
48	324	1000	744	3	98	372	1999	185	7
49	241	1005	424	7	99	251	2032	400	8
50	372	1021	114	10	100	333	2056	233	2

Appendix B

B.1 DP World Sokhna Vessel Map (Baplie file)



DP WORLD
Sokhna
Planning Office

Vessel Map

Vessel Name: (In / Out of) Window LOA: - M

Agent Name: Voyage No: Service:

E.T.A.: E.T.D.:

A.T.A.: A.T.D.:

Berthing Position:

Bollard from To

Particulars:-
(Cranes, IMO's, Off standards, Cut & Run Time ...etc.)

Discharge	
Restows x 2	
Hatch Cover x 2	
Gear Boxes x 2	
Load	
Total Moves	

Satus	Full			MTY			Total
	20'	40'	Ref.	20'	40'	Ref.	
Discharge							
Load							
Total							

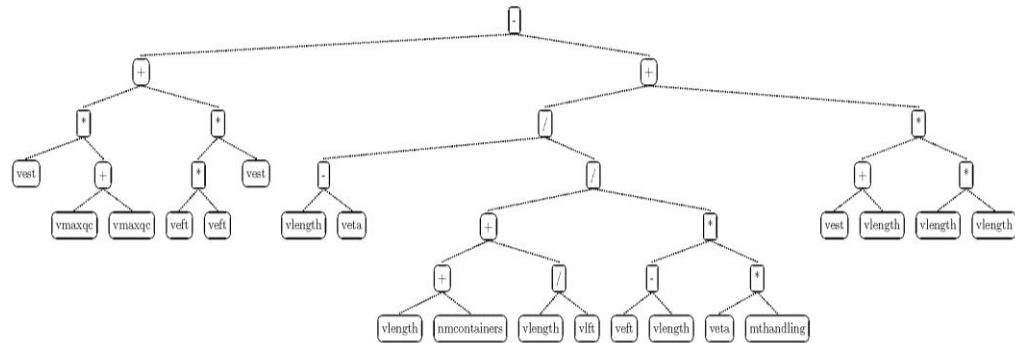
Planning Check list:

Received Baplie Recap and Baplie matching
 Checked Manifests in System Restows processed in Sparcs
 Checked with Precheck all certificates 'group codes' in System
 Received '10%-List' Discharge planned to the yard
 Received Load file 'Stiff / Movin ' Recap and Load file are matching
 Load instructions processed and matching containers at Terminal
 Crane split optimized and discussed with Operations Supervisor
 Checked with Technical Dept. all equipment ready to be used

Started Operations at Finished Operations at
 Vessel departed at

Pre-Stow plan and crane split to Ship "before start loading"
 Final Stow plan to Ship Floppy to Ship "before departure"
 Productivity reports all shifts and Vessel Operation report received
 Final file to Agent
 Vessel Reports "Performance Report, Gantry Crane Productivity, Accountant Report and Vessel's"

B.2 Sample of CDR created by BACAP_GP_DP to solve instance number 1

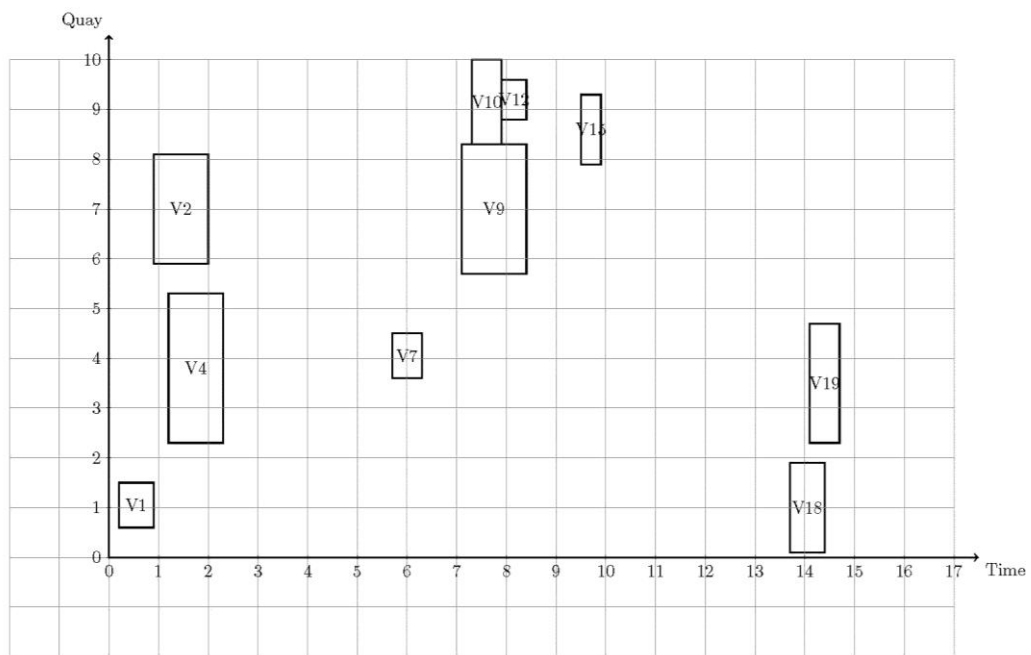


Appendix C

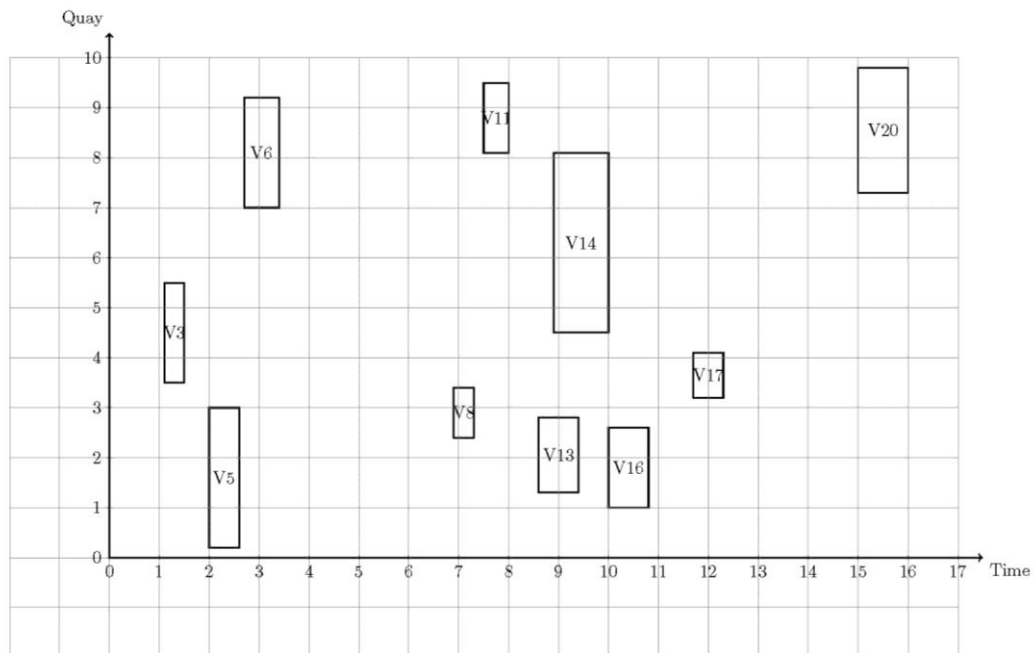
C.1 Example of multiple ports BACAP in chapter 6

The following table is the Instance number 8 dataset which solved for two quays with optimal solution equal to 93.1, the following figures show the distribution for the 20 vessels on quay1 and quay2 to get this optimality.

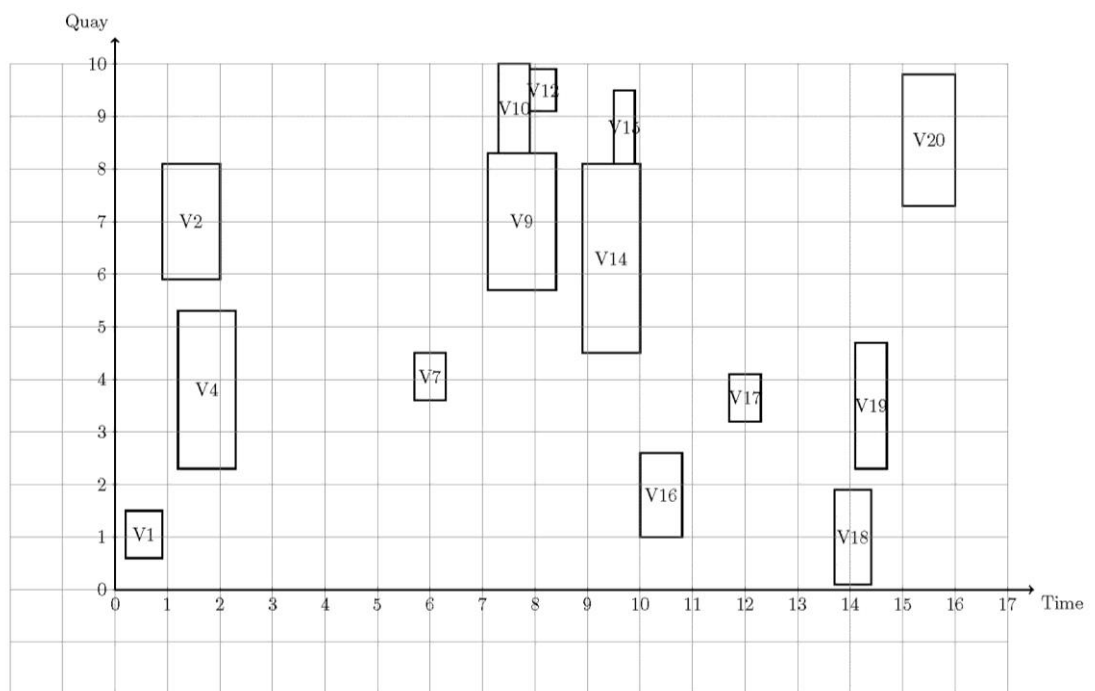
ID	l	b_1^0	b_2^0	m	ETA	EST	EFT	LFT	r^{min}	r^{max}	c^1	c^2	c^3	c_1^{ETA}	c_2^{ETA}
1	9	6	6	12	2	2	9	12	1	2	1	1	3	0	2
2	22	59	59	38	9	8	20	25	2	4	2	2	6	4	5
3	20	35	35	7	11	10	15	17	1	2	1	1	3	2	1
4	30	23	23	54	12	11	23	28	4	6	3	3	9	8	8
5	28	2	2	19	20	18	26	28	2	4	2	2	6	5	4
6	22	70	70	24	27	25	34	37	2	4	2	2	6	5	3
7	9	36	36	11	57	52	63	66	1	2	1	1	3	1	1
8	10	24	24	6	69	62	73	74	1	2	1	1	3	2	1
9	26	62	62	43	71	64	84	90	2	4	2	2	6	3	4
10	17	81	81	10	73	66	79	81	1	2	1	1	3	0	2
11	14	81	81	9	75	68	80	82	1	2	1	1	3	1	0
12	8	88	88	9	79	71	84	86	1	2	1	1	3	1	1
13	15	13	13	14	86	78	94	97	1	2	1	1	3	2	1
14	36	45	45	55	89	80	100	105	4	6	3	3	9	7	7
15	14	79	79	6	95	86	99	100	1	2	1	1	3	1	2
16	16	10	10	14	100	90	108	111	1	2	1	1	3	2	2
17	9	32	32	10	117	106	123	125	1	2	1	1	3	2	2
18	18	1	1	12	137	124	144	147	1	2	1	1	3	0	2
19	24	23	23	20	141	127	147	150	2	4	2	2	6	4	5
20	25	73	73	34	150	135	160	165	2	4	2	2	6	3	3



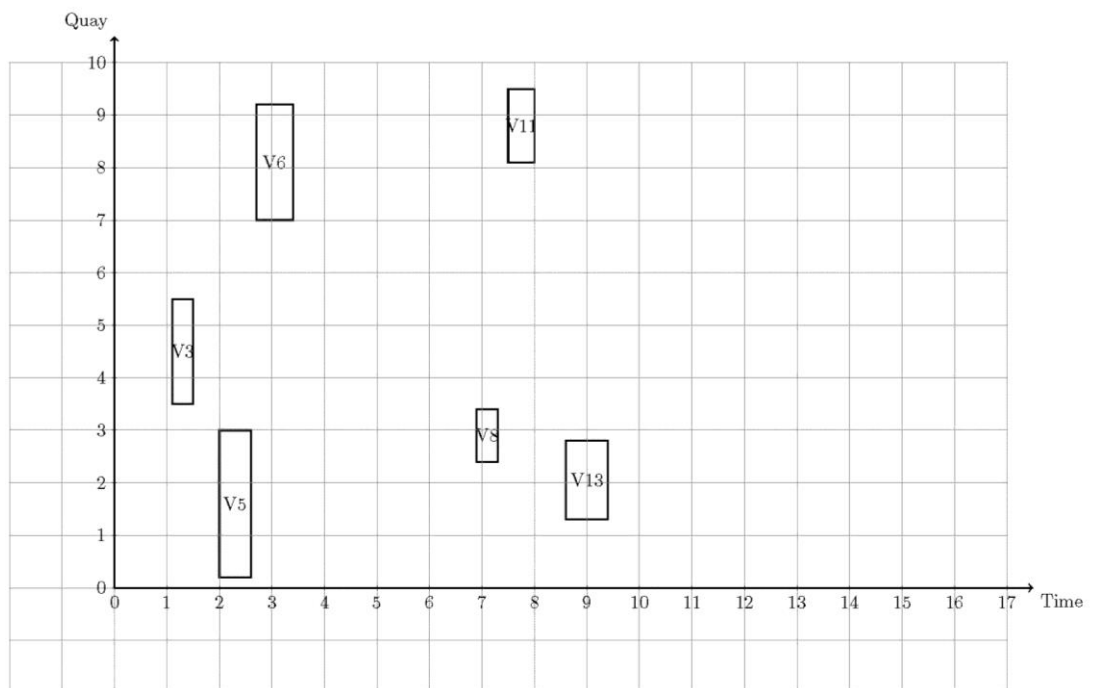
Quay number 1 vessels' distribution



Quay number 2 vessels' distribution



Quay number 1 vessels' distribution using MultiP_BACAP_GA



Quay number 2 vessels' distribution using MultiP_BACAP_GA

C.2 CPLEX code for Multiple Ports BACAP

```
/******  
  
* OPL 12.7.0.0 Model  
  
* Author: Tamer  
  
* Creation Date: 27 Nov 2016 at 04:33:55  
  
*****/  
  
//using CP;  
  
execute {  
  
    // to run Tune1.ops inside the model code  
  
    cplex.fraccuts = 2;  
  
    cplex.mipemphasis = 4;  
  
    cplex.parallelmode = 1;  
  
    cplex.threads = 5;  
  
    cplex.lbheur = true;  
  
    //  
  
    //run.run_processfeasible = true;  
  
    //cplex.intsollim = 1;  
  
    //cplex.objllim = 65;  
  
    // cplex.mcfcuts = 1;  
  
    //cplex.feasoptmode = 3;  
  
    //cplex.tilim = 300; // set time limit to 3600 seconds  
  
    //cplex.mipemphasis = 2; //0 default //1 feasibility over optimality  
                                     //2 optimality over feasibility  
    //3 moving best bound //4 finding hidden feasible solutions  
  
    //cplex.epgap = 0.05;  
  
  
    //cplex.fraccuts = 2;  
  
    // cplex.fpheur = 1;  
  
    // cplex.preind = 0;
```

```

// cplex.simdisplay = 2;

//cplex.getBestObjValue();

// cplex.tuningtilim = 50;

    // cp.param.Workers = 1;

    // cp.param.timeLimit=300;

    //cp.param.searchType = "DepthFirst";
}

int H=...;

float alpha=...; float beta=...; float c4=...;

int M=...;

int nX=...;

int n=...;

range V = 1..n; range T = 0..H-1; range X = 1..nX;

int L[X]=...;

int Q[X]=...;

int id[V]=...;

int l[V]=...;

int b0x[V][X]=...;

int m[V]=...;

int ETA[V]=...;

int EST[V]=...;

int EFT[V]=...;

int LFT[V]=...;

int r_min[V]=...;

int r_max[V]=...;

int c1[V]=...;

int c2[V]=...;

int c3[V]=...;

```

```

int Cx[V][X]=...;

range R = 1..10;

//dvar int+ R[i in V] in r_min[i in V]..r_max[i in V];

// variables

dvar int+ D_ETA[V];

dvar int+ D_EFT[V];

dvar int+ e[V];

dvar int+ D_b[V];

//dvar int+ R[V];

dvar int+ b[V];

dvar int+ s[V];

//

dvar boolean r[V][T][R];

dvar boolean rVT[V][T];

dvar boolean u[V];

dvar boolean y[V][V];

dvar boolean z[V][V];

dvar boolean md[V][X];

// expresiones

dexpr float Z = sum (i in V) (c1[i]*D_ETA[i] + c2[i]*D_EFT[i] +
c3[i]*u[i] + c4 * sum (t in T, q in r_min[i]..r_max[i]) (q*r[i][t][q]))
+ sum (x in X, i in V)(Cx[i][x]*md[i][x]);

// model

minimize Z;

subject to {

    forall(i in V) // (3)

        sum (t in T, q in r_min[i]..r_max[i]) (q ^ alpha)*r[i][t][q]
>= (1+beta*D_b[i])*m[i];

    forall(t in T, x in X) //(4)

```

```

sum (i in V, q in r_min[i]..r_max[i]) q*r[i][t][q]*md[i][x]
<= Q[x];

forall(t in T, i in V) //(5)
    sum (q in r_min[i]..r_max[i]) r[i][t][q] == rVT[i][t];

forall(i in V) //(6)
    sum (t in T) rVT[i][t] == e[i] - s[i];

forall(i in V,t in T) //(7)
    (t+1)*rVT[i][t] <= e[i];
forall(i in V,t in T) //(8)
    t*rVT[i][t] + H*(1-rVT[i][t]) >= s[i];
forall(i in V) //(9)
    D_b[i] >= b[i] - sum(x in X)(b0x[i][x]*md[i][x]);
forall(i in V) //(10)
    D_b[i] >= sum(x in X)(b0x[i][x]*md[i][x]) - b[i];
forall(i in V) //(11)
    D_ETA[i] >= ETA[i] - s[i];
forall(i in V) //(12)
    D_EFT[i] >= e[i] - EFT[i];
forall(i in V) //(13)
    M*u[i] >= e[i] - LFT[i];
forall(i,j in V : i!=j) //(14)
    b[j]+M*(1-y[i][j]) >= b[i]+l[i];
forall(i,j in V : i!=j) //(15)
    s[j]+M*(1-z[i][j]) >= e[i];
forall(i,j in V : i!=j, x in X) //(16)
    y[i][j]+y[j][i]+z[i][j]+z[j][i] >=md[i][x]+md[j][x]-1;

```

```

// forall(i in V)
//      d_min[i]      =      ((1+beta*D_b[i])*m[i])/      Math.pow
(r_max[i],alpha));

forall(i in V) //(17)
{
    b[i]+l[i] <= sum(x in X)(md[i][x]*L[x]);
}

forall(i in V) //(18)
{
    sum(x in X)(md[i][x]) == 1;
}

forall(i in V) //(19)
{
    s[i] <= H;
    s[i] >= EST[i];
}

forall(i in V) //(19-2)
{
    e[i] <= H;
    e[i] >= EST[i];
}

/*forall(i in V, x in X) //(20)
{
    b[i] <= L[x]-l[i];
    b[i] >= 0;
} */

/* forall(i in V) //( )
{
    R[i] <= r_max[i];
    R[i] >= r_min[i];
} */

```

```

    } */

    //forall(i in V) //(19)
    //    D_ETA[i] >=0;
    /*
    forall(i in V) //(19-2)
        D_EFT[i] >=0;
    */

}

/*execute {

    //forall(t in T, x in X) //(4)
    //    writeln(r[]); //sum (i in V, q in r_min[i]..r_max[i])
    q*r[i][t][q] <= Q[x];

    //writeln(z);
    //cplex.getBestObjValue();
}*/

/*main {
    thisOp1Model.generate();
    cplex.solve();
    var ofile = new IloOp1OutputFile("modelRun.txt");
    ofile.writeln(thisOp1Model.printExternalData());
    ofile.writeln(thisOp1Model.printInternalData());
    ofile.writeln(thisOp1Model.printSolution());
    ofile.close();
}*/

```

Appendix D

D.1 Ethical Statement



Certificate of Ethics Review

Project Title:	Heuristic Optimization Methods for Modelling and Solving Maritime Logistics Problems
User ID:	690644
Name:	Tamer El Boghdadly
Application Date:	24/08/2015 14:58:10

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative for the School of Computing is Carl Adams

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- University Policy
- Safety on Geological Fieldwork

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

SchoolOrDepartment: SOC

PrimaryRole: PostgraduateStudent

SupervisorName: Dr. Mohammed Bader-El-den

HumanParticipants: No

PhysicalEcologicalDamage: No

HistoricalOrCulturalDamage: No

HarmToAnimal: No

HarmfulToThirdParties: No

OutputsPotentiallyAdaptedAndMisused: No

Confirmation-ConsideredDataUse: Confirmed


Confirmation-ConsideredImpactAndMitigationOfPontentialMisuse: Confirmed

Confirmation-ActingEthicallyAndHonestly: Confirmed

Certificate Code: 968D-9503-A770-8E2D-E159-84D2-A988-01C2 Page 1

Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy.

Supervisor signature: 


Date: 24/8/2015

FORM UPR16

Research Ethics Review Checklist

Please include this completed form as an appendix to your thesis (see the Postgraduate Research Student Handbook for more information)



Postgraduate Research Student (PGRS) Information		Student ID:	690644
PGRS Name:	Tamer El-boghdady		
Department:	School of Computing	First Supervisor:	Mohamed Bader-El-Den
Start Date: (or progression date for Prof Doc students)	1/10/2013		
Study Mode and Route:	Part-time <input type="checkbox"/>	MPhil <input type="checkbox"/>	MD <input type="checkbox"/>
	Full-time <input checked="" type="checkbox"/>	PhD <input type="checkbox"/>	Professional Doctorate <input type="checkbox"/>
Title of Thesis:	Evolutionary Optimisation Approach for the Single and Multiple-Port Berth Allocation and Quay Crane Assignment Problem		
Thesis Word Count: (excluding ancillary data)	45,542		
<p>If you are unsure about any of the following, please contact the local representative on your Faculty Ethics Committee for advice. Please note that it is your responsibility to follow the University's Ethics Policy and any relevant University, academic or professional guidelines in the conduct of your study</p> <p>Although the Ethics Committee may have given your study a favourable opinion, the final responsibility for the ethical conduct of this work lies with the researcher(s).</p>			
UKRIO Finished Research Checklist: (If you would like to know more about the checklist, please see your Faculty or Departmental Ethics Committee rep or see the online version of the full checklist at: http://www.ukrio.org/what-we-do/code-of-practice-for-research/)			
a) Have all of your research and findings been reported accurately, honestly and within a reasonable time frame?	YES <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	
b) Have all contributions to knowledge been acknowledged?	YES <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	
c) Have you complied with all agreements relating to intellectual property, publication and authorship?	YES <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	
d) Has your research data been retained in a secure and accessible form and will it remain so for the required duration?	YES <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	
e) Does your research comply with all legal, ethical, and contractual requirements?	YES <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	
Candidate Statement:			
I have considered the ethical dimensions of the above named research project, and have successfully obtained the necessary ethical approval(s)			
Ethical review number(s) from Faculty Ethics Committee (or from NRES/SCREC):	968D-9503-A770-8E2D-E159-84D2-A988-01C2		
If you have <i>not</i> submitted your work for ethical review, and/or you have answered 'No' to one or more of questions a) to e), please explain below why this is so:			
<div style="border: 1px solid black; height: 20px; width: 100%;"></div>			
Signed (PGRS):			Date: 13/11/2018