COMBINING SIMULATION AND MULTI-OBJECTIVE OPTIMISATION FOR EQUIPMENT QUANTITY OPTIMISATION IN CONTAINER TERMINALS

by

Zhougeng Lin

This thesis is submitted in partial fulfillment of the
requirements for the award of the degree of
Doctor of Philosophy

of

University of Portsmouth
Portsmouth, United Kingdom
January 2013

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. (44 words)

Acknowledgements

First of all, I would like to express my sincere gratitude to my first supervisor Dr Dylan Jones, second supervisor Dr Xiang Song and other members in the Logistics and Management Mathematics Group (LMMG) in the Department of Mathematics at the University of Portsmouth for their valuable guidance, insight and support throughout my Ph.D. period. Their teaching and supervision were vital to the progression of my research.

Secondly, I would like to thank my thesis examination committee, Prof. Carlos Romero and Dr Graham Wall, for their constructive feedback.

Thirdly, my sincere appreciation also goes to DP World Southampton for the meeting on 24th February 2010 which provided detailed information of the Southampton Container Port.

Fourthly, I would like to thank my family for their support and understanding.

Last but not least, I would like to thank the people who have provided technical and practical support over the course of the project.

Table of Contents

List of Tables

List of Figures

Abstract

This thesis proposes a combination framework to integrate simulation and multi-objective optimisation (MOO) for container terminal equipment optimisation. It addresses how the strengths of simulation and multi-objective optimisation can be integrated to find high quality solutions for multiple objectives with low computational cost. Three structures for the combination framework are proposed respectively: pre-MOO structure, integrated MOO structure and post-MOO structure. The applications of the three structures under the combination framework for following three problems are discussed in the thesis: internal truck quantity optimisation based on post-MOO structure, multiple equipment quantity optimisation based on post-MOO structure and multiple equipment quantity optimisation based on integrated MOO structure.

The truck quantity optimisation problem in modern container terminals, which aims to improve operational efficiency and reduce cost, is discussed in the thesis. This is a multi-objective problem because multiple factors need to be considered in order to guarantee owner's service quality and profitability. A simulation model and a multi-objective optimisation model are built under the combination framework. According to the combination framework and structures proposed, the "Data Processing" is defined as data fitting which generates a set of fitting coefficients and base functions. Solutions provide a series of choices for container terminal operators.

As a further study based on the truck quantity optimisation problem, a multiple equipment (including trucks) quantity optimisation problem is raised. The problem is discussed and a series of optimisation models based on post-MOO structure for multiple equipment deployment are built for the container terminal daily decision making in the consideration of multiple variables and objectives. Simulation and multi-objective optimisation are combined to build integrated optimisation models under the combination framework. The problem is solved by a genetic algorithm.

Based on the multiple equipment quantity optimisation problem raised above, another combination structure, namely MOO leading integrated structure, is employed

to solve the same problem in order to find good enough solutions with less computational cost. The "Data Processing" in the combination framework is defined as data fitting and the "Searching Techniques" is defined as dynamic MOO search. The data fitting generates a set of fitting coefficients and base functions and the dynamic MOO search is a technique to explore the next searching positions based on the Pareto solutions. The results demonstrate that the integrated MOO structure finds better or close to best solutions comparing to the post-MOO structure and the computational cost is likely to be less.

Chapter 1

Introduction

The integration of the global economy and development of transportation have placed sea ports, which are playing an important role in the global economy, at the centre of marine transportation networks and window of international trading. The market requirement upon marine container transportation strongly relates to the global economy and trade. If the world economy (GDP) increases by 1%, the world marine transportation volumes will correspondingly increase by 1.6%[85]. Despite more challenging economic circumstances in some regions, the investment in the container terminal sector has been rising in recent years and container terminals are being opened up in major trading countries over the world. The rise of merchandise trade volume in recent decades leads to more cross-border interdependence in economies, merchandise circulation, capital and applications in new technology. The fast globalisation of the world's economies in recent years is largely based on the rapid development of science and technologies, has resulted from the environment in which market economic system has been fast spreading throughout the world, and has developed on the basis of increasing cross-border division of labour that has been penetrating down to the level of production chains within enterprises of different countries [38].

However container terminals are facing challenges at the same time. Firstly, the tendency of macro scale and professionalisation of container ships and federalisation of shipping lines enhance the bargaining position of large lines in the market and, as being clients of container terminals, raise terminal operational standards. Secondly, the continuously increasing number of container terminals also leads to fierce local competition. Low cost and fast cargo circulation hence become determinants to maintain core competitiveness. Thirdly, the appearance of new transportation modes, such as seamless transportation, door to door transportation, product distribution, etc., has centralised container operations to hub sea ports, which directly intensifies the shortage of their equipment and land resources.

Container transportation also benefitted from the economic growth. Over the past decades, the size of container ships has continually increased, with many modern ships carrying more than 10,000 TEU's (Twenty-feet equivalent unit containers), which means that large amounts of containers have to be loaded, unloaded and transhipped in a short time span in container terminals. It seems that container ships are very possible to get much bigger in the future. On the other hand, limited land and equipment resources in container terminals have led to crowding of the yard which is filled up with containers, trucks, and cranes. Bottlenecks in loading and discharging operations in busy container terminals may occur at operations such as picking up and grounding containers from and to the trucks, loading and unloading containers between the berths and container ships, and the yard crane handling cargos at peak hours. Acquisition of new devices may help but on the other hand will increase the terminal budget. Even if the budget is adequate, the decision of the numbers of different types of equipment to be purchased needs to be made by management teams. The objectives that may be involved in their decision are to improve operational efficiency and reduce congestion and cost.

Therefore, a problem of how to deploy mechanical equipment in container terminals to achieve operational efficiency related objectives and terminal operational cost related objectives is raised.

This thesis mainly addresses the problem above and proposes a framework to combine simulation and multi-objective optimisation to solve the problem. The combination framework proposed defines the elements involved and the process of data streams amongst elements. Two methods are employed in the combination framework (simulation and multi-objective optimisation) and the integration processes of the two methods and parameters are discussed respectively. Three combination structures are therefore proposed for the combination framework to deal with different problems according to the problem characters and features. The structures are pre-MOO, integrated MOO and post-MOO. The pre-MOO structure is suitable for the problems which have a large number of available data and need to build complicated simulation models. The structure normally provides good quality parameters for simulation models from the multi-objective optimisation to reduce computational cost. The integrated MOO structure is usually applied in the problems which have high requirement upon abilities of quick search. The post-MOO structure can be used for the multiple objective problems which do not have a great number of available data. The integrated MOO and post-MOO structures are used to solve a truck quantity optimisation problem and an equipment quantity optimisation problem in this thesis.

A truck quantity optimisation problem is raised in this thesis to assist container terminal management teams to find the best number of internal truck in terminals. The post-MOO structure is used to combine simulation and multi-objective optimisation to solve the problem. The conflicting objectives considered by container terminal decision makers include efficiency requirements by clients, high utilisation rate of trucks, avoidance of congestion in the yard, low fuel consumption goals and low labour demand goals. Discrete-event simulation is employed and combined with multi-objective optimisation to test a realistic set of scenarios for the model and provide a good representation of the flow of goods through a container terminal in the combination framework. Finally, solutions are discussed in the context of the Southampton Container Terminal in United Kingdom.

As a further study based on truck quantity optimisation, multiple types of equipment are taken into consideration in models. Because container terminals are equipped by various types of mechanical devices, so it is more helpful to terminal operators if the best combination of numbers of different types of equipment is provided to support their daily decision making. Models are developed based on post-MOO combination structure to find the best trade-off solutions to optimise each objective. A discrete-event simulation model is proposed based on terminal daily operations in order to present the general cargo flows in container terminals. Multi-objective optimisation is employed to solve multiple objectives involved in decision making and seek for trade-off solutions. Four objectives related to operational efficiency and cost are discussed respectively and the same number of objective functions are built. A genetic algorithm is employed to solve the multi-objective model and the best combination of numbers of different types of equipment can be obtained from solutions. Finally, solutions are discussed in the context of the Southampton Container Terminal.

Based on the models for the multiple equipment quantity optimisation problem, another combination structure, namely integrated MOO, is employed to solve the problem which aims to get better solutions or shorten computational time comparing to the post-MOO structure.

This thesis is organised as follows. The general introduction to the problem and main method is given out in chapter 1. Then the background information related to container ISO standards, information of British, European and global major container ports are addressed in chapter 2. Chapter 3 gives a review of previous papers in container terminal operations, simulation, multi-objective optimisation, data fitting, genetic algorithms and the combination of simulation and multi-objective optimisation. A description of combination of simulation and multi-objective optimisation is stated in chapter 4. Its applications on three container terminal equipment optimisation problems are modelled and optimised afterwards in chapters 5, 6 and 7. Conclusions are given out in chapter 8. Finally, appendices give source codes of the algorithms used in the thesis.

Chapter 2

Background Information

2.1 Introduction

This thesis proposes a simulation and multi-objective optimisation framework and the combination framework is applied to solve equipment quantity optimisation problems in container terminals. The background information about container industry is given in this chapter for better understanding the modelling processes.

In general terms, container terminals can be described as open systems of material flow with two external interfaces which are the quayside with loading and unloading of ships, and the landside where containers are loaded and unloaded on/off trucks and trains; containers are stored in blocks thus facilitating the decoupling of quayside and landside operation [125]. A typical container terminal normally has berths, container yard, gatehouses and the control centre. To be detailed, berths are the areas to moor container ships in terminals. The container yard is an area to temporarily store containers. Gatehouses are the terminal gates to load containers onto and (or) discharge containers from external trucks. The control centre is responsible for terminal operations and delivers instructions. In modern container terminals, computer information systems are widely used to monitor and manage daily operations. The terminal control system in this thesis denotes the computer information systems in container terminals.

This chapter introduces the distribution of the major container ports in the world, Europe and United Kingdom. The equipment in container terminals is also addressed, followed by a section introducing ISO container standards.

2.2 The Major Container Ports in the World and Europe

There have been big changes in the world in international trading, economic globalisation and technology development since the 1960's. As the global economy has

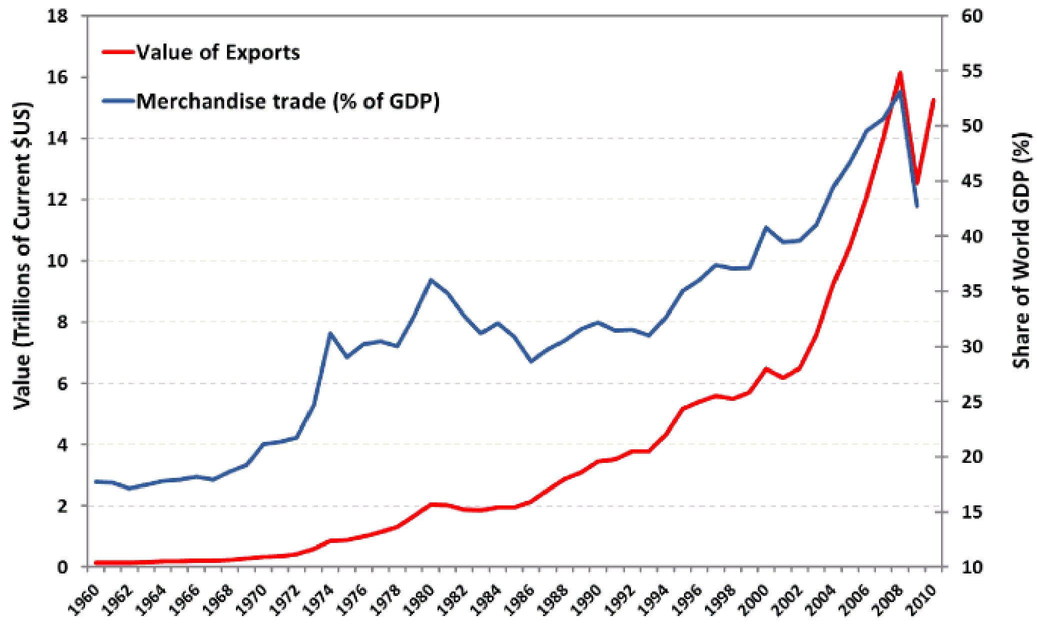

Figure 2.1: World Trade Value from 1960 to 2010 [133]

grown, the world merchandise trade volume has enjoyed swift growth in the past four decades. As shown in figure (2.1), it has experienced two main increase stages: 1970-2002 and 2002-2008. The value of exports was less than 1 trillion U.S. dollars in 1970 and upsurged to over 15 trillion U.S. dollars in 2008. The merchandise trade volume also contributed to Gross Domestic Product (GDP) in the same period. The world GDP has been rising in 1968-1974 and 1993-2008. The world manufacture centre has transferred to Asian countries after the 1970's because of competitive advantages in cheap labour force and raw materials. It resulted in strong performance in exportation in the Far East. Table 2.1 demonstrates the top 10 container ports in the world in 2010 sorted by container throughput. Asian container ports have taken 9 of 10 positions in above ranking, while, seeing from countries, China has 6 container ports being the world's top 10. The port of Rotterdam is the only one from Europe which is the 10^{th}. Table 2.2 shows more detailed data of European container ports. The top 15 container ports in Europe are mainly from traditional industry and marine transportation countries. As being the world's top 10, Rotterdam is on the top in Europe, followed by Hamburg and Antwerp from Germany and Belgium respectively.

Table 2.1: Top 10 Container Ports in the World [114]

Rank	Port	Country	2010 ('000 TEUs)
1	Shanghai	China	29069
2	Singapore	Singapore	28431
3	Hong Kong	China	23699
4	Shenzhen	China	22510
5	Busan	Korea	14194
6	Ningbo-Zhoushan	China	13144
7	Guangzhou	China	12550
8	Qingdao	China	12012
9	Dubai	United Arab Emirates	11600
10	Rotterdam	Netherlands	11100

Above three container ports have more than 8 million TEU container volumes, while Rotterdam has over 10 million. Seeing from the number of ports, Spain has three container ports on the top 15, while Germany, Belgium, United Kingdom and Italy have two.

Table 2.2: Top 15 European Container Port in 2011[105]

Rank	Port	Country	2011 ('000 TEUs)
1	Rotterdam	Netherlands	11877
2	Hamburg	Germany	9014
3	Antwerp	Belgium	8664
4	Bremen	Germany	5915
5	Valencia	Spain	4327
6	Algeciras	Spain	3603
7	Felixstowe	United Kingdom	3265
8	Marsaxlokk	Malta	2360
9	Gioia Tauro	Italy	2338
10	Le Havre	France	2215
11	Zeebrugge	Belgium	2207
12	Barcelona	Spain	2014
13	Genoa	Italy	1847
14	Piraeus	Greece	1680
15	Southampton	United Kingdom	1639

2.3 The Major Container Ports in United Kingdom

There are five major container ports in United Kingdom: Felixstowe, Southampton, Tilbury, Liverpool and Thamesport. As shown in figure (2.2), Felixstowe is the largest

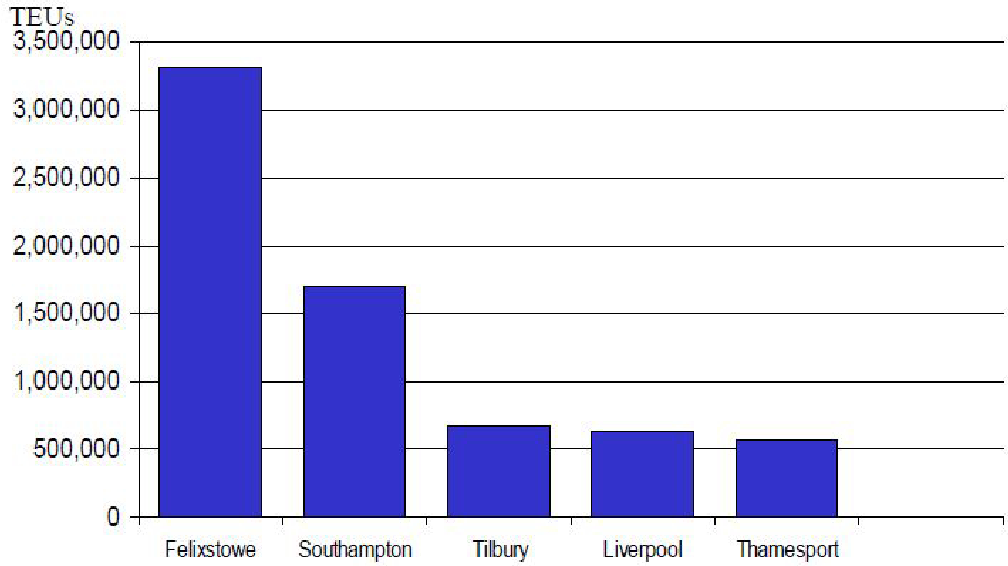

Figure 2.2: United Kingdom Container Port Throughput in 2007 [139]

container port in United Kingdom, while Southampton is the second largest. Seeing from the graph, port of Felixstowe has a substantial lead in container throughput. Port of Southampton has much more volume than the other three ports which are quite close to each other.

2.4 The Major Container Terminal Equipment

Steenken et al. [125] present the types of container terminal equipment: the types of cranes, horizontal transport means and assisting systems. The mechanical equipment discussed in this thesis is quay crane, yard crane and truck.

Firstly, a quay crane is a dockside crane which operates alongside terminal berths loading containers to and discharge containers from container ships. Due to the large size of quay crane, it normally moves on pre-installed tracks or routes. Hence, quay cranes operating for the same container ship are not able to bypass each other. The productivity of quay cranes is usually an important index for shipping lines because quay crane efficiency mainly determines the berthing time of container ships.

Secondly, a yard crane is a dedicated mechanical device to lift containers in the container terminal yard. A large-scale yard may be divided into a number of large areas called zones. In each zone, containers are stacked side by side and one on top of the other to form rectangular shaped heaps called blocks [147]. A typical block may have 6 lanes of containers placed side by side, 5 containers in height for each lane and usually more than 20 containers in length, depending on the geographical shape of the storage yard [147]. The main tasks of yard cranes are to move containers amongst container blocks in the terminal yard, quay crane buffer areas, and gatehouse buffer areas. There are various types of yard cranes such as straddle carrier, sprinter, reach stacker and empty container handler. Straddle carriers lift containers with spreaders and are able to stack containers up to 4 to 6 high. One of the benefits of straddle carriers is that they can work in container blocks because of their structure and height. Another benefit is the width of straddle carriers which stretches over several rows or columns of containers to guarantee its flexibility. Sprinters have a similar structure with straddle carriers except that sprinter is only of one container width. A reach stacker is a vehicle to lift, carry and pile up containers in container blocks in various rows and columns. Reach stackers have more flexibility in movement comparing to straddle carriers and sprinters which only move in container blocks. Then, empty container handlers are dedicated cranes to handle empty containers. In container terminals, empty containers are usually stacked up higher than laden containers because empty containers are lighter than laden containers in weight. Therefore, an empty container handler is a mechanical device to lift less weight but has a longer arm to pile up containers up to 8 high.

Thirdly, trucks are vehicles to transport containers. The trucks work in terminals called internal trucks and those trucks carry containers from and (or) to terminals are called external trucks. The deployment of internal trucks in container terminals is discussed in this thesis.

2.5 Container ISO Standards

Since the appearance of container transportation, there have been various sizes of containers. Many inconveniences are caused therefrom in transportation across different areas and between companies. For instance, container operation equipment is unable to operate containers in different standards; difficulties in transportation; compatibility problems in container storage. The International Organisation for Standardisation have therefore passed a series of standards for container industry, such

Table 2.3: ISO Container Standard Sizes

Size	Length			Width	Height	
Dimensions	6058mm (20’)	12192mm (40’)	13716mm (45’)	2438mm (8’)	2591mm (8’6”)	2896mm (9’6”)
Minimum Internal Dimensions	5867mm	11998mm	13532mm	2330mm	2350mm	2655mm

as ISO 668, ISO 830, ISO 1496, ISO 2308 [25]. Container transportation therefore has manifested its huge advantages in cross-ocean transportation and multi-modal transportation. The standardised dimensions and internal dimensions of containers according to ISO 668 are given in table 2.3 [45]. There might be other sizes of containers appear in container terminals but only 20, 40 and 45 inches standard containers are under consideration in this thesis because the other sizes of container operations have very little proportion of container throughput.

2.6 Summary

This chapter gives information in container transportation industry. Container cargos are mainly carried to hub container ports and distribute to other smaller ports. The major ports in the world and Europe are very busy therefore they need to improve their operations and management to keep pace with the growth of container throughput. Having been experiencing decades’ development, container transportation industry nowadays is highly standardised in structures. The loading and unloading equipment and transportation tools are also highly standardised. Therefore, the problems of equipment optimisation in container terminals are raised in this background.

Chapter 3

Literature Review

3.1 Introduction

Container ports have attracted considerable academic attention in recent decades. Murty et al. [102] give a very detailed introduction to container ports considering multiple objectives which may be involved in decision making. Yard operations, the work flows of outbound and inbound containers, quay crane and yard crane operations, key performance measures of a container terminal, the optimal deployment of yard cranes, the optimal allocation of quay cranes are discussed in the paper.

Comprehensive literature reviews of container terminal operations are given by Vis and de Koster[135], Steenken et al.[125], Maloni and Jackson[91], Vacca et al.[134] and Stahlbock and Voβ[124]. Vis and de Koster[135] present an overview of the processes of containers being transhipped within container terminals. Steenken et al.[125] classify the main logistical processes and operations in container terminals and present a survey of methodologies for container terminal optimisation. Maloni and Jackson[91] review the existing container network capacity literature and discuss the factors impacting the capacity of the container terminals. Vacca et al.[134] present the trends in the literature of container terminal optimisation, which are specialisation on a single problem, combination of problems and integration, and simulation and queuing theory for complete terminals. Additionally, the ports of Antwerp in Belgium and Gioia Tauro in Italy are studied as cases in their paper.

The sections below will discuss the literature review in following areas: simulation for container terminal optimisation, the applications of multi-objective optimisation in container terminals, data fitting, genetic algorithms, and the combination of simulation and multi-objective optimisation.

3.2 Literature Review in Combination of Simulation and Multi-objective Optimisation

This thesis employs a method of combining simulation and multi-objective optimisation to solve the container terminal equipment optimisation problem. Willis and Jones [138] propose a SimMOp framework, which is a simulation based technique to reduce the simulation replications and guarantee the solution goodness, for multi-objective simulation optimisation. It combines a search algorithm with an embedded multi-objective optimisation technique, and database technologies to generate good quality solutions. An inventory case study is given in the paper. Jones and Tamiz [60] (pages 126-128) address three ways to combine goal programming and simulation.

There are some papers developing simulation optimisation to solve multi-objective problems. Asteris et al.[5] propose a multi-objective discrete-event model constructed within Micro Saint Sharp simulation package to examine the flow of UK-bound shipping traffic through the Western continental seaboard system. Baesler and Sepulveda [6] propose a simulation model for cancer treatment centre facility, which is created and integrated to a multi-objective optimisation heuristic with the purpose of finding the best combination of control variables that optimise the system performance. Liu et al.[89] develop simulation models for an automated guided vehicle system in container terminals and employ the method multi-attribute decision making(MADM) to assess the performance of the terminals. Almeder et al. [1] present an approach to combine simulation models and optimisation models to support the operational decisions for supply chain networks. Lin and Kwok [88] apply multi-objective metaheuristics for vehicle routing problems on real and simulated data. Yalcinkaya and Bayhan [140] give an example of optimisation of simulation parameters. They propose a model and solution approach based on discrete-event simulation and response surface methodology to optimise the average passenger travel time for a metro line. The main parameters for this model are rate of carriage fullness and headways with outputs including average travel time. The Derringer-Suich multi-response optimisation procedure [31] is used to determine the rate of carriage fullness to minimise the average travel time.

General simulation optimisation problems are also discussed by a number of papers. Legato et al. [79] propose a simulation based optimisation framework for container loading and discharging operations in container terminals to find the optimal assignment and optimal sequencing (schedule) of bays (jobs) processed by a fixed number of available cranes (machines). In the combination framework, the QCSP (quay crane scheduling problem) configuration is searched by a search process by simulated annealing and the outcome of container loading/discharging plans are evaluated by an evaluation process by simulation. Zeng and Yang [146] also present a framework of simulation optimisation. A simulation optimisation model for scheduling loading operations in container terminals is developed to find good container loading sequences which are improved by a genetic algorithm through an evaluation process by simulation model to evaluate objective function of a given scheduling scheme. Meanwhile, a surrogate model based on an neural network is designed to predict the objective function and filter out potentially poor solutions, thus to decrease the times of running the simulation model. The results show that the simulation optimisation method can solve the scheduling problem of container terminals efficiently.

Some similar studies outside the field of container terminal management are given as follows. Oddoye et al.[107] present a detailed simulation model for healthcare planning in a medical assessment unit (MAU) of a general hospital to test different scenarios to eliminate bottlenecks in order to achieve the optimal clinical workflow. Their paper proposes a new model for healthcare planning in a medical assessment unit and this method combines simulation and weighted goal programming (GP) for efficient resource planning. Results from the simulation are input into a GP model with different weights applied to positive deviations from five objectives based on management preferences for trade-off analysis of the results. Results from different experiments performed in GP are shown in their paper. Sensitivity analysis on weights is conducted to analyse the effects of attaching different weights to the deviations.

3.3 Literature Review in Simulation

A number of papers address the discrete-event simulation method for container terminal optimisation. Simulation in management decision making is defined as simply the use of a computer model to 'mimic' the behaviour of a complicated system and thereby gain insight into the performance of that system under a variety of circumstances [131]. Jahangirian et al.[56] give a literature review in simulation in manufacturing and business from 1997 to 2006. Angeloudis and Bell [4] provide a review of container terminal simulation models.

There are a number of papers which apply simulation in the container industry. Yun and Choi [145], Henesey et al.[49], Liu et al.[89], Li et al.[80], Petering et al. [110], Petering and Murty [111] give papers to this topic. Yun and Choi [145] propose an objective oriented simulation model for container terminals by using computer programming language SIMPLE++ and the model is tested by setting parameters and inputting data from the real terminal of Pusan east in South Korea. Henesey et al.[49] address a container terminal berth allocation problem and stacking policy. A method of Multi Agent Based Simulation for a container terminal is proposed which aids in the evaluation of operational policies for transhipment of containers. A multi-agent based simulator called SimPort is developed to compare the results of two berthing and stacking policies which are Berth Closest To the Stack (BCTS) policy and Overall Time Shortening (OTS) policy respectively. Liu et al.[89] study the automated guided vehicle system in container terminals and develop simulation models to demonstrate the impact of different terminal yard layouts on the terminal performance. The real operational data from the Port of Rotterdam in Netherlands is collected for the simulation scenario parameters. The models are run by using different numbers of vehicles and the results are compared based on different layouts and operations. Vis et al.[136] develop an integer linear programming model to determine the minimum number of vehicle requirements under time-window constraints. A simulation model is built for a container terminal to study the performance of the analytical model and validate the estimates of the vehicle fleet size by the analytical model. The analytical model provides a good estimate of the number of vehicles required. The minimum number of simulation replications is also discussed in their paper. Li et al.[80] build a simulation model for a container terminal logistics system to validate the rationality and creditability of the mathematical model proposed in their paper.

3.4 Literature Review in Multi-objective Optimisation

The applications of multi-objective optimisation to container terminal problems are widely studied by researchers. Gass and Saaty [39] provide the first approach applicable to multi-objective programming in 1955 [24]. Geoffrion and Dyer [40] propose an interactive mathematical programming approach to multi-criterion optimisation and give an application to the operation problem of an academic department. Sawaragi et al. [116] and Steuer [126] address the theories of multi-objective optimisation. Miettinen [97] gives the theories and methods of nonlinear multi-objective optimisation. Multi-objective programming is a method to solve problems with more than one objective which are conflicting. It is, therefore, applicable to decision making in container terminals as the decisions in a modern container terminal are usually made taking account of multiple objectives [24]. Vis and de Koster [135] address an overview of the processes by which containers are transhipped within container terminals and present the introduction of a multi-objective approach related to machine scheduling problems. Marler and Arora [94] give a survey of continuous nonlinear multi-objective optimisation for engineering. Their paper addresses the foundation of fundamental concepts, methods that involve a priori articulation of preferences, methods with a posteriori articulation of preferences, methods that require no articulation of preferences and genetic global algorithms. Ehrgott [33] gives theories of multi-criteria optimisation. Mula et al.[100] give a review on mathematical programming models for supply chain production and transport planning, and conclude the papers using multi-objective programming for planning.

The papers related to multi-objective optimisation in truck optimisation, quay crane optimisation and yard crane optimisation are given in the following sections.

3.4.1 Quay Crane Optimisation

Some papers address multiple objectives involved in quay crane optimisation. The models given by Imai et al. [55] involve the considerations of two objectives. Their paper addresses a simultaneous berth and quay crane allocation problem at a multi-user container terminal. A model for berth and quay crane allocation is built based on a model for berth allocation problem to minimise the total service time and the constraints of the quay crane allocation. Detailed solution procedures are given for the genetic algorithm-based heuristic, which iterates the procedure of determination of berth scheduling and quay crane scheduling at the same time, to solve the problem. Liang et al. [84] discuss the number of quay cranes employed in container terminals. Their paper addresses the problem of determining the berthing position, duration of berthing of a ship and the number of quay cranes assigned to each ship. A model is built in order to minimise the sum of the handling time, waiting time and the delay time for every ship. A hybrid approach which combines the genetic algorithm with heuristic is proposed to dynamically search the near Pareto optimal solutions. The near Pareto optimal solutions are the solutions close to the Pareto frontier and have good approximation of the Pareto optima [54]. Computational experiments show that the proposed approaches are applicable to solve this difficult but essential terminal operation problem. Chang et al. [18] propose a multi-objective model which aims to minimise the moves of quay cranes, the time delay of ship departures, and the energy consumption during quay crane assignment. A rule-based joint berth allocation and quay crane assignments model has been developed based on a rolling-horizon approach. The results show that the ship departure times are earlier and the number of quay cranes employed in container terminals is less.

The mixed integer programming is widely used to model the quay crane problems, such as Lee et al. [75], Zhang and Kim [148] and Kim and Park [67]. Lee et al. [75] propose models to integrate quay crane and yard truck scheduling for container terminals and formulate the problem as a two-stage flexible flow shop with sequence dependent setup time and block in a mixed integer programme. The objective is to minimise the makespan of dispatching the containers allocated to quay cranes subject to a series of constraints related to quay crane and yard crane operations. The results show that the makespan of the proposed strategy that integrates quay crane and yard truck scheduling into a whole can be reduced by a large percentage, ranging from 23% to 115%, compared with the benchmark strategy. Kim and Park [67] discusses a quay crane scheduling problem, a mixed-integer programming model is given to find the best sequence to load and discharge cargos in order to minimise the makespan of operations. A branch and bound method and a heuristic search algorithm (greedy randomised adaptive search procedure (GRASP) [36]) are proposed to find solutions. The results show that the final objective values derived by GRASP did not exceed those derived by the branch and bound method by more than 10%. But GRASP reduced the computational times to 3% on average when the number of quay cranes and the number of tasks exceeded 3 and 20, respectively. Zhang and Kim [148] address quay crane scheduling problems and propose a mixed integer programming model to minimise the number of operation cycles of a quay crane for discharging and loading containers in a ship-bay (i.e. to reduce the number of dual cycle operations). The problem is broken down into two main phases, i.e. intra-stage optimisation (sequencing all stacks in one hatch), and inter-stage optimisation (sequencing all hatches). A hybrid heuristic approach which combines an effective gap-based local search technique with reformulated heuristic approaches is proposed to find the solutions. Kim and Park [67] propose a mixed-integer programming model for the quay crane scheduling problem subject to various constraints related to the quay crane operations. A branch and bound (B & B) method is used to obtain the optimal solutions. The greedy randomised adaptive search procedure (GRASP) is also used to search the near Pareto solutions to reduce computational time. According to the numerical experiment, objective values derived by GRASP did not exceed those derived by the B & B method by more than 10% when the parameters of GRASP have values within specified ranges. GRASP reduced the computational times to 3% on average when the number of quay cranes and the number of tasks exceeded 3 and 20, respectively.

Heuristic methods are widely used to solve the quay crane scheduling problems. Kaveshgar et al. [63], Yang and Wang [141], Bierwirth and Meisel [10], Le et al. [72], Yang et al. [142], Jin and Li [59], Golias et al. [42] and Lee et al. [75] give papers in applying heuristics, most of them use genetic algorithms, in quay crane scheduling problems. Jin and Li [59] present a quay crane deployment problem in container terminals that how to assign quay cranes to container ships and schedule the quay cranes for the tasks of each container ship simultaneously. A non-linear model is built to minimise the total turnaround time of all the container vessels subject to constraints such as the non-crossing of quay cranes and the serving relation constraints of pairs of tasks. A genetic algorithm is proposed to solve the problem. The results for a single ship case are compared with literature and the results for multiple ships are compared with the single ship case by implementing in simulation.

3.4.2 Yard Crane Optimisation

The optimisation for yard operations is widely discussed. Lee and Chao [78], Kim and Bae [64], Bortfeldt and Forster [14], Exposito-Izquierdo et al. [35] and Hirashima

[50] address how to improve the pre-marshalling operations in container terminals.

There are a certain portion of papers discussing yard crane scheduling problems, building models to minimise operational time or cost and then solve the problems by using heuristic methods due to the large size and complexity of the problems. The papers are given by Li et al. [83], He et al. [48], Li et al. [82], Bish [11], He et al. [47], Chang et al. [19], Huang et al. [53] and Javanshir and Ganji [58]. He et al. [48] develop a dynamic scheduling model using objective programming for yard cranes based on rolling-horizon approach. Two objectives are considered in their paper: minimising the total delayed workload among all blocks at each planning horizon and minimising the total times that yard cranes move from one to another block at each planning horizon. A multi-objective function based on the two objectives is proposed subject to a series of constraints. A hybrid algorithm employs heuristic rules and a parallel genetic algorithm is proposed to solve the problem. The parameters for the parallel genetic algorithm are selected by several groups of tests. The parameter settings for this problem are shown as follows. The population and sub-population size are set as 100 and 50, respectively. The crossover probability is set as 0.8. The mutation probability is set as 0.05. The maximum elapsed generation is set as 40. The percentage of replacement is set as 0.25. The migration frequency and number are set as 3 and 5, respectively. Bish [11] discusses a multiple-crane-constrained vehicle scheduling and location problem. A model to analyse the effectiveness of vehicle pooling policies, which allow a set of vehicles to be shared between ships. The problem is NP-hard therefore a simple heuristic called the the transshipment problem based list scheduling heuristic is proposed. Li et al. [83] propose models for yard crane scheduling to decide if container move m is assigned to the yard crane c during time interval t. Heuristics and a rolling-horizon algorithm are employed to solve the problem. The results from three algorithms DMIP1, DMIP2 and DMIP3 show that the model size is greatly reduced systematically and the solution time is shorted from days to seconds. The algorithm yields higher solution quality in a very short time compared to other heuristics used in the literature. Li et al. [82] integrate a continuous time model with heuristics and rolling-horizon algorithm for yard crane scheduling. The model requires far fewer integer variables than previous work. The results show that it significantly improves the solution quality compared to the existing discrete time models and other heuristics found in the literature. The model size is reduced significantly and the solution time is shorted from days to seconds.

The mixed-integer programming is also used for yard crane scheduling problems. Cao et al. [16] propose a mixed-integer programming model for yard crane scheduling to minimise the makespan of loading all outbound containers in the planning horizon. The problem is NP-hard so it is unlikely to be solved by exact solution algorithms. Therefore, the general Benders' cut-based (GBC-based) method and the combinatorial Benders' cut-based (CBC-based) method are proposed to solve the problem. In average, the computational time of CPLEX method is about 1.57 times longer than the time required when the GBC-based method is used to solve the problem.

Kim and Kim [65] have a discussion about the optimal amount of storage space and transfer cranes for import containers. The paper develops a model to minimise the cost and optimise the amount of storage space and number of transfer cranes for imported containers in the yard. The space cost, transfer crane cost and outside truck cost are considered in the model. The results show that the optimal space amount decreases as the space cost increases, but the optimal number of transfer cranes is insensitive to the change of the space cost; both the optimal number of transfer cranes and the optimal space amount increase as the cost of outside trucks increases.

3.4.3 Truck Optimisation

A literature review on multi-objective vehicle routing problems is given by Jozefowiez et al.[62]. Besides, Tan et al.[129] [128], Muller [101] and Lee et al.[76] address their models minimising the number of vehicles. Tan et al.[129] study a truck-trailer-separated situation in container terminals and depots, and use a hybrid multi-objective evolutionary algorithm to solve truck and trailer vehicle routing problems. 20 and 40 TEU trailers, and, importation and exportation situations, are taken into consideration respectively in modelling with consideration of time slots. The number of trailers available for pick-up in a particular time slot is equal to the number of trailers in the previous time slot, plus the trailers returned in the previous time slot and minus those picked up in the previous time slot. The models determine the number of trailer exchange points (TEPs) and minimise the total number of trailers at each

TEP. Tan et al.[128] discuss a vehicle routing problem from a depot to geographically dispersed customers with multiple objectives: minimum travel distance, driver remuneration, and number of vehicles, which subject to a number of constraints such as time windows and vehicle capacity. The modelling of vehicle transportation cost and vehicle driver remuneration is discussed in their paper. Muller [101] discusses a vehicle routing problem with time windows to minimise the number of vehicles and travelling distance. Solutions are obtained in two stages: firstly, Solomon's route construction heuristic I1 [119] is applied to obtain an initial (current) solution and then, route improvement heuristics, Or-opt exchange and 2-opt procedure are repeatedly applied to the first stage solutions to decrease the number of used vehicles and to obtain savings in distance. Lee et al.[76] address job scheduling for truck optimisation in the container terminal environment and present a multi-objective optimisation model to minimise the number of trucks and cost of early arrivals or delays. A case study based on a Chicago transportation carrier is given in their paper. Klundert and Otten [68] discuss utilisation rate, which presents a model for increasing the utilisation of scheduled road transportation activities by accepting extra loads, and study a minimising cost model. Reducing cost problems with multiple goals are discussed by Nishimura et al.[104], Nunkaew and Phruksaphanrat [106], Tan et al.[129] and Tzur and Drezner [132]. Nishimura et al.[104] address the yard trailer routing problem at a maritime container terminal, develop a new routing scheme achieving container handling cost savings and propose a more efficient trailer assignment method 'dynamic routing'. A heuristic method is developed and a wide variety of computational experiments are conducted to solve the problem. The results of the experiments demonstrate that the dynamic routing reduces travel distance and generates substantial savings in the trailer fleet size and overall cost by 15%. Nunkaew and Phruksaphanrat [106] propose a multi-objective programming model to improve the quality of depot transportation service. The multi-objective programme consists of two objectives: minimising the total transportation cost and overall independence value between customers which means the consideration of customer to customer relationship. The latter is quantified by the pairwise comparison matrix. Lexicographic goal programming is applied to the models by setting the total transportation cost as the first goal and the overall independence value as the second goal. Tan et al.[129] address a multi-objective model, which aims to minimise cost and the number of trucks, to solve truck routing problems in container terminals. Tzur and Drezner [132] present a vehicle routing problem for distribution systems with consideration of travelling cost, waiting cost and fixed cost associated with feasible weight and volume capacity utilisation of the vehicles, minimising the number of trips and hence the travelling cost and the sum of the reduction (in percentages) in the slack of all tasks. Lee et al.[74] propose a novel approach to integrate the yard truck scheduling problems and storage allocation problems into a whole. The model proposed minimises the weighted summation of total travel time of yard trucks and total delay of loading and discharging requests. A hybrid insertion algorithm is employed to explore the solutions. Li et al.[80] present a model of the container terminal yard trailer dynamic dispatching problems. The paper compares the similarities between a container terminal logistics system and a computer system, and utilises the hybrid flow shop with blocking based on attributes to build a model based on Harvard architecture and multi-agent. The results of three dispatching policies, which are semaphore mechanism, genetic algorithms and static scheduling respectively, are compared in their paper.

3.5 Literature Review in Genetic Algorithms

In multi-objective optimisation area, evolutionary algorithms are widely applied due to the complexity and time consuming nature of finding Pareto optimal solutions for multiple objectives. Deb [28] and Deb et al. [29] propose the non-dominated sorting genetic algorithm-II (NSGA-II) for multi-objective optimisation. Deb [27] addresses the classical methods and evolutionary algorithms for the multi-objective optimisation. Jones et al.[61] address a survey of multi-objective meta-heuristics papers in the 1990's. Coello [23] gives a review on evolutionary multi-objective optimisation. Branke et al. [15] provide a guidance for evolutionary multi-objective optimisation.

As an algorithm of evolutionary algorithms, the genetic algorithm is a stochastic search technique based on the mechanism of natural selection and natural genetics [75]. Genetic algorithms are very popular for hard to solve problems because their strengths in quick searches for the near Pareto optimal solutions and they are also widely used in the area of container terminal optimisation. Whitley [137] gives a tutorial for genetic algorithms. Konak et al. [69] give a tutorial for multi-objective optimisation using genetic algorithms. Tamaki et al. [127] present a review on multi-objective optimisation by genetic algorithms. Fonseca and Fleming [37] also have a discussion on genetic algorithms for multi-objective optimisation. Hartmann [46] uses a genetic algorithm for an optimisation model for scheduling jobs at container terminals.

Altiparmak et al. [2], Jaszkiewicz [57], Bazzazi et al. [9], Horn et al. [51], Min et al.[98] and Li et al. [81] give papers for genetic algorithms for multi-objective problems. Li et al. [81] present a falling tide algorithm which is applied under a multi-objective optimisation framework. The major advantage of their approach lies in its ease of implementation and its use of a small number of intuitive parameters that are easy to understand by users. A nurse rostering experiment is implemented by the algorithm. The results show that it achieves significantly better results for 11 out of 12 test instances with less computational time. Min et al.[98] demonstrate that genetic algorithm's flexible solution search process that can convert constrained problems into unconstrained problems and then cross the feasibility boundary to find near optimal or optimal solutions in an "intelligent" (probabilistic) manner rather than relying on random enumerations or iterations. In particular, a genetic algorithm is chosen over other meta-heuristics procedures such as tabu search due to its ability to generate a collection of solutions rather than a single solution at each stage. Chung et al. [22] addresses a mixed integer programming model for the train-sequencing problem. A hybrid genetic algorithm is employed to solve the problem. A modified elite group technique is used in the genetic algorithm, which means that the best two chromosomes are preserved in the next generation without changes in its genes. Also, a fixed number of chromosomes in each group are maintained in the next generation. The rest of the population is created anew in the next generation by using random one-point crossover and mutation operations. Additionally, a penalty function that includes route-distance levelling and penalty terms is used as the fitness function. Martin [95] presents a hybrid genetic algorithm/mathematical programming heuristic for the n-job, m-machine flowshop problems with lot streaming. The problem is solved by a genetic algorithm. In this paper, a set of candidate sequence vectors based on the parameters of the problem are randomly generated. They are then evaluated by solving the mixed-integer programming model. The best candidates are selected to form an initial population. The parameters for genetic algorithm are set as follows. The initial number of candidates is 50. Population size is 20. Mating pool size is set at 10. Mutation rate is 10%. Termination occurs after 100 generations.

3.6 Summary

There are a great number of papers discussing optimisation for container terminals. Simulation and multi-objective optimisation are widely used in this area. Continuous multi-objective optimisation is a potentially powerful tool to solve container terminal optimisation problems. Additionally, the method of simulation is effective in modelling the flow of real world entities. The combination of multi-objective optimisation and simulation is therefore hypothesised to be an excellent and practical tool for decision makers of container terminals. For the papers discussing multi-objective optimisation, reducing cost is normally one of objectives. Besides, genetic algorithm is a popular heuristic method to solve multi-objective models. Furthermore, combining simulation and multi-objective optimisation to integrate the strengths of the two methods for container terminal optimisation is a new area which is worth exploring.

Chapter 4

Combination Framework

4.1 Introduction

This chapter discusses the combination of simulation and multi-objective optimisation under the background of container terminal optimisation. Firstly, a general statement is given in section 4.2 which addresses the advantages and disadvantages of simulation and multi-objective optimisation respectively, and then presents the benefits and expected effect of combining the two methods are stated. Section 4.3 states the ways to integrate simulation and multi-objective optimisation and proposes three combination structures. The "Data Processing" and "Searching Techniques" are both discussed. Additionally, The "Data Processing" and "Searching Techniques" are defined for the MOO leading integrated structure and post-MOO structure because they are employed to solve problems in chapters 5, 6 and 7.

4.2 Combination Framework of Simulation and Multi-objective Optimisation

Simulation is the imitation of the operation of a real-world process or system over time [7]. Simulation can virtually implement ideas or decisions instead of real actions which may save money and verify the feasibility of decisions. Simulation is able to test different sets of systematic parameters and provide decision makers with practical feedback which allows them to determine the optimal modes to run terminals without occurrence of any considerable cost. It is normally used in where common models are not applicable because of systematic complexity and uncertainty. Simulation can well describe dynamical systems and continuously respond to new events. Additionally, simulation generates a great number of stochastic values and provides a mass of useful data at a very detailed level. It can be set to repeatedly execute to a certain number of iterations or at a certain level of detail. Therefore simulation might be computational time consuming if a large number of iterations are needed.

The process of optimising systematically and simultaneously a collection of objective functions is called multi-objective optimisation or vector optimisation[94]. In other words, the multi-objective optimisation is the process to find the Pareto optimal solutions to satisfy more than one objective subject to a series of constraints. Most multi-objective optimisation algorithms are able to find the approximate Pareto optimal solutions mathematically. In the combination framework, a great deal of data from simulation can be processed in order to set the parameters for multi-objective optimisation. In order to guarantee the Pareto optimality of solutions, reduce overall computational time and ensure realistic model of work flow under random conditions, simulation and multi-objective optimisation is therefore combined in order to achieve better performance.

Jones and Tamiz [60] address three ways to combine goal programming and simulation which are pre-goal programming, integrated goal programming and post-goal programming. The combination of simulation and more general multi-objective optimisation may also have efficient performance due to the similarities between goal programming and some other multi-objective optimisation techniques. Applying the above combinations to multiple objective problems, then the combinations between multi-objective optimisation and simulation could be concluded as pre-multi-objective optimisation, integrated multi-objective optimisation and post-multi-objective optimisation. This chapter proposes a combination framework to integrate simulation and multi-objective optimisation as shown in figure (4.1). Simulation and multi-objective optimisation are two main methods used in the combination framework. Normally, simulation needs parameters to start to simulate, while these parameters can be set by a series of experimental data, historical data, optimised scenario data etc. If models start from simulation, its parameters need to be collected, processed and prepared from historical data and (or) experimental data and (or) other data available. If multi-objective optimisation models have provided a large number of data to be extracted parameters for simulation scenarios, simulation will obtain the Pareto optimal parameters to run the models. On the other hand, if models begin from multi-objective optimisation models, its parameters need to be set by historical and (or) experimental data. Otherwise, simulation models need to provide data to be

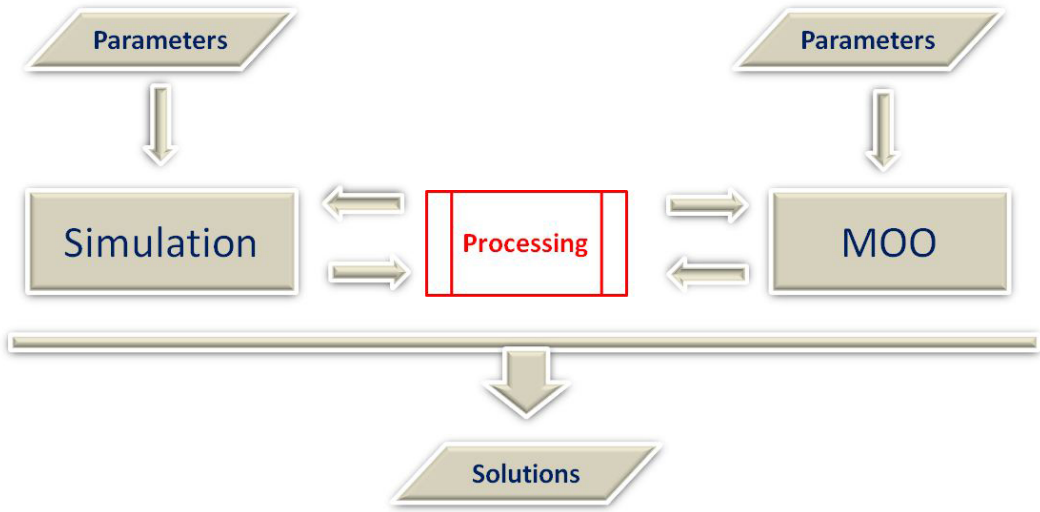

Figure 4.1: Combination Framework

processed to set parameters for multi-objective optimisation to find the Pareto optimal solutions. As the centre box shown in figure (4.1), simulation and multi-objective optimisation need communications (or called "data interchange") between them to continually adjust mutual optimisation processes. The data interchange requires a series of data processing mechanism between them in different sequences. Therefore, the data interchange is discussed in the following section.

4.3 Combination Structures

In this thesis, the equipment quantity optimisation for container terminals is studied and both of simulation and multi-objective optimisation methods are used. Therefore simulation and multi-objective optimisation models need to be developed under container terminal background. The simulation results are for testing a realistic set of scenarios for the models and providing a good representation of container terminal operation processes. Multi-objective optimisation models also need to be built for decision makers because of the multiple objectives which might be raised. Firstly, s-ingle objective formulations are built on the basis of a series of data for each objective. Then a multi-objective optimisation model is proposed subject to constraints based on single objective formulations. Finally, solutions are solved by an exact or heuristic method. The communication between the two methods could proceed in different ways and sequences. Moreover, the starting point of models could be simulation or multi-objective optimisation. Therefore, three combination structures are proposed in this section to represent different situations of optimisation:

- Pre-MOO: Multi-objective optimisation is used prior to simulation which aims to obtain ideal parameters for simulation models.
- Integrated MOO: Multi-objective optimisation is embedded in the simulation or the simulation is embedded in multi-objective optimisation in order to increase the effectiveness and integration of the model(s) and reduce the running time.
- Post-MOO: Multi-objective optimisation is employed posterior to simulation for the purpose of making proper scenario(s) for Multi-objective optimisation and better meeting the requirements of the decision maker(s).

The tests and comparisons of results from different combination structures need to be conducted to choose an appropriate one for problems. The three structures are discussed in the following sections.

4.3.1 Pre-MOO Structure

Pre-MOO structure denotes that multi-objective optimisation is executed prior to simulation in the combination framework to obtain good parameters for simulation. It starts from multi-objective optimisation. The parameters for multi-objective optimisation are pre-set by collecting and processing historical and (or) the previous experimental data. Then the solutions from multi-objective optimisation are processed to determine the parameters for simulation. Simulation models run with the optimised parameters to find the best modes to represent dynamic real world systems.

The details of "Processing" box in figure (4.1) for this structure are demonstrated in figure (4.2). The data stream is delivered between "Simulation" and "MOO" boxes for the communication between the two methods in the combination framework.

4.3.2 Integrated MOO Structure

Integrated MOO Structure denotes that simulation and multi-objective optimisation are embedded in the combination framework to collaborate closely to increase the

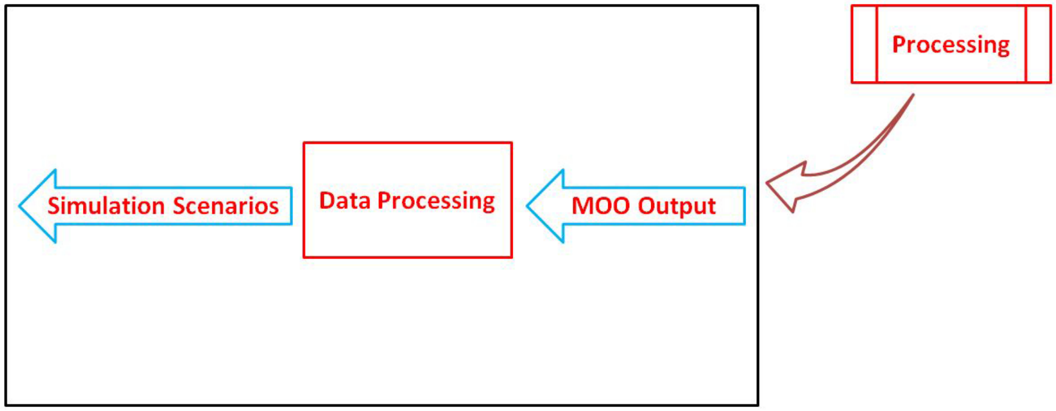

Figure 4.2: Pre-MOO Combination Structure

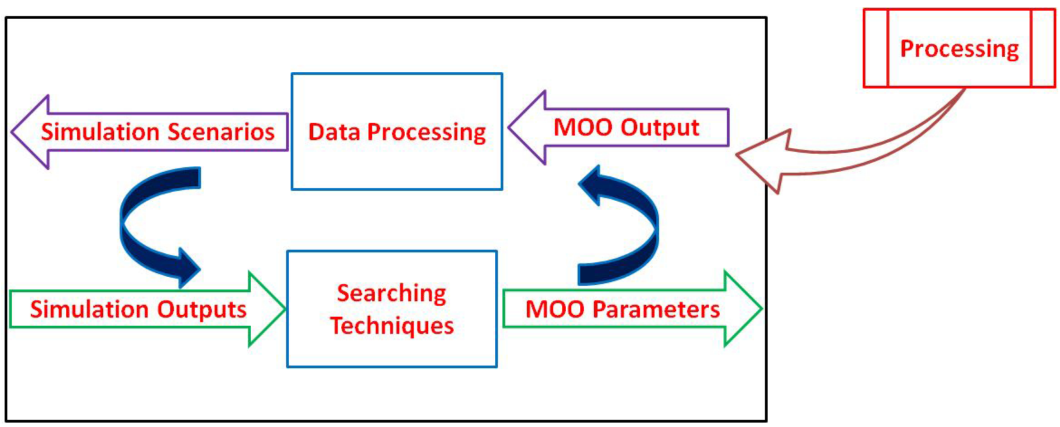

Figure 4.3: Simulation Leading Integrated Structure

effectiveness of the model and reduce the running time. According to the sequences of data streams, there are two sub-structures for integrated MOO structure, namely simulation leading integrated structure and MOO leading integrated structure.

Simulation Leading Integrated Structure

As shown in figure (4.3), the simulation leading integrated structure denotes that simulation and multi-objective optimisation are embedded in the combination framework and the starting point is multi-objective optimisation. According to figures (4.1) and (4.3), the multi-objective optimisation output is delivered to simulation to set simulation scenarios to search the Pareto optimal values of decision variables for multi-objective optimisation, therefore, it is a simulation leading structure. The data stream is generated from simulation to the "Data Processing", and then delivered to multi-objective optimisation; and then from multi-objective optimisation to the "Searching Techniques", and then come to simulation. This is one iteration. The data stream will keep iterating until stopping criteria are met. The stopping conditions are normally pre-set in terms of characters of problems and requirement upon decision makers. For example, the searching loop stops when it could not search better values for a pre-set number of iterations or it goes in an endless loop in local searches.

The details of "Processing" box in figure (4.1) are demonstrated in figure (4.4). The starting point is the multi-objective optimisation in this structure. The initial parameters for the multi-objective optimisation are required to start to run the combination framework. After the first iteration, simulation parameters are set by the Pareto optimal solutions from multi-objective optimisation and the parameters for multi-objective optimisation are set by the processed data from simulation results until the process stop when all stopping criteria are met.

MOO Leading Integrated Structure

As shown in figure (4.4), MOO leading integrated structure denotes that simulation and multi-objective optimisation are embedded in the combination framework and the starting point is the simulation models. According to figures (4.1) and (4.4), simulation output is delivered to multi-objective optimisation to find the Pareto optimal solutions and the solutions are used to search the next decision variables for simulation. Namely, multi-objective optimisation mainly determines the values of the decision variables for the next iteration. Therefore, it is called MOO leading integrated structure.

The data stream is generated from simulation, then delivered to the "Data Processing", and to multi-objective optimisation; and then from multi-objective optimisation to the "Searching Techniques", and to simulation. The data stream will keep iterating until stopping conditions pre-set are met.

The details of "Processing" Box in figure (4.1) are demonstrated in figure (4.4). The combination structure starts from simulation in the first iteration. The initial parameters for simulation are required to start the models. After the first iteration, simulation parameters are set by the Pareto optimal solutions from multi-objective optimisation and the parameters for multi-objective optimisation are obtained from

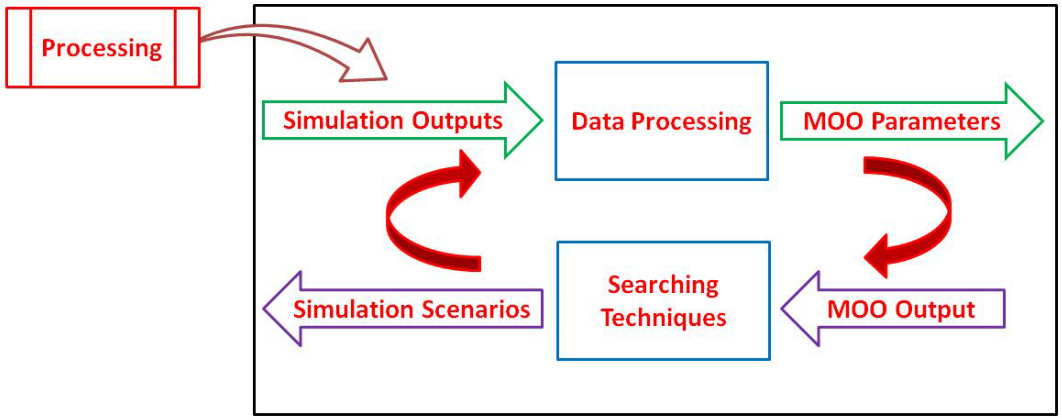

Figure 4.4: MOO Leading Integrated Structure

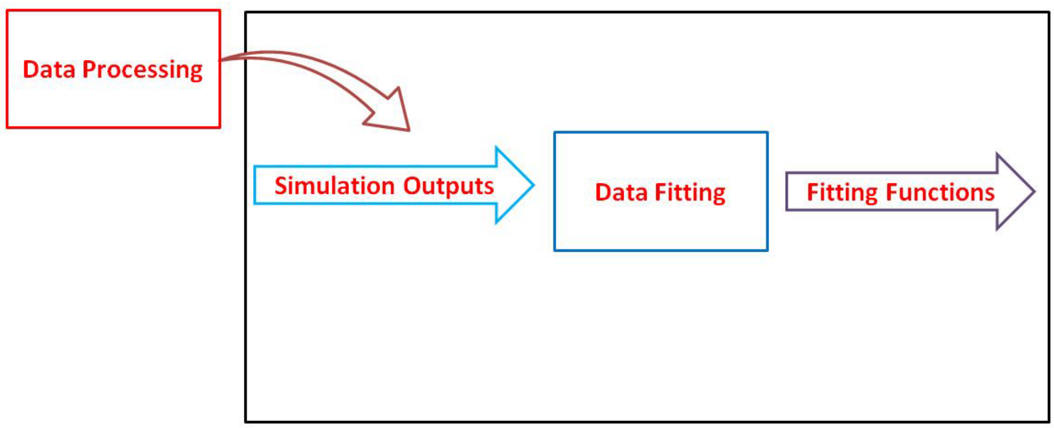

Figure 4.5: Data Processing

the processed data from simulation results.

MOO leading integrated structure is employed in chapter 7, therefore, the specific definitions and methods for "Data Processing" and "Searching Techniques" on the graph (4.4) need to be defined.

(1) "Data Processing"

The data processing is the process to convert the data stream of simulation output on the graph (4.4) into data stream that can be accepted by multi-objective optimisation. Figure (4.5) describes the internal structure of "Data Processing" for the MOO leading integrated structure. Data stream is processed in "Data Processing" box and converted from simulation outputs to multi-objective optimisation parameters as shown in figure (4.4). In other words, the data stream of simulation output is converted to acceptable data stream to multi-objective optimisation. Data fitting is employed to process the data stream. Data fitting is a method to find the mathematic function that have the shortest distance to original data point in geometric space. Many models have been proposed and used for fitting and analysing dielectric-system or conductive-system frequency response data [93]. As a main method to quantify experimental data for single objective formulations in this thesis, data fitting is widely applied in many areas. Yoshimoto et al. [143], Michalowicz and Vlaic [96], Smets et al. [118] and Yoshimoto et al. [144]. The method of least squares is a method to find the best fitted fitting coefficients for base functions to minimise the sum of squared residuals, which is one of methods for data fitting and is mentioned in papers by Ma and Kruth [92], Bode and Shannon [13], Hanbay et al. [44] and Baylar et al. [8].

In the graph (4.5), the models start from simulation which will generate large amounts of data stream (simulation outputs) going through "Data Fitting" box and then being converted to a series of fitting coefficients and base functions be the parameters for multi-objective optimisation models.

As the combination framework starts from simulation, the initial decision variables need to be pre-set to run the simulation. For the problem to be solved in chapter 7, decision variables are the numbers of quay cranes, yard cranes and internal trucks. N_{InVal} denotes the number of initial values. $InValQc_i$, $InValYc_i$ and $InValTr_i$ denote the i^{th} initial values of decision variables for the numbers of quay cranes, yard cranes and internal trucks respectively, where $i \in (0, N_{InVal}]$. $InValQc_i$, $InValYc_i$ and $InValTr_i$ are positive integers. Their equations are given below.

$$InValQc_i = \frac{i \cdot (maxN_{qc} - minN_{qc})}{N_{InVal} + 1} + minN_{qc} \tag{4.1}$$

$$InValYc_i = \frac{i \cdot (maxN_{yc} - minN_{yc})}{N_{InVal} + 1} + minN_{yc} \tag{4.2}$$

$$InValTr_i = \frac{i \cdot (maxN_{tr} - minN_{tr})}{N_{InVal} + 1} + minN_{tr} \tag{4.3}$$

Where $maxN_{qc\,j}$ denotes the upper bound of N_{qc}; $minN_{qc\,j}$ denotes the lower bound of N_{qc}; $maxN_{yc\,j}$ denotes the upper bound of N_{yc}; $minN_{yc\,j}$ denotes the lower bound of N_{yc}; $maxN_{tr\,j}$ denotes the upper bound of N_{tr}; $minN_{tr\,j}$ denotes the lower bound of N_{tr}. $InValQc_i$, $InValYc_i$ and $InValTr_i$ are integers so they need to be rounded to integers if they are decimals.

(2) "Searching Techniques"

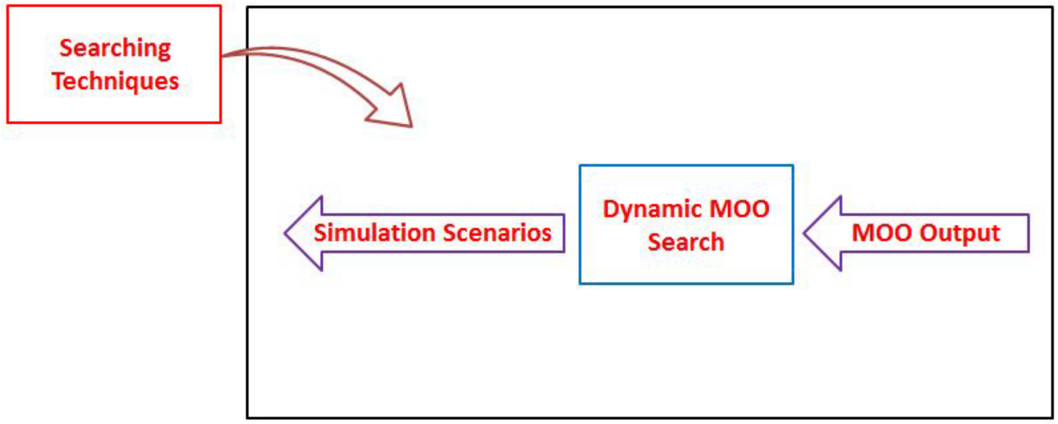

Figure 4.6: Dynamic MOO Search

As shown by figure (4.4), multi-objective optimisation runs when it accepts parameters from data streams from the simulation output. The output stream from multi-objective optimisation goes into the “Searching Techniques” to detect the next values of the decision variables. An algorithm of dynamic MOO search is employed to search the best values for decision variables as shown in figure (4.6). Dynamic MOO search denotes a dynamic search mechanism in the system utilising the output data stream from multi-objective optimisation to dynamically search the near Pareto optimal solutions as seen from the graph (4.6). The multi-objective optimisation functions for the search are dynamically updated by the data stream from the “Data Processing” as new data is continuously added into the data stream. Dynamic MOO search explores the near best values for decision variables for the simulation to start the next iteration. It is expected that the values for decision variables obtained from the dynamic MOO search are quickly close to good values as the search lead by the predicted results from the dynamic MOO search.

(3) Iteration stopping criteria

As the “Data Processing” (defined as data fitting) and “Searching Techniques” (defined as dynamic MOO search) iterate to detect the near best values for decision variables, namely the numbers of terminal mechanic equipment, it is necessary to set appropriate stopping criteria to terminate the loop. The stopping criteria for the problem in chapter 7 are set as follows.

- Predicted values of decision variables obtained from the search already exist in

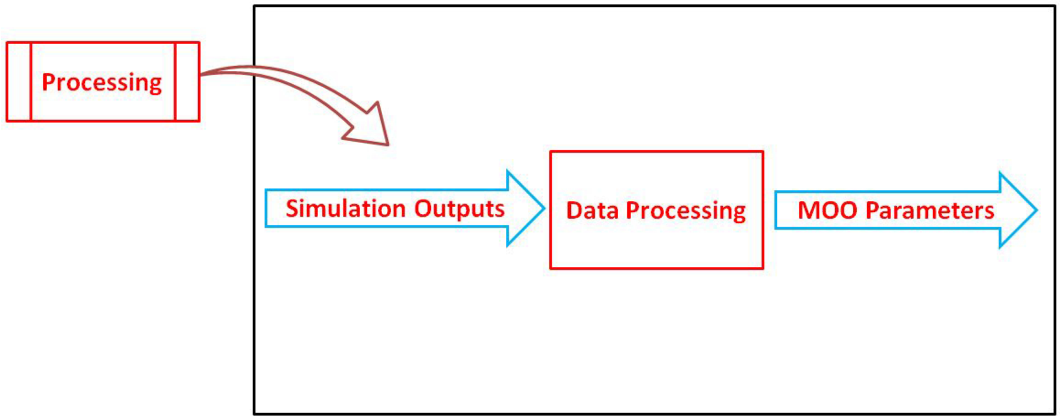

Figure 4.7: Post-MOO Combination Structure

searching list.

- The search could not find better solutions in a certain number of consecutive iterations comparing to the best existed value(s).

4.3.3 Post-MOO Structure

Post-MOO structure denotes that multi-objective optimisation is executed posterior to simulation in the combination framework for the purpose of better presentation of realistic system changes and making appropriate scenarios for multi-objective optimisation in this structure. The models start from simulation. The parameters for simulation are pre-set by collecting historical and (or) the previous experimental data. The results from simulation models are provided for multi-objective optimisation scenarios. Furthermore, the solutions to multi-objective optimisation models are then solved by searching the Pareto frontier.

The details of “Processing” box in figure (4.1) for this structure are demonstrated in figure (4.7). The data stream from simulation goes into the “Data Processing” box in the middle and then goes to multi-objective optimisation.

Post-MOO structure is employed in chapters 5 and 6 to solve internal truck quantity optimisation problem and multiple terminal equipment quantity optimisation problem respectively. Additionally, the definition of “Data Processing” is given in section 4.3.2 that data fitting is used to process data streams.

4.4 Summary

Simulation is a potentially appropriate tool to model container terminals and provide a great deal of data the next step optimisation. On the other hand, multiple objectives are involved in daily management in container terminals such as cost related objectives and productivity related objectives. Multi-objective optimisation is a powerful tool to explore the Pareto optimal solutions for multiple objective problems which have been proved in other areas. In this chapter, a combination framework is proposed to integrate the strengths of the two methods and three combination structures are also proposed for the two methods to work under the combination framework. As MOO leading integrated structure and post-MOO structure are applied in the truck quantity optimisation problem and multiple equipment quantity optimisation problem in chapters 5, 6 and 7, the internal structure of MOO leading integrated structure and post-MOO structure are defined respectively.

Chapter 5

Combining Simulation and Multi-Objective Optimisation by Post-MOO Structure for Container Terminal Truck Quantity Optimisation

5.1 Introduction

Marine transportation has been quickly developing and modernising over the past decades. Larger scale operations, namely a large amount of containers have to be loaded, unloaded and transhipped within a short time span in container terminals, therefore present more requests for higher standards of operations. On the other hand, the limitations of equipment resources and pressure of cost have been an increasingly important problem for terminal operators. Comparing to the controllability of cost and utilisation of land resource, terminal equipment scheduling is more controllable for daily management. The mechanical equipment quantities in terminals are related to the service productivity, equipment utilisation rate, acquisition cost, maintenance cost and labour cost. A good combination of different equipment quantities in daily equipment scheduling avoids the bottle neck effect in multi-threaded operations and, in the meantime, saves expense on equipment. Therefore, the mechanical equipment optimisation is an important decision to be made, which aims to improve terminal productivity and reduce congestion and cost. A container terminal usually has both internal and external trucks and they assume different responsibilities of transporting containers inside and outside the yard respectively. The internal trucks, also called tractors, are a kind of specially designed and dedicated trucks for internal transportation and pre-marshalling within terminals and mainly take on the most of yard operations including pre-marshalling, loading, pre-loading, discharging and pre-discharging[87]. Therefore, a container terminal usually needs a certain number of internal trucks to maintain its internal operations. The quantity scheduling for internal trucks is a daily decision that a terminal has to make. As one of the factors affecting the terminal productivity, cost and traffic, the number of internal trucks assigned to operational tasks in a terminal may vary on operation volumes, traffic situation, total truck quantity, etc.

This is a multi-objective problem and the relevant factors considered by decision makers include terminal productivity and cost goals. To be detailed, there are five objectives discussed in this chapter and they are the operational efficiency requirements, high utilisation rate of the trucks, avoidance of congestion in the yard, fuel consumption goals and low labour demand goals. These objectives are likely to conflict as the cost related goals clash with the productivity related goals.

The method used in this chapter is a combination of multi-objective optimisation and simulation modelling. This chapter also employs discrete-event simulation for container terminals to build a systematic model, covering the terminal gatehouse, yard and berth, including the terminal mechanical equipment: internal truck, external truck, yard crane and quay crane, through the analysis of modern terminal techniques of production processes, management and operations. In this chapter, a post multi-objective optimisation combination framework is used to combine these two methods in order to test a realistic set of scenarios for the model and provide a good representation of the flow of goods through container terminals.

This chapter is organised as follows. First of all, the introduction, problem description and methodology are addressed in section 5.1. Section 5.2 introduces the background and operations of modern container terminals. Section 5.3 gives a description of simulation models, multi-objective optimisation models and a framework to combine both of them. Section 5.4 addresses the parameters and solutions to simulation models, single objective models and multi-objective optimisation models, based on the Southampton Container Terminal. Finally, the conclusions are given in section 5.5.

5.2 Background Information

The geographical structure of a container terminal includes quayside (where berths locate), yard(for container storage), gatehouse(the entrance for external trucks) and depot(for container storage outside the yard). As discussed in section 2.4, the main terminal mechanical equipment includes quay cranes, yard cranes and trucks. Subdividing yard cranes include straddle carriers, sprinters, reach stackers and empty container handlers, while trucks include internal trucks and external trucks. The operations in a container terminal can be grouped into three types of basic processes: loading operations, discharging operations and pre-marshalling operations. So a typical operational flow of container terminals is described as follows:

- Loading operations: containers for exportation are carried into and stored in the terminal yard from outside. When container ships come, these containers are moved from the yard to the quayside and then loaded onto container ships.
- Discharging operations: the containers on container ships are discharged from ships onto the container yard. For those non-transhipping containers, they are then carried out of terminal to their owners or clients. For transhipping containers, they are stored in the terminal yard and then loaded onto other container ships for re-exportation.
- Pre-marshalling operations: an effective action to utilise the loading/unloading operations and it sorts containers according to certain priorities in advance so that the actual processing times of loading/unloading operations can be decreased [52].

From a geographical point of view, the buffer areas for temporary container storage are normally located in the intermediate zones between the terminal quayside and yard, or between the gatehouse and yard. The external trucks carry containers from outside terminals into the yard and vice versa. On the other hand, the internal trucks carry containers within the yard, from container blocks to other container blocks, from container blocks to the quayside buffer areas, from the gatehouse buffer areas to container blocks, etc. Besides, for the internal trucks, the basic operational processes include waiting for new tasks, driving to destinations, loading operations and discharging operations. The operations of the internal trucks can be described by the following sequence:

All the tasks are put into a task list. The internal trucks wait for a new task and they are put into the task waiting queue. The terminal control system matches the new tasks with the idle internal trucks in the waiting queue. If both the new task list and task waiting queue are not empty, they are matched by the control system. Once matched, tasks are sent to trucks which are driven to the assigned positions and enter a crane operation waiting queue. The terminal control system matches the trucks in the queue with the available cranes. The yard crane operations are loading containers onto the internal trucks and discharging them from trucks. Then the loaded and/or discharged trucks are driven the next assigned locations and enter another crane operation waiting queue. The system matches them again and the trucks in queues wait for the next operations. When they finish their tasks, these trucks go back to the task waiting queue and wait for a new task.

On one hand, from a cost control point of view, the truck quantity is supposed to be kept at a minimum level to lessen the purchasing cost, equipment maintaining cost and labour cost. However, on the other hand, from the point of view of productivity improvement, the truck quantity is in proportion to the operational efficiency to a certain extent, therefore, it is supposed to be maintained at a maximum level. Additionally, the truck quantity also affects the terminal traffic and equipment utilisation rate. This chapter aims to seek a trade-off between these objectives.

5.3 Model Description

The major modelling processes in this section are: building a simulation model and a multi-objective optimisation model, proposing a post multi-objective optimisation framework for combination, and an exploration of the Pareto optimal solutions to the model.

5.3.1 Post-MOO Structure

In consideration of the strengths and weaknesses of simulation and multi-objective optimisation, the combination of both of them guarantees the Pareto optimality of solutions, reduces overall computational time if they are combined efficiently and ensures a realistic model of flow under random conditions.

In this chapter, the truck quantity optimisation for container terminals is based on the post-MOO structure, namely the multi-objective optimisation is conducted posterior to the terminal simulation, as given in section 4.3.3. The data fitting is used to process the simulation output as shown in figure (4.7). The simulation results provide a good representation of the operations and are used to be processed for the parameters for single objective models. The multi-objective optimisation model proposed consists of a formulation for five objectives which are all non-linear. The decision variables of the model have upper bounds and lower bounds. All of variables in the model are non-negative. Finally, the Pareto optimal solutions are explored through the multi-objective optimisation model.

5.3.2 Discrete-event Simulation

Discrete-event simulation concerns the modelling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in the time [71]. A system is a collection of related elements and relationships amongst these elements. The states of these elements in a system constitute the states of a system. In a discrete-event system, system states are only changed when new events occur, where the events are those behaviours changing systematic element states, and therefore changing the system states, in other words, the system states are not changed in the period between two contiguous events. And these events happen at random and discrete time points, therefore the system step sizes are also random. Discrete-event simulation is particularly apt to describe the inner working processes of a terminal and it can be employed to calibrate a continuous black-box model for the terminal [115]. A discrete-event simulation model for truck quantity optimisation in container terminals is built in the simulation integration environment of the Micro Saint Sharp simulation package [130]. Micro Saint Sharp is a general purpose tool that can be used to provide solutions ranging from queuing problems involving hospital waiting rooms to complex human decision processes involving future command and control systems [12]. Micro Saint Sharp is a powerful simulation tool for the problems to be solved in this thesis because:

- It provides a visual interface and multiple simulation modules for productive model creation.

- It supports a variety of data types.

- It is fast and flexible with the support of C# coding.

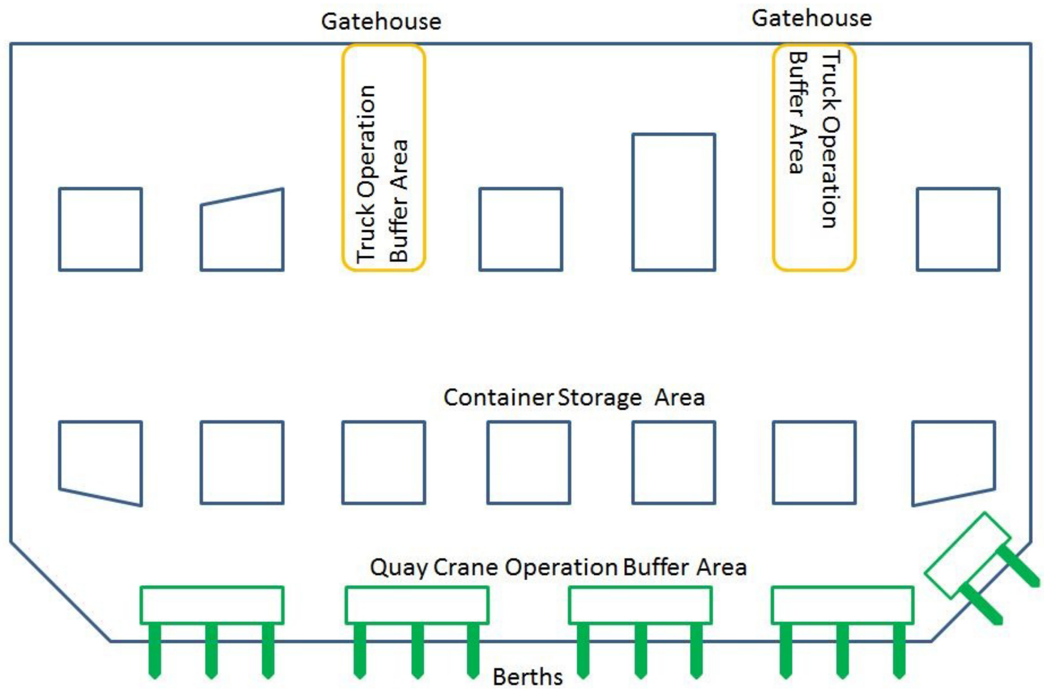

Figure 5.1: Terminal Layout [70]

- It has low requirements on system. A computer with a 90-Megahertz Intel Pentium-class processor, a 64 MB RAM and 150 MB of free hard disk space is able to run the package [112].

In the Micro Saint Sharp simulation package, the "Release Condition" is where controls events to be processed. Conditions are pre-set in the "Release Condition" and events are processed only when events meet the conditions.

The layout of a general container terminal is given in figure (5.1). Berths are the places along the quays where vessels moor and cargos are loaded to and (or) discharged from. Containers are discharged to the inland route transportation and (or) loaded to the water route transportation through sea ports. Gatehouses, where connect terminals to the hinterland, are the other side of sea ports comparing to berths. Containers are unloaded from container ships and transported the inland transportation such as road and railway transportation at the gatehouses and then delivered to customers or transported from the inland transportation and loaded onto container ships. The container terminal yard is the buffer area between berths and gatehouses, and has facilities for temporary storage of containers. Yard operations include discharging of containers from vessels, loading of containers onto vessels, shuffling

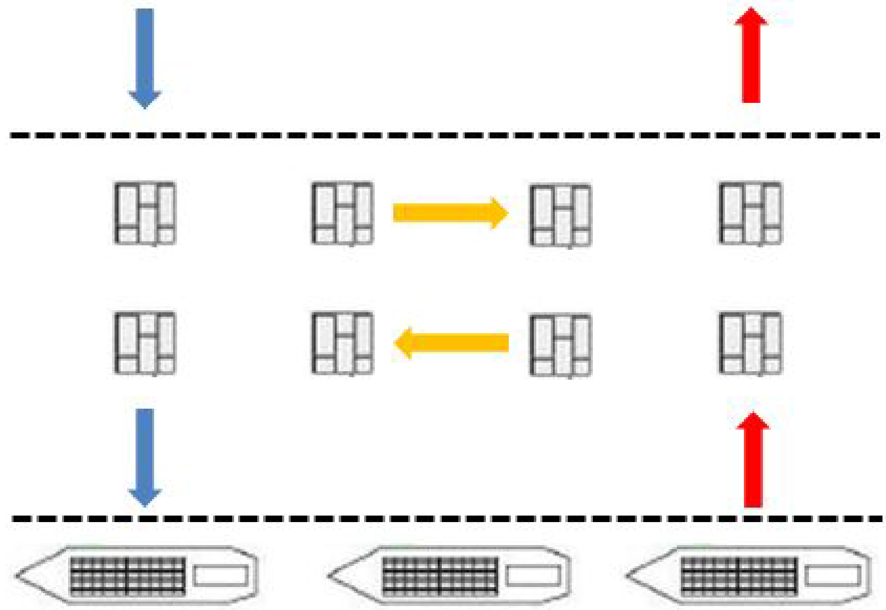

Figure 5.2: Operational Flow Chart

of containers in container blocks, re-distribution of containers to other blocks (yard shifting) for more efficient loading onto the next vessel and inter-terminal haulage when containers need to be moved to other yards in another terminal [34].

A general operational flow chart is shown in figure 5.2. In a general container terminal, the whole terminal operations can be classified to following three modules:

- Loading process module (LDM): loading containers from the yard onto container ships as shown by the down arrows in figure 5.2;
- Unloading process module (ULDM): unloading containers from container ships to the yard as shown by the up arrows in figure 5.2;
- Pre-marshalling process module (MSHM): moving containers from original positions to new positions in the yard as shown by the left and right arrows in figure 5.2.

In the simulation model, the operations of internal trucks can be broken down and summarised to following 8 basic operations:

- Assigning trucks (AT): matching the tasks from the task list with the trucks from the free truck list;

- Truck transportation (TT): empty trucks moving to the assigned destinations for operations;
- Transporting laden containers (TL): trucks transporting laden containers;
- Transporting empty containers (TE): trucks transporting empty containers;
- Pickup laden containers (PL): Cranes picking up laden containers from the yard or trucks;
- Pick up empty containers (PE): Cranes picking up empty containers from the yard or truck;
- Grounding laden containers (GL): Cranes grounding laden containers onto the yard or trucks;
- Grounding empty containers (GE): Cranes grounding empty containers onto the yard or trucks.

As shown by equation (5.1), R is a set of 8 basic operations listed above. AT and TT are indispensable for all operational modules, while the value of each pair of $PL||PE$, $TL||TE$ and $GL||GE$ has to be either one of them.

$$R = [AT, TT, PL||PE, TL||TE, GL||GE] \tag{5.1}$$

There are several lists in the simulation model to control the equipment allocation: waiting task list, executed task list, free truck list, occupied truck list, free yard crane list, occupied yard crane list, free quay crane list and occupied quay crane list.

At the beginning of the simulation, all equipment is in idle state, i.e., the task list is empty. At the same time, all of the trucks, yard cranes and quay cranes are in the free truck list, free yard crane list and free quay crane list respectively. When the simulation begins, new operation tasks are given by the terminal control centre and these tasks go into the waiting task list. Then, the terminal control centre assigns trucks from the free truck list to the new tasks. If the free truck list is not empty, the control system matches the free trucks with the new tasks, namely assigning available trucks with new tasks. Otherwise, the new tasks enter the waiting task list and wait for free trucks. Once the free trucks are assigned to new tasks, they are transferred from the free truck list into the occupied truck list, and in the meantime, these tasks are transferred from the waiting task list to the executed task list. Then destination positions are sent to truck drivers, where trucks load from or discharge to or move to. Then the empty trucks without containers are driven to the appointed locations and enter queues waiting for crane operations. If the free crane list is not empty, the cranes from the list are assigned to the trucks and the tasks will be executed. Trucks then carry containers to their next appointed destinations and enter queues waiting for next crane operations. Finally, if the free crane list is not empty, containers are picked up from trucks and grounded onto ships or the yard or external trucks. Equation (5.1) is a general set that describes the above operational flow. In equation (5.2), R' is a set of 8 basic operation actions.

$$R' = [AT, TT, PL, PE, TL, TE, GL, GE] \tag{5.2}$$

$$\begin{aligned} &U = [u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8] \\ &u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8 \in \{0, 1\} \\ &u_3 + u_4 \leq 1, u_5 + u_6 \leq 1, u_7 + u_8 \leq 1 \end{aligned} \tag{5.3}$$

In equation (5.3), U is a binary coefficient matrix, where the value 0 means the according operation is not included in the according operational module. While the value 1 means the operation is applicable to the according operational module. In equation (5.1), $PL||PE$, $TL||TE$ and $GL||GE$ are alternative operations, which means that only one of two operations is executed.

$$LDM = ULDM = [U, U]^T \; [R', R'] \tag{5.4}$$

$$MSHM = U^T \; R' \tag{5.5}$$

As shown by equation (5.4), both of the loading and unloading operations contain two operation set R'. But there is only one R' for the pre-marshalling operations in equation (5.5). A simulation model is built in the Micro Saint Sharp simulation package based on equations (5.4) and (5.5) and the modelling network is shown in figure 5.3. The loading processes are picking up containers from the external trucks in the buffer areas (i.e. "Loading-pickup from truck(19)" in figure 5.3), grounding them onto the yard ("Loading-ground onto yard(20)"), then picking up from the yard ("Loading-pickup from the yard(9)") and lifting onto container ships ("Loading-ground onto the ship(11)"). The discharging processes indicate that picking up containers from container ships ("Discharging-pickup from the ship (12)"), grounding them onto the yard ("Discharging-ground onto the yard(13)"), then picking up from the yard ("Discharging-pickup from the yard(15)") and grounding onto the buffer areas ("Discharging-ground onto truck or train(17)"). The pre-marshalling processes involve a basic operation of R' which is picking up containers from their original locations in the yard ("Remarshaling-pickup from the yard (24)") and then moving them to other positions in the yard ("Remarshaling-ground onto yard(26)"). Therefore, the non-zero operation elements in equations (5.4) and (5.5) are those corresponding tasks in figure 5.3. To satisfy the constraints in equation (5.3), the coefficients of equations (5.4) and (5.5) are shown in table 5.1.

Table 5.1: The coefficients of equations (5.4) and (5.5)

Operation Type	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
Empty container operations	1	1	1	0	1	0	1	0
Laden container operations	1	1	0	1	0	1	0	1

The simulation parameters for this model include stochastic and non-stochastic parameters.

A discrete-event simulation system reflects the interactions amongst a number of stochastic factors and repetitively deals with randomness, such as the number of events, the occurrence time, the number of entities generated, etc. These stochastic variables have different probability distributions and provide parameters for the system by random sampling, such as task interval time, crane operation time, truck travelling time, etc. Tasks interval time is denoted by X_1; truck assigning time is denoted by X_2; truck travelling time is denoted by X_3; the time of quay crane pick-up operation from the yard is denoted by X_4; the time of quay crane grounding operation to the yard is denoted by X_5; the time of yard crane pick-up operation from the yard is denoted by X_6; the time of yard crane grounding operation to the yard is denoted

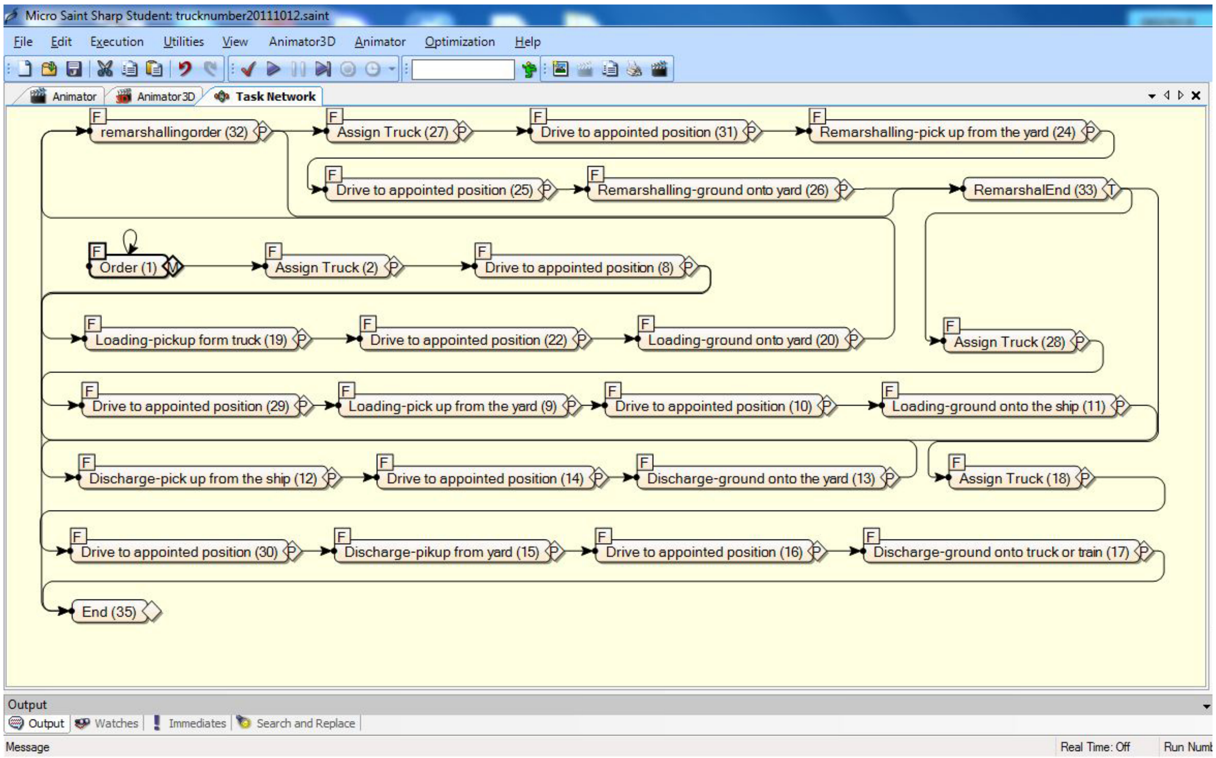

Figure 5.3: General simulation network

by X_7. The probability distribution function for the i^{th} stochastic variable is denoted by $X_i \sim F_i(x)$. θ_i denotes a set of distribution parameters for the i^{th} stochastic variable. The input of parameters include the selection of probability distribution function $F_i(x)$ and estimation of function parameters θ_i. The values of stochastic variables involved in this simulation modelling are generated by the different distributions by the system, such as task interval time distribution, truck assigning time distribution, truck travelling time distribution, quay crane pick-up operation time distribution, quay crane grounding operation time distribution, yard crane pick-up operation time distribution and yard crane grounding operation time distribution.

On the other hand, the non-stochastic parameters are those parameters related to terminal layout, operation processes, equipment quantities and properties, loading/discharging ratio, empty/laden container ratio, transhipment/all container ratio, pre-marshalling probability, simulation duration and the number of simulation iterations.

The output parameters of the simulation model are given out in table 5.2.

Table 5.2: Simulation Output Parameters

Output Parameter	Definition
$N_{optedcntr}$	Number of containers operated by quay cranes
$T_{qc\ m\ i}$	Operational time of m^{th} container by i^{th} quay crane
$Tidle_{qc\ m\ i}$	Idle or waiting time when m^{th} container is operated by i^{th} quay crane
$T_{tsk\ i\ j}$	Time of executing task by i^{th} truck in j^{th} shift
$T_{wo\ i\ j}$	Time of waiting for tasks of i^{th} truck in j^{th} shift
$T_{cr\ i\ j\ k}$	Operational time of k^{th} crane handling j^{th} container when i trucks are employed in the terminal yard
$T_{ladc\ i\ j\ k}$	Operational time of k^{th} laden container of j^{th} task by i^{th} truck
$T_{epyc\ i\ j\ k}$	Operational time of k^{th} empty container of j^{th} task by i^{th} truck
$T_{etr\ i\ j\ k}$	Operational time of k^{th} route of j^{th} task for i^{th} truck
$T_{wkr\ i\ j\ k}$	Working hours of k^{th} worker on i^{th} truck in j^{th} shift

5.3.3 Formulation for Objectives

Container terminal operators, in order to provide first class and competitive service to their clients and yield a highest possible profit for their companies, normally consider two aspects of targets: maintaining lowest cost and pursuing highest productivity. The maintaining cost in daily operations includes fuel consumption and labour cost. The operational efficiency might involve the equipment service rate for clients, equipment utilisation rate and traffic situation. However, the cost objectives and productivity objectives are conflicting to a certain extent. The increase of number of trucks, on one hand enhances the service quality and equipment utilisation rate, on the other hand aggravates cost burden and might cause traffic congestion problems in the yard. Contrarily, the decrease of number of trucks has an opposite effect because the efficiency and cost are interacted and restrained with each other. Given any level of conflict among the objectives, which is a commonplace occurrence, not all of them can be simultaneously optimised [32]. Therefore, the primary focus of this chapter is to find trade-offs amongst these five objectives to satisfy both client requirements and cost control. Five individual objective functions are discussed respectively in this section and a multi-objective optimisation model is presented by the end. These five objectives are:

1. Terminal quay crane efficiency

2. Truck utilisation rate

3. Traffic congestion probability

4. Unit fuel consumption

5. Unit labour cost

First Objective Function: Terminal Quay Crane Efficiency

For the first objective, optimising the terminal operational efficiency is one of very important goals for container terminals as the quality of client services is core competitiveness in a highly competitive market. Regarded as an interface accept containers from and transfer containers to container ships, the performance of quay cranes for container ships is usually highlighted against other terminal operations and considered as a core competence because of the requirements upon loading and discharging time due to the intensive voyages of container vessels. 'Moves per crane-hour' is a common productivity measure for container terminal quay cranes [73], therefore in this chapter, the quay crane performance is quantified as a terminal operational efficiency measure.

The quay crane operation rate is a dependent variable affected by various factors, such as operational time, truck waiting time, equipment idle time. In this case, it is less realistic to describe the relationships between these variables and the dependent variables in a pure mathematical model because even the number of these factors is not certain, while the combination of mathematical models and simulation models is a more feasible method to model the terminal operational efficiency. The container ship operation rate data R_{qc} are derived from table 5.2 output parameters, where R_{qc} is the quay crane operation rate and it is expressed as:

$$R_{qc} = \frac{N_{optedcntr}}{\sum_{i=1}^{N_{qc}} \sum_{m=1}^{N_{ops_i}} (T_{qc\,m\,i} + Tidle_{qc\,m\,i})} \tag{5.6}$$

Where N_{qc} is the number of quay cranes and N_{ops_i} is the number of operation for the i^{th} quay crane. $N_{optedcntr}$, $T_{qc\,m\,i}$ and $Tidle_{qc\,m\,i}$ have been declared in table 5.2. Assume equation (5.7) is a data set which describes the relationships between R_{qc} and N_{tr}(the number of internal trucks).

$$((R_{qc_1}, N_{tr_1}), (R_{qc_2}, N_{tr_2}), \cdots, (R_{qc_m}, N_{tr_m})) \tag{5.7}$$

Where m is the number of data elements in equation (5.7). Assume the fitting function $\varphi(x) = \alpha_1\varphi_1(x) + \alpha_2\varphi_2(x) + \cdots + \alpha_n\varphi_n(x)$ is an expression for variable N_{tr} and dependent variable R_{qc} based on the data set (5.7). Where α_j is a fitting coefficient; $\varphi_j(x)$ is a base function $x^{n'}$; n denotes the number of equation terms; $n' = 1, 2, 3, \cdots, n$. The base function set $\{\varphi_1(x), \varphi_2(x), \cdots, \varphi_n(x)\}$ is determined on the basis of the distribution features of data (5.7). $f_{rqc}(x)$ denote a function mapping N_{tr_m} to R_{qc_m} in the data set (5.7). The residual values between $\varphi(x)$ and $f_{rqc}(x)$ are denoted by $\triangle_{x_i}$ which is described as an error function: $\| \triangle_{x_i} \|^2 = \sum_{i=1}^{m}[\varphi(x_i) - f_{rqc}(x_i)]^2$. Fitting functions are to find the best fit functions for experimental data, i.e., a minimum distance to the original functions in the geometric space for residual values. Therefore, the fitting coefficient set $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ is determined by the minimum value of $\| \triangle_{x_i} \|^2$, i.e. $\frac{\partial}{\partial\alpha}\| \triangle_{x_i} \|^2 = 0$, which is the least squares method shown as follows.

The first objective function is denoted by equation (5.8) which is to be maximised:

$$f_{eff}(x) = \sum_{j=1}^{n} \varphi_j(x)\alpha_j \tag{5.8}$$

Where f_{eff} denotes quay crane service rate; j denotes the j^{th} term of the equation; n is the number of terms of the equation, $n \in N$; $\varphi_j(x)$ denotes the j^{th} base function. The constraints for the first objective function are: $R_{qc} > 0$, $N_{optedcntr} > 0$, $N_{qc} > 0$, $N_{ops_i} > 0$, $T_{qc\,m\,i} > 0$, $Tidle_{qc\,m\,i} > 0$.

$(\varphi_k(x_i), \varphi_j(x_i))$ is the inner product of $\varphi_k(x_i)$ and $\varphi_j(x_i)$, i.e. $(\varphi_k(x_i), \varphi_j(x_i)) = \sum_{i=1}^{m} \varphi_k(x_i)\varphi_j(x_i)$. Where k and j denotes the k^{th} and j^{th} term of the equation; m is the number of data elements which equals to the number of equation terms. Then the inner product of $\varphi_k(x_i)$ and $f_{rqc}(x_i)$ is denoted by $(\varphi_k(x_i), f_{rqc}(x_i))$. Equation (5.9) can be deduced from the above formula:

$$\sum_{i=1}^{n}(\varphi_k, \varphi_i)\alpha_i = (\varphi_k, f_{rqc}) \tag{5.9}$$

$$\begin{bmatrix} (\varphi_1, \varphi_1) & (\varphi_1, \varphi_2) & \cdots & (\varphi_1, \varphi_n) \\ (\varphi_2, \varphi_1) & (\varphi_2, \varphi_2) & \cdots & (\varphi_2, \varphi_n) \\ \vdots & \vdots & & \vdots \\ (\varphi_n, \varphi_1) & (\varphi_n, \varphi_2) & \cdots & (\varphi_n, \varphi_n) \end{bmatrix} \times \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} (\varphi_1, f_{rqc}) \\ (\varphi_2, f_{rqc}) \\ \vdots \\ (\varphi_n, f_{rqc}) \end{bmatrix} \tag{5.10}$$

Equation (5.10) is a matrix for the system of linear equations deduced from equation (5.9). The best-fit fitting coefficient set α is then deduced from equation (5.10).

Second Objective Function: Truck Utilisation Rate

The acquisition cost normally is a large portion of the terminal budget, therefore the improvement of truck utilisation rate and reduction of the purchased number of trucks are goals for terminal operators. T_{rtr} denotes the total running time of all trucks.

$$T_{rtr} = \sum_{i=1}^{N_{tr}} \sum_{j=1}^{N_{shf\ i}} T_{tsk\ i\ j} \tag{5.11}$$

Where $N_{shf\ i}$ is the number of shifts for the i^{th} truck during the simulation; $T_{tsk\ i\ j}$ and $T_{wo\ i\ j}$ below are defined in table 5.2. T_{ttlrtr} denotes a sum of truck running time and waiting time, i.e. the total simulation time.

$$T_{ttlrtr} = \sum_{i=1}^{N_{tr}} \sum_{j=1}^{N_{shf\ i}} (T_{tsk\ i\ j} + T_{wo\ i\ j}) \tag{5.12}$$

The average utilisation rate of internal trucks is quantified in equation (5.13), namely the proportion total truck running time accounts for.

$$f'_{utl} = \frac{T_{rtr}}{T_{ttlrtr\ i}} \cdot 100\% \tag{5.13}$$

The second objective function is denoted by equation (5.14), which is to be maximised and fitted by the data set $((f'_{utl_1}, N_{tr_1}), (f'_{utl_2}, N_{tr_2}), \cdots, (f'_{utl_n}, N_{tr_n}))$.

$$f_{utl}(x) = \sum_{j=1}^{n} \varphi_j(x)\alpha_j \tag{5.14}$$

$\varphi_j(x)$ is determined by simulation data and α_j is fitted by equation (5.10). The constraints for the second objective function are: $N_{tr} > 0$, $N_{shf\ i} > 0$, $T_{tsk\ i\ j} > 0$, $T_{wo\ i\ j} > 0$.

Third Objective Function: Traffic Congestion Probability

For the third objective, traffic congestion is another factor considered by terminal decision makers. In order to increase the land area for container storage in the yard, terminals often reduce the other area of roads and unnecessary space between container blocks, hence, traffic congestion may happen in a bounded space, especially in peak hours. As the container terminal yard has a capacity to contain a certain number of internal trucks operating in the yard, if truck quantity exceeds this equilibrium point, the traffic situation in the yard is likely to get heavy. Assume that every single truck moves in the yard at the same and a uniform velocity, therefore the distances between two trucks keep the same throughout until they stop for queuing up. If the times that trucks need to spend on travelling these distances amongst trucks in the yard are longer than crane operational times, queues do not exist in the system and traffic is smooth. Conversely, if the times are shorter than crane operational times, stopping and queuing up occur in the system, and then the former truck will affect the latter, therefore congestion happens. Distances amongst internal trucks are quantified by time. The arrival and departure times of each truck are recorded, and then the time interval between any two trucks can be described as time distance between them. Let the average operational time of cranes be the ideal value of time intervals of internal trucks, then accordingly, if time intervals are less than this value, trucks might need stopping and queuing up, namely the probability of waiting and queuing will then increase. On the contrary, if the time intervals are greater than this value, the probability of waiting and queuing will correspondingly decrease. X is the time interval between a random truck and its following truck at the beginning of their tasks and assume X is exponentially distributed, i.e. $X \sim E(\lambda)$, i.e. $p(x) = \lambda e^{-\lambda x}$. The mathematical expectation $E(X) = \frac{1}{\lambda}$ is equal to the average value of operational time per container, namely:

$$\lambda = \frac{thrpt \cdot R_{thrcntr} \cdot N_{tr\ j}}{T_{thr}}$$

Where *thrpt* is the throughput of the container terminal during a period T_{thr}; $R_{thrcntr}$ denotes the ratio of container number to container throughput; $N_{tr\ j}$ denotes different truck numbers in scenarios, where j denotes a positive integer from 1 to n. In equation (5.15), $T'_{cr\ i}$ denotes the total crane operational time when i trucks are employed. $T_{cr\ i\ j\ k}$ is defined in table 5.2.

$$T'_{cr\ i} = \sum_{j=1}^{N_{qc}+N_{yc}} \sum_{k=1}^{N_{cntr\ j}} T_{cr\ i\ j\ k} \tag{5.15}$$

Where N_{qc} is the number of quay cranes; N_{yc} is the number of the yard cranes; $N_{cntr\ j}$ denotes the container number operated by j^{th} crane during simulation time. The average value of $T'_{cr\ i}$ is denoted by $T_{cr\ i}$, which is the average crane operational time for a container, i.e.

$$T_{cr\ i} = \frac{T'_{cr\ i}}{\sum_{k=1}^{N_{qc}+N_{yc}} N_{cntr\ j}} \tag{5.16}$$

$f_{ts}(x)$ denotes the fitting function for $T_{cr\ i}$, namely

$$f_{ts}(x) = \sum_{j=1}^{n} \varphi_j(x)\alpha_j \tag{5.17}$$

If the interarrival time between two consecutive trucks is longer than the crane operational time of the former truck, then the former operation finishes before when the latter truck arrives, namely the latter is operated once it arrives at the position without waiting in a queue. Conversely, the latter stops and waits in a queue for the next operation. In another word, if the crane operational time $f_{ts}(N_{tr\ j})$ is longer than the truck departure time interval X_i, i.e. $X_i < f_{ts}(N_{tr\ j})$, then the i^{th} truck arrives at a crane when the previous truck is still in its operation. The i^{th} truck then needs to wait in a queue. And the latter trucks might also need to wait in queues if their arrival times are before the operation finishing time of their former trucks. The probability of $X_i < f_{ts}(N_{tr\ j})$ is denoted by $P\{X_i < f_{ts}(N_{tr\ j})\}$ which is also the objective function for the third objective shown by equation (5.18) which is to be minimised.

$$f_{con}(N_{tr\ i}) = P\{X_i < f_{ts}(N_{tr\ j})\} = \int_{-\infty}^{f_{ts}(N_{tr\ j})} \lambda e^{-\lambda x}\, dx \tag{5.18}$$

The constraints for the third objective function are: $thrpt > 0$, $R_{thrcntr} > 0$, $N_{tr\ j} > 0$, $T_{thr} > 0$, $N_{qc} > 0$, $N_{yc} > 0$, $N_{cntr\ j} > 0$, $T_{cr\ i\ j\ k} > 0$.

Fourth Objective Function: Unit Fuel Consumption

For the fourth objective, fuel consumption of trucks varies with different truckloads such as a truck carrying containers consumes more fuel than an empty truck and a truck carrying a laden container costs more fuel than carrying an empty container. b_1, b_2 and b_3 are the fuel consumption coefficients for empty trucks, trucks carrying empty containers and trucks carrying laden containers respectively. Therefore, the truck moving time multiplying its fuel coefficient is equal to its fuel consumption volume. Equation (5.19) is the function for truck fuel consumption cost.

$$\begin{aligned} f''_{fcost} &= b_1 \sum_{i=1}^{N_{tr}} \sum_{j=1}^{N_{ord\ i}} \sum_{k=1}^{N_{ladops\ i\ j}} T_{ladc\ i\ j\ k} \\ &+ b_2 \sum_{i=1}^{N_{tr}} \sum_{j=1}^{N_{ord\ i}} \sum_{k=1}^{N_{eptops\ i\ j}} T_{eptc\ i\ j\ k} \\ &+ b_3 \sum_{i=1}^{N_{tr}} \sum_{j=1}^{N_{ord\ i}} \sum_{k=1}^{N_{etrops\ i\ j}} T_{etr\ i\ j\ k} \end{aligned} \tag{5.19}$$

Where f''_{fcost} denotes the weighted truck operational time; $T_{ladc\ i\ j\ k}$, $T_{eptc\ i\ j\ k}$ and $T_{etr\ i\ j\ k}$ are defined in table 5.2; $N_{ord\ i}$ is the number of orders(tasks) of the i^{th} truck; $N_{ladops\ i\ j}$ is the number of operation times of laden containers of the j^{th} order of the i^{th} truck; $N_{eptops\ i\ j}$ is the number of operation times of empty containers of the j^{th} order of the i^{th} truck; $N_{etrops\ i\ j}$ is the number of operation times of empty trucks of the j^{th} order(task) of the i^{th} truck. $f'_{fcost}(x)$ is the average operational time for both quay cranes and yard cranes defined by equation (5.20).

$$f'_{fcost}(x) = \frac{f''_{fcost}(x)}{N_{tr}} \tag{5.20}$$

The data set between average fuel consumption and truck numbers is denoted by $((f'_{fcost_1}, N_{tr_1}), (f'_{fcost_2}, N_{tr_2}), \cdots, (f'_{fcost_n}, N_{tr_n}))$. The fourth objective function, which is to be minimised, is denoted by equation (5.21) and $\varphi_j(x)$ is determined by the data features and α_j is fitted by equation (5.10).

$$f_{fcost}(x) = \sum_{j=1}^{n} \varphi_j(x)\alpha_j \tag{5.21}$$

The constraints for the fourth objective function are: $b_1 > 0$, $N_{tr} > 0$, $N_{ord\ i} > 0$, $N_{ladops\ i\ j} > 0$, $T_{ladc\ i\ j\ k} > 0$, $b_2 > 0$, $N_{eptops\ i\ j} > 0$, $T_{eptc\ i\ j\ k} > 0$, $b_3 > 0$, $N_{etrops\ i\ j} > 0$, $T_{etr\ i\ j\ k} > 0$.

Fifth Objective Function: Unit Labour Cost

For the fifth objective, labour cost for truck quantity optimisation involves wages paid for terminal staff. The working time $T_{wkr\ i\ j\ k}$ (defined in table 5.2) multiplying the unit wage is equal to the unit labour cost, which is shown in equation (5.22).

$$f''_{lab} = \sum_{i=1}^{N_{tr}} \sum_{j=1}^{N_{shf\ i}} \sum_{k=1}^{N_{wkr\ i\ j}} W \cdot T_{wkr\ i\ j\ k} \tag{5.22}$$

where f''_{lab} denotes the total labour cost; W is the unit wage; $N_{wkr\ i\ j}$ denotes the number of workers on the i^{th} truck in the j^{th} shift. In a modern terminal, each truck is normally provided at least two workers including a driver and a tally clerk, however, in some cases this number could be one or more than two. f'_{lab} is the average labour cost per container shown in the following equation.

$$f'_{lab} = \frac{f''_{lab}}{N_{tr}} \tag{5.23}$$

The fifth objective function, which is to be minimised, is denoted by equation (5.24), which is fitted by the data set $((f'_{lab_1}, N_{tr_1}), (f'_{lab_2}, N_{tr_2}), \cdots, (f'_{lab_n}, N_{tr_n}))$. $\varphi_j(x)$ is determined by the data features and α_j is fitted by equation (5.10).

$$f_{lab}(x) = \sum_{j=1}^{n} \varphi_j(x)\alpha_j \tag{5.24}$$

The constraints for the fifth objective function are: $N_{tr} > 0$, $N_{shf\ i} > 0$, $N_{wkr\ i\ j} > 0$, $W > 0$, $T_{wkr\ i\ j\ k} > 0$.

5.3.4 Multi-objective Optimisation

Marler and Arora [94], Chanas and Kuchta [17], Chen and Lee [21], Goh et al.[41] and Miettinen [97] present their multi-objective optimisation functions. The multi-objective optimisation function is based on a classical multi-objective programming formulation proposed by Miettinen [97] which is described as follows:

$$\begin{aligned} &\textit{minimise} \quad && \{f_1(x), f_2(x), \cdots, f_k(x)\} \\ &\textit{subject to} \quad && x \in S \end{aligned} \tag{5.25}$$

There are $k(k \geq 2)$ objective functions $f_i : R^n \to R$. The vector of objective functions are denoted by $f(x) = (f_1(x), f_2(x), \cdots, f_k(x))^T$. The decision vector is denoted by

$x = (x_1, x_2, \cdots, x_n)^T$ and $x \in R^n$, where R^n is the decision variable space. S is a series of constraints. For this problem, the decision vector is $x = (N_{tr})^T$ and the vector of objective functions is $f(x) = (-f_{eff}(x), -f_{utl}(x), f_{con}(x), f_{fcost}(x), f_{lab}(x))^T$.

$$\begin{aligned} &minimise \quad && \{-f_{eff}(x), -f_{utl}(x), f_{con}(x), f_{fcost}(x), f_{lab}(x)\} \\ &subject\ to \quad && x \in S \end{aligned} \tag{5.26}$$

5.4 Results and Discussion

The above section discusses building truck quantity decision models for container terminals and this section discusses the execution of proposed models based on the Southampton Container Terminal.

5.4.1 Simulation Parameters and Results

Although the public data from the Southampton Container Terminal are collected, processed and imported for the simulation model, some data is unavailable due to commercial confidentiality reasons. Therefore, those unavailable data need to be estimated to set the simulation scenarios. For the stochastic parameters, Demirci [30], Parola and Sciomachen [109] and Shi et al. [117] have discussion on using exponential distribution to describe the interarrival time in container terminals. Kim and Kim [66] and Lee et al. [77] use the gamma distribution for crane operational times in container terminals. The distribution functions and parameters of stochastic variables are estimated and listed in table 5.3. Therefore, assume that X_1 in table 5.3 follows an exponential distribution and X_2, X_3, X_4, X_5, X_6 and X_7 follow gamma distributions. In table 5.3, θ_1 is:

$$\theta_1 = \frac{simulationtime}{(percThrputToCntrNum \cdot throughput/365)}$$

Where *simulationtime* denotes the length of simulation duration in hour. *percThrput ToCntrNum* denotes the ratio of container number to throughput volume. The detailed operational data for X_1, X_2, X_3, X_4, X_5, X_6 and X_7 from the British container terminal is unavailable due to its confidentiality to the terminal as it is a business organisation. Therefore the simulation parameters θ_2, θ_3, θ_4, θ_5, θ_6 and θ_7 can not be obtained from historical data. An appropriate estimation for the distribution parameters is therefore important to the model. Because of the unavailability of detailed data from the Southampton Container Terminal, the values of the distribution parameters in this chapter are estimated by the knowledge of container terminal operations and the parameter settings from the papers by Hadjiconstantinou and Ma [43], Yun and Choi [145], Ambrosino and Tanfani [3], Shi et al. [117] Rizzoli et al. [115] and Zhu [149]. The parameters can be re-set as long as the realistic data is available. The distribution parameters are set as follows: $\theta_2 = \theta_1$, $\theta_3 = 3.0$ minutes, $\theta_4 = 4.0$ minutes, $\theta_5 = 2.0$ minutes, $\theta_6 = 2.5$ minutes, $\theta_7 = 2.0$ minutes.

Table 5.3: Stochastic Parameters for Simulation

Stochastic Parameters	$F_i(x)$	θ_i
New task interval time distribution	$X_1 \sim E(x)$	θ_1
Truck assigning time distribution	$X_2 \sim \Gamma(x)$	$(\theta_2, \theta_2/2)$
Truck travelling time distribution	$X_3 \sim \Gamma(x)$	$(\theta_3, \theta_3/2)$
Time distribution of quay crane pick-up operation from the yard	$X_4 \sim \Gamma(x)$	$(\theta_4, \theta_4/2)$
Time distribution of quay crane grounding operation to the yard	$X_5 \sim \Gamma(x)$	$(\theta_5, \theta_5/2)$
Time distribution of yard crane pick-up operation from the yard	$X_6 \sim \Gamma(x)$	$(\theta_6, \theta_6/2)$
Time distribution of yard crane grounding operation to the yard	$X_7 \sim \Gamma(x)$	$(\theta_7, \theta_7/2)$

For the non-stochastic parameters, the values of parameters are given in table 5.4. The data of the container throughput of Southampton Container Terminal is shown in table D.1 in Appendix D. The number of quay cranes is shown by table D.2. The number of yard cranes is given by straddle carriers, sprinters, reach stackers and empty container handlers in table D.3. The "Loading:discharging ratio" in the table is an approximate figure obtained from a meeting with Southampton Container Terminal. The data of "Empty containers:laden containers ratio", "Transhipment containers: all containers ratio" and "Pre-marshalling percentage" is unavailable at the moment but needed to run the model. Therefore, the figures in the table are estimated by personal knowledge and recognition of container terminals, which can be changed according to real data.

The values of $N_{tr\,j}$ are set as follows, where $N_{tr\,j}$ is the value of N_{tr} for the j^{th} simulation iteration, j denotes the j^{th} simulation iteration and N_{SimIte} is the number of simulation iterations.

$$\begin{aligned} N_{tr\ j} = \quad & j \quad \cdot (Upper\ bound\ of\ N_{tr} \\ & - \quad Lower\ bound\ of\ N_{tr})/N_{SimIte} \\ & + \quad Lower\ bound\ of\ N_{tr} \end{aligned} \tag{5.27}$$

The simulation time in table 5.4 needs to be as short as possible to save computa-

Table 5.4: Non-Stochastic Parameters for Simulation [86] [120] [113] [122]

Parameter	Value
Container throughput	Data from 2000 to 2009
Quay crane quantity	Data from 2000 to 2009
Yard crane quantity	Data from 2000 to 2009
Loading:discharging ratio	6:4
Empty containers:laden containers ratio	3:7
Transhipment containers: all containers ratio	0.5:10
Pre-marshalling percentage	20%
Simulation time	10 hours
Upper bound of N_{tr}	400
Lower bound of N_{tr}	40
Value of N_{SimIte}	19

tional time, but on the other hand, needs be sufficient in order to reach the expected accuracy level. It is inevasible that simulation normally has noisy data especially at the warm-up stage. The instability of stochastically variables might affect simulation results. Therefore, in order to get rid of these noisy data, a sufficient simulation period is a must to guarantee the accuracy of results. However, from the point of view of computational cost saving, it is better to shorten the simulation time. In this thesis, simulation time and simulation period indicate the simulation time to be simulated in the real world, while computational time indicates the time taken to implement simulation models on computers. In order to shorten the simulation time to a minimum level to reduce the computational time but maintain the expected accuracy level, experiments are implemented to test the value of simulation period. According to equation (5.27) and table 5.4, 20 experiments are implemented to test the stability of simulation results and variant parameters are obtained from equation (5.27). Simulation results are shown in figures (5.4), (5.5), (5.6), (5.7) and (5.8), while data 1 to 20 on the graphs are the results from simulation experiments 1 to 20 with different

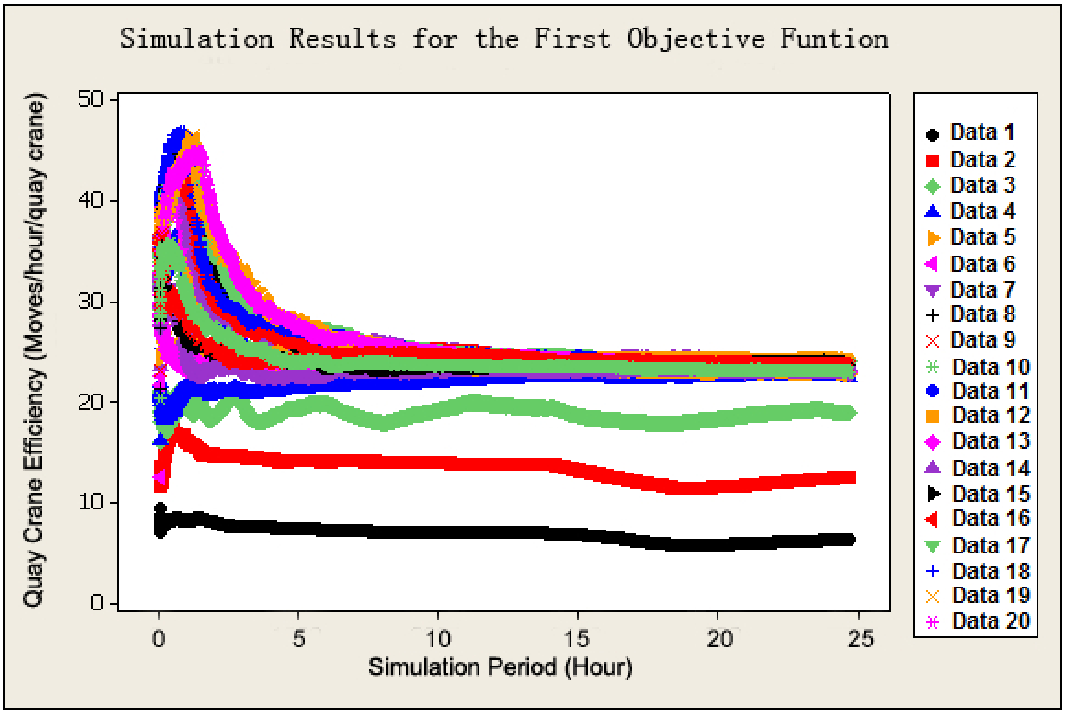

Figure 5.4: Simulation Results for the First Objective Function

variant values from equation (5.27). For the first objective, in figure (5.4), noisy data is mainly in the beginning 5 hours and data become smoother afterwards. Instable data in figure (5.5) are mainly in the first 4 hours for the second objective. Noisy data in figures (5.6), (5.7) and (5.8) are generated in the first 2 hours. The results from simulation need to be processed and noisy data need to be avoided. Therefore simulation time is supposed to be at least 5 hours to get the stable data. So 10 hours simulation time is set to run the simulation models as shown in table 5.4.

The output of the simulation is provided from the execution of simulation model according to table 5.2 and equations (5.6), (5.15), (5.17), (5.20) and (5.23). The results are given in table 5.5.

5.4.2 Single Objective Formulation Parameters and Results

For equation (5.8) in section 5.3.3, the type of base function set $\{\varphi_1(x), \varphi_2(x), \cdots, \varphi_n(x)\}$ and the value of n are determined by the distribution features of simulation data from table 5.2. Therefore, seeing from data graphs, assume $(n-1)$ powered functions are good representations for the above simulation data for equations (5.8), (5.14), (5.17),

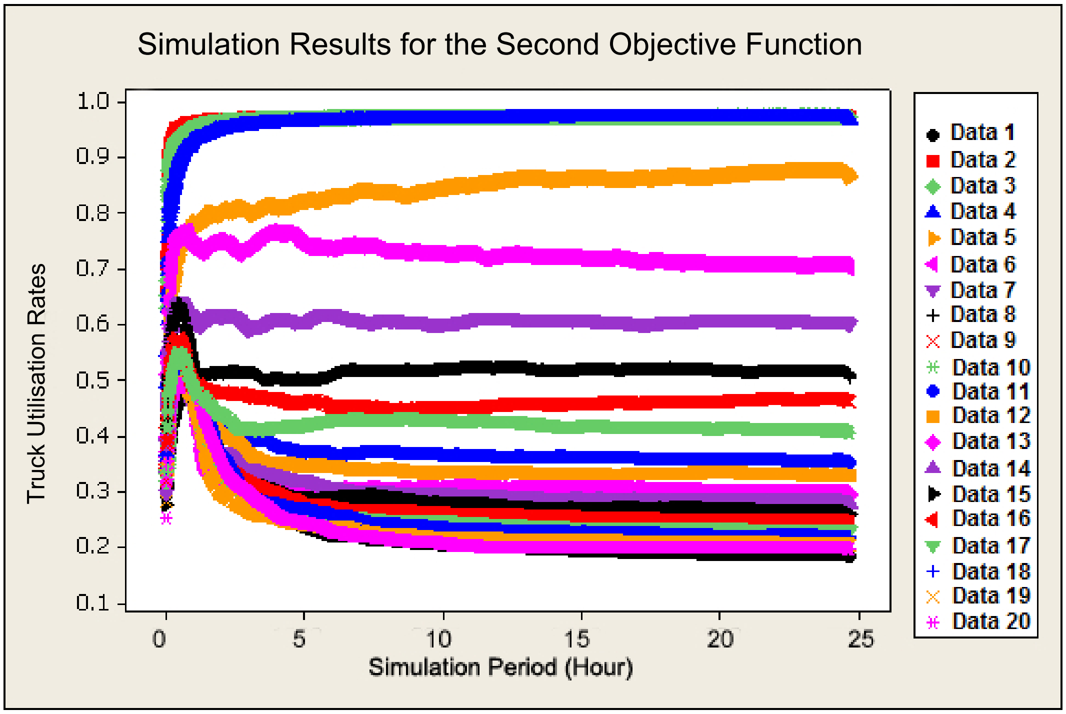

Figure 5.5: Simulation Results for the Second Objective Function

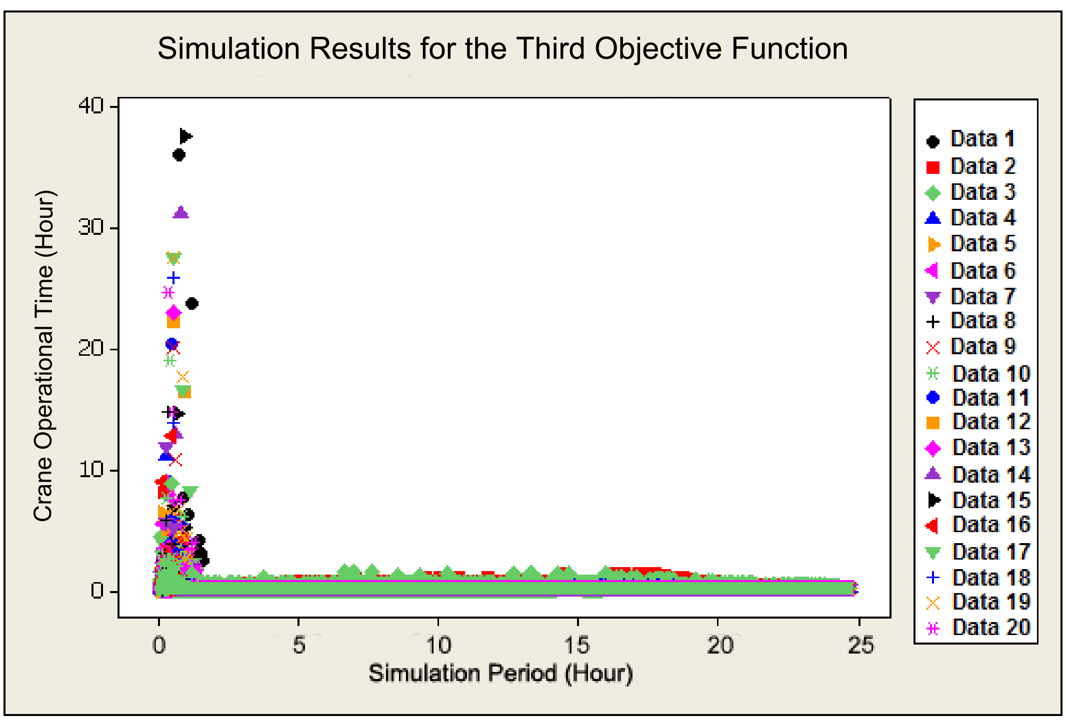

Figure 5.6: Simulation Results for the Third Objective Function

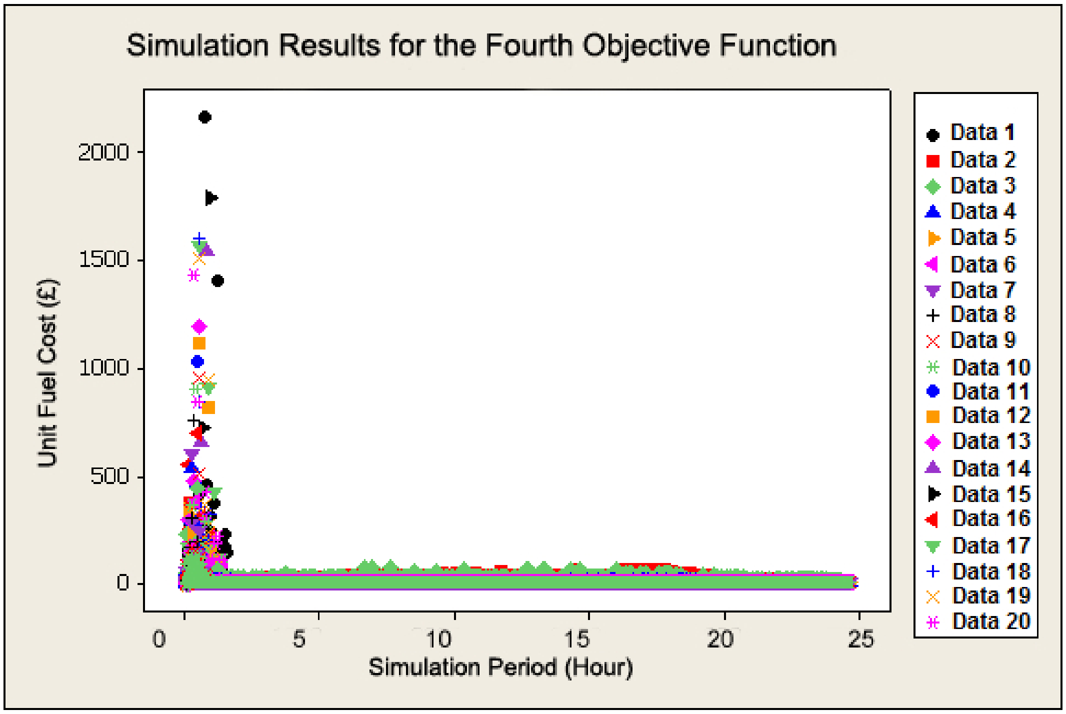

Figure 5.7: Simulation Results for the Fourth Objective Function

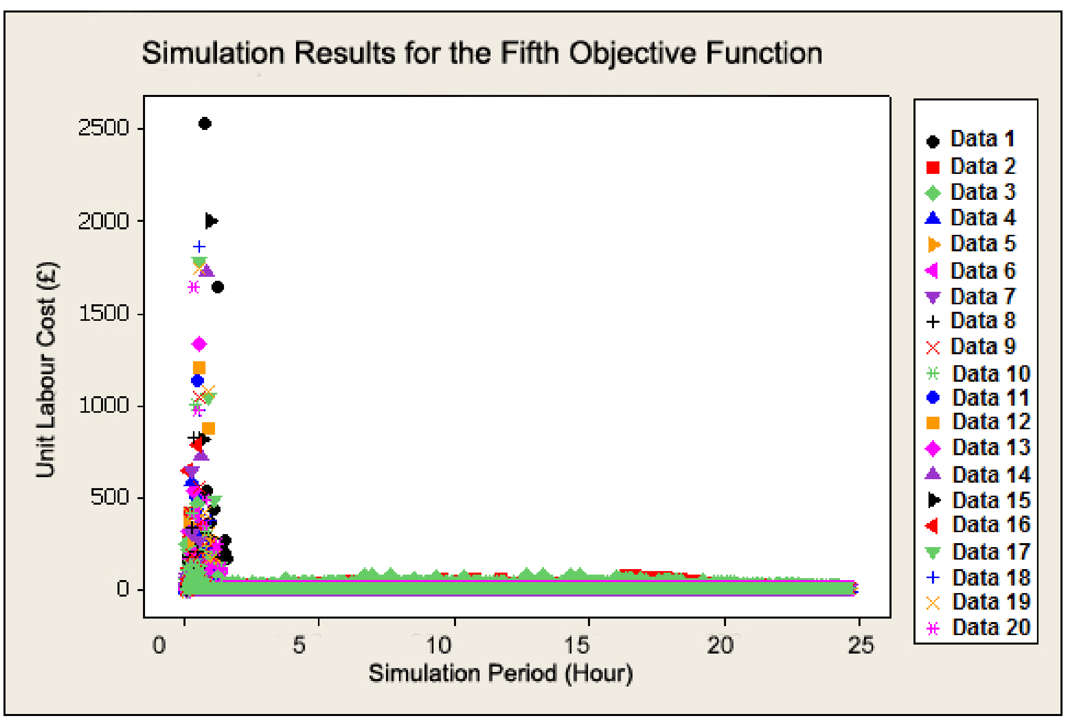

Figure 5.8: Simulation Results for the Fifth Objective Function

Table 5.5: Simulation Results

N_{tr}	R_{qc} (Container/ Crane/Hour)	f'_{utl}	$T'_{cr\ i}$(Hour)	f'_{fcost}(£)	f'_{lab}(£)
40	12.54687315	0.971352718	0.182165646	8.543670125	9.167742329
60	18.89973767	0.965342043	0.171088875	7.972810574	8.564404568
80	22.39256737	0.865179222	0.16744958	7.82420116	8.416769596
100	22.80852256	0.701288155	0.166598276	7.815462336	8.380598942
120	23.21247009	0.598148787	0.16836295	7.900786105	8.479286146
140	22.71417376	0.505343752	0.169210962	7.906908401	8.494977912
160	23.75569816	0.459986691	0.168814357	7.853466927	8.431030495
180	23.48810793	0.404044256	0.168645483	7.844956697	8.431287438
199	22.89956687	0.352800526	0.16611649	7.786144273	8.358429853
220	23.30935551	0.326967563	0.167521601	7.899256638	8.461388431
240	22.84265168	0.292357668	0.16728612	7.854525918	8.435826386
260	23.05794219	0.275840336	0.168112575	7.902267322	8.484674645
280	23.57311275	0.258219027	0.167761641	7.836316177	8.430516841
300	23.09830098	0.238239918	0.166778844	7.831396589	8.427471439
320	23.57766614	0.230908068	0.169162248	7.892895166	8.509722167
340	23.78789388	0.217557498	0.165562909	7.852579818	8.468765462
360	23.60586872	0.203241221	0.16710013	7.847154312	8.434990724
380	23.97224522	0.196332251	0.166657212	7.943286678	8.537419232
400	23.28862359	0.183065927	0.16735693	7.941567596	8.577450368

(5.21) and (5.24), which are shown by following equations respectively.

$$f_{eff}(x) = \sum_{j=1}^{n} N_{tr}^{j-1} \alpha_{j+10} \tag{5.28}$$

$$f_{utl}(x) = \sum_{j=1}^{n} N_{tr}^{j-1} \alpha_{j+20} \tag{5.29}$$

$$f_{ts}(x) = \sum_{j=1}^{n} N_{tr}^{j-1} \alpha_{j+30} \tag{5.30}$$

$$f_{fcost}(x) = \sum_{j=1}^{n} N_{tr}^{j-1} \alpha_{j+40} \tag{5.31}$$

$$f_{lab}(x) = \sum_{j=1}^{n} N_{tr}^{j-1} \alpha_{j+50} \tag{5.32}$$

Numerical experiments have been done to test values of R^2 to find good fitting powers for the functions. The values of n for each above equation are shown in table 5.6 respectively. Furthermore, the data obtained from simulation are input as parameters

Table 5.6: Fitting Powers

$f(x)$	$f_{eff}(x)$	$f_{utl}(x)$	$f_{ts}(x)$	$f_{fcost}(x)$	$f_{lab}(x)$
n	4	3	4	4	4

into the above equations (5.28), (5.29), (5.30), (5.31) and (5.32). The fitting coefficient set α is deduced by equation (5.10) from simulation data. The value of α is computed and exported to equations (5.28), (5.29), (5.30), (5.31) and (5.32). Similarly, the fitting coefficients for the objective functions are given out in table 5.7.

In this container terminal, the number of drivers in a single truck is 2 and the average unit labour cost is £64.5/person/day [90].

5.4.3 Multi-objective Optimisation Results

This is a multiple objective problem which has five objectives, two of which are to be maximised while the other three to be minimised. The objective space is therefore denoted as a five dimensional space O. The variant, namely the truck quantity, is belonging to a decision space D, which in this problem is integer and bounded. Therefore O is accordingly a bounded space because of the variant value range. So

Table 5.7: Fitting Coefficients

α_j	Fitting Values	R^2	α_j	Fitting Values	R^2
	Qc Rate Objective Function: $f_{eff}(x)$			Truck Utilisation Rate Objective Function: $f_{utl}(x)$	
α_{11}	$-1.00608607389221 \times 10^{-8}$		α_{21}	0	
α_{12}	$1.00172577039788 \times 10^{-5}$		α_{22}	$-1.85164922615276 \times 10^{-8}$	
α_{13}	- 0.00350495307286781	0.9464	α_{23}	$2.11313871749685 \times 10^{-5}$	0.9892
α_{14}	0.504609782642747		α_{24}	-0.00836945848863550	
α_{15}	-1.62424413488179		α_{25}	1.34277065923323	
	Average Crane Operational Time Function: $f_{ts}(x)$			Unit Fuel Consumption Objective Function: $f_{fcost}(x)$	
α_{31}	$1.43611966786207 \times 10^{-11}$		α_{41}	$7.34184693898015 \times 10^{-10}$	
α_{32}	$-1.40544943061360 \times 10^{-8}$		α_{42}	$-7.06754055408318 \times 10^{-7}$	
α_{33}	$4.80442768722124 \times 10^{-6}$	0.7824	α_{43}	0.000239076661804849	0.7810
α_{34}	- 0.000670752843376622		α_{44}	-0.0330186354664595	
α_{35}	0.199458851793086		α_{45}	9.38680082103357	
	Unit Labour Cost Objective Function: $f_{lab}(x)$				
α_{51}	$7.63548962659333 \times 10^{-10}$				
α_{52}	$-7.37173666689088 \times 10^{-7}$				
α_{53}	0.000250874980866436	0.7872			
α_{54}	- 0.0348859425028811				
α_{55}	10.0628641598034				

the computational volume to explore all possible solutions in the space O is bounded and acceptable. So, in order to obtain every possible solution, explicit enumeration is employed to explore the solutions on the Pareto frontier. The source codes of explicit enumeration are given in appendix A.

461 solutions are obtained from equation (5.25). The solutions are normalised on axes into a range from 0 to 1, due to the values and value spans of solutions to each objective function are different. Equation (5.33) is used to normalise the solutions, where RS_{ij} denotes the j^{th} normalised solution for the i^{th} objective function; S_{ij} is the j^{th} original solution for the i^{th} objective function; $Smin_i$ is the minimum solution for the i^{th} objective function; $Smax_i$ denotes the maximum solution for the i^{th} objective function.

$$RS_{ij} = \frac{S_{ij} - Smin_i}{Smax_i - Smin_i} \tag{5.33}$$

The normalised solutions to equation (5.25) are shown in figure (5.9). Solution information must be substantial enough to allow an informed decision to be made by the decision maker but not so large as to overwhelm him/her with information [99]. Therefore some representative solutions are selected from figure (5.9) and given in figures (5.10), (5.11), (5.12) and (5.13) for decision supporting. Figure (5.10) shows 6 best solutions to the first objective function, i.e. the terminal quay crane efficiency function. Solutions 1, 2, 3, 4, 5 and 6 in figure (5.10) have the best values for the first objective, very good values for the third objective and also good values for the fourth and fifth objectives, but the values for the second objectives are almost the worst. On figure (5.11), the solutions on the graph are the best for the second and third objective functions, namely the truck utilisation rate and congestion probability. However, the solutions have very poor values for the other three objectives. Solutions on figure (5.12) have the lowest values for the fourth and fifth objectives which are to be minimised. The values for the quay crane performance are very close to 1, while the values on the second and third axis are close to 0.5, i.e. the average value. The above three figures show good solutions to each single objective function, but they do not necessarily provide a balance between all five objectives. This might not be the best option in some circumstances to achieve the best values for one or some objectives at the cost of sacrificing others. On the other hand, if decision makers prefer balanced solutions to each objective, the solutions from figure (5.13) are a choice. Solutions

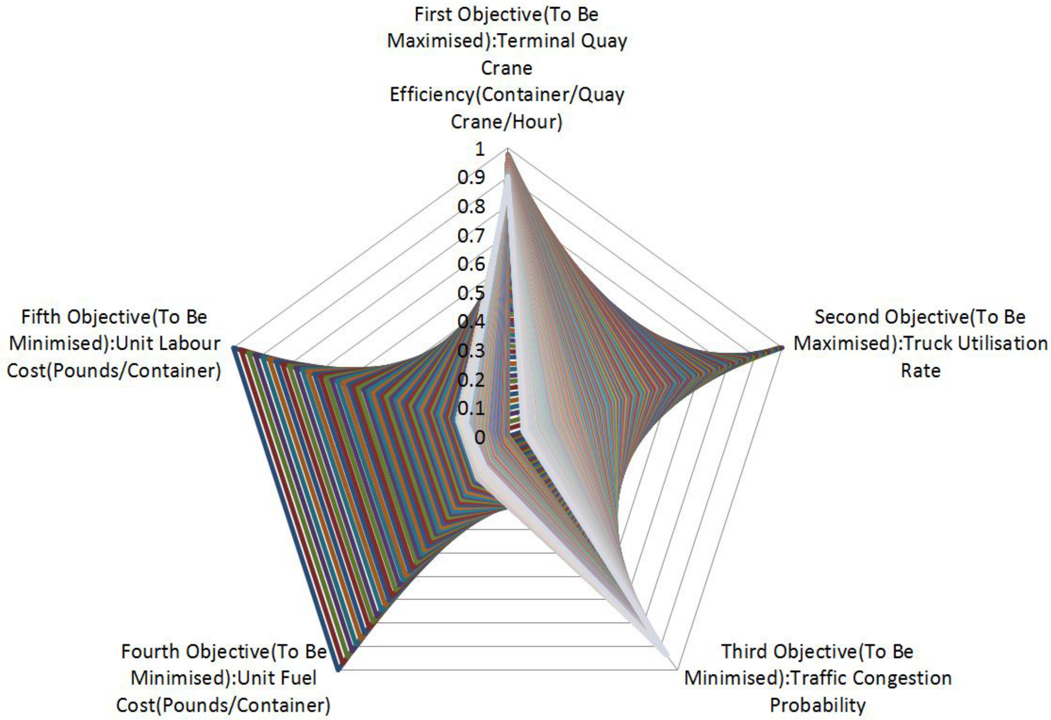

Figure 5.9: Normalised Solutions Graph (Each colour of pentagon shows a solution)

19, 20, 21, 22, 23 and 24 in figure (5.13) have values close to the average value, i.e. 0.5, for the first, fourth and fifth objective functions, with very good values for the second objectives (close to 0.9) and good values for the third objective function (close to 0.2).

On the basis of these solutions, decision makers have some good solutions to choose rather than too much information. They choose appropriate solutions to different situations based on multiple references and at the end the truck quantities can be derived from the selected solutions.

5.4.4 Computational Considerations

The computational time of the optimisation consists of the first stage i.e. simulation computational time and the second stage i.e. multi-objective optimisation computational time. The information of computer hardware to run the models is given as follows: processor: Intel i3 M350 at 2.27 GHz; memory: 3.00GB. Software environment is 64-bit Microsoft Windows 7. The simulation model is run in the Micro

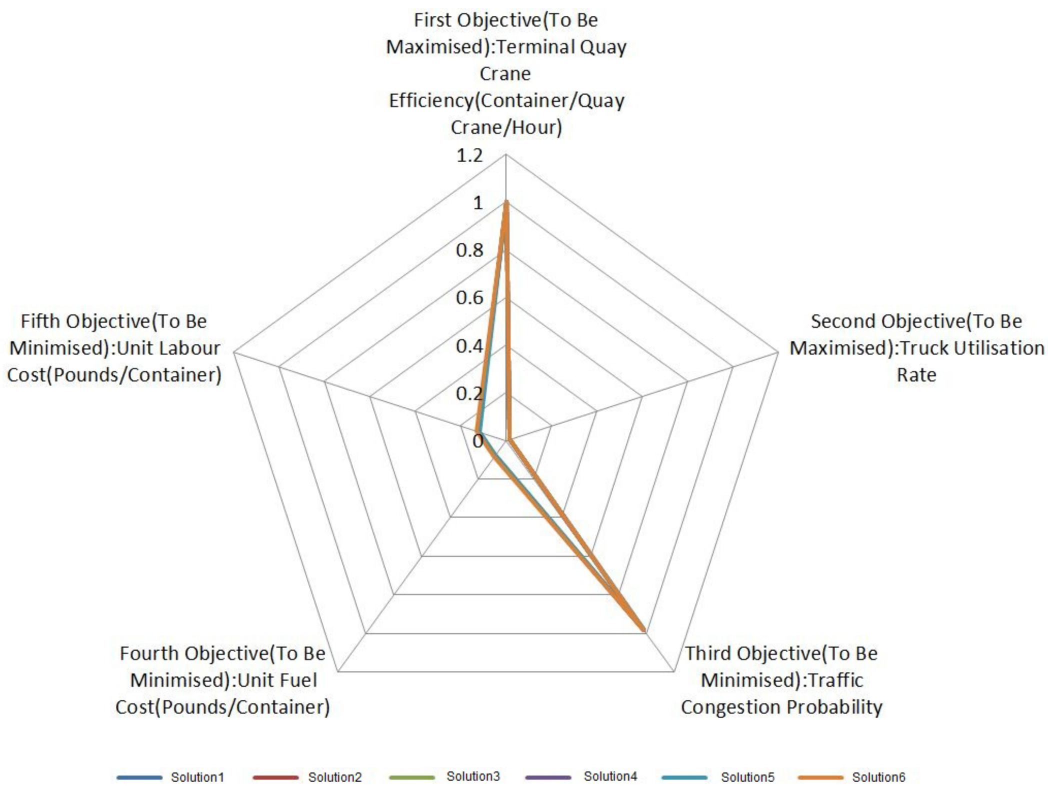

Figure 5.10: The Best Normalised Solutions to the First Objective Function

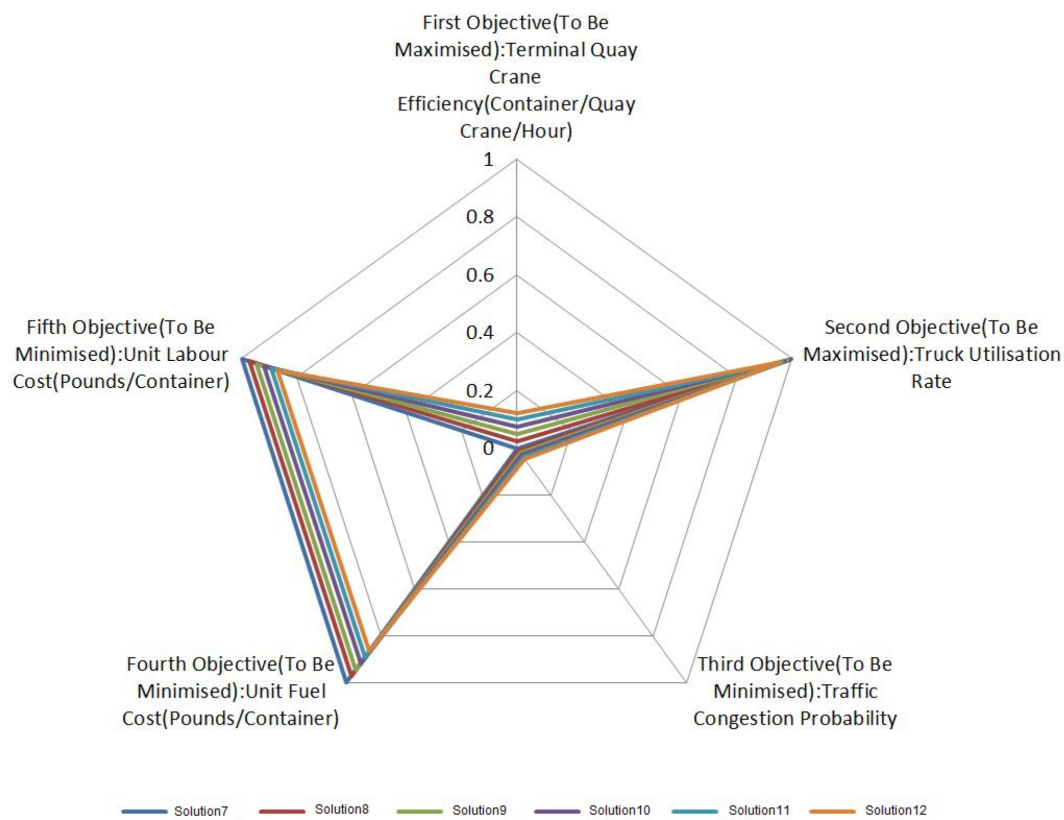

Figure 5.11: The Best Normalised Solutions to the Second and Third Objective Functions

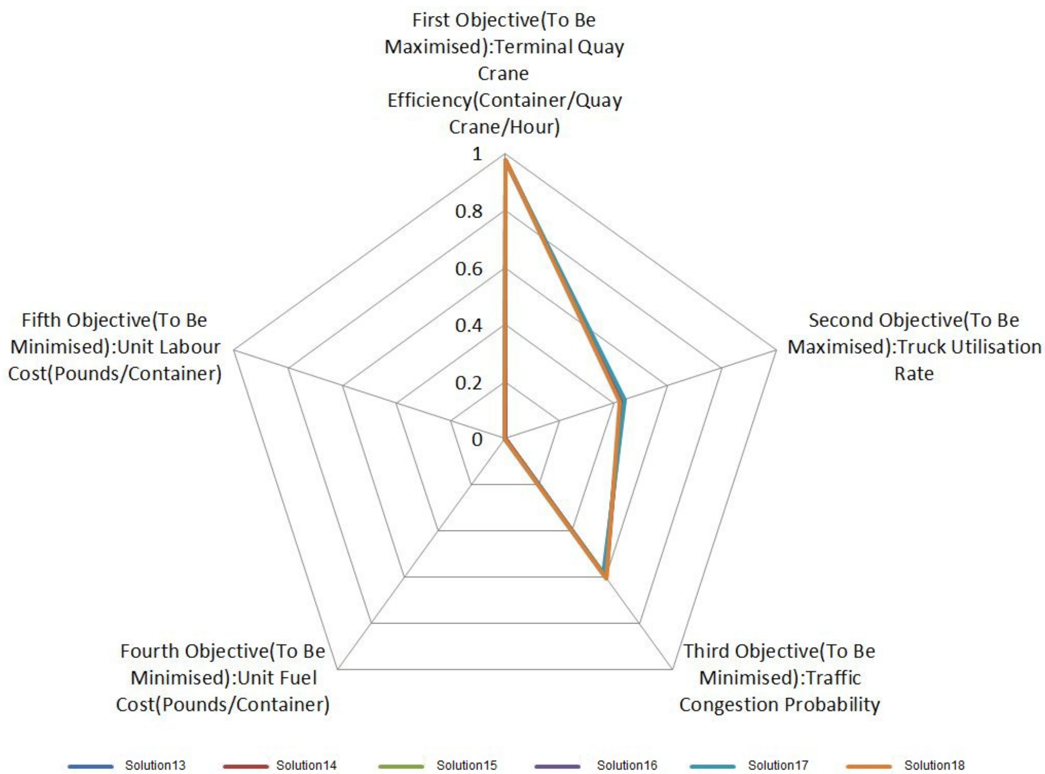

Figure 5.12: The Best Normalised Solutions to the Fourth and Fifth Objective Functions

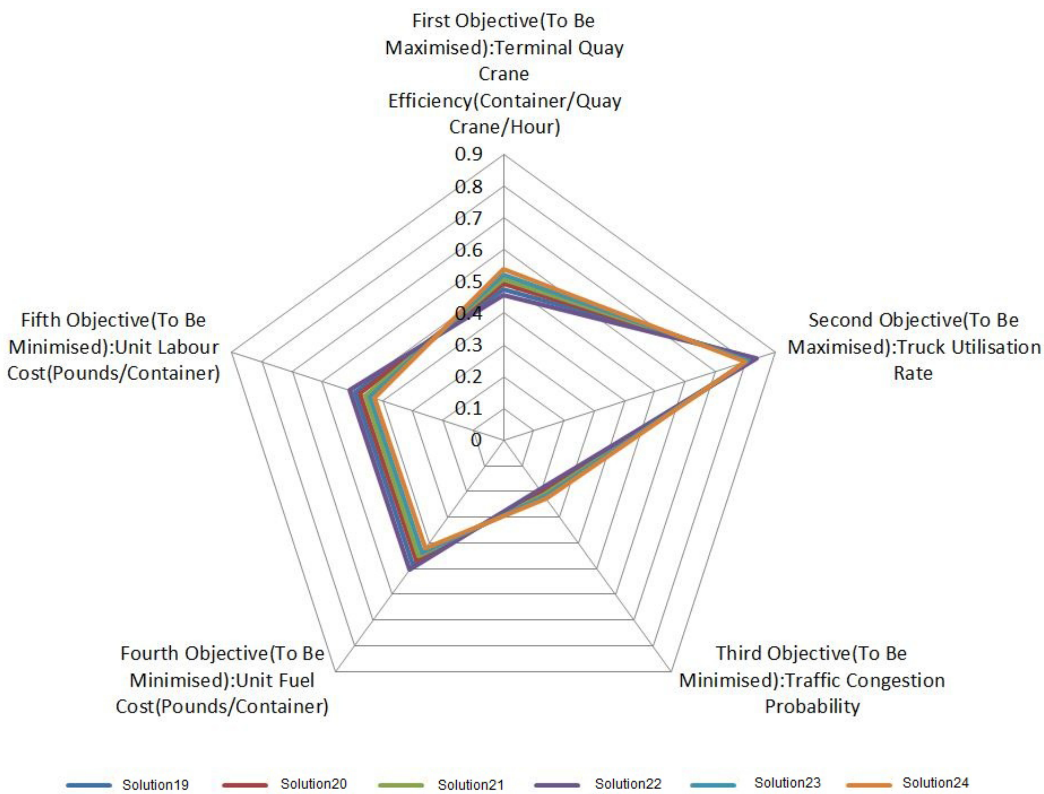

Figure 5.13: The Most Balanced Solutions to Five Objectives

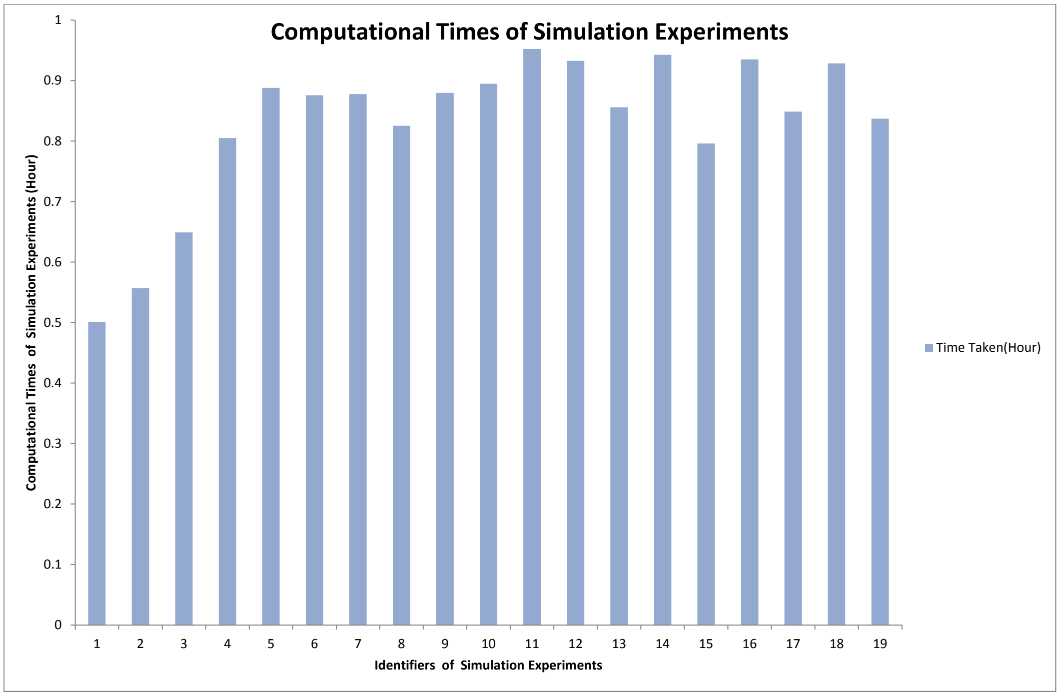

Figure 5.14: Computational Times of Simulation Experiments

Saint Sharp 3.0 Windows version. Explicit enumeration is implemented in C # coded in Microsoft Visual Studio 2010 Express [26]. 19 simulation iterations are run synchronously on one computer to reduce the total time cost. The computational times for each experiment are shown in the graph (5.14). The total computational time for simulation is the experiment using the longest time which is 0.95 hours. Besides, the running time of multi-objective optimisation is 0.03 seconds because the computational volume is very small. Therefore the total time taken is 0.95 hours.

5.5 Summary

This chapter addresses the truck quantity optimisation problem in a container terminal background, proposes a model for the truck quantity decision making for container terminal operators and discusses solutions. In this chapter, simulation and multi-objective optimisation are combined based on a simulation model for container terminal truck operations and a multi-objective optimisation modelling with five conflicting objectives to support terminal daily decision making. Simulation has strengths in its ability to effectively implement complex stochastic processes of container terminals and provide real time data, on the other hand, multi-objective optimisation has better performance in computational time and the Pareto guaranteed solutions. Therefore, a combination is proposed to integrate the strengths of two methods in order to provide optimised truck quantities for daily decision making in a container terminal in effective and efficient ways. The model is applied based on the data from the Southampton Container Terminal and the solutions show that the model offers various and effective solutions to container terminal operators. The computational time is around an hour which is acceptable in the context of the application.

Chapter 6

Combining Simulation and Multi-Objective Optimisation by Post-MOO Structure for Multiple Container Terminal Equipment Optimisation

6.1 Introduction

The terminal operational efficiency denotes the how fast cargos move in and out of terminals. For specific terminal equipment, it means an index to measure the number of containers handled in a unit of time or the time taken to handle a certain number of containers. The equipment productivity is normally measured by a series of indices in term of the needs and requirements from clients. Two types of efficiency objectives are considered in this chapter and they are quay crane operational efficiency and yard crane operational efficiency. Smooth traffic is also considered as one of the efficiency goals as it influences container circulation time in the yard. On the other hand, cost related objectives in terminal daily expenses considered are fuel consumption and unit labour cost.

A combination of simulation and multi-objective optimisation is employed to integrate the strengths of simulation and multi-objective optimisation to deal with dynamic systems and explore the near Pareto optimal solutions to multiple objective problems. On the other hand, the computational cost of direct simulation-based optimisation method is normally very high [108] for some applications because of the number of additional simulation replications [20]. The combination with multi-objective optimisation may reduce the computational cost of simulation without effects on representation of dynamic terminal operations.

In this chapter, a simulation model is developed involving multiple types of terminal equipment in section 6.3.2. A detailed level of output data is able to be obtained from the model for further analysis and use. A multi-objective optimisation is also built for the four main objectives considered in this chapter on the basis of the combination of the two methods. Objective functions are addressed in section 6.3.3, while a general multi-objective optimisation function is stated in section 6.3.4.

Four objectives and three decision variables are considered in the models, therefore the solutions are to be found in a four dimensional objective space and a three dimensional decision space. The genetic algorithm is employed to explore the near Pareto optimal solutions for the multi-objective optimisation model because the problem is non-linear and the computational time of solving the problem by an explicit enumeration algorithm is more than 24 hours.

The main structure of this chapter is described below. Firstly, the introduction to the processes of this chapter is given in section 6.1. Background statement, container terminal operations with multiple equipment and the problem description are addressed in section 6.2. A model for container equipment quantity optimisation is described in section 6.3. Parameters and results are discussed in section 6.4. Finally, conclusions are given in section 6.5.

6.2 Background Information

Being connection points between maritime transportation and inland transportation, container terminals are buffer and storage areas for containers transferred from water route to land route transportation or contrarily from land route to water route transportation. Containers usually have short stays in terminals for the next operations. The velocity of cargo circulation through container terminals is supposed to adapt to the velocity of cargo circulation in the whole supply chain to avoid bottlenecks effects. Otherwise, if cargos are delayed in container terminals, the bottleneck happening in terminals may influence the whole supply chain efficiency.

According to the layout of a container terminal shown in figure (5.1), in order to guarantee smooth and efficient operations, a container terminal normally contains flowing main facilities and mechanical equipment:

- Terminal areas: berth, container yard, gatehouse, buffer area, administration and control centre;
- Mechanical equipment: quay crane, yard crane, internal truck;

On the other hand, as discussed in section 2.4, the main mechanical equipment in container terminals are:

- Equipment alongside the berths: quay crane;
- Equipment in the yard: straddle carrier, sprinter, reach stacker and empty container handler;
- Equipment in the yard gatehouses and berths: Internal truck.

For terminal equipment, quay cranes are the tools to move containers between the water side and land side at berths. Yard cranes are those cranes picking up and grounding containers in the terminal yard, which mainly have to four types of cranes: straddle carrier, sprinter, reach stacker and empty container handler. Internal trucks are the container carriers within terminals.

Container terminal operations normally consists of a series of loading, discharging, location shifts and pre-marshalling movements of containers to achieve the goals of inter-modal transportation under a whole supply chain. General container terminal operations can be broken down to three sub-modules which are:

- Importation operations (Discharging operations)
- Exportation operations (Loading operations)
- Pre-marshalling operations

Categorised by different types of equipment, the operations of container terminals can be categorised to:

- Quay crane operations
- Yard crane operations
- Internal truck operations

Importation and exportation operations include quay crane, yard crane and internal truck operations, while pre-marshalling operations involve the above operations other than quay crane operations.

The objectives to be optimised in this chapter are efficiency related and cost related objectives as listed below:

- Efficiency related objectives

  High quay crane productivity

  High yard crane productivity

  Low terminal yard congestion probability

- Cost related objective

  Short truck travelling distance in the yard (related to fuel consumption and emissions)

6.3 Model Description

This section addresses the processes of developing equipment quantity optimisation models for container terminals under modern management. Terminal operations are analysed, categorised, summarised and then integrated into the modelling. A simulation model is built to present terminal work flows and a multi-objective optimisation model is also proposed for decision makers in the post-MOO structure under the combination framework. A genetic algorithm is developed to explore the near Pareto optimal solutions. Parameter settings and results are also discussed.

In this section, simulation modelling details are addressed in section 6.3.2 and the formulations of objectives are presented in section 6.3.3. The multi-objective optimisation function is given in section 6.3.4. The genetic algorithm is employed to solve the function in section 6.3.5.

6.3.1 Post-MOO Structure

In this chapter, the post-MOO structure is employed to integrate simulation and multi-objective optimisation under the combination framework. The definitions of post-MOO are given section 4.3.3. The data fitting is used to process the simulation output as shown in figure (4.7). The processed data is sent out as the parameters for multi-objective optimisation.

6.3.2 Discrete-event Simulation

This section proposes simulation models for container terminal general operations for equipment quantity optimisation. Two simulation models are developed in this section: a simulation model of general container terminal operations for equipment quantity optimisation and a simulation model for internal truck travelling distance optimisation. The simulation model for general terminal operations is built in the Micro Saint Sharp simulation package and the model for truck travelling distance is developed in C#.

The simulation model of internal truck travelling distance needs a large number of iterations to achieve a high level of accuracy because the model generates a great deal of stochastic numbers. In order to increase the stability of simulation results, a certain number of iterations are necessary to reduce the effect of uncertainty of stochastic numbers and noisy data. However, on the other hand, a large number of iterations are computational time consuming. Therefore, in order to run the model for enough iterations and meanwhile reduce the time taken for the model, the simulation model for internal truck travelling distance optimisation is built separately in C# in Microsoft Visual Studio 2010 Express because a number of applications of visual interfaces in the Micro Saint Sharp package increases the computational time especially in the case of that a large number of iterations are required. In order to reduce the time taken by visual interfaces, the model for truck travelling distance is coded in a pure console mode in C# to avoid the repetitive computation for visual interfaces in each iteration.

The source codes for the model of truck travelling distance are given in appendix B, while the model for general container terminal operations built in the Micro Saint Sharp is discussed as follows.

Simulation of Container Terminal Operations

The elements involved in the simulation model include event, entity, equipment resource, list, variable, parameter, process ("tasks" in the model) as shown in table 6.1. The boxes appearing in the simulation network in figure(6.1) are the tasks representing basic container terminal operations which may be triggered by the occurrence of new event. Containers are the entities as a flow goes through simulation tasks in the model. The flow triggers the executions of different tasks or, in other words, the tasks are driven by these events. The occurrence of events is controlled by execution conditions ("Release Conditions" in the Micro Saint Sharp simulation package) which are like switches to trigger tasks when container flows go through them. Before containers enter tasks, judgements need to be made: only those containers satisfying the release conditions are allowed to enter the tasks and processed by the simulation package.

Another element is equipment resource, namely, the decision variables to be decided in this model. Therefore the values of the decision variables, i.e. equipment numbers, need to be set in simulation scenarios. The equipment considered is quay cranes, yard cranes and internal trucks.

Furthermore, lists in the model are another element of simulation. Lists are a tool to control the numbers of terminal devices to imitate terminal operations in reality. For instance, containers wait in queues if the available terminal devices are zero. The length of queues and average waiting in queues can be observed and recorded by the package. Additionally, in order to implement real systems on computers, variables are needed to control the numbers of equipment and set the parameters of their operational time distributions. These parameters need to be set in "Scenarios" in the Micro Saint Sharp and their values can be tested by realistic data from container terminals.

Table 6.1: Simulation Elements

Entity	container
Equipment resource	quay crane, yard crane and truck
List	waiting task list executed task list free yard crane list occupied yard crane list waiting lists for each yard crane free quay crane list occupied quay crane list waiting lists for each quay crane free truck list occupied truck list

Assume all of terminal equipment is in idle status at the initial state of simulation, namely all of free lists are full while all of executed lists are empty. Terminal internal trucks and yard cranes stop at the truck operation buffer areas in figure (5.1). Quay cranes are at stand-by positions on their tracks. The waiting task and executed task lists are empty. All of yard cranes, quay cranes and trucks are waiting for tasks, namely, they are in the free yard crane list, and free quay crane list and free truck list respectively. When the simulation starts to run, new operation tasks are given by the terminal control system and these tasks enter the waiting task list.

When new tasks are generated, the container terminal control centre allocates idle trucks to new tasks from the free truck list. If the free truck list is not empty, the control system matches the free trucks with new tasks, namely assigning available trucks with new tasks. Otherwise, the new tasks enter the waiting task list and wait for the next available truck. Once free trucks are assigned to new tasks, they are transferred from the free truck list to the occupied truck list, and meanwhile, these tasks are transferred from the waiting task list to the executed task list. Then truck drivers are noticed with the positions to picked up from and the destinations to send to. The trucks are then driven to their destinations and wait for crane operations. If the free crane list is not empty, namely at least one crane is available, the first truck on the list will be operated. Otherwise they will be in the waiting lists. Trucks then carry containers to their next appointed destinations and enter queues waiting for next crane operations. If the free crane list is not empty, containers are picked up from trucks and grounded onto ships or the yard or external trucks.

When trucks are waiting for yard crane operations, the container terminal control centre allocates idle yard cranes to the trucks from the free yard crane list. If it is not empty, the control system matches the free yard cranes with the trucks from the waiting lists. If the free yard crane list is empty, the trucks will be still staying in the waiting list and wait for the next available yard crane. Once available yard cranes are assigned to the trucks, they will be transferred from the free yard crane list to the occupied yard crane list, and the trucks are taken out from the waiting lists for yard cranes.

If trucks are waiting for quay crane operations, the container terminal control centre allocates idle quay cranes to the trucks from the free quay crane list. As long as the free quay crane list has one or more than one quay crane(s) available, the control system matches it(them) with the trucks from the waiting lists. Otherwise, the trucks need to wait for the next available quay crane. Once available quay cranes are assigned to the trucks, they will be transferred from the free quay crane list to the occupied quay crane list, and the trucks are taken out from the waiting list.

The simulation model for general operations is built in the Micro Saint Sharp simulation environment. The simulation model consists of several sub-networks and sub-sub-networks as each sub-network or sub-sub-network describes a basic terminal operation. A general simulation network is given in figure (6.1). The model consists of five modules: two modules belong to importation, two modules belong to exportation and one module represents pre-marshalling operations. Two importation modules are the sub-networks "(2)" and "(3)" in the green boxes in figure (6.1). The sub-network "(2)" represents the operations of containers moving from outside into the terminal yard, while the sub-network "(3)" represents the operations of containers moving from the yard onto container ships. Two exportation modules are the sub-networks "(4)" and "(5)". The sub-network "(4)" represents the operations of containers moving from container ships into the terminal yard, while the sub-network "(5)" represents the operations of containers moving from the yard to outside. Above five modules consists of three basic operational processes, namely quay crane operation process, yard crane operation process and internal truck operation process, while the internal truck process has two sub-modules which are importation truck operation process and exportation truck operation process.

New operation tasks are generated by the terminal control system (i.e. "Operation Orders (1)" in figure (6.1)) and the tasks are sent to be made judgements in "Im/ExPort(10)" that they are either importation tasks or exportation tasks. Importation containers are sent to the sub-network "(2)" and exportation containers are sent to "DirectExp (11)". The processes are discussed separately by importation and exportation below.

The importation process includes the sub-networks "(2)", "(6)" and "(3)". If containers are direct imported, namely cargo owners need to pick up their cargos without any stop in terminals when the containers are discharged from container ships, the sub-networks "(2)" transfers containers from container ships to external trucks

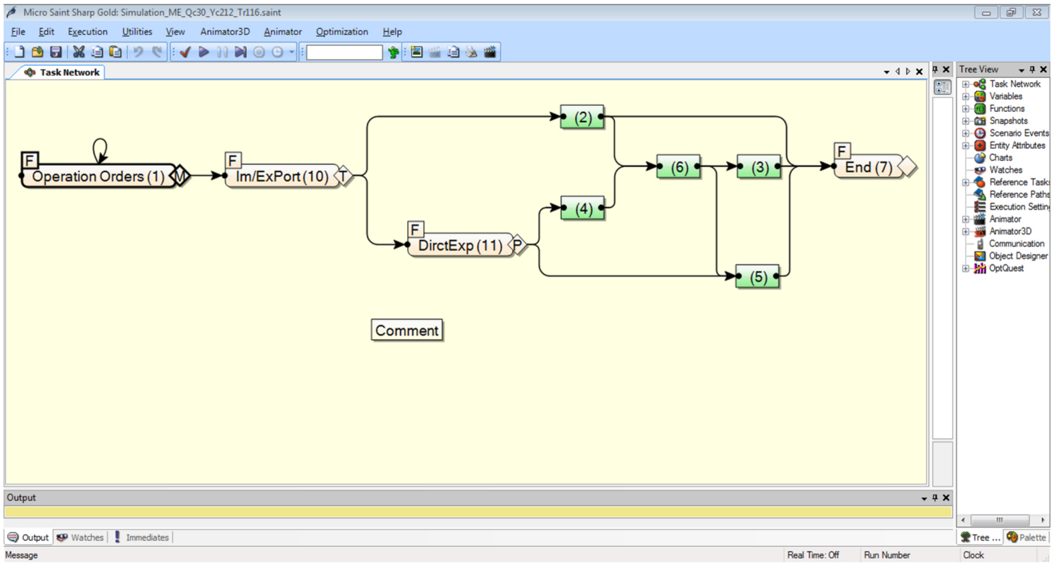

Figure 6.1: General Simulation Network

waiting in the truck operation buffer areas in figure (5.1). Otherwise, if containers are not direct imported, the sub-networks “(2)” transfers containers from container ships to the terminal yard for temporarily storage. Pre-marshalling operations may happen during this period as shown in the sub-network “(6)”. When cargo owners arrange external trucks to pick up their cargos, containers are moved from terminals to the truck operation buffer areas as shown in the sub-network “(3)”. The external trucks will pick up the containers from the truck operation buffer areas.

The exportation process includes the sub-networks “(4)”, “(6)” and “(5)”. If containers are direct exported, namely containers are delivered onto ships without storage in terminals from outside, the sub-network “(4)” transfers containers from external trucks from the truck operation buffer areas onto container ships. Otherwise, containers are delivered into terminals for temporarily storage as shown in the sub-network “(4)”. Pre-marshalling operations may be needed during this period as shown in the sub-network “(6)”. When container ships come to load the containers, the containers are transported to the quay crane operation buffer areas to be loaded onto ships according to the sub-network “(6)”.

The details of the sub-network “(2)” is shown in figure (6.2). It contains a truck operation, a quay crane operation and a yard crane operation. The sub-sub-network “2_16” is a quay crane operation process and sub-sub-network “2_17” is a yard crane

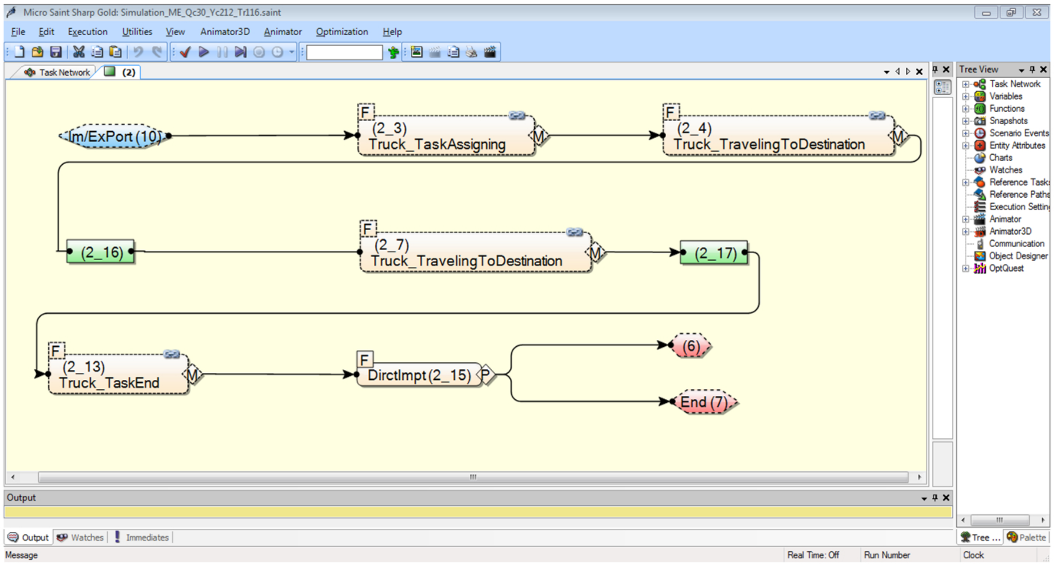

Figure 6.2: Sub-network 2 of Simulation

operation. The sub-sub-network "2_16" is shown in figure (6.3). It describes a quay crane operation process. The sub-sub-network "2_17" is demonstrated in figure (6.4). It shows a yard crane operation process.

The details of the sub-network "(3)" is shown in figure (6.5). The sub-sub-networks "3_11" and "3_12" represent yard crane operations. The sub-sub-networks "3_11" and "3_12" are shown in figures (6.6) and (6.7) respectively.

The details of the sub-network "(4)" is shown in figure (6.8). The sub-sub-networks "4_15" and "4_16" represent yard crane operations. The sub-sub-networks "4_15" and "4_16" are shown in figures (6.9) and (6.10).

The details of the sub-network "(5)" is shown in figure (6.11). The sub-sub-network "5_15" is a quay crane operation process and sub-sub-network "5_14" is a yard crane operation. The sub-sub-network "5_14" is shown in figure (6.12). It describes a yard crane operation process. The sub-sub-network "5_15" is shown in figure (6.13). It describes a quay crane operation process.

The details of the sub-network "(6)" is shown in figure (6.14). The sub-sub-networks "6_15" and "6_16" represent yard operations. The sub-sub-networks "6_15" and "6_16" are shown in figures (6.15) and (6.16) respectively.

There are two types of parameters in the simulation model: stochastic parameters and non-stochastic parameters.

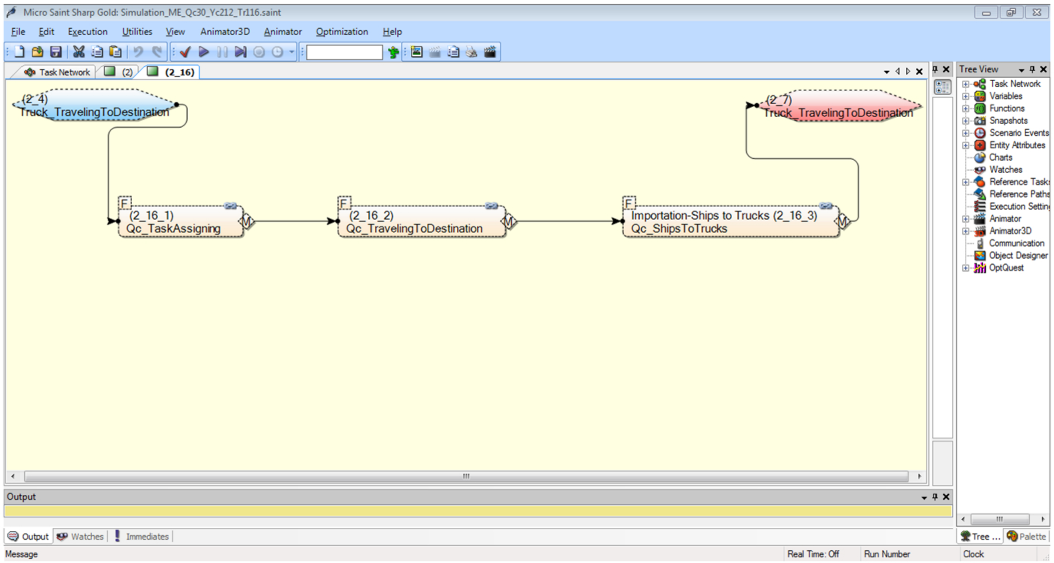

Figure 6.3: Sub-network 1 of Sub-network 2 of Simulation

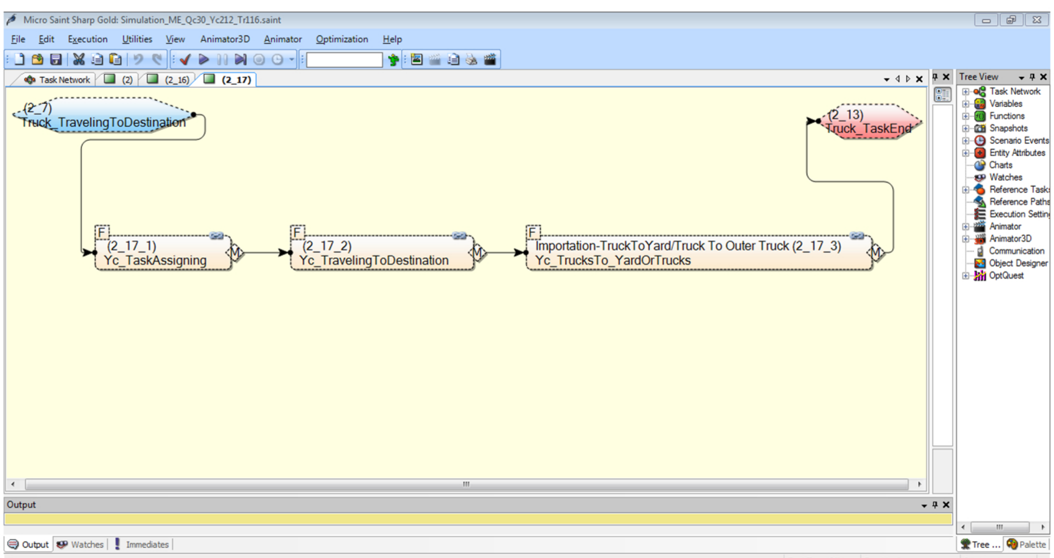

Figure 6.4: Sub-network 2 of Sub-network 2 of Simulation

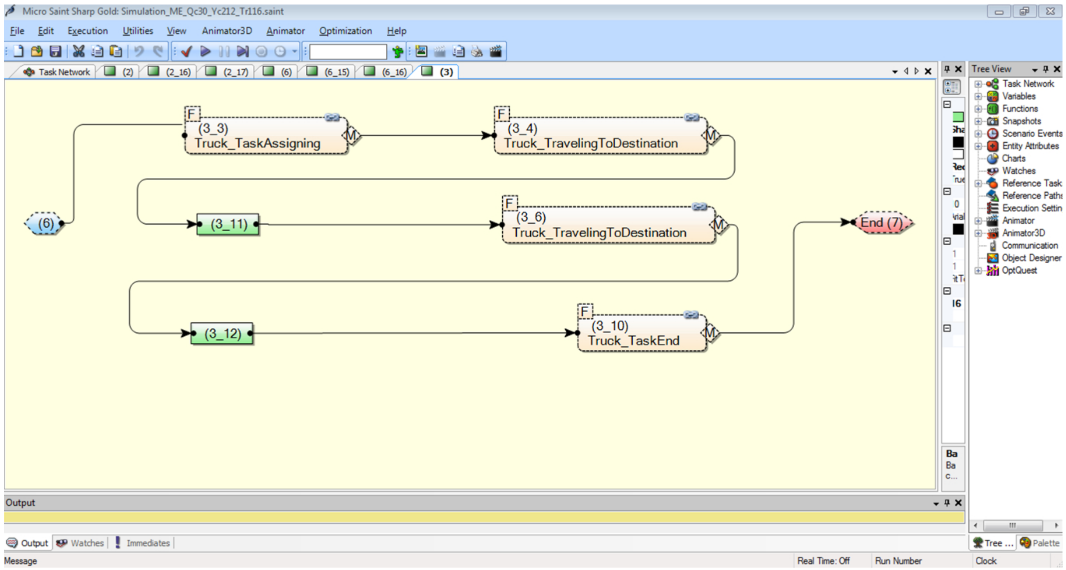

Figure 6.5: Sub-network 3 of Simulation

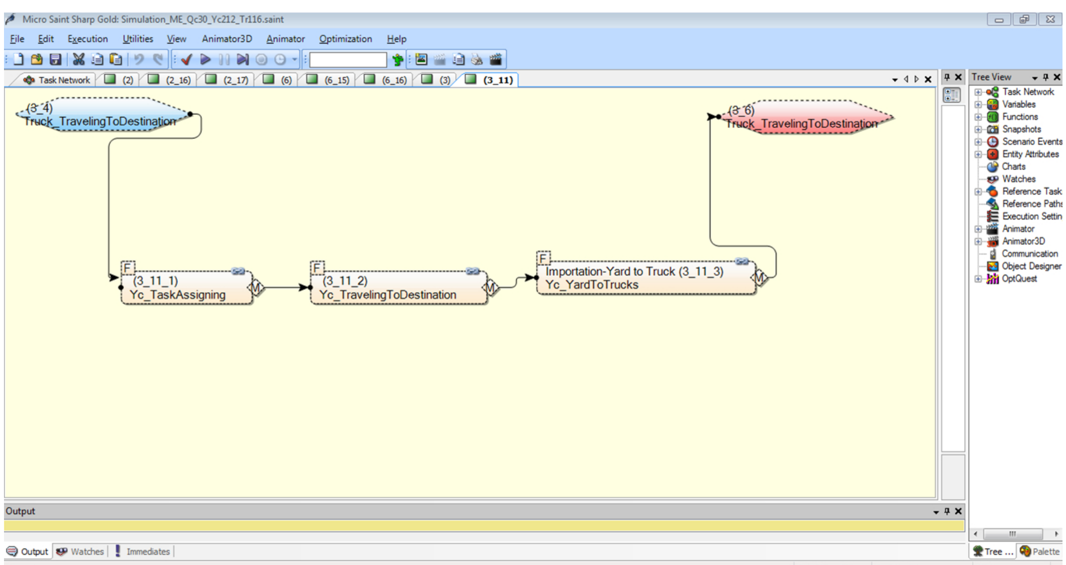

Figure 6.6: Sub-network 1 of Sub-network 3 of Simulation

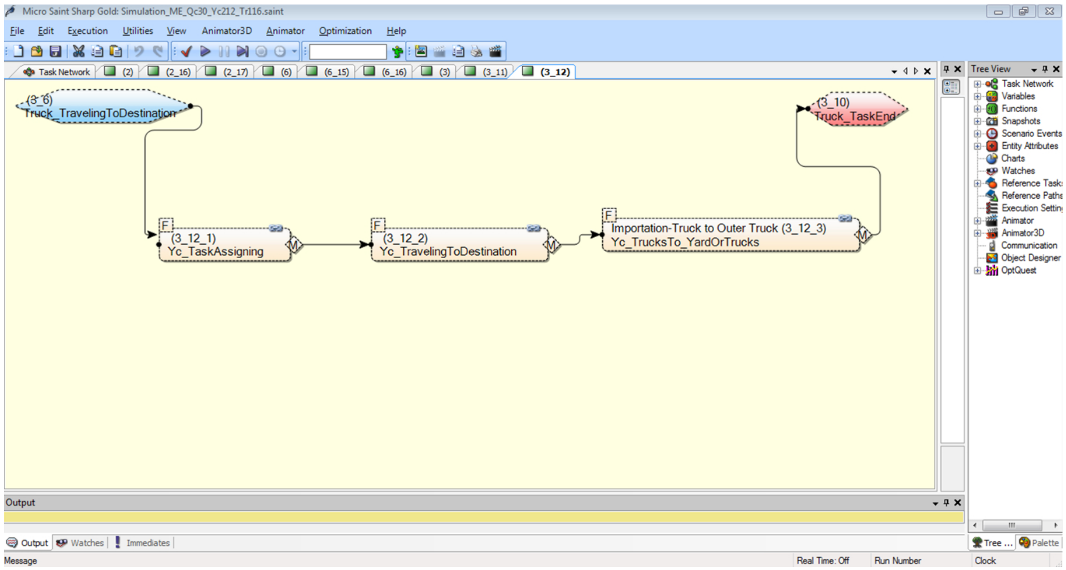

Figure 6.7: Sub-network 2 of Sub-network 3 of Simulation

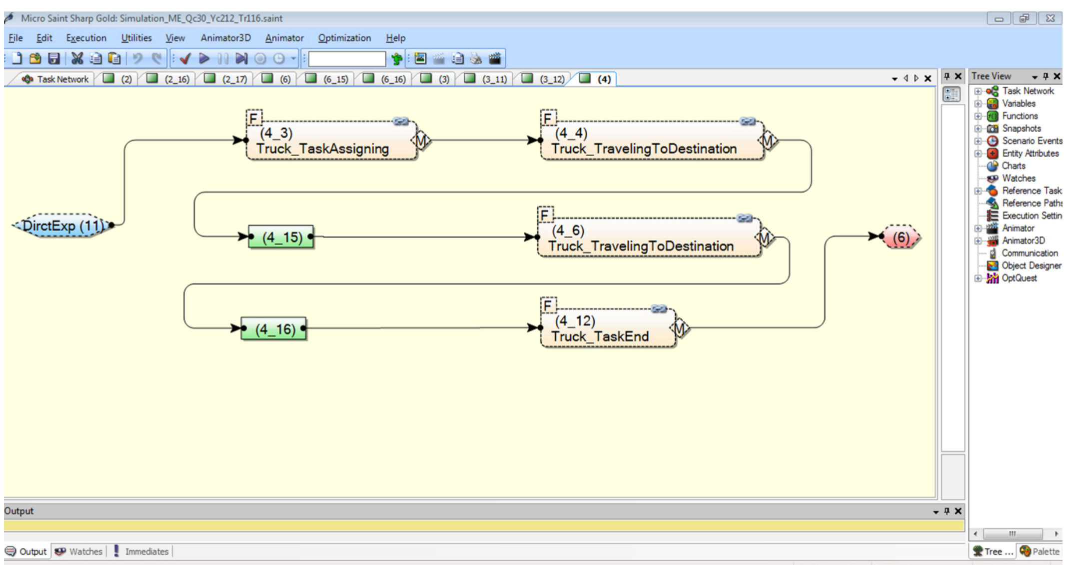

Figure 6.8: Sub-network 4 of Simulation

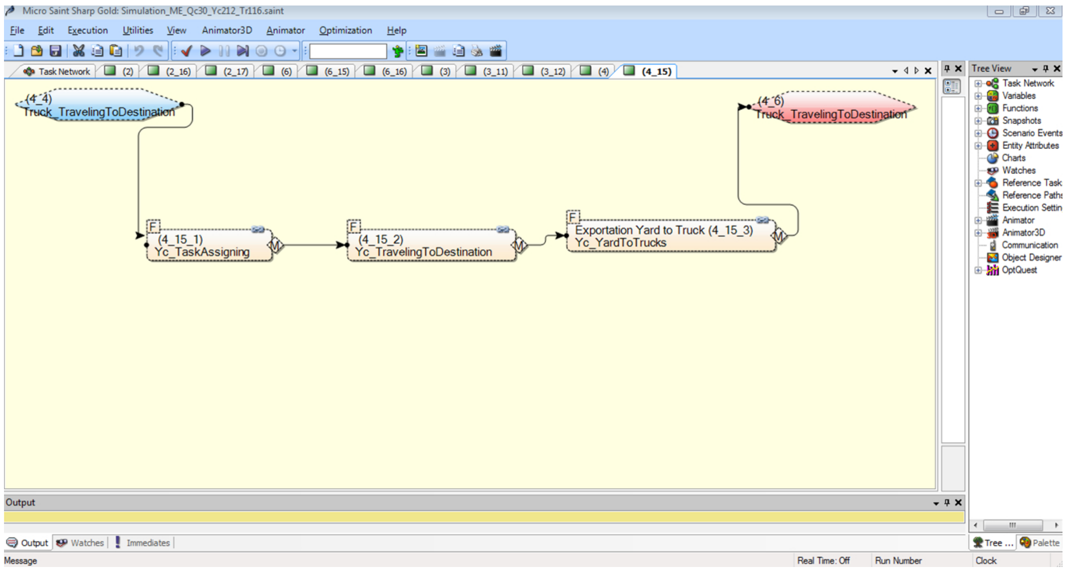

Figure 6.9: Sub-network 1 of Sub-network 4 of Simulation

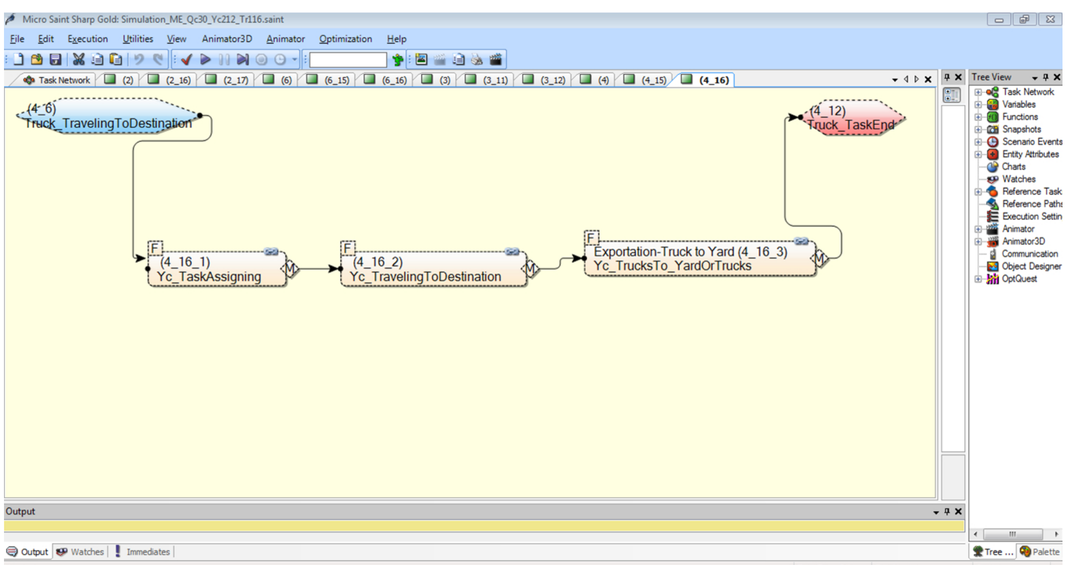

Figure 6.10: Sub-network 2 of Sub-network 4 of Simulation

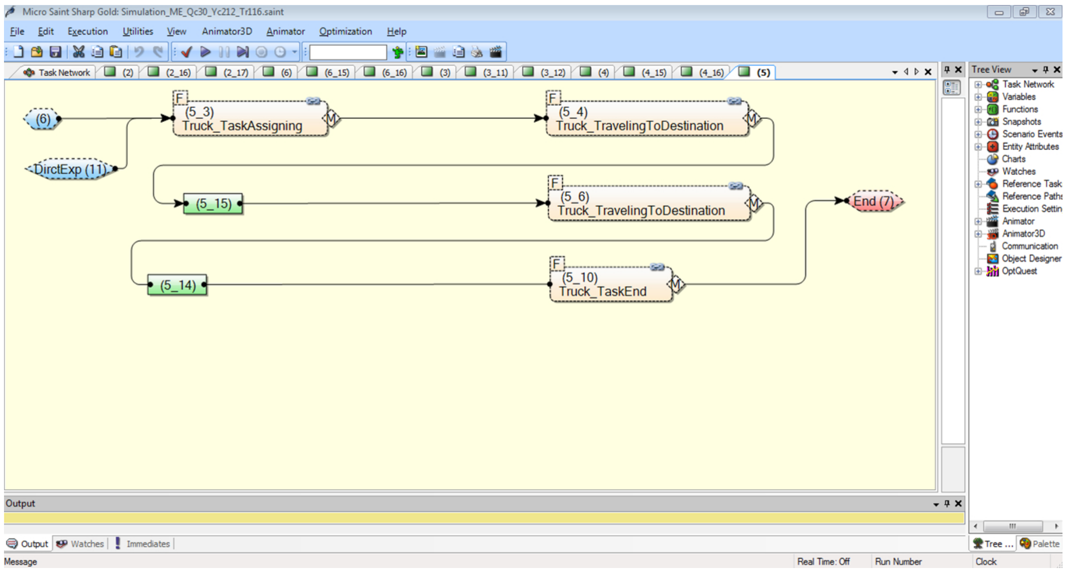

Figure 6.11: Sub-network 5 of Simulation

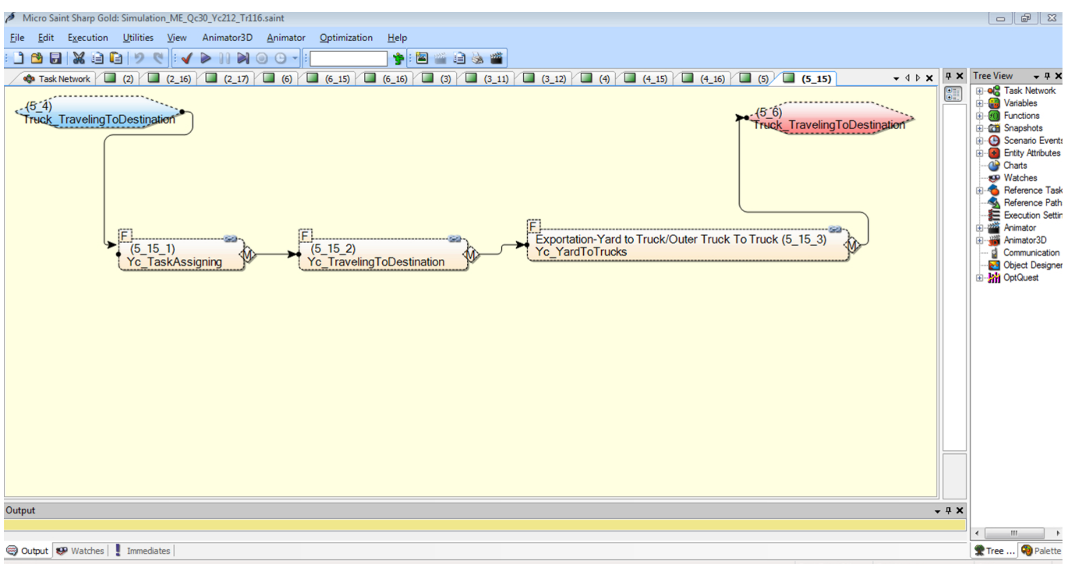

Figure 6.12: Sub-network 1 of Sub-network 5 of Simulation

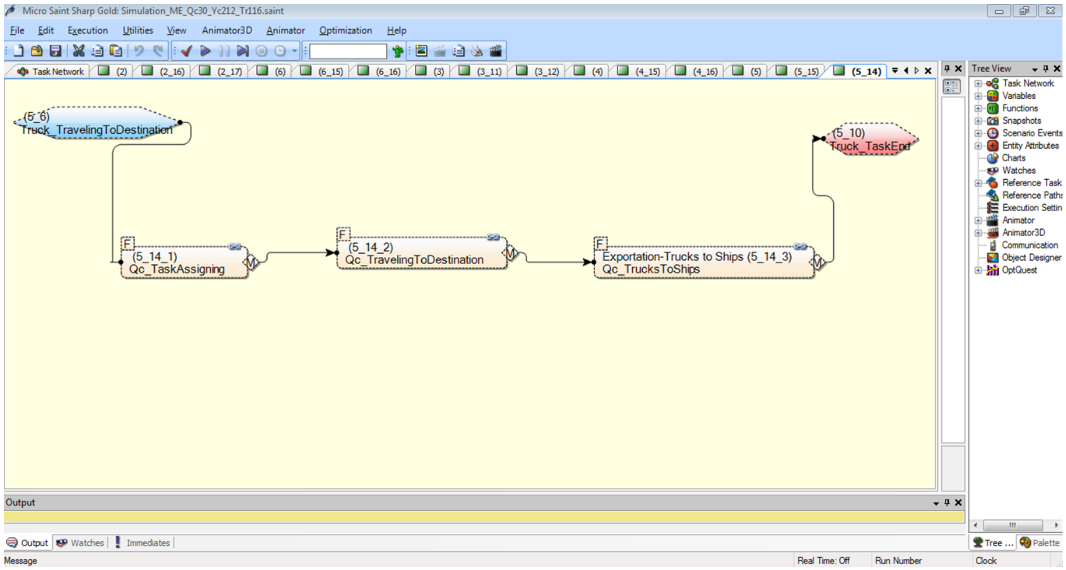

Figure 6.13: Sub-network 2 of Sub-network 5 of Simulation

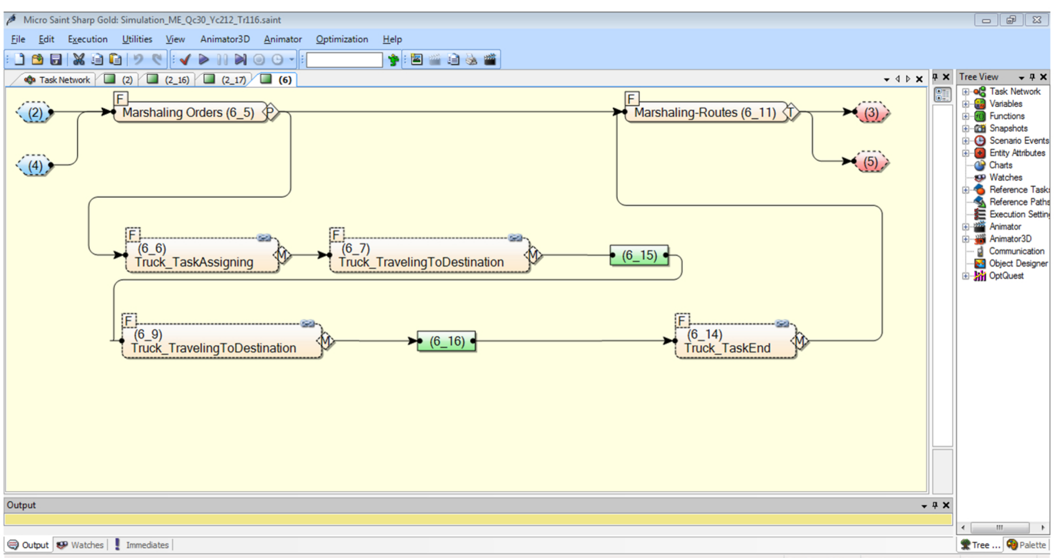

Figure 6.14: Sub-network 6 of Simulation

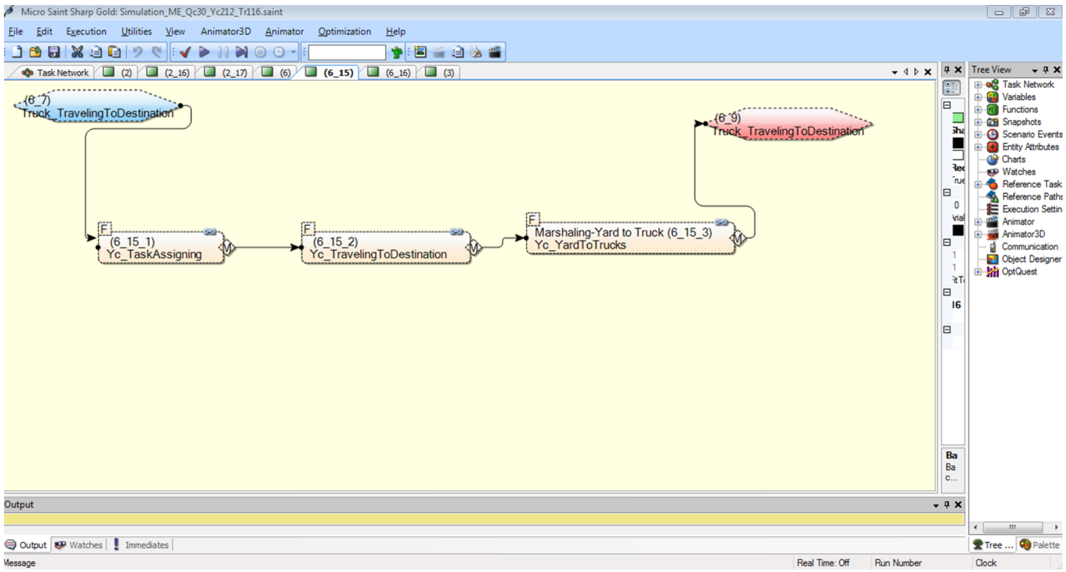

Figure 6.15: Sub-network 1 of Sub-network 6 of Simulation

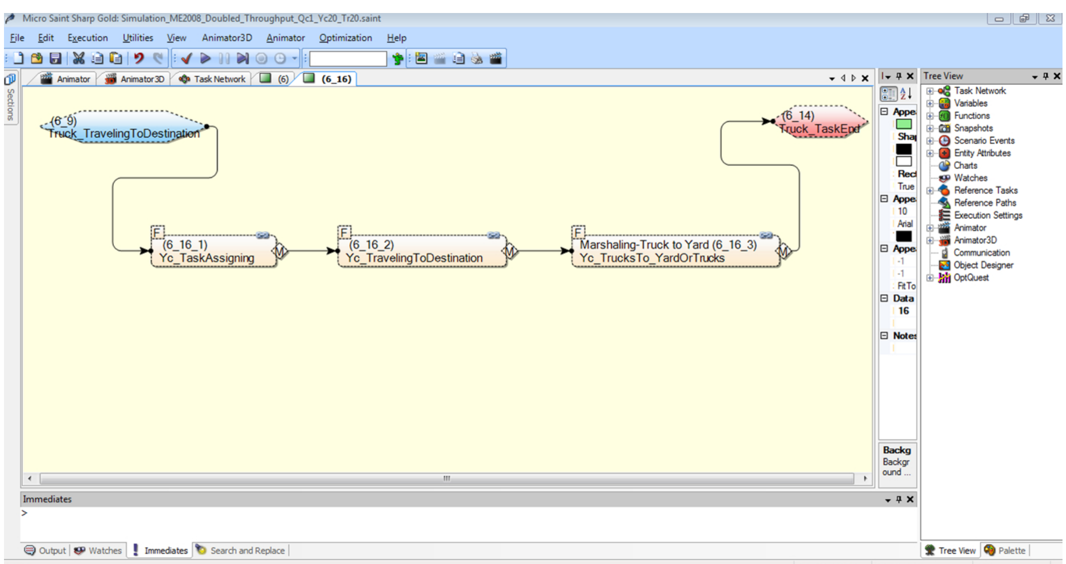

Figure 6.16: Sub-network 2 of Sub-network 6 of Simulation

The operational times of every operation are generated by stochastic variables based on time distribution functions. The definitions of these stochastic variables are declared in table 6.2. The probability distribution function for the i^{th} stochastic variable is denoted by $X_i \sim F_i(x)$, where θ_i denotes a set of distribution parameters for $F_i(x)$. The setting of parameters for time distributions for terminal devices determines

Table 6.2: Time Distribution Definitions

Time Distributions	Definitions
X_1	Tasks interval time distribution
X_2	Truck assigning time distribution
X_3	Truck travelling time distribution
X_4	Time distribution of yard crane operations from yard to trucks
X_5	Time distribution of yard crane operations from trucks to yard or trucks
X_6	Time distribution of quay crane operations from ships to trucks
X_7	Time distribution of quay crane operations from trucks to ships
X_8	Quay crane assigning time distribution
X_9	Yard crane assigning time distribution
X_{10}	Quay crane travelling time distribution
X_{11}	Yard crane travelling time distribution

the values of stochastic variables, namely the operational time of equipment. So parameters need to be assigned with appropriate values to reflect realistic systems. The input of parameters include the selection of probability distribution functions $F_i(x)$ and estimation of function parameter sets θ_i. The values of stochastic variables involved are then generated by computers on the basis of their distribution functions and parameters.

On the other hand, the non-stochastic parameters are those parameters related to terminal layout, operation processes, equipment quantities and properties, loading/discharging ratio, empty/laden container ratio, transhipment/all container ratio, pre-marshalling probability, simulation duration and the number of simulation iterations.

The results of simulation are processed for the scenarios for multi-objective optimisation. The output of the simulation model is given out in table 6.3.

Table 6.3: Simulation Output Parameters

Output Parameter	Definition
$cntrTimQC_{ijk}$	The loading and(or) unloading operational time of the i^{th} lift by j^{th} quay crane for k^{th} ship
$cntrTidleQC_{ijk}$	The idle or waiting time of the i^{th} lift by j^{th} quay crane for k^{th} ship
$NcnQCLft_{ijk}$	The number of containers lifted in the i^{th} lift by j^{th} quay crane for k^{th} ship
$TiYcInTr_{ijk}$	The loading and(or) unloading operational time of the i^{th} lift operation of j^{th} yard crane for k^{th} internal truck
$TiYcExTr_{ijk}$	The loading and(or) unloading operational time of the i^{th} lift operation of j^{th} yard crane for k^{th} external truck
$TiYcCY_i$	The loading and(or) unloading operational time of the i^{th} lifting operation for the containers in the terminal yard
$NcnYcInTr_{ijk}$	The number of containers lifted in the i^{th} lift of j^{th} yard crane for k^{th} internal truck
$NcnYcExTr_{ijk}$	The number of containers lifted in the i^{th} lift of j^{th} yard crane for k^{th} external truck
$NcnYCLft_{ij}$	The number of containers lifted in the i^{th} lift by j^{th} yard crane for pre-marshalling operations in the yard
$tQqc_{ijk}$	The waiting time of the i^{th} container to be lifted by j^{th} quay crane for k^{th} ship
$QYcInTr_{ijk}$	The waiting time of the i^{th} container operated by j^{th} yard crane for k^{th} internal truck
$QYcExTr_{ijk}$	The waiting time of the i^{th} container operated by j^{th} yard crane for k^{th} external truck
$QYcCY_{ij}$	The waiting time of the i^{th} container operated by j^{th} yard crane for pre-marshalling operations

Simulation of Internal Truck Travelling Distance

When a new task is generated, the terminal control system locates the position of loading and discharging, and then, seeks for the closest truck to assume the task. The position of the truck is its previous discharging position. There are three major positions to determine the truck travelling distance for every task: the position to discharge the previous container(s), the position to load the current container(s) according to the task and the position to discharge the current container(s). When a container moved from one container block to another, the previous discharging position, current loading position and current discharging position might be all in the same crane operation area, or two of them are from the same crane operation area but the other one is from a different area, or three of them are all from different crane operation areas. Travelling in the same crane operation areas normally avoids the extra travelling distances. Extra travelling distance means the distance caused by that trucks need to detour round container blocks to go to destinations.

When new tasks are sent to internal truck drivers, the information normally contains loading and discharging positions. (xld_i, yld_i) denotes the position where the i^{th} task loads container(s) from. $(xdisch_i, ydisch_i)$ is the position where the i^{th} task discharge(s) the container(s) to. The travelling distance of the i^{th} task includes: firstly, the travelling distance from the positions of trucks, i.e. the previous container discharging positions, to the current loading positions; secondly, the travelling distance from the container loading positions to the container discharging positions; thirdly, the extra travelling distance. The travelling distance of loading operation of the i^{th} task is denoted by Dld_i which is given out in equation (6.1). It equal to the distance difference on x and y axes.

$$Dld_i = \sqrt{(xld_i - xdisch_{i-1})^2} + \sqrt{(yld_i - ydisch_{i-1})^2} \tag{6.1}$$

The travelling distance of discharging operation of the i^{th} task is denoted by $Ddisch_i$, which is equal to the distance difference on x and y axes.

$$Ddisch_i = \sqrt{(xld_i - xdisch_{i+1})^2} + \sqrt{(yld_i - ydisch_{i+1})^2} \tag{6.2}$$

In container terminals, travelling routes are unlikely to be point-to-point straight lines because trucks run on roads rather than cross container blocks directly. Therefore extra travelling distance is normally inevitable. Suppose the positions of container blocks are described by a four dimensional coordinate, i.e. $(xCnStkUp_j, xCnStkDn_j, yCnStkup_j, yCnStkDn_j)$ which denotes the coordinates of the j^{th} container block in the yard. Therefore the lower left corner of the j^{th} container block is $(xCnStkDn_j, yCnStkDn_j)$. The lower right corner of the j^{th} container block is $(xCnStkUp_j, yCnStkDn_j)$. Similarly, the up left and up right corners of the container block are located at $(xCnStkDn_j,\ yCnStkUp_j)$ and $(xCnStkUp_j, yCnStkUp_j)$ respectively. DEx_i denotes the extra distance between (xld_{i-1}, yld_{i-1}) and (xld_i, yld_i). If the straight lines $x_1 = xld_{i-1}$ and $x_2 = xld_i$ both cross the container block area $(xCnStkUp_j, xCnStkDn_j, yCnStkup_j, yCnStkDn_j)$, then the container i needs to travel an extra distance to detour round container blocks. In this case, trucks have to choose the shortest route to detour container blocks.

DEx_{min} is defined as the minimum distance to detour the container block j and DEx_{max} be the maximum distance.

$$DEx_{min} = min(xCnStkup_j, xCnStkDn_j) \tag{6.3}$$

$$DEx_{max} = max(xCnStkup_j, xCnStkDn_j) \tag{6.4}$$

$$\begin{aligned} DEx_i = 2 \cdot min\ (&\sqrt{(DEx_{min} - xld_{i-1})^2}, \sqrt{(DEx_{max} - xld_{i-1})^2}, \\ &\sqrt{(DEx_{min} - xld_i)^2}, \sqrt{(DEx_{max} - xld_i)^2},) \end{aligned} \tag{6.5}$$

On the other hand, if the straight lines $y_1 = yld_{i-1}$ and $y_2 = yld_i$ both cross the container block area $(xCnStkUp_j, xCnStkDn_j, yCnStkup_j, yCnStkDn_j)$, then the container i needs to travel an extra distance to detour round road-blocks. In this case, trucks have to choose the shortest route to detour container blocks.

$$DEx_{min} = min(yCnStkup_j, yCnStkDn_j) \tag{6.6}$$

$$DEx_{max} = max(yCnStkup_j, yCnStkDn_j) \tag{6.7}$$

$$\begin{aligned} DEx_i = 2 \cdot min\ (&\sqrt{(DEx_{min} - yld_{i-1})^2}, \sqrt{(DEx_{max} - yld_{i-1})^2}, \\ &\sqrt{(DEx_{min} - yld_i)^2}, \sqrt{(DEx_{max} - yld_i)^2},) \end{aligned} \tag{6.8}$$

The total travelling distance for the i^{th} container is denoted by DTr_i when k internal trucks are employed in the terminal yard.

$$DTr_{ik} = (Dld_i + Ddisch_i + DEx_i) \tag{6.9}$$

DTr_k is defined as the total truck travelling distance during simulation period when k internal trucks are employed in the terminal yard.

$$DTr_k = \sum_{i=1}^{N_{cntr}} DTr_{ik} \tag{6.10}$$

In this simulation model, the positions of container blocks ($xCnStkUp_j, xCnStkDn_j, yCnStkup_j, yCnStkDn_j$) are a parameter need to be given according to the terminal layout. The loading and discharging positions on tasks, i.e. (xld_i, yld_i) and ($xdisch_i, ydisch_i$) are generated from the model by generating a great deal of random numbers in the container block areas. The random numbers are generated on the basis of millisecond of current time to avoid the same random numbers. Additionally, a certain number of iterations to run the model is necessary in order to reduce the random errors caused by random numbers. Therefore the parameters involved in this simulation model include the terminal yard coordinates and the number of simulation iterations. Simulation output is the truck travelling distance, i.e. DTr_k.

6.3.3 Formulation for Objectives

Both of efficiency related objectives and cost related objectives are considered in this chapter, where efficiency related goals include high quay crane productivity, high yard crane productivity and low terminal yard congestion probability, and the cost related goal is to find the shortest internal truck travelling distance in the yard (related to the fuel consumption and emissions).

Viewing a container terminal as a whole, berths and gatehouses are like two interfaces to accept and hand over containers from and to terminal clients. Firstly, the ability of cargos to go through container terminals influences the quick change of transportation modes in intermodal transportation across continents and oceans. The loading and discharging rates at the berths normally determine the berthing time of container vessels in terminals and container handling time which might affect the voyage schedules of terminal shipping clients. Secondly, for the terminal gatehouses, cargo handling rates are a determinant to the cargo circulation speed in the yard. If there is a bottleneck at the gatehouse, cargos are unable to evacuate shortly from terminals and be carried into the yard before vessel arrivals. Thirdly, reducing the number of queues will improve the traffic ability in the yard. The road areas are limited in terminals because the main area of the yard is for container storage. Therefore the berth productivity, gatehouse productivity and truck waiting probability are considered as efficiency objectives in this chapter. Other than efficiency related objectives, cost related objectives also need to be taken into consideration in this model, such as minimise travelling distance to reduce fuel cost and equipment emissions. Single objective functions for each management objective are given respectively as follows and then a multi-objective function will be given. By the end of this section, a genetic algorithm to explore the near Pareto optimal solutions to the model will be addressed.

First Objective Function: Quay Crane Service Rate

The first objective is to maximise the berth operational efficiency which reflects how fast terminals transfer containers between terminals and vessels. Petering and Murty [111] state that the average number of lifts achieved at a terminal per QC (quay crane) working hour is known as the GCR (gross crane rate, QC rate); GCR is perhaps the most important performance measure of a terminal. Li et al. [83] address that GCR measures the average rate at which the QCs (quay cranes) transfer containers between vessels and shores and is the most significant performance measure of a CT (container terminal) operation. Le-Griffin and Melissa [73] give a set of common productivity indices for container terminals to measure the equipment, berth, yard, gate and gang efficiency, such as crane productivity is measured by 'moves per crane-hour'. Therefore the index of quay crane productivity is one of important objectives which needs to be considered in this model. It is the number of lifts per quay crane in an unit of time. It equals to the number of lifts over the number of quay cranes and then over the time taken. Equation (6.11) below is the total quay crane operational time.

$$vTQC_w = \sum_{k=1}^{N_{vsl\ w}} \sum_{j=1}^{N_{qc\ k}} \sum_{i=1}^{N_{qclft\ j\ k}} (cntrTimQC_{ijk} + cntrTidleQC_{ijk}) \tag{6.11}$$

Where $vTQC_w$ is the total loading and(or) unloading time of all containers handled by quay cranes from and(or) onto container vessels during the time window w; $N_{vsl\ w}$ is the number of container vessels loading and(or) unloading on the terminal berths in the time window w; $N_{qc\ k}$ is the number of quay cranes loading and(or) discharging cargos onto and(or) from the k^{th} container vessel; $N_{qclft\ j\ k}$ means the j^{th} quay crane has $N_{lft\ j\ k}$ lifts for the k^{th} vessel; $cntrTimQC_{ijk}$ and $cntrTidleQC_{ijk}$ are declared in table 6.3.

Due to the quick development of technology, modern quay cranes are usually able to lift two containers in the same time in a single lift to improve the single machine efficiency. However, most terminals usually also have single-lift quay cranes purchased before other than dual-lift quay cranes. Even for the dual-lift quay cranes, it does not mean every lift is a dual lift in the consideration of the physical locations of containers. For instance, if the containers on the same task are distant to each other, quay cranes could not grab two containers at the same time with a longer distance than the width of crane spreaders. Thus only adjacent containers can be lifted by dual-lifts. $NcnQCLft_{ijk}$ (defined in table 6.3) is the number of containers in a quay crane lift. Equation (6.12) shows the number of containers operated by quay cranes during the time window w:

$$NcnQC_w = \sum_{k=1}^{N_{vsl\ w}} \sum_{j=1}^{N_{qc\ k}} \sum_{i=1}^{N_{qclft\ j\ k}} NcnQCLft_{ijk} \tag{6.12}$$

Where $NcnQC_w$ denotes the number of containers loaded and(or) discharged onto and(or) from container vessels in the time window w.

qR denotes how fast quay cranes handling containers, namely the number of containers handled per quay crane per hour, which equals to quay crane operation time over operated container number.

$$qR = \frac{NcnQC_w}{vTQC_w} \tag{6.13}$$

Suppose equation (6.14) below is a data set which describes the relationships between qR and (N_{tr}, N_{yc} ,N_{qc}).

$$\begin{aligned}((qR_1, (N_{tr_1}, N_{yc_1}, N_{qc_1})), (qR_2, (N_{tr_2}, N_{yc_2}, N_{qc_2})),\\ \cdots, (qR_m, (N_{tr_m}, N_{yc_m}, N_{qc_m})))\end{aligned} \tag{6.14}$$

Where N_{qc} N_{yc} N_{tr} are the numbers of quay cranes, yard cranes and internal trucks respectively. m is the number of the data elements in (6.14). Assume the fitting function $f_{(x)} = \alpha_1\varphi_1(x_1)+\alpha_2\varphi_2(x_1)+\alpha_3\varphi_3(x_1)+\cdots+\alpha_n\varphi_n(x_1)+\beta_1\chi_1(x_2)+\beta_2\chi_2(x_2)+$

$\beta_3\chi_3(x_2) + \cdots + \beta_n\chi_n(x_2) + \gamma_1\psi_1(x_3) + \gamma_2\psi_2(x_3) + \gamma_3\psi_3(x_3) + \cdots + \gamma_n\psi_n(x_3)$ is a mathematical expression for variables x_1 (i.e. N_{qc}), x_2 (i.e. N_{yc}) and x_3 (i.e. N_{tr}), and dependent variable qR based on the data set (6.14). Where $\varphi_i(x_1)$ is a base functions, i.e. $\varphi_i(x_1) = x_1^i$; $i = 1, 2, 3, \cdots, n$; n denotes the number of equation terms; α_i is a fitting coefficient. $\chi_i(x_2)$ is a base functions, i.e. $\chi_i(x_2) = x_2^i$; β_i is a fitting coefficient. $\psi_i(x_3)$ is a base functions, i.e. $\psi_i(x_3) = x_3^i$; γ_i is a fitting coefficient.

The base function set $f_b(x) = \{\varphi(x), \chi(x), \psi(x)\}$ is determined on the basis of the distribution features of data (6.14), where $\varphi(x) = \{\varphi_1(x), \varphi_2(x), \cdots, \varphi_n(x)\}$; $\chi(x) = \{\chi_1(x), \chi_2(x), \cdots, \chi_n(x)\}$; $\psi(x) = \{\psi_1(x), \psi_2(x), \cdots, \psi_n(x)\}$. $f_{qR}(x)$ is a function mapping (N_{qc}, N_{yc}, N_{tr}) to qR in the data set (6.14). The residual value between $f(x_i)$ and $f_{qR}(x_i)$ is denoted by $\triangle_{x_i}$ which is described as an error function: $\| \triangle_{x_i} \|^2 = \sum_{i=1}^{m}[\varphi(x_i) - f_{qR}(x_i)]^2$. Fitting functions are used to find the best fit functions for the experimental data, in another word, the closest functions to the original functions in geometric space. Therefore, the fitting coefficient set $b = \{\alpha, \beta, \gamma\}$ is determined by the minimum value of $\| \triangle_{x_i} \|^2$, i.e. $\frac{\partial}{\partial\alpha\partial\beta\partial\gamma}\| \triangle_{x_i} \|^2 = 0$, which is the least squares method shown as follows, where $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$; $\beta = \{\beta_1, \beta_2, \cdots, \beta_n\}$; $\gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_n\}$.

Equation (6.15) defines the first objective function which is to be maximised:

$$f_{qR}(x) = \sum_{i=1}^{n_{11}} \varphi_i(x_1)\alpha_i + \sum_{j=1}^{n_{12}} \chi_j(x_2)\beta_j + \sum_{k=1}^{n_{13}} \psi_k(x_3)\gamma_k \tag{6.15}$$

Where $f_{qR}(x)$ denotes quay crane service rate function, namely quay crane operational efficiency; i, j and k denote the i^{th}, j^{th} and k^{th} terms of the base functions $\varphi_i(x)$, $\chi_j(x)$ and $\psi_k(x)$ respectively. n_1 denotes the number of terms of equation of $\alpha_1\varphi_1(x_1) + \alpha_2\varphi_2(x_1) + \alpha_3\varphi_3(x_1) + \cdots + \alpha_n\varphi_n(x_1)$, n_2 denotes the number of terms of equation of $\beta_1\chi_1(x_2) + \beta_2\chi_2(x_2) + \beta_3\chi_3(x_2) + \cdots + \beta_n\chi_n(x_2)$ and n_3 denotes the number of terms of equation of $\gamma_1\psi_1(x_3) + \gamma_2\psi_2(x_3) + \gamma_3\psi_3(x_3) + \cdots + \gamma_n\psi_n(x_3)$.

The constraints for the first objective function are: $N_{vsl\,w} > 0$, $N_{qc\,k} > 0$, $N_{qclft\,j\,k} > 0$, $cntrTimQC_{ijk} > 0$, $cntrTidleQC_{ijk} > 0$, $NcnQCLft_{ijk} > 0$.

$(\varphi_k(x_i), \varphi_j(x_i))$ is the inner product of $\varphi_k(x_i)$ and $\varphi_j(x_i)$, i.e. $(\varphi_k(x_i), \varphi_j(x_i)) = \sum_{i=1}^{m} \varphi_k(x_i)\varphi_j(x_i)$. Where k and j are the k^{th} and j^{th} term of their respective e-quations; m denotes the number of data elements, namely the number of equation terms. $(\varphi_k(x_i), f_{rqc}(x_i))$ denotes the inner product of $\varphi_k(x_i)$ and $f_{rqc}(x_i)$. C denotes a matrix of inner products on the basis of experimental data. Λ is a matrix of fitting coefficients. R is a matrix of base functions and experimental data.

$$C = \begin{bmatrix} (\varphi_1, \varphi_1) & (\varphi_1, \varphi_2) & \cdots & (\varphi_1, \varphi_n) \\ (\varphi_2, \varphi_1) & (\varphi_2, \varphi_2) & \cdots & (\varphi_2, \varphi_n) \\ \vdots & \vdots & & \vdots \\ (\varphi_n, \varphi_1) & (\varphi_n, \varphi_2) & \cdots & (\varphi_n, \varphi_n) \\ \\ (\chi_1, \chi_1) & (\chi_1, \chi_2) & \cdots & (\chi_1, \chi_n) \\ (\chi_2, \chi_1) & (\chi_2, \chi_2) & \cdots & (\chi_2, \chi_n) \\ \vdots & \vdots & & \vdots \\ (\chi_n, \varphi_1) & (\chi_n, \chi_2) & \cdots & (\chi_n, \varphi_n) \\ \\ (\psi_1, \psi_1) & (\psi_1, \psi_2) & \cdots & (\psi_1, \psi_n) \\ (\psi_2, \psi_1) & (\psi_2, \psi_2) & \cdots & (\psi_2, \psi_n) \\ \vdots & \vdots & & \vdots \\ (\psi_n, \psi_1) & (\psi_n, \psi_2) & \cdots & (\psi_n, \psi_n) \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \beta_1 & \beta_2 & \ldots & \beta_n \\ \gamma_1 & \gamma_2 & \ldots & \gamma_n \end{bmatrix}$$

$$R = \begin{bmatrix} (\varphi_1, f_{qR}) & (\varphi_2, f_{qR}) & \ldots & (\varphi_n, f_{qR}) \\ (\chi_1, f_{qR}) & (\chi_2, f_{qR}) & \ldots & (\chi_n, f_{qR}) \\ (\psi_1, f_{qR}) & (\psi_2, f_{qR}) & \ldots & (\psi_n, f_{qR}) \end{bmatrix}$$

$$\Lambda^T C = R^T \tag{6.16}$$

Equation (6.16) is a matrix for a system of linear equations. The best-fit fitting coefficient set b is deduced from it.

Second Objective Function: Yard Crane Service Rate

The container terminal yard is normally a keep-out area to external trucks, because of customs supervision and terminal safety reasons. Therefore, employing internal trucks to assume internal container operations is a must. Containers transferred at the gatehouses between internal and external trucks are carried out by yard cranes. Additionally, yard cranes also execute the tasks transferring containers from container blocks to container blocks and internal trucks within the container yard. The performance of yard cranes highly affects the container movements inside terminals as well as the cargo circulation between terminals and outside. Container terminal operations are often bottlenecked by slow yard crane movements [83]. Ng and Mak [103] state that yard cranes very often generate bottlenecks in the container flow in a terminal because of their slow operations. As one of objectives considered in this model, it aims to deploy terminal resources to increase yard crane operational efficiency. Yard cranes are mainly to take responsibility of the pre-marshalling operation and container movements from the gatehouses onto internal trucks. The container movements at the gatehouses denote the container movements from external trucks to internal trucks and, in reverse, from internal trucks to external trucks. Esmer [34] presents that equipment productivity is the number of container moves made per working hour, either for an individual machine or for a particular type of machines and the number of moves can be deduced from data collected. The time of yard crane operations for internal trucks, external trucks and pre-marshalling operations are obtained from the output data in table 6.3.

$tInTrYC_w$ denote the loading and(or) unloading time of containers handled by yard cranes for internal trucks in the time window w.

$$tInTrYC_w = \sum_{k=1}^{N_{InTrYc\,w}} \sum_{j=1}^{N_{yc\,k}} \sum_{i=1}^{N_{YcLftTr\,j\,k}} TiYcInTr_{ijk} \tag{6.17}$$

Where $N_{InTrYc\,w}$ is the number of internal trucks loaded and(or) unloaded by yard cranes in the time window w; $N_{yc\,k}$ is the number of yard cranes loading and(or) discharging cargos to and(or) from the k^{th} truck; $N_{YcLftTr\,j\,k}$ is the number of lifts of the j^{th} yard crane for the k^{th} truck; $TiYcInTr_{ijk}$ is defined in table 6.3.

$tExTrYC_w$ denote the loading and (or) unloading time of containers handled by yard cranes for external trucks in the time window w.

$$tExTrYC_w = \sum_{k=1}^{N_{ExTrYc\,w}} \sum_{j=1}^{N_{yc\,k}} \sum_{i=1}^{N_{YcLftTr\,j\,k}} TiYcExTr_{ijk} \tag{6.18}$$

Where $N_{ExTrYc\,w}$ is the number of External trucks loaded and(or) unloaded by yard cranes in the time window w; $TiYcExTr_{ijk}$ is declared in table 6.3.

$tCYYC_w$ is defined as the loading and(or) unloading time of containers handled by yard cranes in the terminal yard in the time window w.

$$tCYYC_w = \sum_{i=1}^{N_{YdLft}} TiYcCY_i \tag{6.19}$$

Where N_{YdLft} is the number of lifting operations of yard cranes to containers in the terminal yard in the time window w (for pre-marshalling operations transferring containers from container blocks to other blocks in the terminal yard); $TiYcCY_i$ is given in table 6.3.

The total operational time of yard cranes for internal trucks, external trucks and the containers in the terminal yard is denoted by tYc_w which is given in equation (6.20) below.

$$tYc_w = tInTrYC_w + tExTrYC_w + tCYYC_w \tag{6.20}$$

Twin-lift hoist technology is widely applied not only in quay cranes but also in yard cranes as it doubles the number of containers lifted. New twin-lift yard cranes, which means a single crane spreader handles two containers in the same time, are also purchased and employed in some terminals to work with single-lift cranes and achieve good crane productivity and shorter lifting cycle.

$$NcnYcInTr_w = \sum_{k=1}^{N_{InTrYc\ w}} \sum_{j=1}^{N_{yc\ k}} \sum_{i=1}^{N_{YcLftTr\ j\ k}} NcnYcInTr_{ijk} \tag{6.21}$$

$NcnYcInTr_w$ is the number of containers loaded and(or) discharged by yard cranes for internal trucks in the time window w. The definition of $NcnYcInTr_{ijk}$ is given in table 6.3.

$$NcnYcExTr_w = \sum_{k=1}^{N_{ExTrYc\ w}} \sum_{j=1}^{N_{yc\ k}} \sum_{i=1}^{N_{YcLftTr\ j\ k}} NcnYcExTr_{ijk} \tag{6.22}$$

$NcnYcExTr_w$ is the number of containers loaded and(or) discharged by yard cranes for external trucks in the time window w. $NcnYcExTr_{ijk}$ is declared in table 6.3.

$$NcnYcYd_w = \sum_{j=1}^{N_{ycYd}} \sum_{i=1}^{N_{YdLft}} NcnYCLft_{ij} \tag{6.23}$$

$NcnYcYd_w$ is the number of containers loaded and (or) discharged in pre-marshalling operations in the yard by yard cranes in the time window w. N_{ycYd} is the number of yard cranes operate for pre-marshalling operations in the yard. $NcnYCLft_{ij}$ is defined in table 6.3.

$$NcnYC_w = NcnYcInTr_w + NcnYcExTr_w + NcnYcYd_w \qquad (6.24)$$

Where $NcnYC_w$ is the total number of containers operated by yard cranes for internal trucks, external trucks and pre-marshalling. yR denotes the productivity of yard cranes which is the number of containers handled per yard crane per hour, which is equal to yard crane operation time over operated container number.

$$yR = \frac{NcnYC_w}{tYc_w} \qquad (6.25)$$

The following equation is defined as the second objective function (6.26) which is to be maximised.

$$f_{yR}(x) = \sum_{i=1}^{n_{21}} \varphi_i(x_1)\alpha_i + \sum_{j=1}^{n_{22}} \chi_j(x_2)\beta_j + \sum_{k=1}^{n_{23}} \psi_k(x_3)\gamma_k \qquad (6.26)$$

which is fitted by the data set

$$((yR_1, (N_{tr_1}, N_{yc_1}, N_{qc_1})), ((yR_2, (N_{tr_2}, N_{yc_2}, N_{qc_2})), \cdots, ((yR_m, (N_{tr_m}, N_{yc_m}, N_{qc_m})) \qquad (6.27)$$

Base functions $\varphi_i(x)$, $\chi_j(x)$ and $\psi_k(x)$ are determined by the data from table 6.3 and α_i, β_j and γ_k are fitted by equation (6.16).

The constraints for the second objective function are: $N_{InTrYc\,w} > 0$, $N_{yc\,k} > 0$, $N_{YcLftTr\,j\,k} > 0$, $TiYcInTr_{ijk} > 0$, $N_{ExTrYc\,w} > 0$, $TiYcExTr_{ijk} > 0$, $N_{YdLft} > 0$, $TiYcCY_i > 0$, $N_{InTrYc\,w} > 0$, $NcnYcInTr_{ijk} > 0$, $N_{ExTrYc\,w} > 0$, $NcnYcExTr_{ijk} > 0$, $N_{ycYd} > 0$, $N_{YdLft} > 0$, $NcnYCLft_{ij} > 0$.

Third Objective Function: Terminal Yard Congestion Probability

The increase and decrease of equipment quantities might also affect the traffic situation in the yard. An insufficient amount of equipment might cause long waiting times while excess equipment in the yard might increase the traffic burden due to the limited terminal capacity. This tension becomes more apparent in peak hours and the balance is easily damaged.

From a viewpoint of container terminal general layout, land space between container blocks are reserved for traffic use and the gaps amongst container blocks are for yard cranes to lift containers. On the quay side, quay crane displacement only occurs in horizonal direction by moving on its tracks or dedicated roads alongside the berths. Straddle carriers and sprinters, as two types of yard cranes, are able to straddle container blocks and move in block gaps. While another two types of yard cranes, namely reach stackers and empty container handlers, are regularly working in some buffer areas in the yard, they sometimes, but not frequently, share the roads in the yard with internal trucks. If taking the internal trucks into main consideration for waiting time and traffic burden, the length of queues and waiting times are related to the number of internal trucks.

Assume that all internal trucks move in the yard at the same and in an uniform velocity, therefore the distance between internal truck i and its following truck $i+1$, denoted by $Dtr_{i,i+1}$, keeps the same throughout, until they stop for queuing up. $tDtr_{i,i+1}$ denotes the times for the trucks $i+1$ to spend on travelling a distance $Dtr_{i,i+1}$. If the crane operational time of the truck i is shorter than $Dtr_{i,i+1}$, the truck $i+1$ arrives when the truck i finishes its crane operation. Therefore queues do not exist in the system. Conversely, if the crane operational time is longer than $Dtr_{i,i+1}$, stopping and queuing up occur in the system, and then stopped truck will affect its following trucks, congestion therefore happens.

The arrival and departure times of each truck are recorded, and then the time interval between any two trucks can be described as time distances between them. Let the average operational time of cranes be the ideal value for $tDtr_{i,i+1}$, then accordingly, if $tDtr_{i,i+1}$ is less than this value, trucks might need to stop and queue up in front of cranes, namely the probability of waiting and queuing will then increase. On the contrary, if $tDtr_{i,i+1}$ is greater than this value, the probability of waiting and queuing will correspondingly decrease. X denotes the time interval between a random truck and its following truck at the beginning of each task and assume X is exponentially distributed, i.e. $X \sim E(\lambda)$, i.e. $p(x) = \lambda e^{-\lambda x}$. Its mathematical expectation is expressed as $E(X) = \frac{1}{\lambda}$. Let the average value of crane operational time per container be the mathematical expectation, namely the average unit quay crane and yard crane operation time.

$$\lambda = \frac{thrpt \cdot R_{thrcntr} \cdot N_{tr}}{T_{thr}} \tag{6.28}$$

Where $thrpt$ is the throughput of container terminal during the period T_{thr}; $R_{thrcntr}$

denotes the ratio of container number to container throughput.

The average time of a container waiting in queues for quay crane operations is denoted by $tQqc$, which is:

$$tQqc = \frac{\sum_{k=1}^{N_{vsl\ w}} \sum_{j=1}^{N_{qc\ k}} \sum_{i=1}^{N_{qclft\ j\ k}} tQqc_{ijk}}{NcnQC_w} \tag{6.29}$$

Where $tQqc_{ijk}$ is defined in table 6.3.

$tQInTrYC_w$ is defined as the time of containers waiting for yard cranes for internal trucks in the time window w. The total waiting time of internal trucks is given in the following equation.

$$tQInTrYC_w = \sum_{k=1}^{N_{InTrYc\ w}} \sum_{j=1}^{N_{yc\ k}} \sum_{i=1}^{N_{YcLftTr\ j\ k}} QYcInTr_{ijk} \tag{6.30}$$

Where $QYcInTr_{ijk}$ is defined in table 6.3.

$tQExTrYC_w$ denotes the time of containers waiting for yard cranes for external trucks in the time window w. The total waiting time of external trucks is:

$$tQExTrYC_w = \sum_{k=1}^{N_{ExTrYc\ w}} \sum_{j=1}^{N_{yc\ k}} \sum_{i=1}^{N_{YcLftTr\ j\ k}} QYcExTr_{ijk} \tag{6.31}$$

Where $QYcExTr_{ijk}$ is defined in table 6.3.

The total container waiting time for pre-marshalling operations in the yard is denoted by $tQCYYC_w$ which is equal to the equation below.

$$tQCYYC_w = \sum_{j=1}^{N_{ycYd\ w}} \sum_{i=1}^{N_{YdLft\ j}} QYcCY_{ij} \tag{6.32}$$

Where $N_{ycYd\ w}$ is the number of yard cranes employed to terminal yard pre-marshalling operations in the time window w. $N_{YdLft\ j}$ denotes the number of containers lifted by the j^{th} yard crane. $QYcCY_{ij}$ is given out in table 6.3.

The total time of a container waiting in queues for yard crane operations in the time window w is denoted by $tQYc_w$, which equals to summation of waiting times of internal trucks, external trucks and pre-marshalling containers.

$$tQYc_w = tQInTrYC_w + tQExTrYC_w + tQCYYC_w \tag{6.33}$$

The average time of a container waiting in queues for quay crane operations is denoted by $tQyc$, namely total waiting time over the number of containers.

$$tQyc = \frac{tQYc_w}{NcnYC_w} \tag{6.34}$$

$uTQc$ is defined as the unit quay crane operational time per container.

$$uTQc = \frac{1}{qR} \tag{6.35}$$

Similarly, $uTYc$ is defined as the unit yard crane operational time per container.

$$uTYc = \frac{1}{yR} \tag{6.36}$$

The average time that a container spend on quay crane operations, waiting for quay cranes, yard crane operations and waiting for yard cranes is denoted by $tOpsPerCn$. The value of $tOpsPerCn$ is given out in equation (6.37).

$$tOpsPerCn = uTQc + uTYc + tQqc + tQyc \tag{6.37}$$

$f_{ts}(x)$ is defined as the fitting function for the data set below:

$$((tOpsPerCn_1, (N_{tr_1}, N_{yc_1}, N_{qc_1})), ((tOpsPerCn_2, (N_{tr_2}, N_{yc_2}, N_{qc_2})), \\ \cdots, ((tOpsPerCn_m, (N_{tr_m}, N_{yc_m}, N_{qc_m})) \tag{6.38}$$

Namely, $f_{ts}(x)$ is:

$$f_{ts}(x) = \sum_{i=1}^{n_{31}} \varphi_i(x_1)\alpha_i + \sum_{j=1}^{n_{32}} \chi_j(x_2)\beta_j + \sum_{k=1}^{n_{33}} \psi_k(x_3)\gamma_k \tag{6.39}$$

If the crane operational time $f_{ts}(N_{tr_i}, N_{yc_j}, N_{qc_k})$ is longer than the truck departure time interval X_i, i.e. $X_i < f_{ts}(N_{tr_i}, N_{yc_j}, N_{qc_k})$, then the probability of the i^{th} truck arriving at a quay crane or yard crane when the previous truck is still in its operation is high. The i^{th} truck then very possibly needs to wait in a queue. And the latter trucks might also need to wait in queues if their arrival times are before the operation finishing time of the trucks in front of them. The cumulative probability of $X_i < f_{ts}(N_{tr_i}, N_{yc_j}, N_{qc_k})$ is denoted by $P\{X_i < f_{ts}(N_{tr_i}, N_{yc_j}, N_{qc_k})\}$ which is also the objective function for the third objective shown by equation (6.40) which is to be minimised.

$$P\{X_i < f_{ts}(N_{tr_i}, N_{yc_j}, N_{qc_k})\} = \int_{-\infty}^{f_{ts}(N_{tr_i}, N_{yc_j}, N_{qc_k})} \lambda e^{-\lambda x}\, dx \tag{6.40}$$

The constraints for the third objective function are: $thrpt > 0$, $R_{thrcntr} > 0$, $N_{tr\,j} > 0$, $T_{thr} > 0$, $N_{vsl\,w} > 0$, $N_{qc\,k} > 0$, $N_{qclft\,j\,k} > 0$, $tQqc_{ijk} > 0$, $NcnQC_w > 0$, $N_{InTrYc\,w} > 0$, $N_{yc\,k} > 0$, $N_{YcLftTr\,j\,k} > 0$, $QYcInTr_{ijk} > 0$, $N_{ExTrYc\,w} > 0$, $QYcExTr_{ijk} > 0$, $N_{ycYd\,w} > 0$, $N_{YdLft\,j} > 0$, $QYcCY_{ij} > 0$.

Fourth Objective Function: Internal Truck Travelling Distance

Terminal quay cranes work alongside container vessels on the berths, while some types of yard cranes such as straddle carriers work in the container blocks as they move over the top of container blocks and others use roads in the yard. The terminal control centre normally gives instructions to each quay crane and yard crane to stay in certain areas in the yard to avoid making frequent crane movements. Quay cranes are fixed alongside the berths on tracks or dedicated roads, while some of yard cranes are dedicated to the gatehouses to serve the external trucks. Even for the rest of yard cranes in the yard, they usually work in certain areas and then move to another when their work is finished. A lot of movements have been made by internal trucks because container positions are normally very scattered in the yard. Cranes are only in charge of operations in their areas and it is not their responsibilities when containers move out of their areas. But internal trucks are another case. When a truck loads and carries a container, it must takes the container to appointed position to discharge, and then it is free to load the next container. Therefore, the travelling routes of trucks are more scattered than quay cranes and yard cranes.

The purpose of this objective function is to simulate the operations of internal trucks and find out the travelling distances which are to be minimised. Reducing truck travelling distances also satisfies truck fuel consumption and emission targets.

The fourth objective function is defined by equation (6.42), which is to be minimised and fitted by the data set following.

$$((DTr_{k_1}, N_{tr_1}), (DTr_{k_2}, N_{tr_2}), \cdots, (DTr_{k_n}, N_{tr_n})) \tag{6.41}$$

Base functions $\varphi_i(x)$, $\chi_j(x)$ and $\psi_k(x)$ are determined by above data set and α_i, β_j and γ_k are fitted by equation (6.16).

$$f_d(x) = \sum_{i=1}^{n_{41}} \varphi_i(x_1)\alpha_i \tag{6.42}$$

The constraints for the fourth objective function are: DTr_{k_i}¿0, $N_{tr_i} > 0$, where $i = 1, 2, 3, ..., n$.

6.3.4 Multi-objective Optimisation

As discussed in section 5.3.4, the multi-objective optimisation model for this chapter is based on equation (5.25):

$$\begin{aligned} &minimise \quad \{-f_{qR}(x), -f_{yR}(x), f_{con}(x), f_d(x)\} \\ &subject\ to \quad x \in S \end{aligned} \tag{6.43}$$

Where $f_{qR}(x)$ and $f_{yR}(x)$ are to be maximised, and $f_{con}(x)$ and $f_d(x)$ are to be minimised. The decision vector is $x = (N_{qc}, N_{yc}, N_{tr})^T$.

6.3.5 A Genetic Algorithm to Solve the Problem

Exact algorithms cannot solve larger scale problem in acceptable duration [75]. The explicit enumeration algorithm proposed in section 5.4.3 is not suitable for this problem because its complexity and it is computational time consuming as it needs a large number of traversals to find all possible solutions. An experiment using an explicit enumeration algorithm for this problem is implemented and the time taken is more than 24 hours. Genetic algorithms are well suited to solve multi-objective optimisation problems and have been the most popular heuristic approach to multi-objective design and optimisation problems [69]. Jones et al. [61] state that 70% of the multi-objective meta-heuristic articles between 1991 and 1998 utilise genetic algorithms as the primary meta-heuristic.

Zitzler and Thiele [150] propose a fitness function using a Pareto ranking approach. In this chapter, a Pareto ranking method to set fitness function is employed which is the number of solutions dominate in the elements in population over the population size. $f_{fitness}(i, p, t)$ is the fitness function, where i denotes one of chromosomes from the population p at the t^{th} generation.

$$f_{fitness}(i, p, t) = \frac{dp(i, p, t)}{N_p} \tag{6.44}$$

Where $dp(i, p, t)$ is the number of solutions could not dominate the i^{th} solution in the population p at the t^{th} generation. In other words, the i^{th} solution could not be dominated by $dp(i, p, t)$ solutions in the population p at the t^{th} generation.

Roulette wheel selection is used to choose chromosomes to evolve in this chapter. The probability to be selected is:

$$P_{chromo_i} = \frac{f_{fitness}(i,p,t)}{\sum_{i=1}^{N_p} f_{fitness}(i,p,t)} \tag{6.45}$$

Where P_{chromo_i} is the probability of the i^{th} chromosome to be selected to evolve. $P_{chromo_i} >= 0$, $\sum_{i=1}^{N_p} P_{chromo_i} = 1$.

The parameters crossover rate, mutation rate, population size and number of generations need to be determined based on a series of tests. The output of the algorithm is a set of solutions to the model.

6.4 Parameters and Results

In this section, the parameters and results for the models proposed are discussed based on the Southampton Container Terminal. The parameters for simulation model, single objective optimisation functions, the multi-objective optimisation model and the genetic algorithm are analysed and addressed as follows. The results for simulation model are processed for multi-objective optimisation parameters. Finally the solutions to the multi-objective model are explored by a genetic algorithm and the elite solutions to the models are categorised and discussed.

6.4.1 Simulation Parameters and Results

The historical data from the terminal are collected and processed for the scenarios for the simulation models. The parameters for simulation include stochastic and non-stochastic parameters as described in section 6.3.2. The parameters for the simulation model of container terminal operations and simulation model for internal truck travelling distance are discussed respectively.

For the simulation model of container terminal operations, there are a series of parameters to be set. For the stochastic parameters, as discussed in section 5.4.1, suppose X_1 is exponential distributed with a parameter θ_1. Other distributions follow the gamma distribution with parameters (θ_i, θ_{i+1}). The distribution functions and parameters of stochastic variables are estimated and listed in table 6.4. In table 6.4,

θ_1 is expressed as:

$$\theta_1 = \frac{simulationtime}{(percThrputToCntrNum \cdot throughput/365)} \tag{6.46}$$

Where $simulationtime$ denotes the length of simulation duration in hour. $percThrput$ $ToCntrNum$ denotes the ratio of container number to throughput volume. As the discussion on the settings of distribution parameters in section 5.4.1, an appropriate estimation of distribution parameters is important to the model. Assume that $\theta_2 = \theta_1$, $\theta_3 = 3.0$ minutes, $\theta_4 = 4.0$ minutes, $\theta_5 = 2.0$ minutes, $\theta_6 = 2.5$ minutes, $\theta_7 = 2.0$ minutes, $\theta_8 = \theta_1$, $\theta_9 = \theta_1$, $\theta_{10} = 1.5$ minutes, $\theta_{11} = 3$ minutes. For the non-stochastic

Table 6.4: Stochastic Parameters for Simulation

Stochastic Parameters	$F_i(x)$	θ_i
New task interval time distribution	$X_1 \sim E(x)$	θ_1
Truck assigning time distribution	$X_2 \sim \Gamma(x)$	$(\theta_2, \theta_2/2)$
Truck travelling time distribution	$X_3 \sim \Gamma(x)$	$(\theta_3, \theta_3/2)$
Time distribution of yard crane operations from yard to trucks	$X_4 \sim \Gamma(x)$	$(\theta_4, \theta_4/2)$
Time distribution of yard crane operations from trucks to yard or trucks	$X_5 \sim \Gamma(x)$	$(\theta_5, \theta_5/2)$
Time distribution of quay crane operations from ships to trucks	$X_6 \sim \Gamma(x)$	$(\theta_6, \theta_6/2)$
Time distribution of quay crane operations from trucks to ships	$X_7 \sim \Gamma(x)$	$(\theta_7, \theta_7/2)$
Quay crane assigning time distribution	$X_8 \sim \Gamma(x)$	$(\theta_8, \theta_8/2)$
Yard crane assigning time distribution	$X_9 \sim \Gamma(x)$	$(\theta_9, \theta_9/2)$
Quay crane travelling time distribution	$X_{10} \sim \Gamma(x)$	$(\theta_{10}, \theta_{10}/2)$
Yard crane travelling time distribution	$X_{11} \sim \Gamma(x)$	$(\theta_{11}, \theta_{11}/2)$

parameters, they are given in table 6.5. Some of data in table 6.5 is the same as table 5.4. The number of internal trucks in Southampton Container Terminal is 100 in 2009 [123]. In table 6.5, the "Direct Importation container percentage" and "Direct Exportation container percentage" are unavailable at the moment but they are needed to increase the reliability of models. Therefore, the values for them in table 6.5 are estimated by personal knowledge and recognition of container terminals, which can be changed according to real situations. The "Imported containers:Exported containers ratio" is the same as the "Empty containers:laden containers ratio" in table 5.4. The "Simulation Time" is also set as the same value as table 5.4.

The values of numbers of equipment resources $N_{qc\,j}$, $N_{yc\,j}$ and $N_{tr\,j}$ are set as follows, where $N_{qc\,j}$, $N_{yc\,j}$ and $N_{tr\,j}$ are the values of N_{qc}, N_{yc} and N_{tr} for the j^{th} simulation iteration respectively, where j denotes the j^{th} simulation iteration, $j = 0, 1, 2, \cdots, N_{sim}$, and N_{sim} is the number of simulation experiments.

$$N_{qc\,j} = \frac{j \cdot (maxN_{qc} - minN_{qc})}{N_{sim}} + minN_{qc} \tag{6.47}$$

$$N_{yc\,j} = \frac{j \cdot (maxN_{yc} - minN_{yc})}{N_{sim}} + minN_{yc} \tag{6.48}$$

$$N_{tr\,j} = \frac{j \cdot (maxN_{tr} - minN_{tr})}{N_{sim}} + minN_{tr} \tag{6.49}$$

Table 6.5: Non-Stochastic Parameters for Simulation [86] [120] [113] [122]

Parameter	Value
Container throughput	Data from 2000 to 2009
Quay crane quantity	Data from 2000 to 2009
Yard crane quantity	Data from 2000 to 2009
Internal truck quantity	Data from 2009
Direct Importation container percentage	35 %
Direct Exportation container percentage	20 %
Empty containers:laden containers ratio	3:7
Imported containers:Exported containers ratio	7:3
Transhipment containers: all containers ratio	0.5:10
Pre-marshalling percentage	20%
$maxN_{qc}$	50
$minN_{qc}$	1
$maxN_{yc}$	500
$minN_{yc}$	20
$maxN_{tr}$	500
$minN_{tr}$	20
Value of N_{values}	5
Simulation time	10 hours

The output variables are given in table 6.3 and the results are obtained from the execution of simulation model. The size of simulation results are too large to be shown, so in order to reduce the data size, the simulation results are processed to show objective function values for each iteration according to equations (6.15), (6.26), (6.39) and (6.42). The results are given in table 6.6.

For the simulation model for internal truck travelling distance, the accurate positions of container blocks in the Southampton Container Terminal have not been obtained. Therefore a set of simple position data is used to run the model and they can be replaced by realistic data as long as the realistic data is available. Assume the distance unit is metre and the coordinate of the container terminal yard is (0,1800, 0,1800). Assume that there are nine container blocks in this container terminal: (300,500,300,500), (800,1000,300,500), (1300,1500,300,500), (300,500,800,1000), (800,1000,800,1000), (1300,1500,800,1000), (300,500,1300,1500), (800,1000,1300,1500), (1300,1500,1300,1500).

In order to avoid noisy data and reduce the instability of stochastic variables, 100,000 iterations are run to find out the internal truck travelling distance when a certain number of internal trucks are employed in the yard.

The simulation results of internal truck travelling distance are given out in table 6.6.

Table 6.6: Simulation Results

N_{qc}	N_{yc}	N_{tr}	$f_{qR}(x)$ (Container/ Crane/Hour)	$f_{yR}(x)$ (Container/) Crane/Hour)	$f_{ts}(x)$ (Hour)	DTr_k (Metre)
1	20	20	27.28415591	3.250211316	0.383213308	225722.8831
10	116	116	19.35329776	4.126791575	0.36575516	219335.0686
10	500	116	21.25520572	1.015512384	0.352777665	219335.0686
20	116	20	2.139956685	0.824189473	0.411071814	225722.8831
20	212	212	10.42679301	2.52812848	0.34151678	210888.4621
20	212	308	10.62033723	2.600016112	0.349717374	206700.7097
20	404	212	13.68067875	1.809823919	0.358383725	210888.4621
30	212	116	6.408772594	2.282611217	0.352654364	219335.0686
30	308	308	8.811010035	2.07112758	0.348130246	206700.7097
30	308	404	8.319870111	2.028887167	0.346287645	207997.9636
40	116	404	6.065873182	4.714437551	0.343042864	207997.9636
40	308	212	6.626657054	2.155288507	0.363165667	210888.4621
40	404	404	7.212672493	1.66876446	0.336093756	207997.9636
40	404	500	7.589034613	1.830526738	0.35620818	208142.0858
50	20	500	1.448064966	9.74370683	0.408366748	208142.0858
50	500	500	7.024786228	1.912204551	0.34979966	208142.0858

6.4.2 Single Objective Formulation Parameters and Results

In equations (6.15), (6.26), (6.39) and (6.42) in section 6.3.3, the distribution features of simulation data from table 6.3 determine the type of base function set $f_b(x) = \{\varphi(x), \chi(x), \psi(x)\}$, values of fitting coefficient set $b = \{\alpha, \beta, \gamma\}$ and value of n. According to the features of plane graph of simulation output, suppose n_i powered functions are good mathematical representations for equations (6.15), (6.26), (6.39) and (6.42), which are shown in following respectively.

$$f_{qR}(x) = \sum_{i=1}^{n_{11}} N_{qc}^{i-1}\alpha_{i+10} + \sum_{j=1}^{n_{12}} N_{yc}^{j-1}\beta_{j+10} + \sum_{k=1}^{n_{13}} N_{tr}^{k-1}\gamma_{k+10} \tag{6.50}$$

$$f_{yR}(x) = \sum_{i=1}^{n_{21}} N_{qc}^{i-1}\alpha_{i+20} + \sum_{j=1}^{n_{22}} N_{yc}^{j-1}\beta_{j+20} + \sum_{k=1}^{n_{23}} N_{tr}^{k-1}\gamma_{k+20} \tag{6.51}$$

$$f_{ts}(x) = \sum_{i=1}^{n_{31}} N_{qc}^{i-1}\alpha_{i+30} + \sum_{j=1}^{n_{32}} N_{yc}^{j-1}\beta_{j+30} + \sum_{k=1}^{n_{33}} N_{tr}^{k-1}\gamma_{k30} \tag{6.52}$$

$$f_d(x) = \sum_{i=1}^{n_{41}} N_{qc}^{i-1}\alpha_{i+40} \tag{6.53}$$

The values of n_{11}, n_{12}, n_{13}, n_{21}, n_{22}, n_{23}, n_{31}, n_{32}, n_{33} and n_{41} for each above equation are shown in table 6.7 respectively. Numerical experiments have been done to test values of R^2 to find good fitting powers for the functions. Furthermore, the results

Table 6.7: Values of n_i

i	n_{i+1}	n_{i+2}	n_{i+3}
10	3	3	4
20	3	4	4
30	4	4	5
40	3	Nil	Nil

obtained from the simulation model are used to fit the coefficient set $b = \{\alpha, \beta, \gamma\}$ through the above equations (6.50), (6.51), (6.52) and (6.53). The fitting coefficient set b is computed by equation (6.16) from simulation data. The fitting solutions are then exported to equations (6.50), (6.51), (6.52) and (6.53). The fitting coefficients for the objective functions are given out in table 6.8.

In this container terminal, the number of drivers in a single truck and yard crane are 2, while in a quay crane is 1.

Table 6.8: Fitting Coefficients

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function: $f_{qR}(x)$			Yc Efficiency Objective Function: $f_{yR}(x)$	
α_{11}	1×10^{-9}		α_{21}	0.00138	
α_{12}	-5×10^{-7}		α_{22}	-0.06177	
α_{13}	1.05×10^{-5}		α_{23}	1.19754	
β_{11}	0.076705		β_{21}	-5×10^{-8}	
β_{12}	5×10^{-7}		β_{22}	5×10^{-5}	
β_{13}	-0.0002765	0.79	β_{23}	-0.0236	0.97
γ_{11}	0.07537		β_{24}	3.6338	
γ_{12}	0.01287		γ_{21}	2×10^{-8}	
γ_{13}	-1.077417		γ_{22}	-1.2×10^{-5}	
γ_{24}	28.06		γ_{23}	0.00194	
			γ_{24}	0.37966	
	Average Crane Operational Time Function: $f_{ts}(x)$			Truck Travelling Distance Function: $f_d(x)$	
α_{31}	1×10^{-7}		α_{41}	0.1388	
α_{32}	-4.5×10^{-6}		α_{42}	- 109.71	0.98
α_{33}	1.5×10^{-5}		α_{43}	228541	
α_{34}	0.018755				
β_{31}	-1.35×10^{-10}				
β_{32}	1.5×10^{-7}				
β_{33}	-6×10^{-5}	0.76			
β_{34}	0.060885				
γ_{31}	1.6×10^{-11}				
γ_{32}	-1.6×10^{-8}				
γ_{33}	5.6×10^{-6}				
γ_{34}	-0.00088				
γ_{35}	0.33336				

6.4.3 Genetic Algorithm Parameters

The genetic algorithm is employed to search the near Pareto optimal solutions to the models. The source codes of the genetic algorithm are given in appendix C. There are four major parameters to be set for the genetic algorithm: population size, number of generations, crossover rate and mutation rate. A few of experiments have been conducted to determined appropriate parameters for the algorithm. The general idea is to change one or two of the four parameters and observe the change of results. Then change another parameter and observe the changes.

According to previous researches, crossover rate is usually set between 0.5 and 0.9, while mutation rate is usually around 0.01. Taking the median of common values of crossover rate, 0.7 is set as an initial parameter to the genetic algorithm to explore the near Pareto optimal solutions. And mutation rate is taken 0.01 as an initial value. The first group of tests use different values for population size to run the algorithm and observe the fitness values from 1 to 1000 generations. There are five graphs below showing fitness values with population sizes from 50 to 300 and generation(s) from 1 to 1000. Figures (6.17), (6.18), (6.19), (6.20) and (6.21) record the fitness values for every generation with using different values of population size.

As shown in graphs, the data in graph (6.17) are scattered throughout and most of fitness values are below 0.85. Figure (6.18) shows a steadily rising trend with the increase of number of generations. The fitness values mainly distribute between 0.75 and 0.9 at the generation around 800. Figure (6.19) has an increasing trend as well. The peak values are around the 800^{th} generation and at that point, the data is more focused than figure (6.18). The average fitness value of figure (6.20) is lower than graphs (6.18) and (6.19). In figure (6.21), the trend is rising from 1 to approximately 380 generations, but descending from 380 to roughly 700 generations, and then increasing again. The peak values of fitness are around 380 generations but they are lower than figure (6.19). Seen from the peak values in figure (6.19), the combination of 150 populations and 800 generations seems to be a potentially good choice viewing from the experiments.

The second group of data sets are observed to test different values of crossover rates. Crossover rates are set as 0.5, 0.6, 0.7, 0.8 and 0.9 and the results are shown below. The other parameters are set as: population size is 150; maximum generation

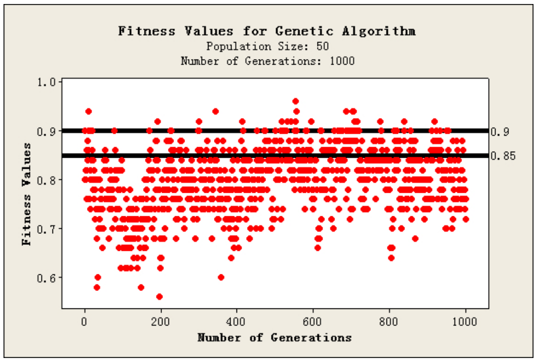

Figure 6.17: Fitness Values for the Genetic Algorithm (Population:50, Generations:1000)

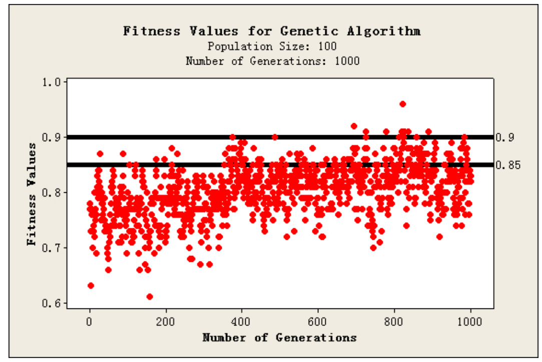

Figure 6.18: Fitness Values for the Genetic Algorithm (Population:100, Generations:1000)

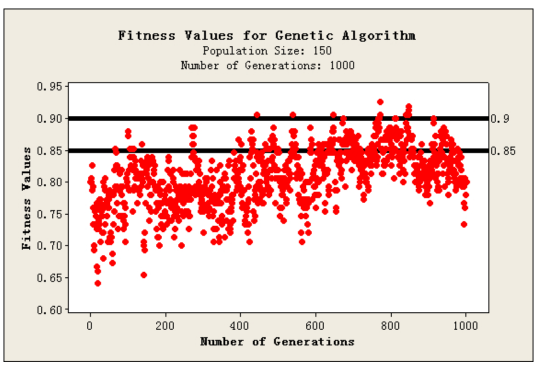

Figure 6.19: Fitness Values for the Genetic Algorithm (Population:150, Generations:1000)

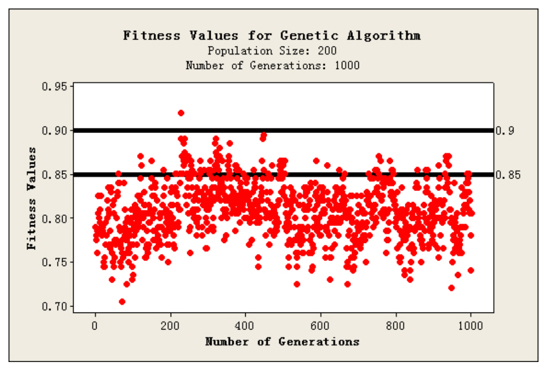

Figure 6.20: Fitness Values for the Genetic Algorithm (Population:200, Generations:1000)

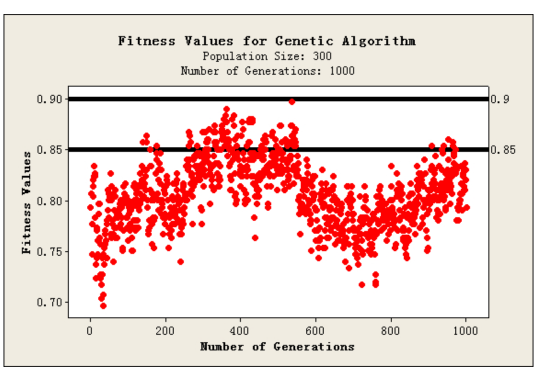

Figure 6.21: Fitness Values for the Genetic Algorithm (Population:300, Generations:1000)

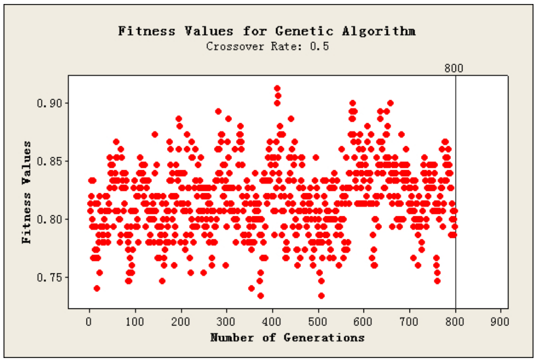

Figure 6.22: Fitness Values for the Genetic Algorithm (Crossover Rate: 0.5)

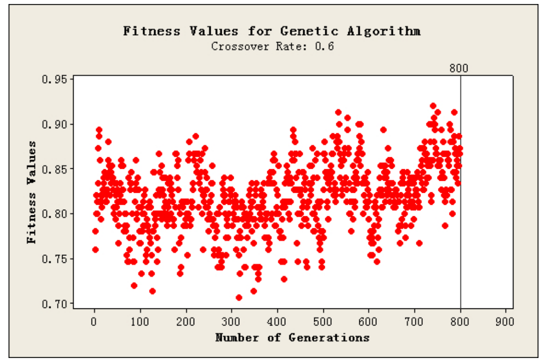

Figure 6.23: Fitness Values for the Genetic Algorithm (Crossover Rate: 0.6)

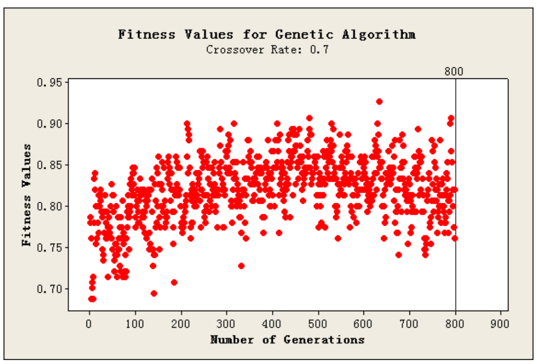

Figure 6.24: Fitness Values for the Genetic Algorithm (Crossover Rate: 0.7)

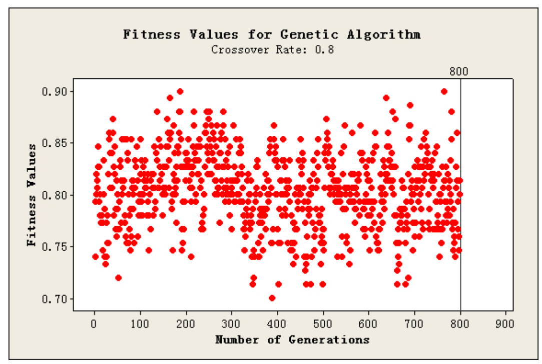

Figure 6.25: Fitness Values for the Genetic Algorithm (Crossover Rate: 0.8)

is 800; mutation rate is 0.01. The results are shown in graphs (6.22), (6.23), (6.24), (6.25), (6.26) respectively. Viewing from the graphs, figures (6.23) and (6.24) show less scattered than the other three at the generation around 800. Furthermore, figure (6.23) seems to have the highest average value at the points close to the 800^{th} generation than other figures.

The third group of data sets is to select an appropriate value for mutation rate. Setting the other three parameters as: population size 150; maximum generation 800 and crossover rate 0.6. The mutation values are tested by using values 0.005, 0.01, 0.02, 0.05, 0.07 and 0.1. The fitness values obtained from the algorithm by using above settings are given out in figures (6.27), (6.28), (6.29), (6.30), (6.31), (6.32) respectively. Figure (6.30) shows better and more focused fitness values than the other five figures at generations close to 800.

Therefore the parameters for genetic algorithm are set as population size 150,

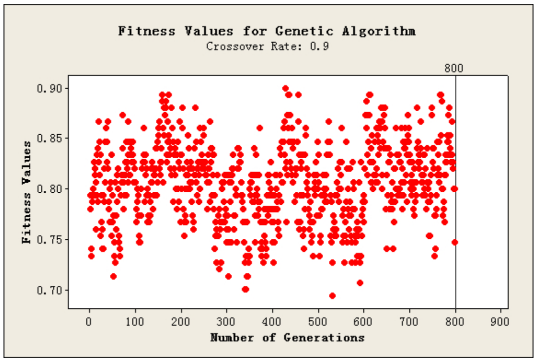

Figure 6.26: Fitness Values for the Genetic Algorithm (Crossover Rate: 0.9)

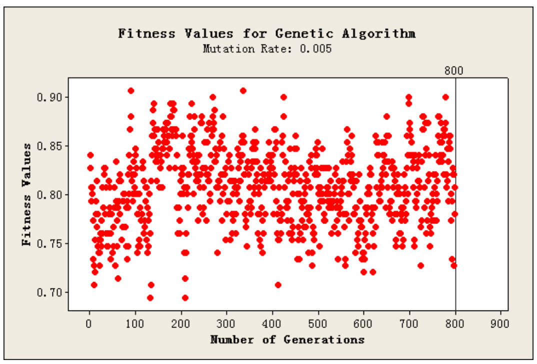

Figure 6.27: Fitness Values for the Genetic Algorithm (Mutation Rate: 0.005)

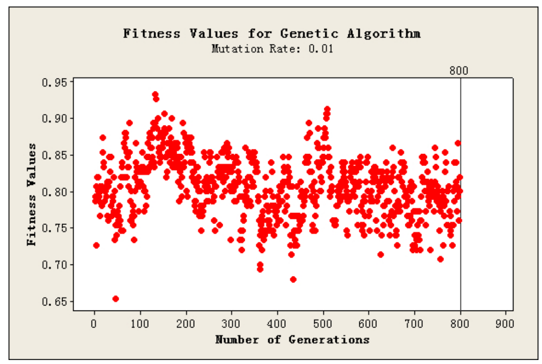

Figure 6.28: Fitness Values for the Genetic Algorithm (Mutation Rate: 0.01)

maximum generation 800, crossover rate 0.6 and mutation rate 0.05.

6.4.4 Multi-objective Optimisation Results

The near Pareto solutions to this four objective problem are explored in the objective space and decision space. O denotes the objective space which is a four dimensional space. Two of four dimensions are to be maximised and the other two are to be minimised. D denotes the decision space which is an integral and three dimensional space.

150 (equals to the population size) solutions are obtained from the genetic algorithm. In order to reduce the number of solutions for decision makers and retain the goodness of solutions, the solutions which have less fitness values than 95% are deleted. Finally, 98 elite solutions are provided to decision makers.

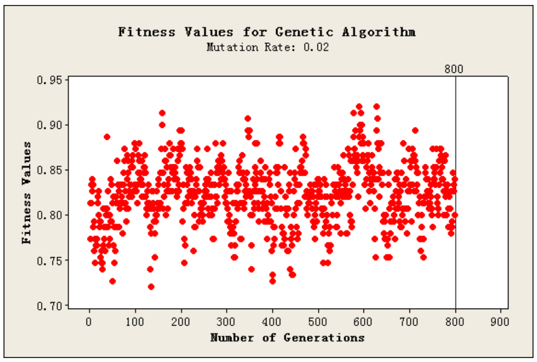

Figure 6.29: Fitness Values for the Genetic Algorithm (Mutation Rate: 0.02)

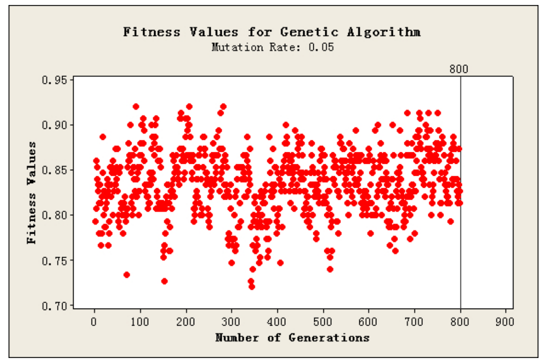

Figure 6.30: Fitness Values for the Genetic Algorithm (Mutation Rate: 0.05)

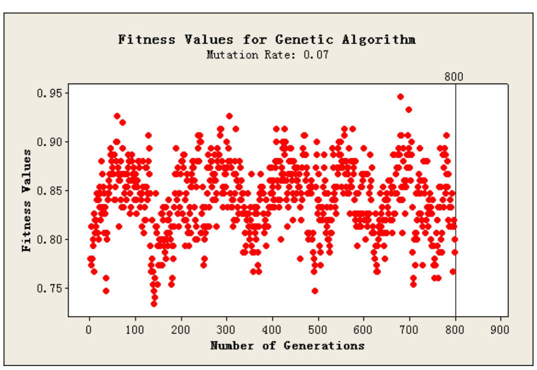

Figure 6.31: Fitness Values for the Genetic Algorithm (Mutation Rate: 0.07)

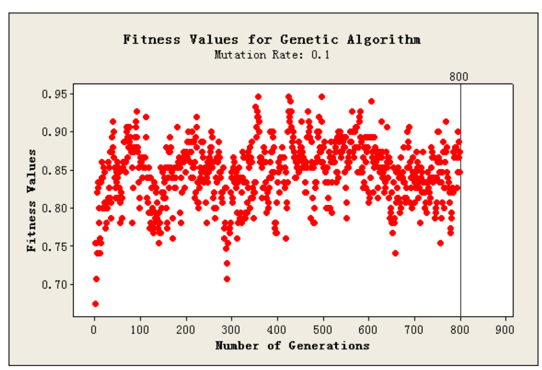

Figure 6.32: Fitness Values for the Genetic Algorithm (Mutation Rate: 0.1)

The solutions are normalised on axes into a range from 0 to 1, to solve the problem of difference in value spans and units of solutions to each objective function. The solutions are normalised by equation (6.54), where RS_{ij} denotes the j^{th} normalised solution for the i^{th} objective function; S_{ij} is the j^{th} original solution for the i^{th} objective function; $Smin_i$ is the minimum solution for the i^{th} objective function; $Smax_i$ denotes the maximum solution for the i^{th} objective function. Table 6.9 gives the maximum and minimum values for each function for equation (6.54).

$$RS_{ij} = \frac{S_{ij} - Smin_i}{Smax_i - Smin_i} \tag{6.54}$$

The normalised solutions to equation (6.43) are shown in figure (6.33). Solution

Table 6.9: The Maximum and Minimum Values of Objective Functions

$Smax_1$	26.84
$Smin_1$	5.35
$Smax_2$	4.07
$Smin_2$	0.09
$Smax_3$	0.6328
$Smin_3$	0.6324
$Smax_4$	226178.21
$Smin_5$	210778.36

information must be substantial enough to allow an informed decision to be made by the decision maker but not so large as to overwhelm him/her with information [99]. Therefore 10 elite and representative solutions for each objective function are selected from figure (6.33) and given in figures (6.34), (6.35), (6.36) and (6.37) for decision supporting. Figure (6.34) gives 10 best solutions to the first objective function (terminal quay crane efficiency function). They are between 0.85 and 1 to the first objective function. Solutions 3, 8 and 9 also have good values for the second objective function, while solutions 2, 6 and 7 are close to 0.4, but solutions 1, 4, 5 and 10 are close to 0.1. These solutions are quite bad for the third objective function. Solutions 1 and 8 have values around 0.75 for the fourth objective function, solution 6 and 10 around 0.4, and others are between 0.1 and 0.3.

The best values for the second objective function are shown in figure (6.35). They all have values greater than 0.8 for the second objective function. For the first objective function, solutions 14, 16 and 18 are between 0.85 and 0.95, solutions 13 and 11 are around 0.7 and 0.55 respectively, and other five solutions are less 0.2.

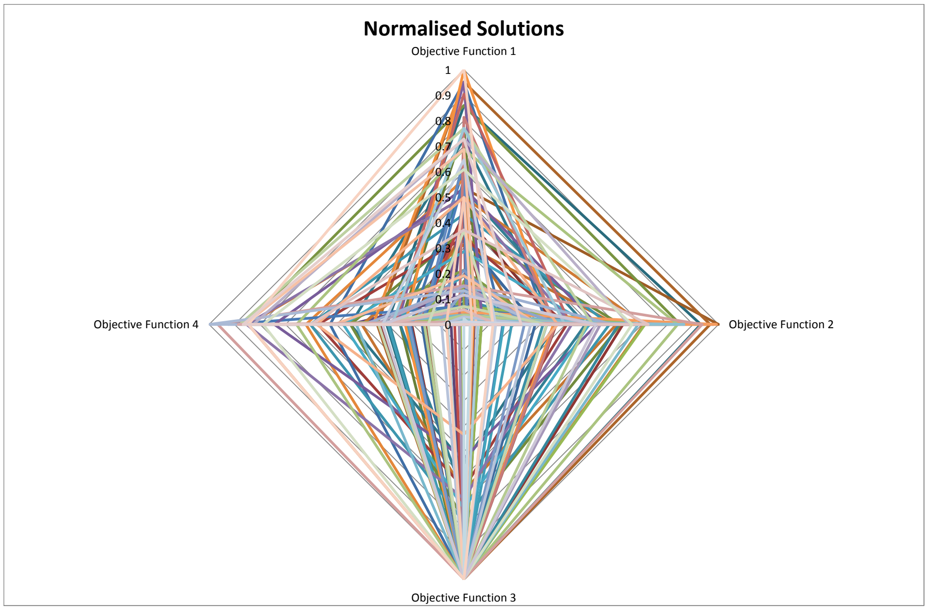

Figure 6.33: Elite Normalised Solutions

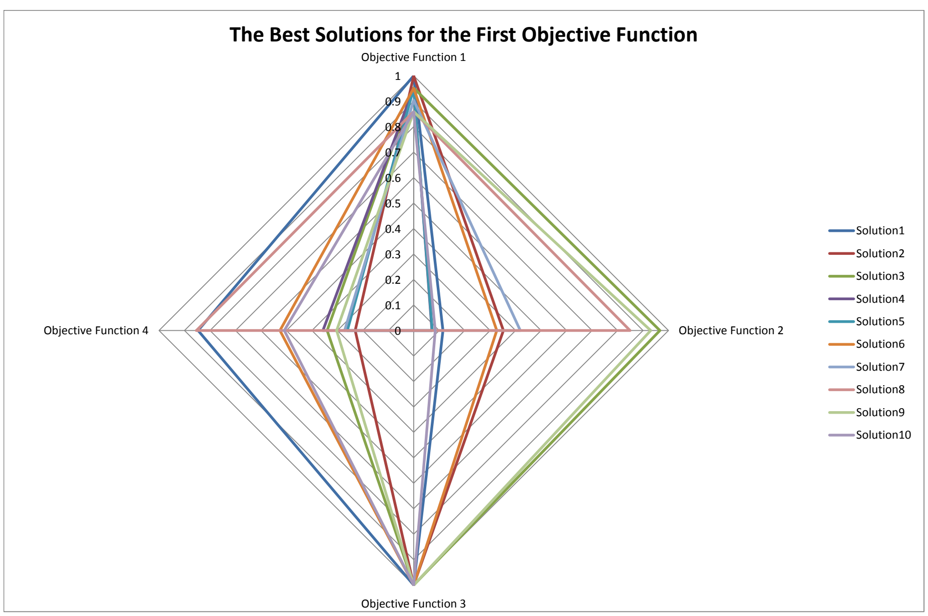

Figure 6.34: The Best Solutions for the First Objective Function

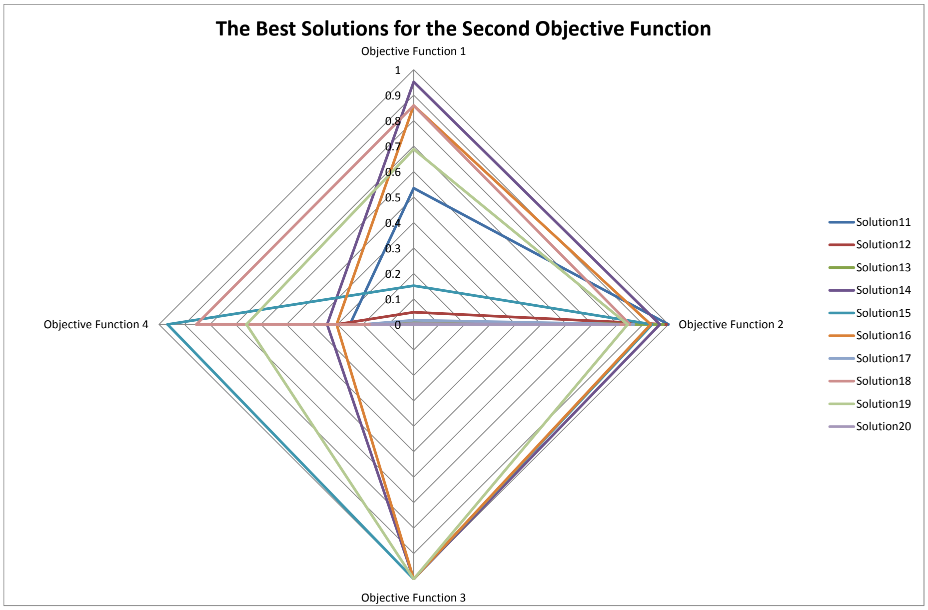

Figure 6.35: The Best Solutions for the Second Objective Function

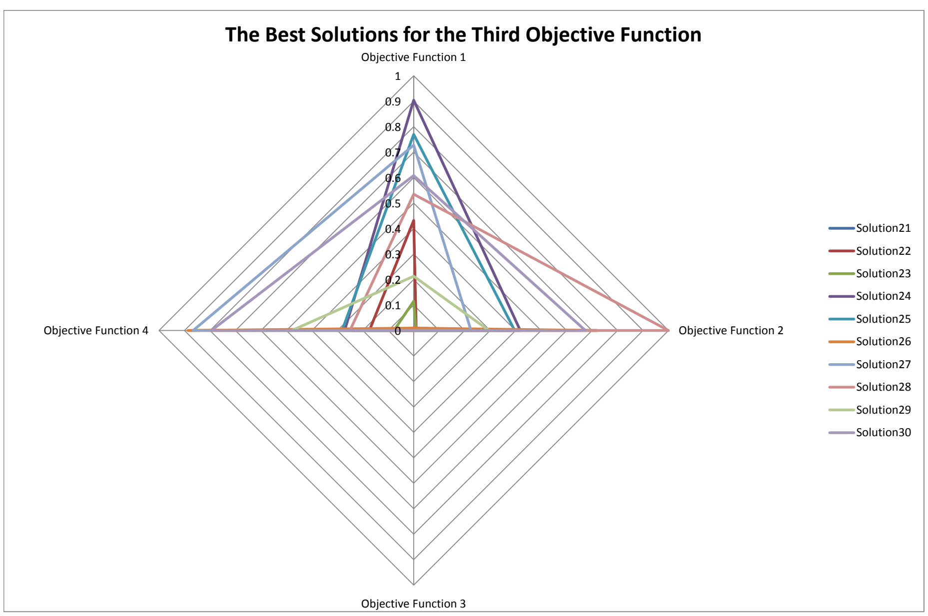

Figure 6.36: The Best Solutions for the Third Objective Function

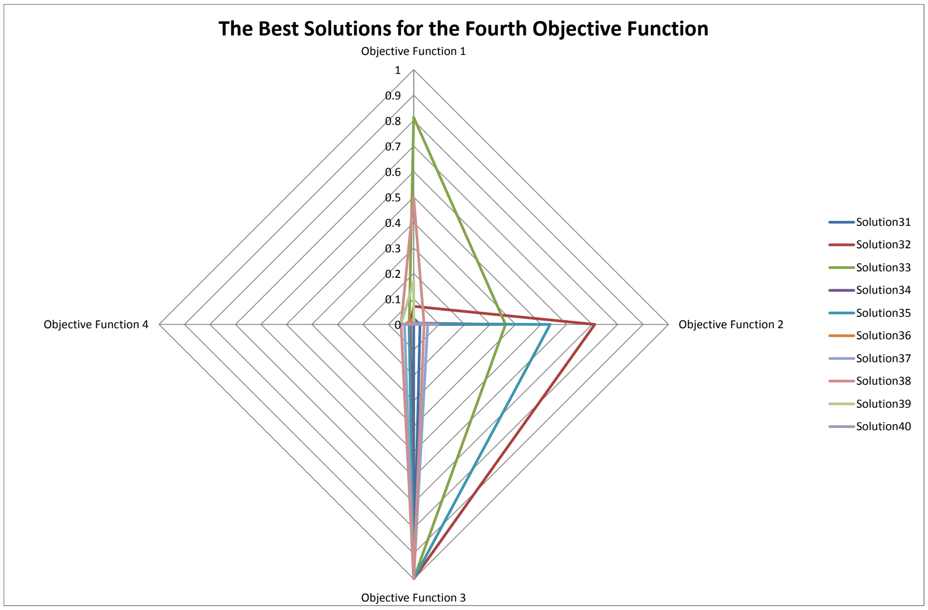

Figure 6.37: The Best Solutions for the Fourth Objective Function

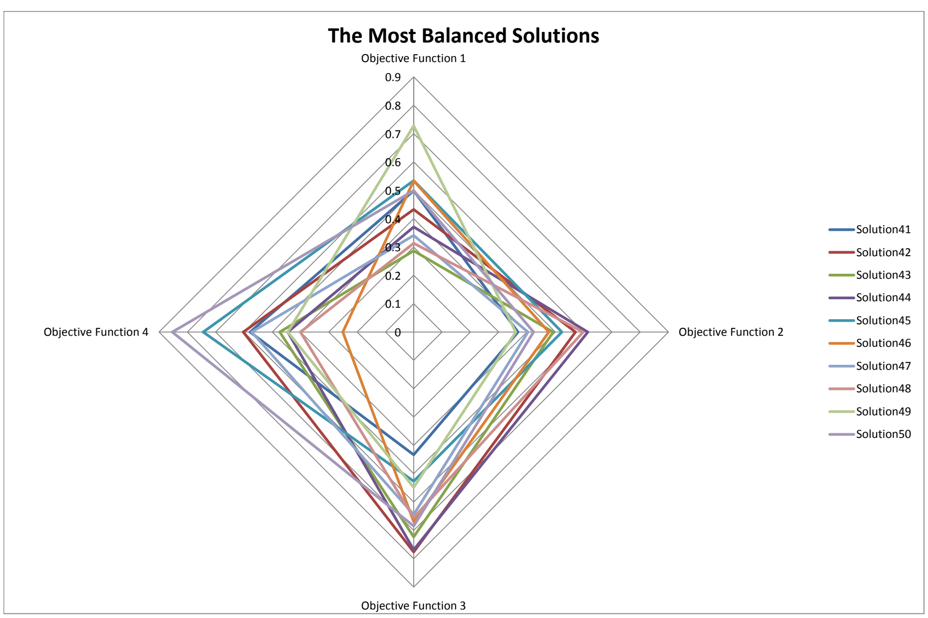

Figure 6.38: The Most Balanced Solutions for All Objective Functions

The best values for the third objective function are shown in figure (6.36). All of them are close to the minimum value. Solution 28 has good values for the second and third objective functions. Solutions 24 has good values for the first and third objective functions as well as adequate values for the other two objective functions. Solutions 25 and 27 show balance in all goals as well.

The fifth graph (6.37) shows the shortest internal truck travelling distance. They have poor values for the other three goals except for solutions 32, 33, 35 and 38. Solutions 32 and 35 have better than average values on the second dimension other than the first dimension. Solution 38 shows an around 0.5 value for the first goal. Solution 33 is much better except for its third dimension.

If decision maker prefer more balance amongst all objectives rather than focusing on one or two of them, figure (6.38) is a better choice. The 10 solutions in the graph have more emphasis on trade-off decisions between terminal productivity and cost. No one of them has the best values for single objective functions but there are still some of them having good values for a single objective and values roughly between 0.3 and 0.7 for other three goals. Solution 43 has 0.73 for the first dimension, 0.36, 0.55 and 0.44 for the other three. Solution 46 has good value for average truck travelling distance (0.25), 0.53 and 0.48 for quay cran and yard crane productivity. Solution 44 has 0.62 for yard crane operational efficiency, 0.37 and 0.44 for quay crane rate and average truck travelling distance per container.

Solutions are simplified and categorised for difference preferences. Appropriate volume rather than overwhelming information is given to decision makers for reference. Different combinations of equipment quantities can be derived from the elite solutions if any of them is chosen by container terminal operators.

6.4.5 Computational Considerations

Computational time consists of three parts: the simulation of general terminal operations built in the Micro Saint Sharp simulation package, the simulation for truck travelling distance and the genetic algorithm to explore the near Pareto solutions. The computer to run the models is given as follows: processor: Intel i3 M350 at 2.27 GHz; memory: 3.00GB. Software environment is 64-bit Microsoft Windows 7. Simulation model is run in the Micro Saint Sharp 3.0 Windows version. Simulation for

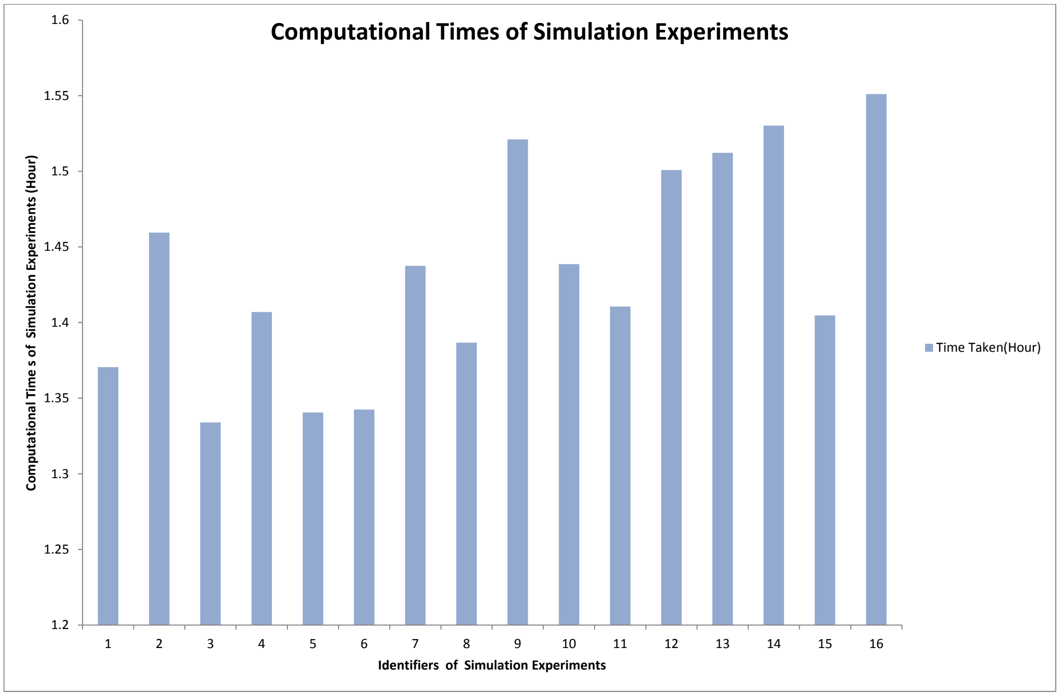

Figure 6.39: Computational Times of Simulation Experiments

truck travelling distance is implemented in C # coded in Microsoft Visual Studio 2010 Express. 16 simulation iterations on the Micro Saint Sharp are run synchronously on one computer. The computational times for each experiment are shown in the graph (6.39). The total computational time for simulation is the experiment using the longest time which is 1.55 hours. For the simulation for internal truck travelling distance, 16 simulation experiments are also run by using the same parameters. Each iteration runs 100000 times to guarantee the accuracy level. Therefore the simulation for truck travelling distance takes 4 minutes and 12.49 seconds. Finally, the genetic algorithm takes 1 minute and 1.93 seconds. In total, the computational time is 1.64 hours.

6.5 Summary

This chapter addresses the equipment optimisation problem in modern container terminals, proposes a model for the equipment quantity decision making for container terminal operators and discusses solutions. In this chapter, simulation and multi-objective optimisation are combined based on simulation models for container terminal operations and a multi-objective optimisation model with four objectives to support terminal daily decision. Simulation has strengths in its ability of effectively implementing complex stochastic processes of container terminals and providing real time data, on the other hand, multi-objective optimisation has better performance in computational time and the exploration of the near Pareto solutions by genetic algorithms. Therefore, a combination is proposed to integrate the strengths of two methods in order to provide optimised equipment quantities for daily decision making in a container terminal in effective and efficient ways. The genetic algorithm used to find the solutions to the models is potentially feasible to solve larger problems in short time. The model is applied based on the data from the Southampton Container Terminal and the solutions show that the model offers various and effective solutions for container terminal operators.

Chapter 7

Combining Simulation and Multi-Objective Optimisation by Integrated MOO Structure for Multiple Container Terminal Equipment Optimisation

7.1 Introduction

Chapter 6 presents a series of methods to solve the multiple mechanical equipment optimisation problem in container terminals by using the post-MOO structure. There are two factors which are very important to the quality of solutions when using post-MOO structure. Firstly, the parameter settings for initial values of independent variables. The initial values of independent variables are set by equations (6.47), (6.48) and (6.49) in chapter 6. However, in the case that decisions are made without knowledge of the solutions because the solutions are what is to be provided to decision makers, the quality of initial values can not be guaranteed to reflect the features of all solutions because of lack of knowledge of the solutions. Secondly, control the value of R^2 of data fitting. As R^2 is used to measure the errors of fitting functions, fitting functions are regarded as valid representation of experimental data only when fitting functions are at a high level of accuracy, i.e. the values of R^2 are close to 1. In order to reduce the influence of initial values and errors of data fitting, another combination structure, namely MOO leading integrated structure, is proposed to solve the problem. The structure decreases the number of initial values to reduce the possible errors caused by initial values and increases the reliability of fitting functions because simulation only runs with optimised parameters which are searched by dynamic MOO search.

This chapter is organised as follows. The integrated MOO structure is addressed in section 7.2. Parameter settings and results are given in section 7.3. A summary is given in section 7.4.

7.2 Integrated MOO Structure

This chapter solves the problem by MOO leading integrated MOO structure in the combination framework shown in figure 4.1. The processes of MOO leading Integrated structure for the multiple equipment optimisation problem and the definitions of the "Data Processing" and "Searching Techniques" shown in figure 4.4 are defined in section 4.3.2. As a further study of the problem raised in chapter 6, this chapter is based on the models proposed in section 6.3. To be detailed, the simulation model is given in section 6.3.2. Formulation for objectives are developed in equations (6.15), (6.26), (6.40) and (6.42). The multi-objective model is given in equation (6.43). Chapter 6 solves the problem by using post-MOO structure, while this chapter aims to solve the same problem by another structure, namely MOO leading integrated structure, to improve the goodness of results and efficiency of computation.

Furthermore, as defined in section 4.3.2, data fitting is employed to find fitting coefficients and based functions from data streams. Additionally, dynamic MOO search is also defined to guide the search for appropriate parameters for the next iteration. The quality of results and computational cost are compared between the two structures at the end of the process. The stopping conditions are defined in section 4.3.2 to control the number of iterations. The number of consecutive iterations that the search could not find better solutions than the current best value(s) is set as three, namely, if the search could not find better solutions than the current best in three consecutive iterations, then the process will stop.

7.3 Parameters and Results

Initial parameters need to be set in advance to run models. The initial parameters are set with the same values for the four objectives if preferences of decision makers are not taken into consideration. Hence, the process of the initial iteration is the same for the four objectives which is given in section 7.3.1. The explorations of the near Pareto optimal solutions for each objective are discussed in sections 7.3.2, 7.3.3, 7.3.4 and 7.3.5 respectively.

7.3.1 The Initial Iteration for All Objectives

As models given out in section 6.3, in order to solve the problem, initial values of decision variables are required to be set at the beginning. The number and values of the initial values of decision variables need to be determined. In order to reduce the influence of initial values on the combination framework performance, the minimum number of initial values of the decision variables needs to be set. The initial values are deduced from equations (4.1), (4.2) and (4.3). Let the number of initial values, i.e. N_{InVal}, equal three. The number of initial values has been reduced to 3 from 16 comparing to the post-MOO structure. Therefore the initial values are shown in table 7.1. Then the combination framework starts from the initial values to search

Table 7.1: Initial Values for Decision Variables

Initial Values	Nqc	Nyc	Ntr
Initial Value 1	13	140	140
Initial Value 2	25	260	260
Initial Value 3	38	380	380

for the near Pareto optimal solutions.

The initial values of the decision variables are input into simulation as parameters as shown in figure (4.1). The simulation model starts with the three groups of initial values. The simulation results are input into the multi-objective optimisation model. The results for the four objectives are shown in table 7.2.

Table 7.2: Simulation Results for the First Iteration for All Objective Functions

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380

The simulation times for the three iterations in table 7.2 are 0.190, 0.214 and 0.247 hours respectively, while the total simulation time for truck travelling distance optimisation for the three experiments is 0.188 hours. Therefore the total simulation time for the first iteration is 0.839 hours.

Then the data stream of simulation output goes into "Data Processing" as demonstrated in figure (4.4). The data fitting is used to fit the data stream as shown in figure (4.5). The values for n_i, as defined in equations (6.50), (6.51), (6.52) and (6.53), are set as shown in table 7.3. A few of experiments have been implemented to test the values of R^2 to find good values for n_i. The fitting coefficients are given in table

Table 7.3: Values of n_i for the First Iteration for All Objective Functions

i	n_{i+1}	n_{i+2}	n_{i+3}
10	3	3	3
20	3	3	3
30	3	3	3
40	3	Nil	Nil

7.4.

As fitting coefficients and n_i have been given, near Pareto optimal solutions to the fitting functions are found by the genetic algorithm. The best values of decision variables for the next iteration can be found from the Pareto solutions. The genetic algorithm based on sections 6.3.5 is employed to search for the near Pareto optimal solutions to the multi-objective optimisation model. The C# source codes for the genetic algorithm for dynamic MOO search in this chapter are very similar to appendix C. The parameter settings for the genetic algorithm are addressed in section 6.4.3. The results of the genetic algorithm are shown in table 7.5. The first row in the table is the best predicted solution for the first objective which has the same best value as the second objective. The best predicted values for the third and fourth objectives are given in the third and fourth rows respectively.

The time taken for the search is 1 minutes and 2 seconds. Therefore the total time of the first iteration including simulation and search is 0.856 hours.

The processes for further explorations of the near Pareto optimal solutions for each objective are presented in following sections.

Table 7.4: Fitting Coefficients for the First Iteration for All Objective Functions

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function: $f_{qR}(x)$			Yc Efficiency Objective Function: $f_{yR}(x)$	
α_{11}	0.0051207235		α_{21}	0.000977871	
α_{12}	−0.3599956362		α_{22}	−0.07367058	
α_{13}	8.7515951463		α_{23}	1.992917592	
β_{11}	0.0000520426		β_{21}	9.76723×10^{-6}	
β_{12}	−0.0373578559	0.99	β_{22}	−0.007558045	0.99
β_{13}	9.1471189320		β_{23}	2.067148623	
γ_{11}	0.0000520426		γ_{21}	9.76723×10^{-6}	
γ_{12}	−0.0373578559		γ_{22}	−0.007558045	
γ_{13}	9.1471189320		γ_{23}	2.067148623	
	Traffic Congestion Probability Function: $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function: $f_d(x)$	
α_{31}	0.0000007861		α_{41}	−0.049159302	
α_{32}	−0.0000400894		α_{42}	63.0208441	0.99
α_{33}	0.2113231676		α_{43}	201811.44	
β_{31}	0.0000000085				
β_{32}	−0.0000044282	0.99			
β_{33}	0.2113878865				
γ_{31}	0.0000000085				
γ_{32}	−0.0000044282				
γ_{33}	0.2113878865				

Table 7.5: Best Predicted Values for the First Iteration for All Objective Functions

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
24.34	5.58	0.6328	
24.34	5.58	0.6328	
8.32	1.92	0.6324	
20.60	4.80	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
203597.70	2	27	29
203597.70	2	27	29
220621.94	43	377	473
203416.75	12	56	26

7.3.2 The Near Optimal Solutions to the First Objective Function

The data processing and dynamic MOO search have been discussed in section 4.3.2. This section addresses the processes of further searches for the solutions to each objective.

Second Iteration for the First Objective Function

The predicted values in table 7.5 are set as simulation parameters, which are input into "Data Processing" (namely data fitting as defined) as shown in figure (4.4). Then two simulation models are executed with the predicted parameters. The simulation results are input into the multi-objective optimisation model as parameters. The simulation results for the four objectives are shown in table 7.6.

Table 7.6: Simulation Results for the Second Iteration for the First and Second Objective Functions

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
24.54	4.68	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207374.80	2	27	29

The time taken for the simulation model to run in the Micro Saint Sharp package is 0.164 hours and the running time for the truck travelling distance simulation is 29 seconds. The total time for simulation is 0.172 hours.

L_{qR} denotes a data list represents equation (7.1) for the first objective function.

$$((qR_1, yR_1, P_1, DTr_{k_1}, N_{tr_1}, N_{yc_1}, N_{qc_1}), (qR_2, yR_2, P_2, DTr_{k_2}, N_{tr_2}, N_{yc_2}, N_{qc_2}), \cdots, (qR_m, yR_m, P_m, DTr_{k_m}, N_{tr_m}, N_{yc_m}, N_{qc_m})) \quad (7.1)$$

Where qR_i denotes the quay crane productivity of the i^{th} experiment, yR_i denotes the yard crane productivity of the i^{th} experiment, P_i denotes the traffic congestion probability in the yard of the i^{th} experiment and DTr_{k_i} denotes the internal truck travelling distance of the i^{th} experiment. $i = 1, 2, 3, \cdots, m$. m is the number of experiments.

The data in table 7.2 is added to L_{qR}. In the second iteration, the new simulation data from table 7.6 is then added into L_{qR}. The updated L_{qR} list is shown in table 7.7. L_{qR} now has four groups of values of the decision variables and objective functions.

Table 7.7: $L_{qR}(L_{yR})$ List for the Second Iteration for the First(Second) Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
24.54	4.68	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207374.80	2	27	29

Then the data is fitted by equation (6.16).

Seeing from the features of simulation output, suppose n_i powered functions shown in equations (6.50), (6.51), (6.52) and (6.53) are good mathematical representations for equations (6.15), (6.26), (6.40) and (6.42) respectively for the data list L_{qR}. A few of experiments have been implemented to test values of R^2 to find good values for n_i and the results are shown in table 7.8. Moreover, the coefficient set $b = \{\alpha, \beta, \gamma\}$ is

Table 7.8: Values of n_i for the Second Iteration for the First and Second Objective Functions

i	n_{i+1}	n_{i+2}	n_{i+3}
10	3	3	3
20	3	3	3
30	4	4	4
40	3	Nil	Nil

fitted by the data via equation (6.16). The fitting coefficients for the first objective function are shown in table 7.9.

As fitting coefficients and n_i have been given, near Pareto optimal solutions to the fitting functions are found by the genetic algorithm. The output from the multi-objective optimisation model are input into the "Searching Techniques" (defined as

Table 7.9: Fitting Coefficients for the Second Iteration for the First and Second Objective Functions

Coeffi-cients	Fitting Values	R^2	Coeffi-cients	Fitting Values	R^2
	Qc Efficiency Objective Function: $f_{qR}(x)$			Yc Efficiency Objective Function: $f_{yR}(x)$	
α_{11}	0.0053508587		α_{21}	0.000464379	
α_{12}	-0.3723870925		α_{22}	-0.046022028	
α_{13}	8.8965838759		α_{23}	1.669410148	
β_{11}	0.0000521355		β_{21}	4.10513×10^{-6}	
β_{12}	-0.0374084785	0.99	β_{22}	-0.004472042	0.99
β_{13}	9.1532061685		β_{23}	1.696064984	
γ_{11}	0.0000534167		γ_{21}	4.2795×10^{-6}	
γ_{12}	-0.0381072057		γ_{22}	-0.004565381	
γ_{13}	9.2373160906		γ_{23}	1.706929995	
	Traffic Congestion Probability Function: $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function: $f_d(x)$	
α_{31}	0.000000047633333		α_{41}	0.021013177	
α_{32}	-0.000002834600000		α_{42}	24.75315068	0.99
α_{33}	0.000044186500000		α_{43}	$206,417.62$	
α_{34}	0.210734808233333				
β_{31}	0.000000000033333				
β_{32}	-0.000000030266667				
β_{33}	0.000004943233333	0.99			
β_{34}	0.210699853833333				
γ_{31}	0.000000000066667				
γ_{32}	-0.000000030866667				
γ_{33}	0.000005084100000				
γ_{34}	0.210689512300000				

dynamic MOO search for this problem) as shown in figure (4.4). The best values of decision variables are then selected from the Pareto solutions for the next iteration. The results shown in table 7.10 are the best predicted values from the Pareto solutions to the multi-objective optimisation model.

Table 7.10: Best Predicted Values for the Second Iteration for the First and Second Objective Functions

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
24.74	4.76	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207101.27	2	23	27

The best predicted values of the decision variables, from which it is possible to obtain the near Pareto optimal solutions, are 2 quay cranes, 23 yard cranes and 27 trucks in the decision space. The best predicted simulation value for the first objective is 24.74. The time taken for the search is 1 minute and 2 seconds.

Third Iteration for the First Objective Function

The processes of following iterations are similar to the second iteration for the first objective function, therefore description in following iterations will be shortened due to their similarities.

The predicted values in table 7.10 are set as simulation parameters, which are input into the “Data Processing”. Then simulation models run with the parameters and the results are shown in table 7.11. The time taken for the simulation model to

Table 7.11: Simulation Results for the Third Iteration for the First and Second Objective Functions

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
23.16	5.24	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207460.33	2	23	27

run in the Micro Saint Sharp package is 0.168 hours and the running time for the truck travelling distance simulation is 26 seconds.

The updated L_{qR} list is shown in table 7.12. The values of n_i are shown in table 7.13. Moreover, the fitting coefficients are shown in table 7.14. The near Pareto optimal values found by dynamic MOO search are shown in table 7.15. The best predicted values of the decision variables are 1 quay crane, 20 yard cranes and 20 trucks and the best predicted simulation value for the first objective is 24.63. The time taken for the search is 1 minute and 2 seconds.

Table 7.12: $L_{qR}(L_{yR})$ List for the Third Iteration for the First(Second) Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
24.54	4.68	0.6324	
23.16	5.24	0.6324	
Truck Travelling Distance (Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207374.80	2	27	29
207460.33	2	23	27

Table 7.13: Values of n_i for the Third Iteration for the First and Second Objective Functions

i	n_{i+1}	n_{i+2}	n_{i+3}
10	3	3	3
20	3	4	2
30	4	4	4
40	3	Nil	Nil

Fourth Iteration for the First Objective Function

The predicted values in table 7.15 are set as simulation parameters, which are input into the “Data Processing”. Then simulation models run with the parameters and the results are shown in table 7.16. The time taken for the simulation model to run

Table 7.14: Fitting Coefficients for the Third Iteration for the First and Second Objective Functions

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function: $f_{qR}(x)$			Yc Efficiency Objective Function: $f_{yR}(x)$	
α_{11}	0.0049391687		α_{21}	0.000617705	
α_{12}	-0.3502199620		α_{22}	-0.054277721	
α_{13}	8.6372129030		α_{23}	1.766007546	
β_{11}	0.0000461830		β_{21}	5.6124×10^{-6}	
β_{12}	-0.0342236292	0.99	β_{22}	-0.0052785	0.98
β_{13}	8.7826740224		β_{23}	1.789890012	
γ_{11}	0.0000482594		β_{24}	5.8529×10^{-6}	
γ_{12}	-0.0353213088		γ_{21}	-0.005415301	
γ_{13}	8.9075615588		γ_{22}	1.807531327	
	Traffic Congestion Probability Function: $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function: $f_d(x)$	
α_{31}	0.000000047633333		α_{41}	0.024558722	
α_{32}	-0.000002834600000		α_{42}	22.83789781	0.99
α_{33}	0.000044186500000		α_{43}	206644.32	
α_{34}	0.210734808233333				
β_{31}	0.000000000033333				
β_{32}	-0.000000029500000				
β_{33}	0.000004770400000	0.86			
β_{34}	0.210710958466667				
γ_{31}	0.000000000066667				
γ_{32}	-0.000000030500000				
γ_{33}	0.000005003433333				
γ_{34}	0.210695034933333				

Table 7.15: Best Predicted Values for the Third Iteration for the First and Second Objective Functions

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
24.63	5.10	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207110.90	1	20	20

Table 7.16: Simulation Results for the Fourth Iteration for the First and Second Objective Functions

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
27.28	3.25	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207218.14	1	20	20

in the Micro Saint Sharp package is 0.151 hours and the running time for the truck travelling distance simulation is 23 seconds.

The updated L_{qR} list is shown in table 7.17. The values of n_i are shown in table

Table 7.17: $L_{qR}(L_{yR})$ List for the Fourth Iteration for the First(Second) Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
24.54	4.68	0.6324	
23.16	5.24	0.6324	
27.28	3.25	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207374.80	2	27	29
207460.33	2	23	27
207218.14	1	20	20

7.18. Moreover, the fitting coefficients are shown in table 7.19.

The near Pareto optimal values found by dynamic MOO search are shown in table 7.20. The best predicted values of the decision variables are 2 quay cranes, 24 yard cranes and 36 trucks and the best predicted simulation value for the first objective is 24.50. The time taken for the search is 1 minute and 2 seconds.

Table 7.18: Values of n_i for the Fourth Iteration for the First and Second Objective Functions

i	n_{i+1}	n_{i+2}	n_{i+3}
10	3	3	3
20	5	5	5
30	4	4	4
40	3	Nil	Nil

Fifth Iteration for the First Objective Function

The predicted values in table 7.20 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.21. The time taken for the simulation model to run in the Micro Saint Sharp package is 0.169 hours and the running time for the truck travelling distance simulation is 34 seconds.

The updated L_{qR} list is shown in table 7.22. The values of n_i are shown in table 7.23. Moreover, the fitting coefficients are shown in table 7.24.

The near Pareto optimal values found by dynamic MOO search are shown in table 7.25. The best predicted values of the decision variables are 1 quay crane, 24 yard cranes and 25 trucks and the best predicted simulation value for the first objective is 26.10. The time taken for the search is 1 minute and 2 seconds.

Sixth Iteration for the First Objective Function

The predicted values in table 7.25 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.26. The time taken for the simulation model to run in the Micro Saint Sharp package is 0.182 hours and the running time for the truck travelling distance simulation is 26 seconds.

The updated L_{qR} list is shown in table 7.27. The values of n_i are shown in table 7.28. Moreover, the fitting coefficients are shown in table 7.29.

The near Pareto optimal values found by dynamic MOO search are shown in table 7.30. The best predicted values of the decision variables are 3 quay cranes, 33 yard cranes and 85 trucks and the best predicted simulation value for the first objective is 22.91. The time taken for the search is 1 minute and 2 seconds.

Table 7.19: Fitting Coefficients for the Fourth Iteration for the First and Second Objective Functions

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function $f_{qR}(x)$			Yc Efficiency Objective Function $f_{yR}(x)$	
α_{11}	0.0055880061		α_{21}	-0.0000570080000	
α_{12}	-0.3827559728		α_{22}	0.0044681456667	
α_{13}	8.9672572075		α_{23}	-0.1101701878333	
			α_{24}	0.8701456903667	
			α_{25}	0.3190171317667	
β_{11}	0.0000532807		β_{21}	-0.0000000066333	
β_{12}	-0.0379317016	0.98	β_{22}	0.0000053640333	0.90
β_{13}	9.1950608579		β_{23}	-0.0013921444333	
			β_{24}	0.1212647864667	
			β_{25}	-0.6655818771333	
γ_{11}	0.0000553010		γ_{21}	-0.0000000066000	
γ_{12}	-0.0389323403		γ_{22}	0.0000053664667	
γ_{13}	9.2943299356		γ_{23}	-0.0013992994667	
			γ_{24}	0.1233844211667	
			γ_{25}	-0.8333558039000	
	Traffic Congestion Probability Function $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function $f_d(x)$	
α_{31}	0.0000000439333		α_{41}	0.025422534	
α_{32}	-0.0000025889000		α_{42}	22.39492425	0.99
α_{33}	0.0000396608000		α_{43}	$206,691.76$	
α_{34}	0.2107547913667				
β_{31}	0.0000000000474				
β_{32}	-0.0000000286399				
β_{33}	0.0000045951618	0.67			
β_{34}	0.2107206268494				
γ_{31}	0.0000000000473				
γ_{32}	-0.0000000286375				
γ_{33}	0.0000046360410				
γ_{34}	0.2107135534186				

Table 7.20: Best Predicted Values for the Fourth Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
24.50	5.20	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207530.93	2	24	36

Table 7.21: Simulation Results for the Fifth Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
27.96	5.40	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207411.73	2	24	36

Table 7.22: L_{qR} List for the Fifth Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
24.54	4.68	0.6324	
23.16	5.24	0.6324	
27.28	3.25	0.6324	
27.96	5.40	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207374.80	2	27	29
207460.33	2	23	27
207218.14	1	20	20
207411.73	2	24	36

Table 7.23: Values of n_i for the Fifth Iteration for the First Objective Function

i	n_{i+1}	n_{i+2}	n_{i+3}
10	3	3	3
20	5	5	5
30	4	4	4
40	3	Nil	Nil

Seventh Iteration for the First Objective Function

The predicted values in table 7.30 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.31. The time taken for the simulation model to run in the Micro Saint Sharp package is 0.179 hours and the running time for the truck travelling distance simulation is 1 minute and 6 seconds.

The updated L_{qR} list is shown in table 7.32. The values of n_i are shown in table 7.33. Moreover, the fitting coefficients are shown in table 7.34.

The near Pareto optimal values found by dynamic MOO search are shown in table 7.35. The best predicted values of the decision variables are 4 quay cranes, 46 yard cranes and 53 trucks and the best predicted simulation value for the first objective is 23.56. The time taken for the search is 1 minute and 2 seconds.

Eighth Iteration for the First Objective Function

The predicted values in table 7.35 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.36. The time taken for the simulation model to run in the Micro Saint Sharp package is 0.186 hours and the running time for the truck travelling distance simulation is 41 seconds.

The updated L_{qR} list is shown in table 7.37. In table 7.37, the fifth iteration obtains the best value for the first objective which is 27.96. The sixth, seventh and eighth iterations obtain worse values than this. The search process stops according to the stopping criteria in section 4.3.2.

The best value for the first objective function searched by the genetic algorithm in the post-MOO structure is 26.84 moves/quay crane/hour, while the best value obtained from the MOO leading integrated structure is higher (27.96 moves/quay

Table 7.24: Fitting Coefficients for the Fifth Iteration for the First Objective Function

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function $f_{qR}(x)$			Yc Efficiency Objective Function $f_{yR}(x)$	
α_{11}	0.0059991849		α_{21}	-0.0000623179000	
α_{12}	-0.4062255054		α_{22}	0.0048770083667	
α_{13}	9.2699324489		α_{23}	-0.1199669625667	
			α_{24}	0.9451162392000	
			α_{25}	0.2534398049333	
β_{11}	0.0000577562		β_{21}	-0.0000000074333	
β_{12}	-0.0403971509	0.97	β_{22}	0.0000059984000	0.97
β_{13}	9.4969592419		β_{23}	-0.0015514921667	
			β_{24}	0.1344600521333	
			β_{25}	-0.8239311124667	
γ_{11}	0.0000581028		γ_{21}	-0.0000000050333	
γ_{12}	-0.0410038594		γ_{22}	0.0000040996333	
γ_{13}	9.6580078106		γ_{23}	-0.0010731408667	
			γ_{24}	0.0945544138333	
			γ_{25}	-0.3209182352000	
	Traffic Congestion Probability Function $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function $f_d(x)$	
α_{31}	0.0000000439000		α_{41}	0.025176244	
α_{32}	-0.0000025905667		α_{42}	22.57702484	0.99
α_{33}	0.0000398754667		α_{43}	206659.79	
α_{34}	0.2107515776667				
β_{31}	0.0000000000475				
β_{32}	-0.0000000286832				
β_{33}	0.0000046066979	0.62			
β_{34}	0.2107196744973				
γ_{31}	0.0000000000452				
γ_{32}	-0.0000000274658				
γ_{33}	0.0000044989030				
γ_{34}	0.2107091673425				

Table 7.25: Best Predicted Values for the Fifth Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
26.10	4.11	0.6324	
Truck Travelling Distance (Metre)	Nqc	Nyc	Ntr
207239.95	1	24	25

Table 7.26: Simulation Results for the Sixth Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
26.03	2.89	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207122.86	1	24	25

Table 7.27: L_{qR} List for the Sixth Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
24.54	4.68	0.6324	
23.16	5.24	0.6324	
27.28	3.25	0.6324	
27.96	5.40	0.6324	
26.03	2.89	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207374.80	2	27	29
207460.33	2	23	27
207218.14	1	20	20
207411.73	2	24	36
207122.86	1	24	25

Table 7.28: Values of n_i for the Sixth Iteration for the First Objective Function

i	n_{i+1}	n_{i+2}	n_{i+3}
10	3	3	3
20	5	5	5
30	4	4	4
40	3	Nil	Nil

crane/hour). Seeing from the computational volume, it takes 16 simulation iterations and 1 genetic algorithm search in post-MOO structure, while the integrated MOO structure takes 10 simulation iterations and 7 genetic algorithm searches. To compare computational time, the computational time for simulation in the post-MOO structure is: the simulation in Micro Saint Sharp takes 1.55 hours and the simulation of truck travelling distance takes 4 minutes and 12.49 seconds. The genetic algorithm takes 1 minute and 1.93 seconds. Therefore it is 1.64 hours in total for the post-MOO structure. On the other hand, the computational time for the MOO leading integrated structure is: the simulation in Micro Saint Sharp takes 1.85 hours and simulation for travelling distance takes 15 minutes and 21 seconds. The dynamic MOO search takes 7 minutes and 14 seconds. Totally, the second structure takes 2.23 hours which is 35.76% more than the first structure but the second structure gets a better solution.

7.3.3 The Near Optimal Solutions to the Second Objective Function

Second Iteration for the Second Objective Function

The predicted values in table 7.5 are set as simulation parameters. The predicted values between first and second objectives are the same in table 7.5, therefore the simulation results by using the same predicted parameters are also the same, which are shown in table 7.6, and the computational is also the same, which is 0.164 hours for the simulation model run in the Micro Saint Sharp and 29 seconds for the model of truck travelling distance.

L_{yR} denotes a data list represents equation (7.1) for the second objective function. Add the data in table 7.2 to L_{yR}. New simulation data from table 7.6 is then added into L_{yR}. The updated L_{yR} list is the same as L_{qR} in table 7.7. Then the data is fitted by equation (6.16).

Seeing from the features of simulation output, suppose n_i powered functions shown

Table 7.29: Fitting Coefficients for the Sixth Iteration for the First Objective Function

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function $f_{qR}(x)$			Yc Efficiency Objective Function $f_{yR}(x)$	
α_{11}	0.0058958092		α_{21}	−0.0001018484500	
α_{12}	−0.4011728633		α_{22}	0.0079684963000	
α_{13}	9.2216501953		α_{23}	−0.1960322617000	
			α_{24}	1.5506821266500	
			α_{25}	0.1733815184000	
β_{11}	0.0000577562		β_{21}	−0.0000000018100	
β_{12}	−0.0403971509	0.97	β_{22}	0.0000014677300	0.92
β_{13}	9.4969592419		β_{23}	−0.0003821057800	
			β_{24}	0.0334366422300	
			β_{25}	−0.1643596899900	
γ_{11}	0.0000581656		γ_{21}	−0.0000000068000	
γ_{12}	−0.0406226753		γ_{22}	0.0000055352800	
γ_{13}	9.5245750797		γ_{23}	−0.0014542991600	
			γ_{24}	0.1299923902800	
			γ_{25}	−0.8350841096000	
	Traffic Congestion Probability Function $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function $f_d(x)$	
α_{31}	0.0000000421000		α_{41}	0.024647287	
α_{32}	−0.0000024737333		α_{42}	22.85352286	0.99
α_{33}	0.0000378119667		α_{43}	206629.03	
α_{34}	0.2107596043667				
β_{31}	0.0000000000475				
β_{32}	−0.0000000287094	0.70			
β_{33}	0.0000046136812				
β_{34}	0.2107190980027				
γ_{31}	0.0000000000449				
γ_{32}	−0.0000000272623				
γ_{33}	0.0000044563852				
γ_{34}	0.2107116271617				

Table 7.30: Best Predicted Values for the Sixth Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
22.91	6.59	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
208749.66	3	33	85

Table 7.31: Simulation Results for the Seventh Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
25.83	5.74	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
208684.33	3	33	85

Table 7.32: L_{qR} List for the Seventh Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
24.54	4.68	0.6324	
23.16	5.24	0.6324	
27.28	3.25	0.6324	
27.96	5.40	0.6324	
26.03	2.89	0.6324	
25.83	5.74	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207374.80	2	27	29
207460.33	2	23	27
207218.14	1	20	20
207411.73	2	24	36
207122.86	1	24	25
208684.33	3	33	85

Table 7.33: Values of n_i for the Seventh Iteration for the First Objective Function

i	n_{i+1}	n_{i+2}	n_{i+3}
10	3	3	3
20	5	5	5
30	4	4	4
40	3	Nil	Nil

in equations (6.50), (6.51), (6.52) and (6.53) are good mathematical representations for equations (6.15), (6.26), (6.40) and (6.42) respectively for the data list L_{yR}. A few of experiments have been implemented to test values of R^2 to find good values for n_i and the results are shown in table 7.8.

Moreover, the coefficient set $b = \{\alpha, \beta, \gamma\}$ is fitted by the data via equation (6.16). The fitting coefficients for the second objective function are shown in table 7.9.

The near Pareto optimal values found by dynamic MOO search are shown in table 7.10. The best predicted values of the decision variables are 2 quay cranes, 23 yard cranes and 27 trucks in the decision space, which gets the maximum value 4.76. The time taken for the search is 1 minute and 2 seconds.

Third Iteration for the Second Objective Function

The predicted values in table 7.10 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.11. The time taken for the simulation model to run in the Micro Saint Sharp package is 0.168 hours and the running time for the truck travelling distance simulation is 26 seconds.

The updated L_{yR} list is shown in table 7.12. The values of n_i are shown in table 7.13. Moreover, the fitting coefficients are shown in table 7.14.

The near Pareto optimal values found by dynamic MOO search are shown in table 7.15. The best predicted values of the decision variables are 2 quay cranes, 20 yard cranes and 20 trucks and the best predicted simulation value for the second objective is 5.10. The time taken for the search is 1 minute and 2 seconds.

Table 7.34: Fitting Coefficients for the Seventh Iteration for the First Objective Function

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function $f_{qR}(x)$			Yc Efficiency Objective Function $f_{yR}(x)$	
α_{11}	0.0058800634		α_{21}	-0.0000848887000	
α_{12}	-0.4031316478		α_{22}	0.0066541862500	
α_{13}	9.3042539533		α_{23}	-0.1641354814500	
			α_{24}	1.2983097827500	
			α_{25}	0.4598541148500	
β_{11}	0.0000584166		β_{21}	-0.0000000018200	
β_{12}	-0.0408973429	0.94	β_{22}	0.0000014727200	0.86
β_{13}	9.5865584807		β_{23}	-0.0003833886700	
			β_{24}	0.0335495738000	
			β_{25}	-0.1663319891300	
γ_{11}	0.0000396974		γ_{21}	-0.0000000051200	
γ_{12}	-0.0345507283		γ_{22}	0.0000041748000	
γ_{13}	9.6605887925		γ_{23}	-0.0010942161200	
			γ_{24}	0.0966436409200	
			γ_{25}	-0.1861490463600	
	Traffic Congestion Probability Function $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function $f_d(x)$	
α_{31}	0.0000000147900		α_{41}	0.025244655	
α_{32}	-0.0000008684400		α_{42}	22.64393211	0.99
α_{33}	0.0000130125800		α_{43}	206629.00	
α_{34}	0.0632287784200				
β_{31}	0.0000000000161				
β_{32}	-0.0000000096955				
β_{33}	0.0000015271890	0.86			
β_{34}	0.0632169163231				
γ_{31}	0.0000000001137				
γ_{32}	-0.0000000691324				
γ_{33}	0.0000112946503				
γ_{34}	0.5056954987565				

Table 7.35: Best Predicted Values for the Seventh Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
23.56	6.58	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207900.04	4	46	53

Table 7.36: Simulation Results for the Eighth Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
22.30	5.32	0.6324	
Truck Travelling Distance (Metre)	Nqc	Nyc	Ntr
207652.61	4	46	53

Table 7.37: L_{qR} List for the Eighth Iteration for the First Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
24.54	4.68	0.6324	
23.16	5.24	0.6324	
27.28	3.25	0.6324	
27.96	5.40	0.6324	
26.03	2.89	0.6324	
25.83	5.74	0.6328	
22.30	5.32	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207374.80	2	27	29
207460.33	2	23	27
207218.14	1	20	20
207411.73	2	24	36
207122.86	1	24	25
208684.33	3	33	85
207652.61	4	46	53

Fourth Iteration for the Second Objective Function

The predicted values in table 7.15 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.16. The time taken for the simulation model to run in the Micro Saint Sharp package is 0.151 hours and the running time for the truck travelling distance simulation is 23 seconds.

The updated L_{yR} list is shown in table 7.17. The values of n_i are shown in table 7.18. Moreover, the fitting coefficients are shown in table 7.19.

The near Pareto optimal values found by dynamic MOO search are shown in table 7.38. The best predicted values of the decision variables are 2 quay cranes, 74 yard

Table 7.38: Best Predicted Values for the Fourth Iteration for the Second Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
22.29	6.84	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207950.10	2	74	53

cranes and 53 trucks and the best predicted simulation value for the second objective is 6.84. The time taken for the search is 1 minute and 2 seconds.

Fifth Iteration for the Second Objective Function

The predicted values in table 7.38 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.39. The time taken for the simulation model to run

Table 7.39: Simulation Results for the Fifth Iteration for the Second Objective Functions

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
27.85	1.85	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207868.99	2	74	53

in the Micro Saint Sharp package is 0.186 hours and the running time for the truck travelling distance simulation is 45 seconds.

The updated L_{yR} list is shown in table 7.40. The values of n_i are shown in table 7.41. Moreover, the fitting coefficients are shown in table 7.42.

Table 7.40: L_{yR} List for the Fifth Iteration for the Second Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
24.54	4.68	0.6324	
23.16	5.24	0.6324	
27.28	3.25	0.6324	
27.85	1.85	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207374.80	2	27	29
207460.33	2	23	27
207218.14	1	20	20
207868.99	2	74	53

Table 7.41: Values of n_i for the Fifth Iteration for the Second Objective Function

i	n_{i+1}	n_{i+2}	n_{i+3}
10	3	3	3
20	5	5	5
30	4	4	4
40	3	Nil	Nil

The near Pareto optimal values found by dynamic MOO search are shown in table 7.43. The best predicted values of the decision variables are 8 quay cranes, 399 yard cranes and 498 trucks and the best predicted simulation value for the second objective is 8.03. The time taken for the search is 1 minute and 2 seconds.

Table 7.42: Fitting Coefficients for the Fifth Iteration for the Second Objective Function

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function $f_{qR}(x)$			Yc Efficiency Objective Function $f_{yR}(x)$	
α_{11}	0.0059858989		α_{21}	-0.0000005725870	
α_{12}	-0.4054671576		α_{22}	0.0000464451180	
α_{13}	9.2601524036		α_{23}	-0.0012061365430	
			α_{24}	0.0100418533080	
			α_{25}	0.0236205238640	
β_{11}	0.0000361453		β_{21}	0.0000000085500	
β_{12}	-0.0327504965	0.92	β_{22}	-0.0000067933800	0.72
β_{13}	9.4677006598		β_{23}	0.0017237602800	
			β_{24}	-0.1570920409800	
			β_{25}	6.6770517440700	
γ_{11}	0.0000514432		γ_{21}	0.0000000008190	
γ_{12}	-0.0387083831		γ_{22}	-0.0000006460830	
γ_{13}	9.6667675692		γ_{23}	0.0001636588440	
			γ_{24}	-0.0149323203180	
			γ_{25}	0.6566584728240	
	Traffic Congestion Probability Function $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function $f_d(x)$	
α_{31}	0.0000000266200		α_{41}	0.025587339	
α_{32}	-0.0000015441200		α_{42}	22.38535684	0.99
α_{33}	0.0000225964000		α_{43}	206675.85	
α_{34}	0.1264708417200				
β_{31}	0.0000000000594				
β_{32}	-0.0000000359259	0.99			
β_{33}	0.0000057564473				
β_{34}	0.2528609123885				
γ_{31}	0.0000000000653				
γ_{32}	-0.0000000395736				
γ_{33}	0.0000063389853				
γ_{34}	0.2528480724527				

Table 7.43: Best Predicted Values for the Fifth Iteration for the Second Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
11.70	8.03	0.6328	
Truck Travelling Distance(Metre)	*Nqc*	*Nyc*	*Ntr*
224169.52	8	399	498

Sixth Iteration for the Second Objective Function

The predicted values in table 7.43 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.44. The time taken for the simulation model to run

Table 7.44: Simulation Results for the Sixth Iteration for the Second Objective Functions

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
25.51	1.25	0.6324	
Truck Travelling Distance(Metre)	*Nqc*	*Nyc*	*Ntr*
211260.11	2	74	53

in the Micro Saint Sharp package is 0.182 hours and the running time for the truck travelling distance simulation is 5minutes and 22 seconds.

In table 7.40, the third iteration obtains the best value for the second objective which is 5.24. The fourth, fifth and sixth iterations obtain worse objective values from simulation results comparing to the third iteration. Therefore the search stops according to the stopping criteria in section 4.3.2.

The best value for the second objective function searched by the genetic algorithm in the Post-MOO structure is 4.07 moves/yard crane/hour, while the best value obtained from the MOO leading integrated structure is better (5.24). Seeing from the computational volume, it takes 16 simulation iterations and 1 genetic algorithm search in Post-MOO structure, while the integrated MOO structure takes 8 simulation iterations and 5 dynamic MOO searches. On the other hand, the computational time by using the MOO leading integrated structure is: the simulation in Micro Saint

Sharp takes 1.43 hours and simulation for travelling distance takes 18 minutes and 40 seconds. The dynamic MOO search takes 5 minutes and 9 seconds. Therefore the second structure totally takes 1.827 hours which is 11.40% longer than using post-MOO structure.

7.3.4 The Near Optimal Solutions to the Third Objective Function

Second Iteration for the Third Objective Function

The predicted values in table 7.5 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.45. The time taken for the simulation model to run in the

Table 7.45: Simulation Results for the Second Iteration for the Third Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
6.29	1.88	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207720.79	43	377	473

Micro Saint Sharp package is 0.23 hours and the running time for the truck travelling distance simulation is 5 minutes and 48 seconds. The total time for simulation is 0.327 hours.

L_p denotes a data list represents equation (7.1) for the third objective function. Add the data in table 7.2 to L_p. In the second iteration, new simulation data from table 7.45 is then added into L_p. The updated L_p list is shown in table 7.46. Then the data is fitted by equation (6.16).

Seeing from the features of simulation output, suppose n_i powered functions shown in equations (6.50), (6.51), (6.52) and (6.53) are good mathematical representations for equations (6.15), (6.26), (6.40) and (6.42) respectively for the data list L_p. A few of experiments have been implemented to test values of R^2 to find good values for n_i and the results are shown in table 7.47. Moreover, the coefficient set $b = \{\alpha, \beta, \gamma\}$ is fitted by the data via equation (6.16). The fitting coefficients for the third objective function are shown in table 7.48.

Table 7.46: L_p List for the Second Iteration for the Third Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
6.29	1.88	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207720.79	43	377	473

Table 7.47: Values of n_i for the Second Iteration for the Third Objective Function

i	n_{i+1}	n_{i+2}	n_{i+3}
10	3	3	3
20	3	3	3
30	4	4	4
40	3	Nil	Nil

The near Pareto optimal values found by dynamic MOO search are shown in table 7.49. The best predicted values of the decision variables are 27 quay cranes, 128 yard cranes and 183 trucks in the decision space, which gets the best value 0.6324. The time taken for the search is 1 minute and 2 seconds.

Third Iteration for the Third Objective Function

The predicted values in table 7.49 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.50. The time taken for the simulation model to run in the Micro Saint Sharp package is 0.192 hours and the running time for the truck travelling distance simulation is 2 minutes and 1 seconds.

The updated L_p list is shown in table 7.51. The values of n_i are shown in table 7.52. Moreover, the fitting coefficients are shown in table 7.53.

The near Pareto optimal values found by dynamic MOO search are shown in table 7.54. The best predicted values of the decision variables are 31 quay cranes, 121 yard cranes and 129 trucks in the decision space, which gets the maximum value 0.6324.

Table 7.48: Fitting Coefficients for the Second Iteration for the Third Objective Function

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function $f_{qR}(x)$			Yc Efficiency Objective Function $f_{yR}(x)$	
α_{11}	0.0034673828		α_{21}	0.0009580929000	
α_{12}	-0.2830101046		α_{22}	-0.0727496460667	
α_{13}	7.9897940223		α_{23}	1.9838046029000	
β_{11}	0.0000464766		β_{21}	0.0000100661333	
β_{12}	-0.0352279951	0.98	β_{22}	-0.0076724201000	0.99
β_{13}	8.9603252943		β_{23}	2.0771795681333	
γ_{11}	0.0000302725		γ_{21}	0.0000086624667	
γ_{12}	-0.0266596471		γ_{22}	-0.0070151402333	
γ_{13}	8.0181938485		γ_{23}	2.0098587571333	
	Traffic Congestion Probability Function $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function $f_d(x)$	
α_{31}	-0.0000000891000		α_{41}	-0.323591427	
α_{32}	0.0000075567333		α_{42}	197.8813427	0.82
α_{33}	0.0001976854000		α_{43}	187580.33	
α_{34}	0.2124233996333				
β_{31}	0.0000000014000				
β_{32}	-0.0000010845000	0.99			
β_{33}	0.0002595808000				
β_{34}	0.1920048078000				
γ_{31}	-0.0000000000667				
γ_{32}	0.0000000542000				
γ_{33}	0.0000154627333				
γ_{34}	0.2121980240000				

Table 7.49: Best Predicted Values for the Second Iteration for the Third Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
12.24	2.99	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
212955.86	27	128	183

Table 7.50: Simulation Results for the Third Iteration for the Third Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
7.16	3.67	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
205449.77	27	128	183

Table 7.51: L_p List for the Third Iteration for the Third Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
6.29	1.88	0.6324	
7.16	3.67	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207720.79	43	377	473
205449.77	27	128	183

Table 7.52: Values of n_i for the Third Iteration for the Third Objective Function

i	n_{i+1}	n_{i+2}	n_{i+3}
10	3	4	5
20	5	3	3
30	5	5	5
40	4	Nil	Nil

Table 7.53: Fitting Coefficients for the Third Iteration for the Third Objective Function

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function $f_{qR}(x)$			Yc Efficiency Objective Function $f_{yR}(x)$	
α_{11}	0.0045799776		α_{21}	0.000102553	
α_{12}	-0.3439655328		α_{22}	-0.012206661	
α_{13}	8.5998603478		α_{23}	0.517754095	
			α_{24}	-9.146143436	
			α_{25}	56.48890036	
β_{11}	0.0000069646		β_{21}	9.1911×10^{-6}	
β_{12}	-0.0053783680	0.88	β_{22}	-0.007145048	0.95
β_{13}	1.2743025290		β_{23}	2.002499291	
β_{14}	-87.1886581917				
γ_{11}	0.0000000083		γ_{21}	8.48213×10^{-6}	
γ_{12}	-0.0000105348		γ_{22}	-0.007186762	
γ_{13}	0.0047899717		γ_{23}	2.118129978	
γ_{14}	-0.9193690053				
γ_{15}	-0.9193690053				
	Traffic Congestion Probability Function $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function $f_d(x)$	
α_{31}	0.0000000190000		α_{41}	-0.002860588	
α_{32}	-0.0000023497000		α_{42}	2.312374413	0.89
α_{33}	0.0001032426333		α_{43}	-535.15	
α_{34}	-0.0018773085333		α_{44}	$245,972.65$	
α_{35}	0.2225115484333				
β_{31}	0.0000000000054				
β_{32}	-0.0000000048103				
β_{33}	0.0000015056722	0.99			
β_{34}	-0.0001960013849				
β_{35}	0.2200008198302				
γ_{31}	-0.0000000000006				
γ_{32}	0.0000000006723				
γ_{33}	-0.0000002709088				
γ_{34}	0.0000445875317				
γ_{35}	0.2083816152915				

Table 7.54: Best Predicted Values for the Third Iteration for the Third Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
9.22	4.19	0.6324	
Truck Travelling Distance(Metre)	*Nqc*	*Nyc*	*Ntr*
209277.78	31	121	129

The time taken for the search is 1 minute and 2 seconds.

Fourth Iteration for the Third Objective Function

The predicted values in table 7.54 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.55. The time taken for the simulation model to run

Table 7.55: Simulation Results for the Fourth Iteration for the Third Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
6.08	3.73	0.6328	
Truck Travelling Distance(Metre)	*Nqc*	*Nyc*	*Ntr*
209608.29	31	121	129

in the Micro Saint Sharp package is 0.189 hours and the running time for the truck travelling distance simulation is 1 minute and 29 seconds.

The best value in L_p list is 0.6324 which is found in the first iteration for the third objective. The second, third and fourth iterations get objective values equal to or worse than the existed best value. Therefore the iteration is stopped.

The best value for the third objective function searched by the Post-MOO structure is 0.6324, and the best value obtained from the MOO leading integrated structure is also 0.6324. Seeing from the computational volume, the integrated MOO structure takes 6 simulation iterations and 3 genetic algorithm searches. The computational time is: the simulation takes 1.26 hours and the dynamic MOO search takes 3 minutes and 6 seconds. Totally, it takes 1.34 hours which is 18.29% less than post-MOO structure.

7.3.5 The Near Optimal Solutions to the Fourth Objective Function

Second Iteration for the Fourth Objective Function

The predicted values in table 7.5 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.56. The time taken for the simulation model to run

Table 7.56: Simulation Results for the Second Iteration for the Fourth Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
4.44	2.27	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207374.80	12	56	26

in the Micro Saint Sharp package is 0.156 hours and the running time for the truck travelling distance simulation is 26 seconds.

L_d denotes a data list represents equation (7.1) for the fourth objective function. Add the data in table 7.2 to L_d. In the second iteration, the new simulation data from table 7.56 is then added into L_d. The updated L_d list is shown in table 7.57. Then the data is fitted by equation (6.16).

Table 7.57: L_d List for the Second Iteration for the Fourth Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
4.44	2.27	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207374.80	12	56	26

Seeing from the features of simulation output, suppose n_i powered functions shown in equations (6.50), (6.51), (6.52) and (6.53) are good mathematical representations for equations (6.15), (6.26), (6.40) and (6.42) respectively for the data list L_d. A few of experiments have been implemented to test values of R^2 to find good values for n_i and the results are shown in table 7.58. Moreover, the coefficient set $b = \{\alpha, \beta, \gamma\}$ is

Table 7.58: Values of n_i for the Second Iteration for the Fourth Objective Function

i	n_{i+1}	n_{i+2}	n_{i+3}
10	4	4	4
20	4	4	4
30	4	4	4
40	3	Nil	Nil

fitted by the data via equation (6.16). The fitting coefficients for the fourth objective function are shown in table 7.59.

The near Pareto optimal values found by dynamic MOO search are shown in table 7.60. The best predicted values of the decision variables are 23 quay cranes, 28 yard cranes and 22 trucks in the decision space, which gets the maximum value 206975.97. The time taken for the search is 1 minute and 2 seconds.

Third Iteration for the Fourth Objective Function

The predicted values in table 7.60 are set as simulation parameters, which are input into the "Data Processing". Then simulation models run with the parameters and the results are shown in table 7.61. The time taken for the simulation model to run in the Micro Saint Sharp package is 0.157 hours and the running time for the truck travelling distance simulation is 26 seconds.

The updated L_d list is shown in table 7.62. The values of n_i are shown in table 7.63. Moreover, the fitting coefficients are shown in table 7.64.

The near Pareto optimal values found by dynamic MOO search are shown in table 7.65. The best predicted values of the decision variables are 23 quay cranes, 22 yard cranes and 22 trucks in the decision space, which gets the maximum value 207287.20. The time taken for the search is 1 minute and 2 seconds.

The iterations are stopped because the searching value is 22 (trucks) which is the same as second iteration. The fourth objective function only has one decision variable, therefore iteration stops due to the predicted value of the decision variable exists in the L_p list.

Table 7.59: Fitting Coefficients for the Second Iteration for the Fourth Objective Function

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function $f_{qR}(x)$			Yc Efficiency Objective Function $f_{yR}(x)$	
α_{11}	0.0109183839		α_{21}	0.0014561901000	
α_{12}	−0.8246764495		α_{22}	−0.1096925779000	
α_{13}	18.9546254039		α_{23}	2.5023297342000	
α_{14}	−126.0904454670		α_{24}	−15.9910303319333	
β_{11}	0.0000010338		β_{21}	0.0000001651667	
β_{12}	−0.0007543134	0.99	β_{22}	−0.0001190656333	0.99
β_{13}	0.1574081265		β_{23}	0.0235600498667	
β_{14}	−5.1522604100		β_{24}	−0.2174877359667	
γ_{11}	0.0000007129		γ_{21}	0.0000001185667	
γ_{12}	−0.0005040397		γ_{22}	−0.0000827221333	
γ_{13}	0.0969574142		γ_{23}	0.0147816954333	
γ_{13}	−0.7140743576		γ_{24}	0.4270037637000	
	Traffic Congestion Probability Function $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function $f_d(x)$	
α_{31}	0.0000000604667		α_{41}	0.021265935	
α_{32}	−0.0000038094000		α_{42}	24.64797932	0.99
α_{33}	0.0000668761667		α_{43}	206423.42	
α_{34}	0.2105764037667				
β_{31}	0.0000000000333				
β_{32}	−0.0000000240333	0.63			
β_{33}	0.0000034363333				
β_{34}	0.2108104876333				
γ_{31}	0.0000000000333				
γ_{32}	−0.0000000198667				
γ_{33}	0.0000024280667				
γ_{34}	0.2108845131000				

Table 7.60: Best Predicted Values for the Second Iteration for the Fourth Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
6.33	2.32	0.6324	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
206975.97	23	28	22

Table 7.61: Simulation Results for the Third Iteration for the Fourth Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
1.77	3.80	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207592.48	23	28	22

Table 7.62: L_d List for the Third Iteration for the Fourth Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
14.81	3.60	0.6328	
8.86	2.27	0.6324	
7.40	1.82	0.6328	
4.44	2.27	0.6328	
1.77	3.80	0.6328	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
209670.84	13	140	140
214873.69	25	260	260
218660.76	38	380	380
207374.80	12	56	26
207592.48	23	28	22

Table 7.63: Values of n_i for the Third Iteration for the Fourth Objective Function

i	n_{i+1}	n_{i+2}	n_{i+3}
10	5	4	4
20	4	5	5
30	5	4	5
40	3	Nil	Nil

Table 7.64: Fitting Coefficients for the Third Iteration for the Fourth Objective Function

Coefficients	Fitting Values	R^2	Coefficients	Fitting Values	R^2
	Qc Efficiency Objective Function $f_{qR}(x)$			Yc Efficiency Objective Function $f_{yR}(x)$	
α_{11}	-0.0017775645		α_{21}	0.0014802019667	
α_{12}	0.1673440564		α_{22}	-0.1115241214000	
α_{13}	-5.5903267686		α_{23}	2.5449617971333	
α_{14}	78.6416848651		α_{24}	-16.2839847119333	
α_{15}	-389.5254985437		β_{21}	0.0000000035667	
β_{11}	0.0000007040		β_{22}	-0.0000028220667	
β_{12}	-0.0005217479	0.95	β_{23}	0.0007102109333	0.76
β_{13}	0.1095516396		β_{24}	-0.0635640152667	
β_{14}	-2.4841220402		β_{25}	2.5503121474333	
γ_{11}	0.0000007572		γ_{21}	0.0000000137667	
γ_{12}	-0.0005364904		γ_{22}	-0.0000109874000	
γ_{13}	0.1041147844		γ_{23}	0.0027927069333	
γ_{14}	-1.1728361123		γ_{24}	-0.2433069418667	
			γ_{25}	5.3824202421667	
	Traffic Congestion Probability Function $P\{X_i < f_{ts}\}$			Truck Travelling Distance Function $f_d(x)$	
α_{31}	0.0000000302333		α_{41}	0.027380187	
α_{32}	-0.0000026000667		α_{42}	21.37779006	0.99
α_{33}	0.0000772462333		α_{43}	206803.64	
α_{34}	-0.0009482993667				
α_{35}	0.2150569867000				
β_{31}	0.0000000000333				
β_{32}	-0.0000000184000				
β_{33}	0.0000022779000	0.70			
β_{34}	0.2108750731333				
γ_{31}	0.0000000000002				
γ_{32}	-0.0000000000869				
γ_{33}	0.0000000120387				
γ_{34}	-0.0000004359443				
γ_{35}	0.2109395034177				

Table 7.65: Best Predicted Values for the Third Iteration for the Fourth Objective Function

Quay Crane Productivity (Move/quay crane/hour)	Yard Crane Productivity (Move/quay crane/hour)	Traffic Congestion Probability	
1.14	4.00	0.632436671784682	
Truck Travelling Distance(Metre)	Nqc	Nyc	Ntr
207287.20	23	22	22

The best value for the fourth objective function searched by the post-MOO structure is 210778.36 metres, while the best value obtained from the MOO leading integrated structure is better (207294.24). Seeing from the computational volume, the integrated MOO structure takes 5 simulation iterations and 3 genetic algorithm searches. The computational time is: the simulation takes 0.96 hours and the dynamic MOO search takes 0.22 hours. Hence, it takes 1.18 hours by using integrated MOO structure which is 28.05% less than using post-MOO structure.

7.4 Summary

This chapter mainly addresses the processes to use the MOO leading integrated structure to solve multiple equipment quantity optimisation problem raised and modelled in chapter 6. It aims to traverse good quality values of the decision variables to find good quality solutions with low computational cost. The results from the MOO leading integrated structure are better than post-MOO structure in the first, second, and fourth objective function, while the best results of the third objective are the same. The solutions to the first, second and fourth objectives obtained by the MOO leading integrated structure have 4.17%, 28.75% and 1.65% improvement comparing to the post-MOO structure. The computational volume by using the MOO leading integrated structure of is less than post-MOO structure for all objectives. Furthermore, the computational time of the first and second objectives by the MOO leading integrated structure are 35.76% and 11.40% longer than post-MOO structure. But for the third and fourth objectives, using the MOO leading integrated structure reduces computational time by 18.29% and 28.05% respectively comparing to the post-MOO structure.

Chapter 8

Summary and Conclusions

This thesis addresses a new way to solve the problem of mechanical equipment quantity optimisation for container terminals. Background information about the container industry is discussed and operational flows of container terminals are analysed. Simulation is able to well describe dynamic systems such as container terminal systems and provide a great deal of detailed data. However simulation for large systems needs a lot of computational volume and may be time consuming. Another method employed, multi-objective optimisation, solves the problems with multiple objectives and provides near Pareto optimal solutions. In order to integrate the strengths of the two methods, a combination framework to integrate simulation and multi-objective optimisation is therefore proposed to solve the problem. The combination framework to solve a container terminal optimisation problem with a series of management objectives is potentially powerful and time saving. Three combination structures are presented under the combination framework for different types of problems.

The applications of the three combination structures on three problems are discussed in this thesis: the internal truck quantity optimisation problem by using post-MOO structure is discussed in chapter 5; the multiple equipment quantity optimisation problem in container terminals by using post-MOO structure is given in chapter 6; the multiple equipment quantity optimisation problem in container terminals by using MOO leading integrated structure is given in chapter 7.

Chapters 5 discusses a single variable problem, namely the internal truck quantity optimisation problem in container terminals, under the combination framework. Five objectives related to cost and operational efficiency are taken into consideration. Simulation and multi-objective optimisation models are developed under the post-MOO combination structure. Explicit enumeration is employed to produce the Pareto optimal solutions.

Chapters 6 further studies a multiple variable problem, namely the multiple equipment quantity optimisation problem, under the post-MOO structure. Two simulation models are built to represent container terminal operations. One of them is developed in the Micro Saint simulation package for general container terminal operations and another is coded separately in C# for internal truck travelling distance to reduce computational time. Multi-objective optimisation models are proposed for four objectives related to terminal productivity and cost. The genetic algorithm is employed to explore the near Pareto optimal solutions due to the complexity of the problem.

As a further study based on the multiple equipment quantity optimisation problem raised, chapter 7 solves the problem by the MOO leading integrated structure in order to get good solutions with low computational cost. The "Data Processing" and "Searching Techniques" in the combination structure are defined as the data fitting and dynamic MOO search respectively to process and deliver the data streams between simulation and multi-objective optimisation in the framework. Solutions are explored until stopping conditions are met.

The results show that post-MOO and MOO leading integrated structures are suitable to solve dynamic problems with multiple objectives like container terminal optimisation problems. Post-MOO is efficient when the computational time of simulation is low because a number of simulation iterations are needed to generate a great number of data to increase the accuracy level of models. On the other hand, if the simulation models are time consuming, MOO leading integrated structure is more effective in exploring good solutions and reducing computational time. Furthermore, MOO leading integrated structure only searches the best solutions for each objective however the post-MOO structure provides much more data for further analysis.

There are two important points to the two combination structures used in this thesis. First of all, as the data fitting is defined as a method for "Data Processing" in the three combination structures in this thesis, the accuracy level achieved by the data fitting determines the searching efficiency of "Searching Techniques" in the integrated MOO structure and the quality of solutions obtained from the post-MOO structure. In the MOO leading integrated structure, the data fitting and dynamic MOO search explore the potential optimal decision values to find the near Pareto optimal solutions. The dynamic MOO search is based on the parameters from the data fitting, therefore, the quality of solutions found by the dynamic MOO search relies on the accuracy level of parameters delivered from the data fitting. Data fitting results with poor R^2 values do not provide good decision values, which may result in poor performance of the framework and poor quality of solutions obtained from the combination. For the post-MOO structure, problems are directly solved based on the fitting functions. Poor quality of fitting functions may lead to great deviations. Secondly, in the MOO leading integrated structure, the dynamic MOO search needs to consider the situations where the search can not escape from a local optimum. In order to reduce the number of initial values of decision variables, the process starts with a minimum number of initial values which are set evenly on the decision axes. The fitting functions may not be a good representation for all solutions, therefore, it is possible that the dynamic MOO search could not predict better decision values to escape from a local optimum.

The models proposed in this thesis need to be improved. Firstly, the range of the congestion probability objective function built in section 6.3.3 is very narrow, therefore the model could be improved to get a wider range of values which is closer to reality. Secondly, the simulation models built in the Micro Saint Sharp package are computational time consuming. The time taken in running simulation models is possible to be shorted if the models are implemented in C#. Thirdly, parameter settings and estimation. The setting of parameters can be improved if more data is obtained. Fourthly, feedback from the Southampton Container Terminal on the objectives, modelling, parameter settings and solutions is valuable for further improvement.

My research interests in the future are still in this area. Firstly, the MOO leading integrated structure and post-MOO structure are employed to solve the problems in this thesis, however, the pre-MOO structure and simulation leading integrated structure have not been defined and applied in this thesis. Therefore, the exploration of applications for the two combination structures may be a new research area. Secondly, techniques for the “Data Processing” and “Searching Techniques” in the combination framework could be substituted by other methods if they are proved to have better performance for the problems. The combination framework is only a framework and its internal structures and methods need to be defined. Therefore, more efficient and effective methods for the "Data Processing" and "Searching Techniques" may be found in the future research. Finally, as discussed in this thesis, the combination framework is able to solve container terminal optimisation problems, its applicability to other problems is still unknown. More research is needed to work out the performance of the framework in other areas.

Appendix A

C# Codes for the Explicit Numeration for Truck Quantity Optimisation

The programme is implemented by "Windows Forms Application" in Visual C# 2010 Express. There are three source code documents: Tr_Program.cs, Tr_Form1.Designer.cs and Tr_Form1.cs. The source codes of "Tr_Program.cs" are shown as follows:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(
                false);
            Application.Run(new Form1());
        }
    }
```

```
}
```

The source codes of “Tr_Program.cs” are shown as follows:

```
namespace WindowsFormsApplication1
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components =
            null;
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources
            should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code
        /// <summary>
        /// Required method for Designer support - do not
           modify
```

```
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.btnSolutions = new System.Windows.Forms.
        Button();
    this.SuspendLayout();

    // btnSolutions
    this.btnSolutions.Location = new System.Drawing.
        Point(214, 98);
    this.btnSolutions.Name = "btnSolutions";
    this.btnSolutions.Size = new System.Drawing.Size
        (139, 31);
    this.btnSolutions.TabIndex = 0;
    this.btnSolutions.Text = "Get Solutions";
    this.btnSolutions.UseVisualStyleBackColor = true;
    this.btnSolutions.Click += new System.
        EventHandler(this.button1_Click);

    // Form1
    this.AutoScaleDimensions = new System.Drawing.
        SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.
        AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(571,
        262);
    this.Controls.Add(this.btnSolutions);
    this.Name = "Form1";
    this.Text = "Multiple Equipment Quantities 
        Optimisation";
```

```
            this.Load += new System.EventHandler(this.
                Form1_Load);
            this.ResumeLayout(false);
        }
        #endregion

        private System.Windows.Forms.Button btnSolutions;
    }
}
```

The source codes of "Tr_Form1.cs" are shown as follows:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Collections; //For ArrayList()
using System.IO;//Create txt files
using System.Diagnostics;//Stopwatch to record programme
    running time

namespace WindowsFormsApplication1
{
    public struct structPareto
    {
        public int Ntr;
        public double[] solutionValue;
```

```
}

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
    }

    private void button1_Click(object sender, EventArgs e
       )
    {
        //Record the programme running time
        Stopwatch programRunningTime = new Stopwatch();
        programRunningTime.Start();

        int obFunNum = 5;//Number of objective functions
        ArrayList arrLi_AllSolutions = new ArrayList();//
           A list of solutions and Ntr.

        //Initialisation
        int upBoundTr = 500;
        int lowBoundTr = 20;

        string path = Environment.GetFolderPath(
           Environment.SpecialFolder.DesktopDirectory) +
           "\\MOO_Solution_Tr.txt";
```

```
StreamWriter txt;
txt = File.CreateText(path);
txt.WriteLine("————————————————All Posible
   Values——————————————————————");

//Find all solutions
for (int int_Temp_i = lowBoundTr; int_Temp_i <
   upBoundTr; int_Temp_i++)
//Calculate all solutions
{
    structPareto stru_Temp = new structPareto();
    stru_Temp.solutionValue = new double[obFunNum
       ];

    //comparison temperary values{fun1,fun2,fun3,
       fun4}
    stru_Temp.Ntr = int_Temp_i;
    stru_Temp.solutionValue[0] = obj_Fun1(
       int_Temp_i);
    stru_Temp.solutionValue[1] = obj_Fun2(
       int_Temp_i);
    stru_Temp.solutionValue[2] = obj_Fun3(
       int_Temp_i);
    stru_Temp.solutionValue[3] = obj_Fun4(
       int_Temp_i);
    stru_Temp.solutionValue[4] = obj_Fun5(
       int_Temp_i);

    arrLi_AllSolutions.Add(stru_Temp);//Important
```

```
        stru_Temp = (structPareto)arrLi_AllSolutions[
            arrLi_AllSolutions.Count - 1];

        txt.WriteLine(Convert.ToString(
            arrLi_AllSolutions.Count));
        writeTxt(txt, stru_Temp);
    }//Finding all solutions ends

    //Explore pareto frontier
    for (int int_Temp_i = 0; int_Temp_i <
        arrLi_AllSolutions.Count; int_Temp_i++)
    //Current Fun Value
    {
        structPareto stru_Temp1 = new structPareto();
        stru_Temp1 = (structPareto)arrLi_AllSolutions
            [int_Temp_i];//Current Fun Value

        for (int int_Temp_j = 0; int_Temp_j <
            arrLi_AllSolutions.Count; int_Temp_j++)
        //Compare with all Values
        {
            structPareto stru_Temp2 = new
                structPareto();
            stru_Temp2 = (structPareto)
                arrLi_AllSolutions[int_Temp_j];//Other
                 values

            if (paretoComparation(stru_Temp1,
                stru_Temp2))//if stru_Temp1 is worse
                than stru_Temp2, then delete
                stru_Temp1
```

```
            { }//not worse or comparing to the same
               values
            else//worse
            {
                arrLi_AllSolutions.RemoveAt(
                   int_Temp_i);//remove stru_Temp1
                int_Temp_i = int_Temp_i - 1;//if the
                   first element 0, int_Temp_i=-1,
                   int_Temp_i++, int_Temp_i=0
                break;
            }
        }
    }
    //Explorign Pareto Frontier ends

    for (int int_Temp_i = 0; int_Temp_i < 20;
       int_Temp_i++)
    { txt.WriteLine("\n"); } //return

    txt.WriteLine("-------------------------Solutions
       ------------------------------------");
    foreach (structPareto str_Temp in
       arrLi_AllSolutions)
    {
        writeTxt(txt, str_Temp);
    }
    txt.WriteLine(Convert.ToString(arrLi_AllSolutions
       .Count));

    programRunningTime.Stop();
```

```
    txt.WriteLine("--------------------------Programme 
       Running Time------------------------------------");
    txt.WriteLine("Total Running Time" +
       programRunningTime.Elapsed);

    txt.Close();

    MessageBox.Show("Solutions have been exported to 
       " + path
        + "\nProgramme Running Time" +
           programRunningTime.Elapsed);
}
//End of function Main

private static void PrintValues(ArrayList
   allSolutions)
{
    throw new NotImplementedException();
}

static double obj_Fun1(int Ntr)
//objective function1: maximise QC Operation Rate
//Unit: moves per hour
{
    double fx, b0, b1, b2, b3, b4;
    b0 = -1.00608607389221 * Math.Pow(10, -8);
    b1 = 1.00172577039788 * Math.Pow(10, -5);
    b2 = -0.00350495307286781;
    b3 = 0.504609782642747;
    b4 = -1.62424413488179;
```

```
    fx = dataFittingFun(b0, b1, b2, b3, b4, Ntr);
    return fx;
}

static double obj_Fun2(int Ntr)
//objective function2: maximise Truck Utilisation
   Rate
{
    double fx, b0, b1, b2, b3, b4;
    b0 = 0;
    b1 = -1.85164922615276 * Math.Pow(10, -8);
    b2 = 2.11313871749685 * Math.Pow(10, -5);
    b3 = -0.00836945848863550;
    b4 = 1.34277065923323;

    fx = dataFittingFun(b0, b1, b2, b3, b4, Ntr);
    return fx;
}

static double obj_Fun3(int Ntr)
//objective function3: mnimise Congestion Probability
{
    double fx, b0, b1, b2, b3, b4;
    b0 = 1.43611966786207 * Math.Pow(10, -11);
    b1 = -1.40544943061360 * Math.Pow(10, -8);
    b2 = 4.80442768722124 * Math.Pow(10, -6);
    b3 = -0.000670752843376622;
    b4 = 0.199458851793086;

    fx = dataFittingFun(b0, b1, b2, b3, b4, Ntr);
    return fx;
```

```
}

static double obj_Fun4(int Ntr)
//objective function4: minimise Fuel Consumption
{
    double fx, b0, b1, b2, b3, b4;
    b0 = 7.34184693898015 * Math.Pow(10, -10);
    b1 = -7.06754055408318 * Math.Pow(10, -7);
    b2 = 0.000239076661804849;
    b3 = -0.0330186354664595;
    b4 = 9.38680082103357;

    fx = dataFittingFun(b0, b1, b2, b3, b4, Ntr);
    return fx;
}

static double obj_Fun5(int Ntr)
//objective function5: minimise Labour Consumption
{
    double fx, b0, b1, b2, b3, b4;
    b0 = 7.63548962659333 * Math.Pow(10, -10);
    b1 = -7.37173666689088 * Math.Pow(10, -7);
    b2 = 0.000250874980866436;
    b3 = -0.0348859425028811;
    b4 = 10.0628641598034;

    fx = dataFittingFun(b0, b1, b2, b3, b4, Ntr);
    return fx;
}
```

```
static double dataFittingFun(double b0, double b1,
   double b2, double b3,
    double b4, int x)
{
    double y;
    y = b0 * Math.Pow(x, 4) + b1 * Math.Pow(x, 3) +
        b2 * Math.Pow(x, 2) + b3 * x + b4;

    return y;
}

static void writeTxt(StreamWriter f_txt, structPareto
    f_stru_Temp)
{
    f_txt.WriteLine("Ntr:" + f_stru_Temp.Ntr + ".Fun1
       ,Fun2,Fun3,Fun4, Fun5:"
        + f_stru_Temp.solutionValue[0] + " " +
           f_stru_Temp.solutionValue[1] + " " +
        f_stru_Temp.solutionValue[2] + " " +
           f_stru_Temp.solutionValue[3] + " " +
        f_stru_Temp.solutionValue[4]);
}

static bool paretoComparation(structPareto
   f_stru_Temp1, structPareto f_stru_Temp2)
{
    bool f_tag = true;

    if (f_stru_Temp1.solutionValue[0] != f_stru_Temp2
       .solutionValue[0] || //Max 1st Obj QC
```

```
    f_stru_Temp1.solutionValue[1] != f_stru_Temp2
        .solutionValue[1] ||//Max 2nd Obj
        Utilisation
    f_stru_Temp1.solutionValue[2] != f_stru_Temp2
        .solutionValue[2] ||//Min 3rd Obj
        Congestion
    f_stru_Temp1.solutionValue[3] != f_stru_Temp2
        .solutionValue[3] ||//Min 4th Obj Fuel
        cost
    f_stru_Temp1.solutionValue[4] != f_stru_Temp2
        .solutionValue[4])
//This is not itself.
{
    if (f_stru_Temp1.solutionValue[0] >
        f_stru_Temp2.solutionValue[0] ||//Max 1st
        Obj QC
         f_stru_Temp1.solutionValue[1] >
            f_stru_Temp2.solutionValue[1] ||//Max
            2nd Obj Utilisation
         f_stru_Temp1.solutionValue[2] <
            f_stru_Temp2.solutionValue[2] ||//Min
            3rd Obj Congestion
         f_stru_Temp1.solutionValue[3] <
            f_stru_Temp2.solutionValue[3] ||//Min
            4th Obj Fuel cost
         f_stru_Temp1.solutionValue[4] <
            f_stru_Temp2.solutionValue[4])  //Min
            5th Obj Labour cost
    { }//f_stru_Temp1 is not worse than
        f_stru_Temp2;
    else
```

```
                {
                    f_tag = false;//f_stru_Temp1 is worse
                        than f_stru_Temp2; delete.
                }
            }
            else { } //f_stru_Temp1 = f_stru_Temp2;
            return f_tag;
        }
    }
}
```

Appendix B

C# Codes for the Simulation for Truck Travelling Distance

This programme is a "Windows Forms Application", which includes three source code documents: SimTrTravelling_Program.cs, SimTrTravelling_Form1.Designer.cs and SimTrTravelling_Form1.cs. The source codes of "SimTrTravelling_Program.cs" are shown as follows:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(
                false);
            Application.Run(new Form1());
        }
```

```
    }
}
```

The source codes of “SimTrTravelling_Program.cs” are shown as follows:

```
namespace WindowsFormsApplication1
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components =
            null;
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources
            should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code
        /// <summary>
```

```
/// Required method for Designer support - do not
   modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.btnSolutions = new System.Windows.Forms.
       Button();
    this.labTr_Txt = new System.Windows.Forms.Label()
       ;
    this.labTr_Value = new System.Windows.Forms.Label
       ();
    this.labFun4_Value = new System.Windows.Forms.
       Label();
    this.labFun4_Txt = new System.Windows.Forms.Label
       ();
    this.SuspendLayout();

    // btnSolutions
    this.btnSolutions.Location = new System.Drawing.
       Point(420, 219);
    this.btnSolutions.Name = "btnSolutions";
    this.btnSolutions.Size = new System.Drawing.Size
       (139, 31);
    this.btnSolutions.TabIndex = 0;
    this.btnSolutions.Text = "Get Solutions";
    this.btnSolutions.UseVisualStyleBackColor = true;
    this.btnSolutions.Click += new System.
       EventHandler(this.button1_Click);
```

```
// labTr_Txt
this.labTr_Txt.AutoSize = true;
this.labTr_Txt.Location = new System.Drawing.
   Point(33, 43);
this.labTr_Txt.Name = "labTr_Txt";
this.labTr_Txt.Size = new System.Drawing.Size(92,
    13);
this.labTr_Txt.TabIndex = 1;
this.labTr_Txt.Text = "Number␣fo␣Trucks";

// labTr_Value
this.labTr_Value.AutoSize = true;
this.labTr_Value.Location = new System.Drawing.
   Point(170, 43);
this.labTr_Value.Name = "labTr_Value";
this.labTr_Value.Size = new System.Drawing.Size
   (39, 13);
this.labTr_Value.TabIndex = 3;
this.labTr_Value.Text = "Values";

// labFun4_Value
this.labFun4_Value.AutoSize = true;
this.labFun4_Value.Location = new System.Drawing.
   Point(170, 89);
this.labFun4_Value.Name = "labFun4_Value";
this.labFun4_Value.Size = new System.Drawing.Size
   (39, 13);
this.labFun4_Value.TabIndex = 15;
```

```
this.labFun4_Value.Text = "Values";

// labFun4_Txt
this.labFun4_Txt.AutoSize = true;
this.labFun4_Txt.Location = new System.Drawing.
    Point(33, 89);
this.labFun4_Txt.Name = "labFun4_Txt";
this.labFun4_Txt.Size = new System.Drawing.Size
    (114, 13);
this.labFun4_Txt.TabIndex = 14;
this.labFun4_Txt.Text = "4th Objective Function";

// Form1
this.AutoScaleDimensions = new System.Drawing.
    SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.
    AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(571,
    262);
this.Controls.Add(this.labFun4_Value);
this.Controls.Add(this.labFun4_Txt);
this.Controls.Add(this.labTr_Value);
this.Controls.Add(this.labTr_Txt);
this.Controls.Add(this.btnSolutions);
this.Name = "Form1";
this.Text = "Multiple Equipment Quantities 
    Optimisation";
this.Load += new System.EventHandler(this.
    Form1_Load);
```

```
            this.ResumeLayout(false);
            this.PerformLayout();
        }
        #endregion

        private System.Windows.Forms.Button btnSolutions;
        private System.Windows.Forms.Label labTr_Txt;
        private System.Windows.Forms.Label labTr_Value;
        private System.Windows.Forms.Label labFun4_Value;
        private System.Windows.Forms.Label labFun4_Txt;
    }
}
```

The source codes of "SimTrTravelling_ Form1.cs" are shown as follows:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;//Create txt files
using System.Diagnostics;//Stopwatch to record programme
   running time

namespace WindowsFormsApplication1
{
    public struct structPareto
```

```
{
    public int Ntr, Nyc, Nqc;
    public double[] solutionValue;
}

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
    }

    private void button1_Click(object sender, EventArgs e
       )
    {
        //Record the programme running time
        Stopwatch programRunningTime = new Stopwatch();
        programRunningTime.Start();

        //Yard Range {x1,x2,y1,y2}:{0,1800,0,1800}
        double[,] cntrBlckCor = new double[,]//
           Coordinates of container blocks {x1,x2,y1,y2}
        {
            {300,500,300,500}, {800,1000,300,500},
               {1300,1500,300,500},
```

```
    {300,500,800,1000}, {800,1000,800,1000},
       {1300,1500,800,1000},
    {300,500,1300,1500},{800,1000,1300,1500},{1300,1500,1300

};

string path = Environment.GetFolderPath(
   Environment.SpecialFolder.DesktopDirectory)
    + "\\Simulation_Fourth_Objective_Function.txt
       ";
StreamWriter txt;
txt = File.CreateText(path);
txt.WriteLine("————————————————Values_for_
   the_Fourth_Objective_Function
   ——————————————————————");

int[] Ntr = new int[9] { 20, 116, 212, 308, 404,
   500, 800, 1000, 1200 };
int numTrvDisIterations = 100000;
txt.WriteLine("Number_of_Iterations:{0}",
   numTrvDisIterations);

foreach (int int_Temp in Ntr)
{
    double dou_Temp = obj_Fun4(int_Temp,
       numTrvDisIterations);
    txt.WriteLine("Number_of_Trucks:_{0};_Value_
       of_the_Fourth_Objective_Function:_{1}.",
        int_Temp, dou_Temp);
```

```
            labTr_Value.Text = Convert.ToString(int_Temp)
                ;
            labFun4_Value.Text = Convert.ToString(
                dou_Temp);
            Refresh();
        }

        programRunningTime.Stop();

        txt.WriteLine("------------------------------Programme␣
            Running␣Time------------------------------------------");
        txt.WriteLine("Total␣Running␣Time:" +
            programRunningTime.Elapsed);

        txt.Close();
        MessageBox.Show("Programme␣Ends.");
    }
    //main function ends

    static double obj_Fun4(int f_Ntr, int
        numTrvDisIterations)
    //objective function4: minimise travelling distance.
    //Unit: metre
    {
        double[][] trPositions = new double[f_Ntr][];
        for (int int_Temp_i = 0; int_Temp_i < trPositions
            .Length; int_Temp_i++)
        {
```

```
        trPositions[int_Temp_i] = new double[2];
        trPositions[int_Temp_i] = ranCoordinates(
            false);//trPositions[int_Temp_i][2]
    }

    double avgTrvDis = 0;

    for (int int_Temp_i = 0; int_Temp_i <
        numTrvDisIterations; int_Temp_i++)
    {
        //Generate new order position(Loading
            containers)
        double[] xyOrd_Load = new double[2];//
            cntrBlckCor[3]={xc,yc,blockNumber}
        xyOrd_Load = ranCoordinates(true);//2
            dimensional array, position of an order

        //Search the closest truck
        int trDisIndex = 0;
        double disOut = seekingShortestTr(xyOrd_Load,
             trPositions, ref trDisIndex);//Shortest
            truck
        trPositions[trDisIndex] = xyOrd_Load;//The
            truck [trDisIndex] goes to xyOrd;

        //Generate the next order position(
            Discharging containers)
        double[] xyOrd_Disch = new double[2];//
            cntrBlckCor[3]={xc,yc,blockNumber}
```

```
            xyOrd_Disch = ranCoordinates(true);
            double disBack = seekingShortestTr(
               xyOrd_Disch, trPositions, ref trDisIndex);
               //Shortest truck
            trPositions[trDisIndex] = xyOrd_Disch;//The
               truck [trDisIndex] goes to xycd;

            avgTrvDis = disOut + disBack + avgTrvDis;
        }
        return avgTrvDis / numTrvDisIterations;

    }
    //Objective functions end

    static double[] ranCoordinates(bool
       IsCoordinateInCntrBlocks)
    //IsCoordinateInCntrBlocks:Orders or truck positions
    {
        double[] xyc = new double[2];
        Random ran = new Random();
        int xCntrBlockLowerBound, xCntrBlockUpperBound,
           xCntrBlockDisInterval,
        yCntrBlockLowerBound, yCntrBlockUpperBound,
           yCntrBlockDisInterval;

        xCntrBlockDisInterval = 500;//Internal is the
           same
        yCntrBlockDisInterval = 500;//Internal is the
           same
```

```
//Generate an order coordinate in container
   blocks
if (IsCoordinateInCntrBlocks)
{
    xCntrBlockLowerBound = 300;//According to
       cntrBlckCor
    xCntrBlockUpperBound = 500;//According to
       cntrBlckCor

    yCntrBlockLowerBound = 300;//According to
       cntrBlckCor
    yCntrBlockUpperBound = 500;//According to
       cntrBlckCor
}
else//Generate an truck position outside
   container blocks
{
    xCntrBlockLowerBound = 0;//According to
       cntrBlckCor
    xCntrBlockUpperBound = 300;//According to
       cntrBlckCor

    yCntrBlockLowerBound = 0;//According to
       cntrBlckCor
    yCntrBlockUpperBound = 300;//According to
       cntrBlckCor
}
```

```
        int int_Temp_x = ran.Next(0, 3);//0-3,Max value
           exclusive
        xyc[0] = ran.Next(xCntrBlockLowerBound +
           xCntrBlockDisInterval * int_Temp_x,
                xCntrBlockUpperBound +
                   xCntrBlockDisInterval * int_Temp_x);
        int int_Temp_y = ran.Next(0, 3);//0-3,Max value
           exclusive
        xyc[1] = ran.Next(yCntrBlockLowerBound +
           yCntrBlockDisInterval * int_Temp_y,
                yCntrBlockUpperBound +
                   yCntrBlockDisInterval * int_Temp_y);

        return xyc;
    }

    static double seekingShortestTr
        (double[] xyOrd, double[][] trPositions, ref int
           trDisIndex)
    {
        //Search the closest truck
        List<double> arrlist_Temp1 = new List<double>();
        List<double> arrlist_Temp2 = new List<double>();
        for (int int_Temp_Tr = 0; int_Temp_Tr <
           trPositions.Length; int_Temp_Tr++)
        {
            //Normal distance
            double trDis = Math.Abs(xyOrd[0] -
               trPositions[int_Temp_Tr][0])
```

```
        + Math.Abs(xyOrd[1] - trPositions[
            int_Temp_Tr][1]);

//Extra distance across container blocks#
double extrDis = 0;
//x=xyOrd[0] and x= trPositions[][0] both
    cross container blocks
if (
     (300 < xyOrd[0]) && (xyOrd[0] < 500) &&
     (300 < trPositions[int_Temp_Tr][0]) && (
         trPositions[int_Temp_Tr][0] < 500)
     )
{
     extrDis = 2 * Math.Min(
          Math.Min(Math.Abs(xyOrd[0] - 300),
              Math.Abs(xyOrd[0] - 500)),
         Math.Min(Math.Abs(trPositions[
             int_Temp_Tr][0] - 300), Math.Abs(
             trPositions[int_Temp_Tr][0] - 500))
     );
}

if (
     (800 < xyOrd[0]) && (xyOrd[0] < 1000) &&
     (800 < trPositions[int_Temp_Tr][0]) && (
         trPositions[int_Temp_Tr][0] < 1000)
     )
{
     extrDis = 2 * Math.Min(
```

```
            Math.Min(Math.Abs(xyOrd[0] - 800),
                Math.Abs(xyOrd[0] - 1000)),
          Math.Min(Math.Abs(trPositions[
              int_Temp_Tr][0] - 800), Math.Abs(
              trPositions[int_Temp_Tr][0] - 1000)
              )
      );
}

if (
      (1300 < xyOrd[0]) && (xyOrd[0] < 1500) &&
      (1300 < trPositions[int_Temp_Tr][0]) && (
          trPositions[int_Temp_Tr][0] < 1500)
      )
{
      extrDis = 2 * Math.Min(
            Math.Min(Math.Abs(xyOrd[0] - 1300),
                Math.Abs(xyOrd[0] - 1500)),
          Math.Min(Math.Abs(trPositions[
              int_Temp_Tr][0] - 1300), Math.Abs(
              trPositions[int_Temp_Tr][0] - 1500)
              )
      );
}
//y
if (
      (300 < xyOrd[1]) && (xyOrd[1] < 500) &&
      (300 < trPositions[int_Temp_Tr][1]) && (
          trPositions[int_Temp_Tr][1] < 500)
```

```
    )
{
    extrDis = 2 * Math.Min(
        Math.Min(Math.Abs(xyOrd[1] - 300),
            Math.Abs(xyOrd[1] - 500)),
        Math.Min(Math.Abs(trPositions[
            int_Temp_Tr][1] - 300), Math.Abs(
            trPositions[int_Temp_Tr][1] - 500))
    );
}

if (
    (800 < xyOrd[1]) && (xyOrd[1] < 1000) &&
    (800 < trPositions[int_Temp_Tr][1]) && (
        trPositions[int_Temp_Tr][1] < 1000)
    )
{
    extrDis = 2 * Math.Min(
        Math.Min(Math.Abs(xyOrd[1] - 800),
            Math.Abs(xyOrd[1] - 1000)),
        Math.Min(Math.Abs(trPositions[
            int_Temp_Tr][1] - 800), Math.Abs(
            trPositions[int_Temp_Tr][1] - 1000)
            )
    );
}

if (
    (1300 < xyOrd[1]) && (xyOrd[1] < 1500) &&
```

```
                    (1300 < trPositions[int_Temp_Tr][1]) && (
                        trPositions[int_Temp_Tr][1] < 1500)
                    )
                {
                    extrDis = 2 * Math.Min(
                         Math.Min(Math.Abs(xyOrd[1] - 1300),
                            Math.Abs(xyOrd[1] - 1500)),
                        Math.Min(Math.Abs(trPositions[
                            int_Temp_Tr][1] - 1300), Math.Abs(
                            trPositions[int_Temp_Tr][1] - 1500)
                            )
                    );
                }

                trDis = trDis + extrDis;
                arrlist_Temp1.Add(trDis);
                arrlist_Temp2.Add(trDis);
            }

            arrlist_Temp1.Sort();//from min to max

            double minDis = arrlist_Temp1[0];//get min
            trDisIndex = arrlist_Temp2.IndexOf(minDis);
            return (minDis);
        }
    }
}
```

Appendix C

C# Codes for the Genetic Algorithm to Explore Pareto Optimal Solutions in Post-MOO Structure

This programme is a "Console Application", which includes one source code document shown as follows:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;//Create txt files
using System.Diagnostics;//Stopwatch to record programme
    running time

namespace Fr_SearchBestFunValue
{
    public struct structChromosome
    {
        public int Nqc, Nyc, Ntr;
        public double valueFun1, valueFun2, valueFun3,
            valueFun4;
        public double fitness;
    }
    class Program
    {
```

```
static void Main(string[] args)
{
    //Record the programme running time
    Stopwatch programRunningTime = new Stopwatch();
    programRunningTime.Start();

    double crossoverRate = 0.6; //Crossover rate
    double mutationRate = 0.05; //Mutation rate
    int xLength = 3;//Length of chromosomes
    int populationSize = 150; //Population size
    int maxNumberOfGenerations = 800; //Number of
        iterations

    int qcLowerBound = 1;
    int qcUpperBound = 50;
    int ycLowerBound = 20;
    int ycUpperBound = 500;
    int trLowerBound = 20;
    int trUpperBound = 200;

    string txtFilePath = Environment.GetFolderPath(
        Environment.SpecialFolder.DesktopDirectory)
         + "\\MOO_Solutions_ME.txt";
    StreamWriter txtFile;
    txtFile = File.CreateText(txtFilePath);
    txtFile.WriteLine("Population Size:{0};Number of 
        Generations:{1};Crossover Rate:{2};Mutation 
        Rate:{3}",
```

```
    populationSize, maxNumberOfGenerations,
        crossoverRate, mutationRate);
//Genetic Algorithm
int currentNumOfGenerations = 0;
List<structChromosome> solutionPopulation = new
    List<structChromosome>();
List<double> douList_Fitness = new List<double>()
    ;
Random randomSeed = new Random(DateTime.Now.
    Millisecond);

//First step: Generate initial population
txtFile.WriteLine("Current number of Generations:
     {0}",
                    Convert.ToString(
                        currentNumOfGenerations));
for (int i = 0; i < populationSize; i++)
{
    bool tag = false;
    structChromosome newChromosome = new
        structChromosome();
    do
    {
        newChromosome.Nqc = randomSeed.Next(
            qcLowerBound, qcUpperBound);
        newChromosome.Nyc = randomSeed.Next(
            ycLowerBound, ycUpperBound);
```

```
            newChromosome.Ntr = randomSeed.Next(
                trLowerBound, trUpperBound);

            foreach (structChromosome strChrom_Temp
                in solutionPopulation)
            {
                tag = threeSameVariants(strChrom_Temp
                    , newChromosome);
            }
        }
        while (tag);
        newChromosome.valueFun1 = obj_Fun1
            (newChromosome.Nqc, newChromosome.Nyc,
                newChromosome.Ntr);
        newChromosome.valueFun2 = obj_Fun2
            (newChromosome.Nqc, newChromosome.Nyc,
                newChromosome.Ntr);
        newChromosome.valueFun3 = obj_Fun3
            (newChromosome.Nqc, newChromosome.Nyc,
                newChromosome.Ntr);
        newChromosome.valueFun4 = obj_Fun4(
            newChromosome.Ntr);

        solutionPopulation.Add(newChromosome);
    }

    //Second step: Evolve "maxNumberOfGenerations"
        generations
```

```
for (; currentNumOfGenerations <
    maxNumberOfGenerations;
    currentNumOfGenerations++)
{
     if (currentNumOfGenerations % 100 == 0)
     {
          Console.WriteLine("Current generation:" +
               currentNumOfGenerations);
     }
     //Results display
     int numFitnessGreaterThan90 = 0;
     double totalFitness = 0;
     //Calculate fitness values
     for (int int_Temp_i = 0; int_Temp_i <
         solutionPopulation.Count; int_Temp_i++)
     {
          //Get fitness value
          solutionPopulation[int_Temp_i] =
               copyChromosome(solutionPopulation[
                  int_Temp_i], solutionPopulation);
          totalFitness = totalFitness +
              solutionPopulation[int_Temp_i].fitness
              ;
          if (solutionPopulation[int_Temp_i].
              fitness > (0.9 * populationSize))
          { numFitnessGreaterThan90++; }
          txtFile.WriteLine(
```

```
        "Nqc: {0}; Nyc: {1}; Ntr: {2};Fitness
           value:{3};Fun1:{4};Fun2:{5};Fun3
           :{6};Fun4:{7};",
        solutionPopulation[int_Temp_i].Nqc,
           solutionPopulation[int_Temp_i].Nyc
           ,
        solutionPopulation[int_Temp_i].Ntr,
           solutionPopulation[int_Temp_i].
           fitness,
        solutionPopulation[int_Temp_i].
           valueFun1, solutionPopulation[
           int_Temp_i].valueFun2,
        solutionPopulation[int_Temp_i].
           valueFun3, solutionPopulation[
           int_Temp_i].valueFun4
        );
}

txtFile.WriteLine("Percentage of elements (
   Fitness values > 90%):{0}",
   numFitnessGreaterThan90);
double dou_Temp = numFitnessGreaterThan90 *
   0.1 * 10 / solutionPopulation.Count;
douList_Fitness.Add(dou_Temp);

//Evolution
```

```
txtFile.WriteLine("Current␣number␣of␣
    Generations:{0}", currentNumOfGenerations)
    ;
//newsolutionPopulation.Clear();//Remove all
    elements
for (int i = 0; i < populationSize; i = i +
    2)
{
     //Roulette wheel selection
     int index1 = rouletteWheelSelection(
         totalFitness, solutionPopulation);
     int index2;
     do
     {
          index2 = rouletteWheelSelection(
              totalFitness, solutionPopulation);
     }
     while (index1 == index2);

     structChromosome chromo1 =
         solutionPopulation[index1];
     structChromosome chromo2 =
         solutionPopulation[index2];

     if (randomSeed.NextDouble() <
         crossoverRate)
     {
          crossoverFun(
```

```
            ref chromo1, ref chromo2,
               solutionPopulation, xLength,
            qcLowerBound, qcUpperBound,
            ycLowerBound, ycUpperBound,
            trLowerBound, trUpperBound
            );
        chromo1 = copyChromosome(chromo1,
           solutionPopulation);
        chromo2 = copyChromosome(chromo2,
           solutionPopulation);

        solutionPopulation[index1] = chromo1;
        solutionPopulation[index2] = chromo2;
    }

    if (randomSeed.NextDouble() <
       mutationRate)
    {
        chromo1 = mutationFun(
            chromo1, solutionPopulation,
               xLength,
            qcLowerBound, qcUpperBound,
            ycLowerBound, ycUpperBound,
            trLowerBound, trUpperBound
            );
        chromo1 = copyChromosome(chromo1,
           solutionPopulation);
        chromo2 = mutationFun(
```

```
                    chromo2, solutionPopulation,
                        xLength,
                    qcLowerBound, qcUpperBound,
                    ycLowerBound, ycUpperBound,
                    trLowerBound, trUpperBound
                    );
                chromo2 = copyChromosome(chromo2,
                    solutionPopulation);
                solutionPopulation[index1] = chromo1;
                solutionPopulation[index2] = chromo2;
            }
        }
    }
    for (int int_Temp_i = 0; int_Temp_i <
        solutionPopulation.Count; int_Temp_i++)
    {
        if (
            (solutionPopulation[int_Temp_i].fitness <
                0.95 * populationSize) ||
            (solutionPopulation[int_Temp_i].valueFun1
                <= 0) ||
            (solutionPopulation[int_Temp_i].valueFun2
                <= 0) ||
            (solutionPopulation[int_Temp_i].valueFun3
                <= 0) ||
            (solutionPopulation[int_Temp_i].valueFun4
                <= 0)
            )
```

```
        {
            solutionPopulation.RemoveAt(int_Temp_i);
            int_Temp_i--;//Move a step backwards
                because the number of elements
                decreases one.
        }
    }
    List<double> douList_BestObj1 = new List<double
        >();
    List<double> douList_BestObj2 = new List<double
        >();
    List<double> douList_BestObj3 = new List<double
        >();
    List<double> douList_BestObj4 = new List<double
        >();
    List<double> douList_Balanced = new List<double
        >();

    List<double> douList_BestObj1_Ind = new List<
        double>();
    List<double> douList_BestObj2_Ind = new List<
        double>();
    List<double> douList_BestObj3_Ind = new List<
        double>();
    List<double> douList_BestObj4_Ind = new List<
        double>();
    List<double> douList_Balanced_Ind = new List<
        double>();
```

```
foreach (structChromosome strChrom_Temp in
   solutionPopulation)
{
    douList_BestObj1.Add(strChrom_Temp.valueFun1)
        ;
    douList_BestObj2.Add(strChrom_Temp.valueFun2)
        ;
    douList_BestObj3.Add(strChrom_Temp.valueFun3)
        ;
    douList_BestObj4.Add(strChrom_Temp.valueFun4)
        ;
}

douList_BestObj1_Ind.AddRange(douList_BestObj1);
douList_BestObj2_Ind.AddRange(douList_BestObj2);
douList_BestObj3_Ind.AddRange(douList_BestObj3);
douList_BestObj4_Ind.AddRange(douList_BestObj4);

douList_BestObj1.Sort();//from min to max
douList_BestObj2.Sort();
douList_BestObj3.Sort();
douList_BestObj4.Sort();

//Fun1
txtFile.WriteLine("The Max Value for Objective
   Function 1:{0};",
```

```
    douList_BestObj1[douList_BestObj1.Count - 1])
        ;
txtFile.WriteLine("The Min Value for Objective 
   Function 1:{0};", douList_BestObj1[0]);
//Fin2
txtFile.WriteLine("The Max Value for Objective 
   Function 2:{0};",
    douList_BestObj2[douList_BestObj2.Count - 1])
        ;
txtFile.WriteLine("The Min Value for Objective 
   Function 2:{0};", douList_BestObj2[0]);
//Fun3
txtFile.WriteLine("The Max Value for Objective 
   Function 3:{0};",
    douList_BestObj3[douList_BestObj3.Count - 1])
        ;
txtFile.WriteLine("The Min Value for Objective 
   Function 3:{0};", douList_BestObj3[0]);
//Fun4
txtFile.WriteLine("The Max Value for Objective 
   Function 4:{0};",
    douList_BestObj4[douList_BestObj4.Count - 1])
        ;
txtFile.WriteLine("The Min Value for Objective 
   Function 4:{0};", douList_BestObj4[0]);
//
```

```
txtFile.WriteLine("————Delete negative 
   solutions and solutions which's fitness < 
   95%————");
txtFile.WriteLine("————Number of Good 
   Normalised Solutions: " +
    solutionPopulation.Count + "——————");
foreach (structChromosome struChro_Temp in
   solutionPopulation)
{
     txtFile.WriteLine(
         "Nqc:{0};Nyc:{1};Ntr:{2};Fun1:{3};Fun2
            :{4};Fun3:{5};Fun4:{6};Fitness:{7}",
         struChro_Temp.Nqc, struChro_Temp.Nyc,
            struChro_Temp.Ntr,
         struChro_Temp.valueFun1, struChro_Temp.
            valueFun2,
         struChro_Temp.valueFun3, struChro_Temp.
            valueFun4,
         struChro_Temp.fitness
         );
}

//Normalisation and sorting
List<structChromosome> list_NormalisedSolutions =
    new List<structChromosome>();
foreach (structChromosome strChro_Temp in
   solutionPopulation)
{
```

```
structChromosome newChromosome = new
   structChromosome();
newChromosome.Nqc = strChro_Temp.Nqc;
newChromosome.Nyc = strChro_Temp.Nyc;
newChromosome.Ntr = strChro_Temp.Ntr;

newChromosome.valueFun1 =
     (strChro_Temp.valueFun1 -
        douList_BestObj1[0]) /
     (douList_BestObj1[douList_BestObj1.Count
        - 1] - douList_BestObj1[0]);
// (value - min) / (max - min);

newChromosome.valueFun2 =
     (strChro_Temp.valueFun2 -
        douList_BestObj2[0]) /
     (douList_BestObj2[douList_BestObj2.Count
        - 1] - douList_BestObj2[0]);

newChromosome.valueFun3 =
     (strChro_Temp.valueFun3 -
        douList_BestObj3[0]) /
     (douList_BestObj3[douList_BestObj3.Count
        - 1] - douList_BestObj3[0]);

newChromosome.valueFun4 =
     (strChro_Temp.valueFun4 -
        douList_BestObj4[0]) /
```

```
        (douList_BestObj4[douList_BestObj4.Count
           - 1] - douList_BestObj4[0]);

    list_NormalisedSolutions.Add(newChromosome);
}

txtFile.WriteLine("——————————————————————
   Normalised Solutions
   ——————————————————————————————");
txtFile.WriteLine("———————Number of Good 
   Normalised Solutions: " +
    list_NormalisedSolutions.Count + "
       ————————————");
foreach (structChromosome struChro_Temp in
   list_NormalisedSolutions)
{
    txtFile.WriteLine(
        "Nqc:{0};Nyc:{1};Ntr:{2};Fun1:{3};Fun2
           :{4};Fun3:{5};Fun4:{6};Fitness:{7}",
        struChro_Temp.Nqc, struChro_Temp.Nyc,
           struChro_Temp.Ntr,
        struChro_Temp.valueFun1, struChro_Temp.
           valueFun2,
        struChro_Temp.valueFun3, struChro_Temp.
           valueFun4,
        struChro_Temp.fitness
        );
}
```

```
douList_BestObj1.Reverse();//from max to min
douList_BestObj2.Reverse();//from max to min

txtFile.WriteLine("-------------------------Best 
   Normalised Solutions for the First Objective 
   Function-----------------------------------");
for (int int_Temp_i = 0; int_Temp_i < 10;
   int_Temp_i++)
{
    int index_Temp = douList_BestObj1_Ind.IndexOf
       (douList_BestObj1[0]);
    //Avoidance of repetitive solutions
    douList_BestObj1.RemoveAt(0);
    douList_BestObj1_Ind[index_Temp] =
       douList_BestObj1[douList_BestObj1.Count -
       1];

    txtFile.WriteLine(
        "Nqc:{0};Nyc:{1};Ntr:{2};Fun1:{3};Fun2
           :{4};Fun3:{5};Fun4:{6}",
        list_NormalisedSolutions[index_Temp].Nqc,
        list_NormalisedSolutions[index_Temp].Nyc,
        list_NormalisedSolutions[index_Temp].Ntr,
        list_NormalisedSolutions[index_Temp].
           valueFun1,
        list_NormalisedSolutions[index_Temp].
           valueFun2,
```

```
            list_NormalisedSolutions[index_Temp].
                valueFun3,
            list_NormalisedSolutions[index_Temp].
                valueFun4
            );
}

txtFile.WriteLine("——————————————Best 
    Normalised Solutions for the Second Objective 
    Function————————————————");
for (int int_Temp_i = 0; int_Temp_i < 10;
    int_Temp_i++)
{
      int index_Temp = douList_BestObj2_Ind.IndexOf
          (douList_BestObj2[0]);
      //Avoidance of repetitive solutions
      douList_BestObj2.RemoveAt(0);
      douList_BestObj2_Ind[index_Temp] =
          douList_BestObj2[douList_BestObj2.Count -
          1];

      txtFile.WriteLine(
           "Nqc:{0};Nyc:{1};Ntr:{2};Fun1:{3};Fun2
               :{4};Fun3:{5};Fun4:{6}",
            list_NormalisedSolutions[index_Temp].Nqc,
            list_NormalisedSolutions[index_Temp].Nyc,
            list_NormalisedSolutions[index_Temp].Ntr,
```

```
            list_NormalisedSolutions[index_Temp].
                valueFun1,
            list_NormalisedSolutions[index_Temp].
                valueFun2,
            list_NormalisedSolutions[index_Temp].
                valueFun3,
            list_NormalisedSolutions[index_Temp].
                valueFun4
            );
}

txtFile.WriteLine("------------------------Best 
    Normalised Solutions for the Third Objective 
    Function--------------------------------");
for (int int_Temp_i = 0; int_Temp_i < 10;
    int_Temp_i++)
{
      int index_Temp = douList_BestObj3_Ind.IndexOf
          (douList_BestObj3[0]);
      //Avoidance of repetitive solutions
      douList_BestObj3.RemoveAt(0);
      douList_BestObj3_Ind[index_Temp] =
          douList_BestObj3[douList_BestObj3.Count -
          1];

      txtFile.WriteLine(
          "Nqc:{0};Nyc:{1};Ntr:{2};Fun1:{3};Fun2
              :{4};Fun3:{5};Fun4:{6}",
```

```
        list_NormalisedSolutions[index_Temp].Nqc,
        list_NormalisedSolutions[index_Temp].Nyc,
        list_NormalisedSolutions[index_Temp].Ntr,
        list_NormalisedSolutions[index_Temp].
          valueFun1,
        list_NormalisedSolutions[index_Temp].
          valueFun2,
        list_NormalisedSolutions[index_Temp].
          valueFun3,
        list_NormalisedSolutions[index_Temp].
          valueFun4
        );
}

txtFile.WriteLine("————————————————Best␣
  Normalised␣Solutions␣for␣the␣Fourth␣Objective␣
  Function——————————————————————");
for (int int_Temp_i = 0; int_Temp_i < 10;
  int_Temp_i++)
{
    int index_Temp = douList_BestObj4_Ind.IndexOf
        (douList_BestObj4[0]);
    //Avoidance of repetitive solutions
    douList_BestObj4.RemoveAt(0);
    douList_BestObj4_Ind[index_Temp] =
        douList_BestObj4[douList_BestObj4.Count -
        1];
```

```
    txtFile.WriteLine(
        "Nqc:{0};Nyc:{1};Ntr:{2};Fun1:{3};Fun2
            :{4};Fun3:{5};Fun4:{6}",
        list_NormalisedSolutions[index_Temp].Nqc,
        list_NormalisedSolutions[index_Temp].Nyc,
        list_NormalisedSolutions[index_Temp].Ntr,
        list_NormalisedSolutions[index_Temp].
            valueFun1,
        list_NormalisedSolutions[index_Temp].
            valueFun2,
        list_NormalisedSolutions[index_Temp].
            valueFun3,
        list_NormalisedSolutions[index_Temp].
            valueFun4
        );
}

foreach (structChromosome strChrom_Temp in
    list_NormalisedSolutions)
{
    douList_Balanced.Add(
        Math.Abs(strChrom_Temp.valueFun1 - 0.5) +
        Math.Abs(strChrom_Temp.valueFun2 - 0.5) +
        Math.Abs(strChrom_Temp.valueFun3 - 0.5) +
        Math.Abs(strChrom_Temp.valueFun4 - 0.5)
        );
}
douList_Balanced_Ind.AddRange(douList_Balanced);
```

```
douList_Balanced.Sort();

txtFile.WriteLine("————————————Most␣
   Balanced␣Normalised␣Solutions
   ————————————————");
for (int int_Temp_i = 0; int_Temp_i < 10;
   int_Temp_i++)
{
    int index_Temp = douList_Balanced_Ind.IndexOf
       (douList_Balanced[0]);
    //Avoidance of repetitive solutions
    douList_Balanced.RemoveAt(0);
    douList_Balanced_Ind[index_Temp] =
       douList_Balanced[douList_Balanced.Count −
       1];

    txtFile.WriteLine(
        "Nqc:{0};Nyc:{1};Ntr:{2};Fun1:{3};Fun2
           :{4};Fun3:{5};Fun4:{6}",
        list_NormalisedSolutions[index_Temp].Nqc,
        list_NormalisedSolutions[index_Temp].Nyc,
        list_NormalisedSolutions[index_Temp].Ntr,
        list_NormalisedSolutions[index_Temp].
           valueFun1,
        list_NormalisedSolutions[index_Temp].
           valueFun2,
        list_NormalisedSolutions[index_Temp].
           valueFun3,
```

```
        list_NormalisedSolutions[index_Temp].
            valueFun4
        );
}

//Fitness Values
txtFile.WriteLine("——————Fitness␣Values␣for
    ␣Each␣Generation—————");
foreach (double dou_Temp in douList_Fitness)
{
    txtFile.WriteLine(dou_Temp);
}

programRunningTime.Stop();

txtFile.WriteLine("———————————
    Programme␣Running␣Time
    ——————————————");
txtFile.WriteLine("Total␣Running␣Time" +
    programRunningTime.Elapsed);

txtFile.Close();
Console.WriteLine("Solutions␣have␣been␣exported␣
    to␣" + txtFilePath
     + "\nProgramme␣Running␣Time" +
        programRunningTime.Elapsed);
Console.WriteLine("Job␣done!");
Console.ReadKey();
```

```
}

//main function ends

private static bool threeSameVariants
    (structChromosome f_Chromo1, structChromosome
       f_Chromo2)
{
    return (
        (f_Chromo1.Nqc == f_Chromo2.Nqc) &&
        (f_Chromo1.Nyc == f_Chromo2.Nyc) &&
        (f_Chromo1.Ntr == f_Chromo2.Ntr)
        );
}
private static structChromosome copyChromosome
    (structChromosome Origin, List<structChromosome>
       f_solutionPopulation)
{
    structChromosome newChromosome = new
       structChromosome();
    newChromosome.Nqc = Origin.Nqc;
    newChromosome.Nyc = Origin.Nyc;
    newChromosome.Ntr = Origin.Ntr;
    newChromosome.valueFun1 = Origin.valueFun1;
    newChromosome.valueFun2 = Origin.valueFun2;
    newChromosome.valueFun3 = Origin.valueFun3;
    newChromosome.valueFun4 = Origin.valueFun4;
```

```
    newChromosome.fitness = fitnessValue(Origin,
       f_solutionPopulation);
    return newChromosome;
}

private static int rouletteWheelSelection
    (double totalFitness, List<structChromosome>
       f_solutionPopulation)
{
    Random randomSeed = new Random(DateTime.Now.
       Millisecond);
    double slice = randomSeed.NextDouble() *
       totalFitness;
    double fitnessSoFar = 0;
    for (int int_Temp_i = 0; int_Temp_i <
       f_solutionPopulation.Count; int_Temp_i++)
    {
        fitnessSoFar = fitnessSoFar +
           f_solutionPopulation[int_Temp_i].fitness;
        if (fitnessSoFar >= slice)
        {
            return int_Temp_i;
        }
    }
    return randomSeed.Next(0, f_solutionPopulation.
       Count);
}
```

```
public static void crossoverFun
    (ref structChromosome f_chromo1, ref
       structChromosome f_chromo2,
    List<structChromosome> f_solutionPopulation, int
       f_xLength,
    int f_qcLowerBound, int f_qcUpperBound, int
       f_ycLowerBound,
    int f_ycUpperBound, int f_trLowerBound, int
       f_trUpperBound)
{
    List<double> listDouArr_Temp_Ntr = new List<
       double>();
    List<double> listDouArr_Temp_Nyc = new List<
       double>();
    List<double> listDouArr_Temp_Nqc = new List<
       double>();

    foreach (structChromosome strChromo_Temp in
       f_solutionPopulation)
    {
        listDouArr_Temp_Nqc.Add(strChromo_Temp.Nqc);
        listDouArr_Temp_Nyc.Add(strChromo_Temp.Nyc);
        listDouArr_Temp_Ntr.Add(strChromo_Temp.Ntr);
    }

    //structChromosome f_chromo_Temp = new
       structChromosome();
    int f_chromo1_Nqc = f_chromo1.Nqc;
```

```
int f_chromo2_Nqc = f_chromo2.Nqc;
int f_chromo1_Nyc = f_chromo1.Nyc;
int f_chromo2_Nyc = f_chromo2.Nyc;
int f_chromo1_Ntr = f_chromo1.Ntr;
int f_chromo2_Ntr = f_chromo2.Ntr;

sub_crossoverFun(
    ref f_chromo1_Nqc, ref f_chromo2_Nqc,
       f_xLength,
    f_qcLowerBound, f_qcUpperBound
    );
sub_crossoverFun(
    ref f_chromo1_Nyc, ref f_chromo2_Nyc,
       f_xLength,
    f_ycLowerBound, f_ycUpperBound
    );
sub_crossoverFun(
    ref f_chromo1_Ntr, ref f_chromo2_Ntr,
       f_xLength,
    f_trLowerBound, f_trUpperBound
    );

f_chromo1.Nqc = f_chromo1_Nqc;
f_chromo1.Nyc = f_chromo1_Nyc;
f_chromo1.Ntr = f_chromo1_Ntr;
f_chromo1.valueFun1 = obj_Fun1(f_chromo1.Nqc,
   f_chromo1.Nyc, f_chromo1.Ntr);
```

```
    f_chromo1.valueFun2 = obj_Fun2(f_chromo1.Nqc,
       f_chromo1.Nyc, f_chromo1.Ntr);
    f_chromo1.valueFun3 = obj_Fun3(f_chromo1.Nqc,
       f_chromo1.Nyc, f_chromo1.Ntr);
    f_chromo1.valueFun4 = obj_Fun4(f_chromo1.Ntr);

    f_chromo2.Nqc = f_chromo2_Nqc;
    f_chromo2.Nyc = f_chromo2_Nyc;
    f_chromo2.Ntr = f_chromo2_Ntr;
    f_chromo2.valueFun1 = obj_Fun1(f_chromo2.Nqc,
       f_chromo2.Nyc, f_chromo2.Ntr);
    f_chromo2.valueFun2 = obj_Fun2(f_chromo2.Nqc,
       f_chromo2.Nyc, f_chromo2.Ntr);
    f_chromo2.valueFun3 = obj_Fun3(f_chromo2.Nqc,
       f_chromo2.Nyc, f_chromo2.Ntr);
    f_chromo2.valueFun4 = obj_Fun4(f_chromo2.Ntr);

}

public static void sub_crossoverFun
    (ref int int_Chrom_1, ref int int_Chrom_2, int
       f_xLength,
    int lowerBound, int upperBound)
{
    int x1 = int_Chrom_1;
    int x2 = int_Chrom_2;
    //MessageBox.Show(str_Chrom_1+";"+ str_Chrom_2);
```

```
Random randomSeed = new Random(DateTime.Now.
    Millisecond);

int cutPosition = randomSeed.Next(f_xLength);
switch (cutPosition)
{
    case 0:
        int int_Temp_i = (x1 / 100) * 100 + (x2 %
            100);
        x2 = (x2 / 100) * 100 + (x1 % 100);
        x1 = int_Temp_i;
        break;
    case 1:
        int int_Temp_j_1 = x1 / 100;
        int int_Temp_j_2 = (x1 % 100) / 10;
        int int_Temp_j_3 = x1 % 10;
        int int_Temp_k_1 = x2 / 100;
        int int_Temp_k_2 = (x2 % 100) / 10;
        int int_Temp_k_3 = x2 % 10;
        x1 = int_Temp_k_1 * 100 + int_Temp_j_2 *
            10 + int_Temp_k_3;
        x2 = int_Temp_j_1 * 100 + int_Temp_k_2 *
            10 + int_Temp_j_3;
        break;
    case 2:
        int int_Temp_m = (x1 / 10) * 10 + (x2 %
            10);
        x2 = (x2 / 10) * 10 + (x1 % 10);
```

```
            x1 = int_Temp_m;
            break;
        default:
            break;
    }
    if ((x1 < lowerBound) || (x1 > upperBound))
    { x1 = randomSeed.Next(lowerBound, upperBound); }

    if ((x2 < lowerBound) || (x2 > upperBound))
    { x2 = randomSeed.Next(lowerBound, upperBound); }

    int_Chrom_1 = x1;
    int_Chrom_2 = x2;
}

public static structChromosome mutationFun
    (structChromosome f_chromo, List<structChromosome
       > f_solutionPopulation,
    int f_xLength, int f_qcLowerBound, int
       f_qcUpperBound, int f_ycLowerBound,
    int f_ycUpperBound, int f_trLowerBound, int
       f_trUpperBound)
{
    structChromosome chromo_Temp = new
       structChromosome();//
    chromo_Temp.Nqc = sub_MutationFun
        (f_chromo.Nqc, f_xLength, f_qcLowerBound,
           f_qcUpperBound);
```

```
        chromo_Temp.Nyc = sub_MutationFun
             (f_chromo.Nyc, f_xLength, f_ycLowerBound,
                f_ycUpperBound);
        chromo_Temp.Ntr = sub_MutationFun
             (f_chromo.Ntr, f_xLength, f_trLowerBound,
                f_trUpperBound);

        chromo_Temp.valueFun1 = obj_Fun1(chromo_Temp.Nqc,
             chromo_Temp.Nyc, chromo_Temp.Ntr);
        chromo_Temp.valueFun2 = obj_Fun2(chromo_Temp.Nqc,
             chromo_Temp.Nyc, chromo_Temp.Ntr);
        chromo_Temp.valueFun3 = obj_Fun3(chromo_Temp.Nqc,
             chromo_Temp.Nyc, chromo_Temp.Ntr);
        chromo_Temp.valueFun4 = obj_Fun4(chromo_Temp.Ntr)
            ;

        return chromo_Temp;
    }

    public static int sub_MutationFun(int int_Chrom, int
       f_xLength,
        int lowerBound, int upperBound)
    {
        string str_Chrom = Convert.ToString(int_Chrom);
        if (str_Chrom.Length < f_xLength)
        {
             str_Chrom = str_Chrom.PadLeft(f_xLength, '0')
                ;
```

```
}

Random randomSeed = new Random(DateTime.Now.
    Millisecond);
string strChange = Convert.ToString(randomSeed.
    Next(0, 10));
int cutPosition = randomSeed.Next(0, str_Chrom.
    Length);

if (cutPosition == 0)
{
    str_Chrom = strChange + str_Chrom;
    str_Chrom = str_Chrom.Substring(0, f_xLength)
        ;
}
else
{
    string str_Chrom_1_Head = str_Chrom.Substring
        (0, cutPosition - 1);
    string str_Chrom_1_Tail = str_Chrom.Substring
        (cutPosition);
    str_Chrom = str_Chrom_1_Head + strChange +
        str_Chrom_1_Tail;
}

int x = int.Parse(str_Chrom);
if ((x < lowerBound) || (x > upperBound))
{ x = randomSeed.Next(lowerBound, upperBound); }
```

```
    return x;
}

public static double fitnessValue
    (structChromosome chromosome, List<
       structChromosome> f_solutionPopulation)
{
    int paretoDominantNum = 0;
    //Fitness value is the number of chromosomes it
       dominates.
    foreach (structChromosome
       chromosomeFromPopulation in
       f_solutionPopulation)
    //Compare with all Values
    {
        if (paretoComparation(chromosome,
           chromosomeFromPopulation))
        //if stru_Temp1 is worse than stru_Temp2,
           return false
        {
            paretoDominantNum++;//not worse or
               comparing to the same values
        }
    }
    return paretoDominantNum;
}
```

```
static bool paretoComparation
    (structChromosome f_strChromosome,
      structChromosome chromosomeFromPopulation)
{
    bool f_tag;
    if ( //not delete itself
        (f_strChromosome.valueFun1 !=
          chromosomeFromPopulation.valueFun1) ||//
          Max 1st Obj Qc Rate
        (f_strChromosome.valueFun2 !=
          chromosomeFromPopulation.valueFun2) ||//
          Max 2nd Obj Yc Rate
        (f_strChromosome.valueFun3 !=
          chromosomeFromPopulation.valueFun3) ||//
          Min 3rd Obj Congestion
        (f_strChromosome.valueFun4 !=
          chromosomeFromPopulation.valueFun4)   //
          Min 4th Obj Travelling Distance
        )
    //This is not itself.Any of elements is not equal
        to.
    {
        if (
            (f_strChromosome.valueFun1 >
              chromosomeFromPopulation.valueFun1) ||
              //Max 1st Obj Qc Rate
```

```
            (f_strChromosome.valueFun2 >
              chromosomeFromPopulation.valueFun2) ||
              //Max 2nd Obj Yc Rate
            (f_strChromosome.valueFun3 <
              chromosomeFromPopulation.valueFun3) ||
              //Min 3rd Obj Congestion
            (f_strChromosome.valueFun4 <
              chromosomeFromPopulation.valueFun4)
              //Min 4th Obj Travelling Distance
            )
        {
            f_tag = true;//f_stru_Temp1 is not worse
              than f_stru_Temp2;
        }
        else
        {
            f_tag = false;//f_stru_Temp1 is worse
              than f_stru_Temp2; delete.
        }
    }
    else { f_tag = true; } //f_stru_Temp1 =
       f_stru_Temp2;keep it
    return f_tag;
}

static double obj_Fun1(int Nqc, int Nyc, int Ntr)
//objective function1: maximise QC Operation Rate
//Unit: moves per hour
```

```
{
    double fx = 0.005 * (2E-07 * Ntr * Ntr * Ntr -
       0.0001 * Ntr * Ntr + 0.0021 * Ntr + 15.341)
    + 0.005 * (0.0001 * Nyc * Nyc - 0.0553 * Nyc +
       15.074)
    + 0.99 * (0.013 * Nqc * Nqc - 1.0883 * Nqc +
       28.06);
    return fx;
}
//
static double obj_Fun2(int Nqc, int Nyc, int Ntr)
//objective function1: maximise YC Operation Rate
//Unit: moves per hour
{
    double fx = 0.3 * (0.0046 * Nqc * Nqc - 0.2059 *
       Nqc + 3.9918) +
         0.5 * (-1E-07 * Nyc * Nyc * Nyc + 0.0001 *
            Nyc * Nyc - 0.0472 * Nyc + 7.2676) +
         0.2 * (1E-07 * Ntr * Ntr * Ntr - 6E-05 * Ntr
            * Ntr + 0.0097 * Ntr + 1.8983);
    return fx;
}
//
static double obj_Fun3(int Nqc, int Nyc, int Ntr)
//objective function3: mnimise Congestion Probability
{
    double fTg = 0.05 * (2E-06 * Nqc * Nqc * Nqc - 9E
       -05 * Nqc * Nqc + 0.0003 * Nqc + 0.3751) +
```

```
        0.15 * (-9E-10 * Nyc * Nyc * Nyc + 1E-06 *
            Nyc * Nyc - 0.0004 * Nyc + 0.4059) +
        0.8 * (2E-11 * Ntr * Ntr * Ntr * Ntr - 2E-08
            * Ntr * Ntr * Ntr + 7E-06 * Ntr * Ntr -
            0.0011 * Ntr + 0.4167);

    double lamda;//The minimum time gap amongst
        trucks (orders)

    lamda = 1 / fTg;
    double LowerLimit = 0;
    double UpperLimit = fTg;
    double fx;
    fx = IntegralFunction(LowerLimit, UpperLimit,
        lamda, fTg);
    return fx;
}

static double obj_Fun4(int Ntr)
//objective function1: maximise YC Operation Rate
//Unit: moves per hour
{
    double fx = 0.0001 * Math.Pow(Ntr, 2) - 0.1069 *
        Ntr + 222.75;
    return fx;
}
//Objective functions end
```

```
        public static double IntegralFunction(double
            LowerLimit, double UpperLimit, double lamda,
            double Tg)
        {
            double result = 0;
            double dx = (UpperLimit - LowerLimit) / 1000;
            for (double x = LowerLimit; x <= UpperLimit; x =
                x + dx)
            {
                double fx = lamda * Math.Exp(-lamda * x);
                result = result + fx * dx;
            }
            return result;
        }
    }
}
```

Appendix D

Data Details

The data of the container throughput of Southampton Container Terminal is shown in table D.1.

Table D.1: Container Throughput of Southampton Container Terminal [122] [113] [86]

Year	2000	2005	2006	2007	2008
Throughput (TEU)	1,000,000	1,382,000	1,500,000	1,900,000	1,500,000

The numbers of quay cranes in Southampton Container Terminal between 2000 and 2009 are given in table D.2.

Table D.2: Number of Quay Cranes in Southampton Container Terminal [120]

Year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Quay Crane	4	6	6	6	6	6	7	7	9	11

The numbers of yard cranes from 2000 to 2009 are given in table D.3.

Table D.3: Number of Yard Cranes in Southampton Container Terminal [121]

Year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Straddle Carrier	35	49	58	67	78	89	94	109	109	124
Sprinter				5	5	5	5	6	6	6
Empty Container Handler				2	4	6	6	7	7	15
Reach Stacker					4	4	4	4	4	4

Bibliography

[1] C. Almeder, M. Preusser, and R. F. Hartl. Simulation and optimization of supply chains: alternative or complementary approaches? *OR Spectrum*, 31:95–119, 2009.

[2] F. Altiparmak, M. Gen, L. Lin, and T. Paksoy. A genetic algorithm approach for multi-objective optimization of supply chain networks. *Computers and Industrial Engineering*, 51:196–215, 2006.

[3] D. Ambrosino and E. Tanfani. A discrete event simulation model for the analysis of critical factors in the expansion plan of a marine container terminal. In *23rd European Conference on Modelling and Simulation*, 2009.

[4] P. Angeloudis and M. G. H. Bell. A review of container terminal simulation models. *Maritime Policy and Management*, 38:523–540, 2011.

[5] M. Asteris, A. Collins, and D. F. Jones. Container port infrastructure in northwest europe: Policy-level modeling. *Journal of Policy Modeling*, 34:312–324, 2012.

[6] F. F. Baesler and J. A. Sepulveda. Multi-objective simulation optimization for a cancer treatment center. In *2001 Winter Simulation Conference*, 2001.

[7] J. Banks, J. S. Carson II, B. L. Nelson, and D. M. Nicol. *Discrete-Event System Simulation*. Prentice Hall, 2001.

[8] A. Baylar, D. Hanbay, and M. Batan. Application of least square support vector machines in the prediction of aeration performance of plunging overfall jets from weirs. *Expert Systems with Applications*, 36:8368–8374, 2009.

[9] M. Bazzazi, N. Safaei, and N. Javadian. A genetic algorithm to solve the storage space allocation problem in a container terminal. *Computers and Industrial Engineering*, 56:44–52, 2009.

[10] C. Bierwirth and F. Meisel. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202:615–627, 2010.

[11] E. K. Bish. A multiple-crane-constrained scheduling problem in a container terminal. *European Journal of Operational Research*, 144:83–107, 2003.

[12] W. K. Bloechle and D. Schunk. Micro saint sharp simulation software: micro saint sharp simulation software. In *35th conference on Winter simulation: driving innovation*, pages 182–187, 2003.

[13] H.W. Bode and C.E. Shannon. A simplified derivation of linear least square smoothing and prediction theory. In *Institution of Radio Engineers*, volume 38, pages 417–425, 1950.

[14] A. Bortfeldt and F. Forster. A tree search procedure for the container premarshalling problem. *European Journal of Operational Research*, 217:531–540, 2012.

[15] J. Branke, T. Kauβler, and H. Schmeck. Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32:499–507, 2001.

[16] J. Cao, D. Lee, J. Chen, and Q. Shi. The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods. *Transportation Research Part E: Logistics and Transportation Review*, 46:344–353, 2010.

[17] S. Chanas and D. Kuchta. Multiobjective programming in optimization of interval objective functions - a generalized approach. *European Journal of Operational Research*, pages 594–598, 1996.

[18] D. Chang, J. He, and H. Zhang. A rule-based joint berth allocation and quay crane assignment. In *2nd International Conference on Industrial and Information Systems (IIS)*, volume 1, pages 464 – 467, 2010.

[19] D. Chang, Z. Jiang, W. Yan, and J. He. Developing a dynamic rolling-horizon decision strategy for yard crane scheduling. *Advanced Engineering Informatics*, 25:485–494, 2011.

[20] C. Chen, D. He, M. Fu, and L. H. Lee. Efficient simulation budget allocation for selecting an optimal subset. *INFORMS Journal on Computing*, 20:579–595, 2008.

[21] C. Chen and W. Lee. Multi-objective optimization of multi-echelon supply chain networks with uncertain product demands and prices. *Computers and Chemical Engineering*, 28:1131–1144, 2004.

[22] J. Chung, S. Oh, and I. Choi. A hybrid genetic algorithm for train sequencing in the korean railway. *Omega*, 37:555–565, 2009.

[23] C.A. Coello. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine*, 1:28–36, 2006.

[24] J. L. Cohon. *Multiobjective Progamming and Planning*. Dover Publications, New York, 2003.

[25] Container-transportation.com. Iso container standards. `http://www.container-transportation.com/iso-container-standards.html`.

[26] Microsoft Corporation. Visual studio 2010 express products. `http://www.microsoft.com/visualstudio/en-gb/express`, 2010.

[27] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons, 2001.

[28] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*, pages 849–858. 2000.

[29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182197, 2002.

[30] E. Demirci. Simulation modelling and analysis of a port investment. *Simulation*, 79:94–105, 2003.

[31] G. Derringer and R. Suich. Simultaneous optimization of several variables. *Journal of Quality Technology*, 12:214–219, 1980.

[32] L. Diaz-Balteiro and C. Romero. Making forestry decisions with multiple criteria: A review and an assessment. *Forest Ecology and Management*, 255:3222–3241, 2008.

[33] M. Ehrgott. *Multicriteria Optimization*. Springer, second edition, 2005.

[34] S. Esmer. Performance measurements of container terminal operations. `http://www.sbe.deu.edu.tr/dergi/cilt10.say%C4%B11/10.1%20esmer.pdf`, 2008.

[35] C. Exposito-Izquierdo, B. Melian-Batista, and M. Moreno-Vega. Pre-marshalling problem: Heuristic solution method and instances generator. *Expert Systems with Applications*, 39:8337–8349, 2012.

[36] T. A. Feo and M. G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–134, 1995.

[37] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Fifth International Conference in Genetic Algorithms*, 1993.

[38] S. Gao. Economic globalization: Trends, risks and risk prevention. `http://www.un.org/esa/analysis/devplan/cdpbackgroundpapers/bp2000_1shangquan.pdf`, 2000.

[39] S. I. Gass and T. L. Saaty. Parametric objective function (part 2)-generalization. *Journal of the Operations Research Society of America*, 3:395–401, 1955.

[40] A. M. Geoffrion, J. S. Dyer, and A. Feinberg. An interactive approach for multi-criterion optimization, with an application to the operation of an academic department. *Management Science*, 19:357–368, 1972.

[41] C.K. Goh, K.C. Tan, D.S. Liu, and S.C. Chiam. A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design. *European Journal of Operational Research*, 202:42–54, 2010.

[42] M. M. Golias, M. Boile, and S. Theofanis. Berth scheduling by customer service differentiation:a multi-objective approach. *Transportation Research Part E: Logistics and Transportation Review*, 45:878–892, 2009.

[43] E. Hadjiconstantinou and N. L. Ma. Evaluating straddle carrier deployment policies: a simulation study for the piraeus container terminal. *Maritime Policy and Management*, 36:353–366, 2009.

[44] D. Hanbay, A. Baylar, and M. Batan. Prediction of aeration efficiency on stepped cascades by using least square support vector machines. *Expert Systems with Applications*, 36:4248–2452, 2009.

[45] Hapag-Lloyd. Container specification. `http://www.hapag-lloyd.com/downloads/press_and_media/publications/Brochure_Container_Specification_en.pdf`.

[46] S. Hartmann. A general framework for scheduling equipment and manpower at container terminals. *OR Spectrum*, 26:51–74, 2004.

[47] J. He, D. Chang, W. Mi, and W. Yan. A strategy for yard crane scheduling based on hybrid parallel genetic algorithm. In *International Symposium on Knowledge Acquisition and Modeling*, pages 678–683, 2008.

[48] J. He, D. Chang, W. Mi, and W. Yan. A hybrid parallel genetic algorithm for yard crane scheduling. *Transportation Research Part E: Logistics and Transportation Review*, 46:136–155, 2010.

[49] L. Henesey, P. Davidsson, and J. A. Persson. Agent based simulation architecture for evaluating operational policies in transshipping containers. *MATES*, 4196:73–85, 2006.

[50] Y. Hirashima. A q-learning system for group-based container marshalling with a-priori knowledge for ship loading. In *ICROS-SICE International Joint Conference*, pages 1728–1733, 2009.

[51] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *First IEEE Conference on Evolutionary Computation*, volume 1, pages 82–87, 1994.

[52] S. Huang and T. Lin. Heuristic algorithms for container pre-marshalling problems. *Computers and Industrial Engineering*, 62:13–20, 2012.

[53] Y. Huang, C. Liang, and Y. Yang. The optimum route problem by genetic algorithm for loading/unloading of yard crane. *Computers and Industrial Engineering*, 56:993–1001, 2009.

[54] K. Ikeda, H. Kita, and S. Kobayashi. Failure of pareto-based moeas: does non-dominated really mean near to optimal? In *2001 Congress on Evolutionary Computation*, volume 2, pages 957–962, 2001.

[55] A. Imai, H. C. Chen, E. Nishimura, and S. Papadimitriou. The simultaneous berth and quay crane allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, 44:900–920, 2008.

[56] M. Jahangirian, T. Eldabi, A. Naseer, L. K. Stergioulas, and T. Young. Simulation in manufacturing and business: A review. *European Journal of Operational Research*, 203:1–13, 2010.

[57] A. Jaszkiewicz. Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, 137:50–71, 2002.

[58] H. Javanshir and S. R. S. Ganji. Yard crane scheduling in port container terminals using genetic algorithm. *Journal of Industrial Engineering International*, 6:39–50, 2010.

[59] Z. Jin and N. Li. Optimization of quay crane dynamic scheduling based on berth schedules in container terminal. *Journal of Transportation Systems Engineering and Information Technology*, 11:58–64, 2011.

[60] D. Jones and M. Tamiz. *Practical Goal Programming*. International series in operations research & management science. Springer New York Dordrecht Heidelberg London, London, 2010.

[61] D.F. Jones, S.K. Mirrazavi, and M. Tamiz. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137:1–9, 2002.

[62] N. Jozefowiez, F. Semet, and E. Talbi. Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189:293–309, 2008.

[63] N. Kaveshgar, N. Huynh, and S. K. Rahimian. An efficient genetic algorithm for solving the quay crane scheduling problem. *Expert Systems with Applications*, 39:13108–13117, 2012.

[64] K. H. Kim and J. W. Bae. Re-marshaling export containers in port container terminals. *Computers and Industrial Engineering*, 35:655–658, 1998.

[65] K. H. Kim and H. B. Kim. The optimal determination of the space requirement and the number of transfer cranes for import containers. *Computers and Industrial Engineering*, 35:427–430, 1998.

[66] K. H. Kim and H. B. Kim. The optimal sizing of the storage space and handling facilities for import containers. *Transportation Research Part B: Methodological*, 36:821–835, 2002.

[67] K. H. Kim and Y. Park. A crane scheduling method for port container terminals. *European Journal of Operational Research*, 156:752–768, 2004.

[68] J. van de Klundert and B. Otten. Improving ltl truck load utilization on line. *European Journal of Operational Research*, 210:336–343, 2011.

[69] A. Konak, D. W. Coit, and A. E. Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*, 91:992–1007, 2006.

[70] E. Kozan. Optimising container transfers at multimodal terminals. *Mathematical and Computer Modelling*, 31:235–243, 2000.

[71] A. M. Law and W. D. Kelton. *Simulation modeling and analysis*. McGraw-Hill Higher Education, 2000.

[72] M. Le, C. Wu, and H. Zhang. An integrated optimization method to solve the berth-qc allocation problem. In *Eighth International Conference on Natural Computation (ICNC)*, pages 753 – 757, 2012.

[73] H. D. Le-Griffin and M. Murphy. Container terminal productivity: experiences at the ports of los angeles and long beach. In *NUF Conference*, pages 1–21, 2006.

[74] D. Lee, J. X. Cao, Q. Shi, and J. H. Chen. A heuristic algorithm for yard truck scheduling and storage allocation problems. *Transportation Research Part E*, 45:810–820, 2009.

[75] D. Lee, J. X. Cao, and Q. X. Shi. Integrated quay crane and yard truck schedule for inbound containers. In *IEEE International Conference on Industrial Engineering and Engineering Management*, pages 1219–1223, 2008.

[76] S. Lee, J. Turner, M. S. Daskin, T. Homem-de Mello, and K. Smilowitz. Improving fleet utilization for carriers by interval scheduling. *European Journal of Operational Research*, 218:261–269, 2012.

[77] T. Lee, N. Park, and D. Lee. A simulation study for the logistics planning of a container terminal in view of scm. *Maritime Policy and Management*, 30:243–254, 2003.

[78] Y. Lee and S. Chao. A neighborhood search heuristic for pre-marshalling export containers. *European Journal of Operational Research*, 196:468–475, 2009.

[79] P. Legato, R. M. Mazza, and R. Trunfio. Simulation-based optimization for discharge/loading operations at a maritime container terminal. *OR Spectrum*, 32:543–567, 2010.

[80] B. Li, W. Li, Y. Yang, Y. Ge, H. C. Chen, and X. Liang. Modeling and simulation of yard trailer dispatching at container teminals. In *International Conference on Automation and Logistics*, pages 29–34, 2009.

[81] J. Li, E. K. Burke, T. Curtois, S. Petrovic, and R. Qu. The falling tide algorithm: A new multi-objective approach for complex workforce scheduling. *Omega*, 40:283–293, 2012.

[82] W. Li, M. Goh, Y. Wu, M.E.H. Petering, R. de Souza, and Y.C. Wu. A continuoustimemodelformultipleyardcraneschedulingwithlastminute job arrivals. *International Journal of Production Economics*, 136:332–343, 2012.

[83] W. Li, Y. Wu, M.E.H. Petering, M. Goh, and R. de Souza. Discrete time model and algorithms for container yard crane scheduling. *European Journal of Operational Research*, 198:165–172, 2009.

[84] C. Liang, Y. Huang, and Y. Yang. A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Computers and Industrial Engineering*, 56:1021–1028, 2009.

[85] Ningbo Marine Group Company Limited. 2007/2008 analysis of china marine container transportation. `http://www.nbmarine.com/hangye/2007614130247.htm`, 2007.

[86] Orient Overseas Container Line Limited. About oocl uk. `http://www.oocl.com/uk/eng/localinformation/aboutoocluk/Pages/default.aspx?print=true&rsimg=`, 2011.

[87] Yantian International Container Terminals Limited. Terminal terms. `http://www.yict.com.cn/2006en/newcomers/terminalterms.asp`, 2006.

[88] C.K.Y. Lin and R.C.W. Kwok. Multi-objective metaheuristics for a location-routing problem with multiple use of vehicles on real data and simulated data. *European Journal of Operational Research*, 175:833–849, 2006.

[89] C. Liu, H. Jula, K. Vukadinovic, and P. Ioannou. Automated guided vehicle system for two container yard layouts. *Transportation Research*, Part C 12:349–368, 2004.

[90] Know The Web Ltd. Lorry driver salary. `http://www.mysalary.co.uk/average-salary/Lorry_Driver_1710`, 2011.

[91] M. M. Maloni and E. C. Jackson. North american container port capacity : A literature review. *Transportation journal*, 44:16–36, 2005.

[92] W. Ma and J. P. Kruth. Parameterization of randomly measured points for least squares fitting of b-spline curves and surfaces. *Computer-Aided Design*, 27:663–675, 1995.

[93] J. R. Macdonald. Impedance spectroscopy: Models, data fitting, and analysis. *Solid State Ionics*, 176:1961–1969, 2005.

[94] R.T. Marler and J.S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26:369–395, 2004.

[95] C. H. Martin. A hybrid genetic algorithm/mathematical programming approach to the multi-family flowshop scheduling problem with lot streaming. *Omega*, 37:126–137, 2009.

[96] A. Michalowicz and G. Vlaic. Multiple solutions in data fitting: a trap in exafs structural analysis and some ideas to avoid it. *Journal of Synchrotron Radiation*, 5:1317–1320, 1998.

[97] K. M. Miettinen. *Nonlinear multiobjective optimization*. Kluwer's International Series. Kluwer Academic Publishers, 1999.

[98] H. Min, H. J. Ko, and C. S. Ko. A genetic algorithm approach to developing the multi-echelon reverse logistics network for product returns. *Omega*, 34:56–69, 2006.

[99] M.Tamiz and D. F. Jones. Interactive frameworks for investigation of goal programming models: theory and practice. *Journal of Multi-Criteria Decision Analysis*, 6:52–60, 1997.

[100] J. Mula, D. Peidro, M. Daz-Madroero, and E. Vicens. Mathematical programming models for supply chain production and transport planning. *European Journal of Operational Research*, 204:377–390, 2010.

[101] J. Muller. Approximative solutions to the bicriterion vehicle routing problem with time windows. *European Journal of Operational Research*, 202:223–231, 2010.

[102] K. G. Murty, J. Liu, Y. Wan, and R. Linn. A decision support system for operations in a container terminal. *Decision Support Systems*, 39:309–332, 2005.

[103] W.C. Ng and K.L. Mak. Yard crane scheduling in port container terminals. *Applied Mathematical Modelling*, 29:263–276, 2005.

[104] E. Nishimura, A. Imai, and S. Papadimitriou. Yard trailer routing at a maritime container terminal. *Transportation Research Part E*, 41:53–76, 2005.

[105] T. Notteboom. Dynamics in port competition in europe implications for north italian ports. `http://www.irso.it/wp-content/uploads/Notteboom-workshop-Milan-18april2012.pdf`, 2012.

[106] W. Nunkaew and B. Phruksaphanrat. A multiobjective programming for transportation problem with the consideration of both depot to customer and customer to customer relationships. In *International MultiConference of Engineers and Computer Scientists 2009*, volume 2, 2009.

[107] J.P. Oddoye, D.F. Jones, M. Tamiz, and P. Schmidt. Combining simulation and goal programming for healthcare planning in a medical assessment unit. *European Journal of Operational Research*, 193:250–261, 2009.

[108] H. Park and X. Dang. *Design and Simulation-Based Optimization of Cooling Channels for Plastic Injection Mold*, chapter 2, pages 19–44. InTech, 2012.

[109] F. Parola and A. Sciomachen. Intermodal container flows in a port system network: Analysis of possible growths via simulation models. *International Journal of Production Economics*, 97:75–88, 2005.

[110] M. E. H. Petering, Y. Wu, W. Li, M. Goh, and R. de Souza. Development and simulation analysis of real-time yard crane control systems for seaport container transshipment terminals. *OR Spectrum*, 31:801–835, 2009.

[111] M. E.H. Petering and K. G. Murty. Effect of block length and yard crane deployment systems on overall performance at a seaport container transshipment terminal. *Computers and Operations Research*, 36:1711–1725, 2009.

[112] Adept Scientific plc. Micro saint. `http://www.adeptscience.co.uk/products/mathsim/microsaint/index.html`, 2012.

[113] Associated British Ports. Port of southampton master plan 2009. `http://www.hythe-hants.org.uk/PDF_Files/ABP_Master_Plan_2009.pdf`, 2009.

[114] Marine Department of the Government of the Hong Kong Special Administration Region. Ranking of container ports of the world. `http://www.mardep.gov.hk/en/publication/pdf/portstat_2_y_b5.pdf`, 2010.

[115] A. E. Rizzoli, N. Fornara, and L. M. Gambardella. A simulation tool for combined rail/road transport in intermodal terminals. *Mathematics and Computers in Simulation*, 59:57–71, 2002.

[116] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of multiobjective optimization*, volume 176 of *Mathematics in science and engineering*. Academic Press, Orlando, 1985.

[117] Y. Shi, X. Yang, Q. He, Q. Li, and Q. Xu. Anylogic-based simulation analysis of queuing system at container terminals. In *2nd International Conference on Information Science and Engineering*, pages 121–124, 2010.

[118] B. F. Smets, A. Jobbgy, R. M. Cowan, and Jr. C.P.L. Grady. Evaluation of respirometric data: Identification of features that preclude data fitting with existing kinetic expressions. *Ecotoxicology and Environmental Safety*, 33:88–99, 1996.

[119] M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. operations research. *Operations Research*, 35:254–265, 1987.

[120] DP World Southampton. Crane specifications. `http://www.dpworldsouthampton.com/portuserzone/specifications/cranes.htm`, 2008.

[121] DP World Southampton. Straddle carrier & other equipment specifications. `http://www.dpworldsouthampton.com/portuserzone/specifications/equipment.htm`, 2008.

[122] DP World Southampton. Dp world southampton announces new managing director. `http://www.dpworldsouthampton.com/mediazone/pressreleases/21102009.htm`, 2009.

[123] DP World Southampton. Press releases. `http://www.dpworldsouthampton.com/mediazone/pressReleases.htm`, 2009.

[124] R. Stahlbock and S. Voβ. Operations research at container terminals: A literature update. *OR Spectrum*, 30:1–52, 2008.

[125] D. Steenken, S. Voβ, and R. Stahlbock. Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, 26:3–49, 2004.

[126] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computations, and Application.* John Wiley and Sons, Inc., New York, 1986.

[127] H. Tamaki, H. Kita, and S. Kobayashi. Multi-objective optimization by genetic algorithms: a review. In *IEEE International Conference on Evolutionary Computation*, pages 517–522, 1996.

[128] K.C. Tan, C.Y. Cheong, and C.K. Goh. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research*, 177:813–839, 2007.

[129] K.C. Tan, Y.H. Chew, and L.H. Lee. A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research*, 172:855–885, 2006.

[130] Alion Science and Techology. Micro saint sharp official website. `http://www.microsaintsharp.com/`, 2011.

[131] A. Thesen and L.E. Travis. *Simulation for decision making.* West Pub. Co., 1992.

[132] M. Tzur and E. Drezner. A lookahead partitioning heuristic for a new assignment and scheduling problem in a distribution system. *European Journal of Operational Research*, 215:325–336, 2011.

[133] Department of Global Studies & Geography at the Hofstra University. World merchandise trade, 1960-2010. `http://people.hofstra.edu/geotrans/eng/ch5en/conc5en/worldexports.html`, 2010.

[134] I. Vacca, M. Bierlaire, and M. Salani. Optimization at container terminals:status, trends and perspectives. In *7th Swiss Transport Research Conference*, 2007.

[135] I. F.A. Vis and R. de Koster. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147:1–16, 2003.

[136] I. F.A. Vis, R.(M.) B. M. de Koster, and M. W. P. Savelsbergh. Minimum vehicle fleet size under time-window constraints at a container teminal. *Transportation Science*, 39:249–260, 2005.

[137] D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85, 1994.

[138] K.O. Willis and D.F. Jones. Multi-objective simulation optimization through search heuristics and relational database analysis. *Decision Support Systems*, 46:277–286, 2008.

[139] D. A. Wilson. Presentation to qbe management forum. `http://www.qbeeurope.com/documents/casualty/risk/forum/nov2008/QBE%20presentation%201%20Hutchison.pdf`, 2008.

[140] O. Yalcinkaya and G. M. Bayhan. Modelling and optimization of average travel time for a metro line by simulation and response surface methodology. *European Journal of Operational Research*, 196:225–233, 2009.

[141] C. Yang and N. Wang. A solution method for berth-quay crane allocation problem. In *2nd International Conference on Computer Engineering and Technology (ICCET)*, volume 2, pages 537–541, 2010.

[142] C. Yang, X. Wang, and Z. Li. An optimization approach for coupling problem of berth allocation and quay crane assignment in container terminal. *Computers and Industrial Engineering*, 63:243–253, 2012.

[143] F. Yoshimoto, T. Harada, and Y. Yoshimoto. Data fitting with a spline using a real-coded genetic algorithm. *Computer-Aided Design*, 35:751–760, 2003.

[144] F. Yoshimoto, M. Moriyama, and T. Harada. Automatic knot placement by a genetic algorithm for data fitting with a spline. In *International Conference on Shape Modeling and Applications*, pages 162–169, 1999.

[145] W. Y. Yun and Y. S. Choi. A simulation model for container-terminal operation analysis using an object-oriented approach. *International Journal Production Economics*, 59:221–230, 1999.

[146] Q. Zeng and Z. Yang. Integrating simulation and optimization to schedule loading operations in container terminals. *Computers and Operations Research*, 36:1935–1944, 2009.

[147] C. Zhang, Y. Wan, J. Liu, and R. J. Linn. Dynamic crane deployment in container storage yards. *Transportation Research Part B*, 36:537–555, 2002.

[148] H. Zhang and K. H. Kim. Maximizing the number of dual-cycle operations of quay cranes in container terminals. *Computers and Industrial Engineering*, 56:979–992, 2009.

[149] M. Zhu. Modeling and simulation of automated container terminal operation. *Journal of Computers*, 5:951–957, 2010.

[150] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms-a comparative case study. In Agoston E. Eiben, Thomas Bck, Marc Schoenauer, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature*, volume 1498, pages 292–301. Springer.