DOCTOR OF PHILOSOPHY

Development of a Constrained Fuzzy Knowledge-Based Optimisation System for the Management of Container Yard Operations in the Logistics Industry

Abbas, Ali

Award date:
2018

Awarding institution:
Coventry University

Development of a Constrained Fuzzy Knowledge-Based Optimisation System for the Management of Container Yard Operations in the Logistics Industry

Ali Abbas

Development of a Constrained Fuzzy Knowledge-Based Optimisation System for the Management of Container Yard Operations in the Logistics Industry

Ali Hadi Hussain Joma Abbas

A thesis submitted in partial fulfilment of the University's requirements for the Degree of Doctor of Philosophy

April 2018

Certificate of Ethical Approval

Applicant:

Ali Abbas

Project Title:

Development of a Constrained Fuzzy Knowledge-Based Optimisation System for the Management of Container Yard Operations in the Logistics Industry.

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

23 April 2018

Project Reference Number:

P70129

LIST OF PUBLICATIONS

Abbas A., Al-Bazi A., and Palade V. (2018). 'A Constrained Fuzzy Knowledge-Based System for the Management of Container Yard Operations'. *International Journal of Fuzzy Systems*, 20 (4), 1205-1223.

ABSTRACT

Owing to the many uncertainties involved, the management of container yard operations is very challenging**.** The storage of containers is one of those operations that require proper management to achieve efficient utilisation of the yard, short handling time and a minimum number of re-handlings.

The aim of this study is to develop a fuzzy knowledge based optimisation system named 'FKB_GA' for the management of container yard operations that takes into consideration factors and constraints that exist in real-life situations. One of these factors is the duration of stay of a container in each stack. Because the duration of stay of containers stored with pre-existing containers varies dynamically over time, an 'ON/OFF' strategy is proposed to activate or deactivate the duration of stay factor in the estimation of departure time if the topmost containers for each stack have been stored for a similar time period. A Genetic Algorithm model based Multi-Layer concept is developed which identifies the optimal fuzzy rules required for each set of fired rules to achieve a minimum number of container re-handlings when selecting a stack. The system was coded using Visual Basic for Applications (VBA) in MS Office Excel.

An industrial case study is used to demonstrate the applicability and practicability of the developed system. The proposed system has the potential to produce more effective storage and retrieval strategies, by reducing the number of re-handlings of containers. The performance of the proposed system is assessed by comparing with other storage and retrieval techniques including Constrained-Probabilistic Stack Allocation "CPSA" and Constrained-Neighbourhood Stack Allocation "CNSA".

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my deepest gratitude to my director of study, Dr Ammar Al-Bazi. I am truly thankful to him for his constant support and encouragement and for his valuable guidance. He has been to me as a friend. It has been a great honour for me to work under his supervision.

I am also grateful to my supervisors, Dr Vasile Palade and Dr Norlaily Yaacob for their valuable comments and discussion.

Special thanks to Mr Steve Parry and Mr Robert Nunnery for their support during data collection and designing of the container yard operations system.

Most importantly, I want to thank my dearest family. I owe my deepest gratitude to my mother and my father, for their selfless love, sacrifice and encouragement.

Finally, I am truly grateful to my wife, for her caring and encouragement during all stages of my study. I cannot thank her enough for her constant support.

TABLE OF CONTENETS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

With the growth in international container shipping worldwide owing to the offshoring of manufacturing, improving the operations in container terminals (Galle et al. 2017) has become a pressing problem.

Containers are shipped into a container terminal and stacked in a yard until requested by a customer (Misliah et al. 2012). A number of special resources are involved in the handling processes for containers such as: gantry cranes, trucks, straddle carriers, etc. (Misliah et al. 2012). A container yard is an important resource within a terminal which is used to store different sizes and types of containers (Bielli et al. 2006).

Containers are large boxes, ranging from 20 to 45 feet that are used for transporting goods from one destination to another. According to the goods being carried in them they come in different types, which include open sided, dry, open top, tank, and refrigerated (Ayachi et al. 2013a).

Container terminals contain a number of different areas each having their own operations. These areas are called rail side, container-yard side, and gate-side, the most important being the container-yard side where the storage and retrieval of containers takes place.

When containers arrive at the yard they have to be stored and when they leave they have to be retrieved. As the containers are stacked one on top of another, efficient storage and retrieval is required to prevent unnecessary re-handling. This is called the container storage problem.

Containers of different type, weight and size arrive by train and have to be unloaded and transferred to the yard until trucks come to transport them to the customer destination. The resource used to unload and transfer the container is called a reach stacker.

1.2 Container Yard Operations Management in the Logistics Industry

Container yard management can be considered to be one of the most complex tasks in the rail industry. This complexity is associated with the interaction of physical and informational constraints and the addition of a number of different resources associated with the exporting and importing of containers. Moreover, operational interaction among the different service processes taking place at the terminal increases the level of complexity (Holguín-Veras and Jara-Díaz 1999).

The management of container yard operations is a complex task due to the inherent uncertainties in the storage and/or retrieval operations of the containers. These uncertainties occur because the arrival of a truck to pick up the container is random, so the departure time of the container is unknown. Storing an import container on the top of another one which is due to go out of the yard first can lead to unnecessary handling by yard cranes which is a costly and time-consuming operation (Ting et al. 2010).

However, another important challenge is the efficient stacking of containers which is of crucial importance for the effective execution of the remaining terminal operations. The efficiency of the stacking operations requires both strategic and operational decisions. Strategic decisions are needed to determine the stacking equipment and yard layout required and operational decisions are needed to achieve efficient container stacking and the scheduling and routing of the stacking equipment (Vis and de Koster 2003). These decisions usually have to be made with respect to the available space, the planned container throughput, the expected container-dwell time, the planned yard utilisation as well as external regulations concerning customs control, environmental protection and occupational safety. In an ideal container terminal, this information is available, enabling effective planning for the stacking of containers. Several constraints also need to be taken into account which includes: the maximum stacking height (i.e., the maximum number of tiers) which further depends on the stacking equipment used and the weight of the containers.

1.2.1 Storage and Retrieval Operations of Containers

Containers are stacked on the ground in the yards and are usually separated into blocks that consist of bays, rows and tiers. Most container terminals form separated blocks according to the attributes of the containers. Two ways of storing containers in the yards can be distinguished. Firstly, containers may be stored on a chassis, which enables direct access to each individual container. Secondly, containers may be stacked on the ground and piled up. Hence, not every single container is directly accessible for retrieval. In order to get access to containers that are stored below others, the upper ones have to be re-handled, which means that they have to be re-located to other storage locations (Vis and de Koster 2003).

1.2.2 Problems Faced in Storage Operation of Containers

The container storage operation is a very important task for achieving efficient utilisation of container yards. Proper storage operation leads to a reduction of the container yard operations cycle time (Zehendner et al. 2017).

Storage space allocation and location assignment are two of the problems that occur in storage operations when containers have an unknown departure date/time (Chen and Lu 2012).

The storage space allocation problem involves the assignment of containers to a block or a bay, while the location assignment problem involves the allocation of containers to stacks (Zhen et al. 2013). The location assignment problem is quite complex because of the uncertainty regarding which container departs first.

However, another important challenge making the stack allocation complex, is that the duration of stay for containers in the yard varies according to the departure time which is sometimes unknown when the stacking operation is performed.

When the duration of stay of import containers is short, the stack allocation process will be simpler, thus the storage and retrieval operations will be less complex and the storage requirement in the yard will be low. A number of researchers have considered the storage space allocation problem when the duration of stay is short. (e.g. (Kim and Kim 1999, Saurí and Martín 2011)).

Because the storage operation is less complex, most terminals around the world offer "free time" and only when the duration of stay is longer than this will a charge be made for container storage.

If the duration of stay is long the higher will be the requirement for storage and the stack allocation for the storage and retrieval operations for containers will be more complex. Stack allocation will be more complex the longer the longer the duration of stay of the container in the yard. This is explained in more detail in the problem identified in the next section.

1.3 Research Problem

Trains arriving at a container yard are loaded with containers of many different types, sizes and weights. When the train arrives at the container yard, the containers need to be unloaded, transported and then stored in the yard. Containers have to be stored in stacks, one on top of the other. These containers are handled by one shared resource called a reach stacker. The reach stacker is used for the storage and retrieval operations of containers in the yard. The retrieval operation might happen during the storage operation. When these two operations are required as the same time, the allocation operation will then be stopped (i.e. terminated) and the retrieval operation will be carried out, because the retrieval operation has priority over the storage operation. Once the retrieval operation is completed, then the storage operation will be resumed. The problem comes when containers staying in the yard for a long time are stacked on top of other containers which need to depart the yard first. If this happens then unnecessary movements of the containers on the top need to take place, to access the container to be despatched. This is more likely to happen when the departure time for containers already in the stack is unknown, and these containers stay for a long time (i.e. maximum of a one-month period), before a notification is sent to customers. This can happen when customers arrange for the collection of their containers by the 3PL companies without any prior notice being given to the yard operators which makes the storage and retrieval operations more challenging. Most of the yard operators are happy for containers to stay for a long time providing customers pay a pre-defined daily storage fee.

When containers are allowed to stay in the yard for a long time, the consideration of the duration that a container spends in the yard is crucial as it relates to its departure time (i.e. the longer the time the container spends in the yard, the greater is the chance of the container leaving). When the departure time is uncertain, identifying the best stacks in which to store containers, by considering only the duration of stay is a challenging task. This task becomes even more challenging when the topmost containers have been stored for roughly the same time. In this case, further factors, including the similarity between containers (i.e. destined for the same customer, etc.), and the number of containers per stack, will need to be taken into account to optimise the storage operation, and subsequently, reduce the number of re-handlings. See figure 1.1 for a schematic representation of the problem.

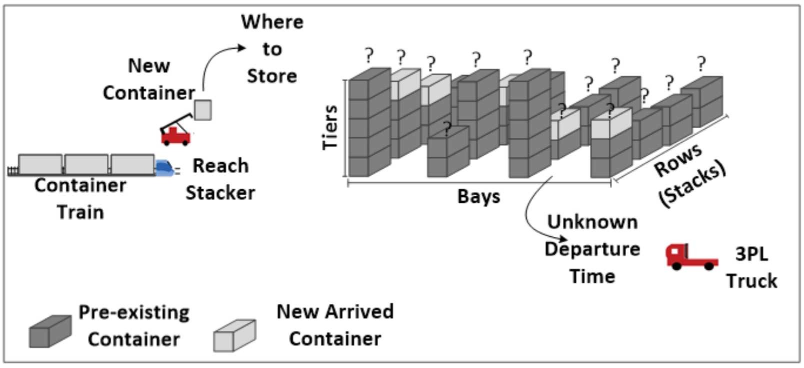


Figure 1.1: Schematic representation for the layout of a pre-existing container yard

A fuzzy knowledge based optimisation system is proposed to model the fuzzy aspect associated with each of the three storage factors both individually and combined. Then, the optimisation process identifies which fuzzy rules allocated per stack should be selected to achieve the best allocation plan for containers. This contributes, in turn, to improving the retrieval operation and minimising the total number of re-handlings for containers.

1.4 Research Question

Will fuzzy knowledge based genetic algorithm system improve the container yard operations in terms of minimising the total number of re-handlings of containers?

This research question is defined in order to clarify the focus of the research. This research question provides an essential path for developing a system to enable the management of container yard operations to achieve the minimum number of re-handlings for containers. This leads to the minimum retrieval time of containers in yards.

1.5 Research Aim

The aim of this research is to develop a fuzzy knowledge based optimisation system to improve the storage and retrieval operations for containers under real life constraints such as a long duration of stay and an unknown time of departure.

This system will then be used to generate an optimal/near optimal allocation plan for the allocation of containers to stacks by considering three fuzzy real life storage factors and a number of constraints. This optimal allocation plan should achieve the minimum number of re-handlings for containers.

1.6 Research Objectives

In order to deliver the aim of the research, the following objectives are established:

1. To conduct a comprehensive review of literature in the area of container yard operations management. This includes approaches and methodologies that are developed to solve different container storage problems in the presence of a number of real life constraints.
2. To understand and then model the logic of the container terminal operations focussing on the storage and retrieval operations of container yard sites.
3. To collect the required data and identify the sources of fuzziness in the container yard operations.
4. To simulate the current container yard operations for a better understanding of these operations, including the capture of container arrival and departure events.
5. To develop a fuzzy knowledge based model for the proper allocation of containers to stacks taking into consideration three fuzzy real life factors and other deterministic constraints.
6. To develop an optimisation module embedded in the developed fuzzy knowledge based model for optimising the allocation plans for the allocation of containers to stacks.
7. To verify and validate the simulated model.

1.7 Research Tools and Techniques

A combination of techniques and tools are utilised to deliver each of the set objectives. These are:

1. A literature review is used to satisfy objective 1.
2. Visual modelling languages such as flowcharts, UMLs including Use Case, Activity, Class, and Sequence Diagrams are used to satisfy objective 2.
3. On-site visits, structured interviews and a stop watch are used to satisfy objective 3.
4. Discrete Event Simulation (DES) is used to simulate the container yard operations and arrival of container trains to satisfy objective 4.
5. A fuzzy knowledge based approach is used to satisfy objective 5
6. Genetic Algorithms (GAs) are used to minimise the total number of re-handling of containers to satisfy objective 6.
7. A real life case study is used to verify and validate the developed system, which includes different yard constraints together with different scenarios for train arrival to satisfy objective 7.

The first technique used in this research was to identify the gap in knowledge for studies related to the storage and retrieval operations of containers, in yards containing pre-existing containers with an unknown departure time. The second technique was used to understand the storage and retrieval operations of containers in the yard. This was presented by developing flowcharts to illustrate the sequence of storage, retrieval and transportation operations and logical movement of containers in the yard. In addition, specification, and data flow diagrams for the container-yard operations-management system were developed. The specification diagrams were developed to provide a system architecture with a visual language for specifying, constructing and documenting the artefacts of the software system. The user interfaces were visually modelled in order to describe the behaviour of the system in response to user interactions. Data flow diagrams were developed for the storage and retrieval operations of containers to clarify how the data was moving between models and profiles in the computer system. The required data and processing times for the storage and retrieval operations of containers in the yard were identified by the third technique. The logic for the storage and retrieval operations of containers data was captured by conducting two onsite visits. The container yard operation times were identified by using a stopwatch tool. The fourth technique was used to simulate train arrivals, container storage and container-retrieval operations, which will provide a better understanding of these processes in the yard sites. To achieve improved yard storage operations for containers with an unknown departure time, three fuzzy real-life factors (i.e. the number, similarity, and duration of stay of containers per stack) were taken into consideration by the fuzzy knowledge based modelling used as the fifth technique. This technique was used to allocate stacks for containers in both the storage and retrieval operations in the yard. The Fuzzy Knowledge Based model consisted of a number of membership functions including the fuzzy sets for each of the input/output factors, together with the crisp constraints. Each input factor was modelled using three fuzzy sets. This technique assessed each stack in the yard by providing an acceptability level value (based on the three input factors described above) for the container storage operation. The container will be stored in the stack that has the highest acceptability level value. The sixth technique was used to optimise the storage and retrieval operations for containers in the yard. This technique was utilised to select optimal/near optimal rules from a set of fuzzy fired rules for each container and stack, which will select the best stacks for the container storage and retrieval operations, and, in turn, reduce the total number of re-handlings of containers during the retrieval operation. The final technique was used to verify and validate the system developed for the container yard operations which can be used by yard operators. In this verification and validation process, different container yard scenarios were considered.

1.8 Research Deliverables

The deliverables of this research are summarised as follow:

1. A comprehensive review of literature for container terminal operations management.
2. A number of logical diagrams including storage, retrieval, transportation operations and container flow diagrams reflecting the flow of container terminal operations, including container yard operations management.
3. A collection of data for process logic and operation times relating to each area of container yard operations.

4. A model that simulates the current container terminal operations focusing on storage and retrieval of containers.
5. A fuzzy knowledge based model incorporating a number of fuzzy storage factors for the most effective container yard operations management given on unknown departure time for containers.
6. A Genetic Algorithm model embedded into the fuzzy knowledge based model above to optimise the allocation plan for allocating containers to stacks.
7. A verified and validated storage and retrieval operations management system.

1.9 Benefits to Academia

The proposed methodology provides two contributions to academia:

- As a theoretical contribution, a new direction and innovative research framework to model, then handle the complexity of the current container-yard operations management problem.
- As a practical contribution, real life scenarios including busy, moderately busy and quiet yard conditions, together with the rushed and slow arrival of containers by train are investigated in detail.

1.10 Benefits to the Logistics Industry

The proposed Fuzzy Knowledge Based Genetic Algorithm ‘FKB_GA’ system is designed to plan the allocation of containers to stacks to achieve optimal/ near optimal storage and retrieval operations in the yard.

The computerised system is designed to be used by container yard operators in order to optimise the storage plan for containers with a long duration of stay and where the departure time is unknown.

1.11 Research Scope

The scope of this research is limited to stack allocation during storage and retrieval operations of import containers in container yard management systems. A set of related data is identified and collected from a container yard at one of the container terminals in the UK. This research considers a yard with containers where the departure times are unknown. These containers have a long duration of stay in the yard which makes the stack allocation process for the storage and retrieval operation of containers more challenging.

1.12 Research Motivation

This section discusses the motivation that led to the undertaking of this PhD study. There is a dearth of research on the problem of the management of container yard operations when there are pre-existing containers in the yard with an unknown departure time.

Parameters affecting this problem such as: the duration of stay of containers; the number per stack; the customer affiliations; and the high level of ambiguity inherited by the behaviour of their departure, have not been studied in depth.

There is a need to provide a computerised system to assist container yard operators when dealing with this problem and the effects of these different parameters.

1.13 Research Output Generalisability

The generic outputs of the research are as follows:

- The system could be used by any container yard operators that have pre-existing containers with a long duration of stay in the yard and an unknown departure time.
- The proposed 'ON/OFF' strategy in this study could be applied to other logistics applications such as the warehousing operations of manufacturing companies.
- The developed GA is applicable for warehousing operations which utilise multiple stacks for items in their storage areas.
- The developed system was shown to be ideal for quiet and busy yard scenarios where noticeable improvements were obtained.

1.14 Thesis Structure

This thesis is presented in three main parts: the identification and review of the container storage and retrieval modelling approaches, the development of a container yard operations methodology and the design of a prototype model to discover the benefits of the proposed system.

The current chapter includes the general background of systems for container yard operations, the problem definition, aims and objectives, tools and techniques, deliverables, benefits and the scope of the research.

Figure 1.2 shows the structure of the thesis and how chapters are interrelated with each other.

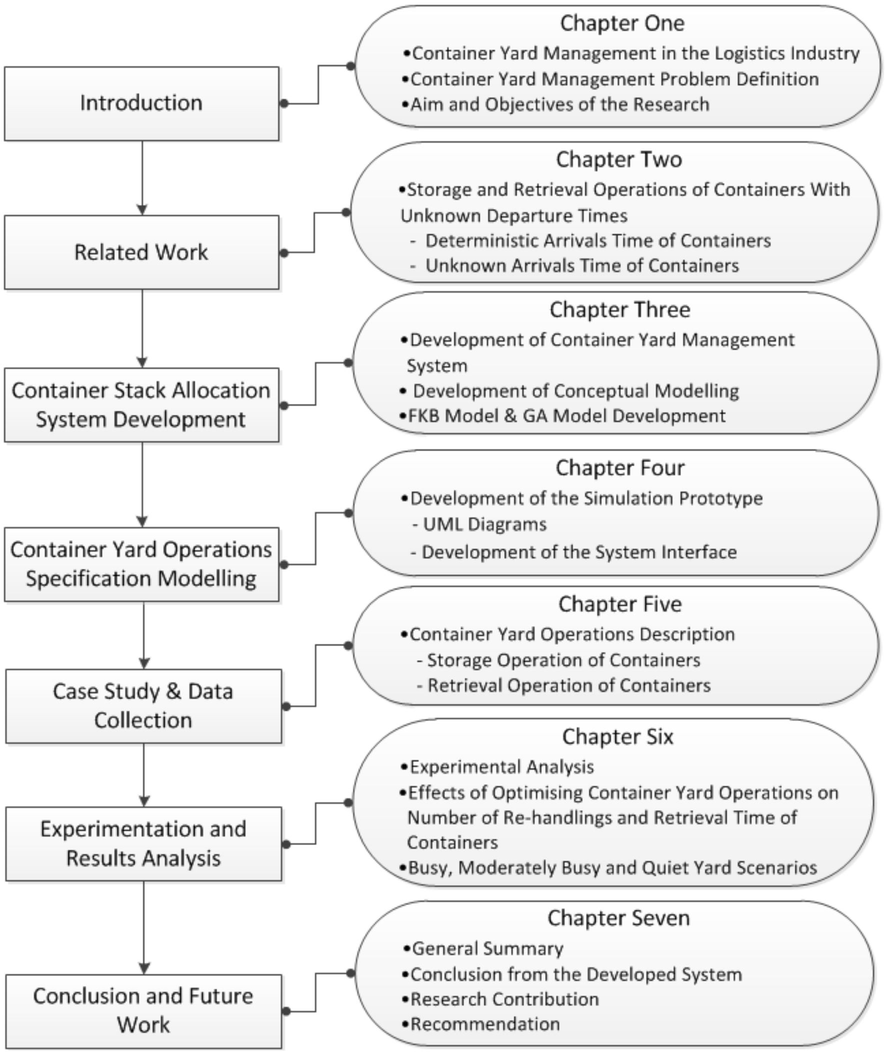


Figure 1.2: Thesis Structure Chart

The remainder of this thesis has been divided into the following chapters.

Chapter Two: demonstrates a number of tools and techniques used in container yard operations management. The outputs from a number of researchers were used to identify tools and techniques that have already been developed and used to solve the container yard operations problems. A critique is provided to show the gaps in current container yard operations techniques, which require further innovation and research.

Chapter Three: demonstrates the development of the proposed container yard operations management system. The framework of the system is presented and discussed in detail. Conceptual modelling is introduced as a system modelling tool. This methodology is used to present the development processes of the container yard operations management system. The development process of the fuzzy knowledge based module is discussed in detail as one of the components of the container yard operations management system. The development process for an ON/OFF strategy is presented which enables the activation and deactivation of one of the input factors of the fuzzy knowledge based module. A neighbourhood algorithm developed for the container re-handling operation is presented. In addition, the process for developing the optimisation model is presented and the multi-layer concept introduced.

Chapter Four: presents the structure of the prototype for the simulation of container yard operations. UML diagrams are presented for description of the system. A system interface development for the user to access and update the prototype.

Chapter Five: describes the container yard operations management. The collected data is presented in order to demonstrate how the yard operations are carried out. The operations, including the storage and retrieval of containers are described in detail.

Chapter Six: illustrates the experimental part of the developed stack allocation system for container yard operations management. The effects of optimising container yard operations on a number of re-handling and retrieval times for containers are indicated by analysing busy, moderately busy and quiet yard scenarios together with the rushed and slow arrival of containers train. A comparison between the ‘Current’, ‘FKB’, and ‘FKB_GA’ storage and retrieval approaches is outlined to prove the concept of the proposed allocation methodology. A comparison study is made with “CPSA” and “CNSA” as storage and retrieval techniques to evaluate the performance of the proposed stack allocation system using a genetic algorithm (GA).

Chapter Seven: summarises the research work. The conclusions drawn from the research, advantages and disadvantages of the adopted methodology, limitations of the research and future recommendation are presented.

1.15 Chapter Summary

The container yard operations in the logistics industry have been defined. The problem relating to the container yard operations management have been explained in detail. The aim and objectives together with the scope and deliverables of this work have been described. Research problem, research question and problems faced in storage operation of containers are presented. Benefits to academia and logistics industry have been addressed as well in this chapter. Both motivation and generic outputs of this research have been explained in this chapter. A thesis structure was outlined and a brief introduction for each chapter has been given.

The next chapter addresses the literature on previous work for container yard operations including the storage and retrieval operations.

CHAPTER 2

PREVIOUS LITERATURE ON CONTAINER STORAGE AND RETRIEVAL OPERATIONS

2.1 Introduction

In this chapter, a summary of methodologies, strategies and contributions of previous studies of container terminal operations management are presented. These studies investigate storage and retrieval operations for containers of a different type, size, width and weight, which are subject to different real life factors and constraints including the unknown departure time of containers and short and long term durations of stay for pre-existing containers. They form the main basis and starting point for the proposed solutions in this thesis.

In the next section, the literature review examines the main publications on the topic of storage and retrieval operations for containers with an unknown departure time.

2.2 Storage and Retrieval Operations with Unknown Departure Time

In this section, the literature for container yard storage and retrieval operations for containers with unknown departure times in both static and dynamic environments is presented. These are classified into containers with same/similar and different arrival times. The purpose of this classification is to explore the impact of the container arrival pattern on the behaviour of its departure. A static environment means the arrivals and departures of containers at yards are fixed (Tang et al. 2015). While a dynamic environment means the arrivals and departures of containers at yards are changing during the storage and retrieval operations (Tang et al. 2015).

2.2.1 Unknown Departure of Containers with Deterministic Arrival Times

A deterministic arrival time means that the arrival time of containers brought by trains or vessels is known in advance (Budipriyanto et al. 2017). A number of studies have investigated this type of behaviour including (Ries et al. 2014) who developed a fuzzy logic based rule model for the storage space allocation problem of containers at a seaport terminal. The main goal was to assign a proper stack location to every incoming container in order to minimise the relocation ratio of containers that depart at random times. Each container was arriving every 5 minutes and departing at times in the future. The fuzzy logic approach was implemented in two phases within a static environment. The first phase was the selection of the block and the second phase was the selection of the stack. A set of criteria was considered in the model which included: the distance from the block to the gate, the block utilisation, the stack height, and the difference in the estimated dispatch time between the newly arrived and topmost containers in the stack. The fuzzy logic model showed a good performance with a low variability for the results obtained. The containers in this paper were assumed to have the same size and type. However, grouping these containers based on other specific attributes, such as number of containers stored at each stack, duration of stay or customer, was not considered by the model.

(Yang et al. 2015) developed a Multi-Objective Integer Programming Model (MOIPM) for solving the container Stacking Position Determination Problem (SPDP) with a duration of stay of a few days run. The unloading of inbound containers with a deterministic arrival time and a random departure time were considered, specifically focusing on the Storage Space Allocation Problem (SSAP) in container terminals. The objective of the model was to increase the container circulation, reduce unbalanced workloads, and reduce the movements of the yard crane. A Genetic algorithm model was developed in order to solve the SPDP problem. The results showed the effectiveness and the robustness of the model and its contribution to the operational efficiency of stack yard operations in container terminals. In the storage process, the model did not take into consideration the weight, size, type or duration of stay of the containers, or the stack height.

(Jin et al. 2004) developed an intelligent neural network based on fuzzy logic for the scheduling of static container yard operations when the arrival time of containers was known in advance. This model included system status evaluation, operation rule and stack height regulation, and operation scheduling to improve the storage of containers, when the container departure times were random and the container stay time was only of a few days. The aim of the model was to reduce the total ship waiting time and total operation time. The model did not take into consideration the weight, size, type or inter-arrival time of containers or the grouping of containers by customer.

A Genetic Algorithm model was developed by (Ayachi et al. 2013a) for optimising the static storage space allocation for import and export containers with known arrival times. The objective of the model was to minimise the re-handling operations and to organise efficiently those containers that were only staying for a short time in the storage space. A number of techniques, as well as a mathematical formula were utilised and some experiments were run to solve the problem. The experimental results confirmed that the method could solve the Storage Space Allocation Problem (SSAP) and showed good results when compared with the Last-In-First-Out (LIFO) rule applied to the same problem. Different types of containers with the same size were considered, however the stack height, and the inter-arrival time of containers, which influence the storage operation, were not considered in the developed model.

(Ayachi et al. 2010b) proposed a Genetic Algorithm model to optimise the storage positions for containers of different types with known arrival times where the delivery times were random / probabilistic. The objective of the model was to identify a static optimal storage plan for containers with a short duration of stay so as to reduce the number of re-handlings and increase the likelihood of meeting the customer delivery deadline. The approach was compared with a Last-In-First-Out (LIFO) rule applied to the same problem and showed good results. Although size, type and the random delivery times for containers were taken into account, other criteria relevant to the storage process, including stack utilisation, grouping containers by customer, or the duration of stay of containers in the yard were not considered.

(de Castillo and Daganzo 1993) used mathematical functions based on a uniform distribution to analyse both segregation and non-segregation strategies for the static container storage problem when the arrival time of containers was deterministic and departure time was random. The main aim was to reduce the handling effort for containers with a duration of stay based on the number of ship arrivals. In the segregation strategy, cargoes from different ships were separated into stacks of different sizes, while in the non-segregation strategy all stacks were kept at the same size. It was concluded that the segregation strategy presented better solutions for the least busy terminals, whereas the non-segregation strategy reduced the operating cost for the busier terminals. Neither of the strategies considered containers of different size and type, departure times or the grouping of containers by customer.

(Kim and Kim 1999a) further improved on the segregation strategy proposed by (de Castillo and Daganzo 1993) by using mathematical difference equations where the size and type of containers were assumed to be the same. This method considered containers with a duration of stay from 3 to 6 days, taking into account arrival patterns (including constant, deterministic, cyclic, and dynamic), and the number of containers stored in each bay in order to achieve efficient storage operation. The containers were assumed to depart randomly in this study. The equations used both an approximate as well as an exact method. The average percentage absolute error of the approximate method compared with the exact method was only 1.88%. The same strategy was explored further by the same authors in 2002, when they studied how to optimise the space needed for a given number of containers. The authors did not consider different container departure times or the grouping of containers by customer.

(Huynh 2008) introduced two stacking methods, mixed and non-mixed, to solve the storage operation problem in static environments. This is to determine whether or not newly arrived containers were stacked on top of containers with known arrival times and unknown departure times. These methods evaluated the effect of container duration of stay on storage policies based on a number of criteria, such as imported throughput, storage density, and re-handling productivity. The Monte Carlo simulation method was used to estimate the expected number of re-handlings of containers. The effect of a short duration of stay on the re-handling productivity was shown by comparing the amount of import deliveries with the total amount of import moves. However, containers of a different weight, size, and type were not considered.

(Saurí and Martín 2011) proposed a mathematical model for both static and dynamic environments based on probability distribution functions to achieve optimal storage for containers with a duration of stay in the yard of 3 to 4 days, while minimising unnecessary the movement of containers. The arrival and departure rates for containers, storage yard characteristics, probability of containers leaving the terminal, and the relationship of that probability to the deterministic inter-arrival time were taken into consideration while developing the optimisation model. The study showed that the choosing of the proper strategy depends on the stack height; inter arrival time of vessels, and the duration of stay for containers. Space allocation criteria such as the grouping of containers based on customer, or stack height, were not taken into account in this approach.

(Kim 1997) suggested a number of mathematical regression equations to solve the static container stacking problem which was used to estimate the total number of re-handlings required to pick up from the bay containers with random departure times. The main variables considered in the regression equations were: the number of containers, the number of rows and the distribution of stacking heights in the bay. All the containers in this study were dispatched after a short period of stay time.
The results showed that the mathematical equations performed better than the Index of Selectivity (IOS) method in evaluating the number of re-handlings for containers during the retrieval process. Kim concluded that the total number of re-handlings for retrieving a container was dependent on the total number of containers in the yard. With this approach, no attention was given to stack utilisation, grouping of containers by customer or the duration of stay of containers.

(LAN et al. 2001) compared ordered and random stacking strategies for the static assignment of the correct slot for 150 containers with a short duration of stay in the yard. The strategies were used to simulate the stacking of containers with both known and unknown departure times in both single and twin storage areas given deterministic arrival times. The aim of the comparison was to establish the number of unproductive movements of containers for each strategy. It was found that the total number of unproductive moves decreased as the ratio of containers with a known departure time increases. The authors did not consider the grouping of containers based on either duration of stay or customer in the proposed strategies.

(Liu et al. 2010) introduced a fuzzy-based optimisation model for optimising the storage space allocation process for containers with deterministic arrival times and uncertain departure times. The purpose of the model was firstly to minimise the unbalanced workloads between yard blocks and secondly to minimise the number of blocks to which a group of containers were split. A planning horizon method was implemented to tackle the storage problem where the number of containers was dynamically changing. The model considered containers with a duration of stay of a few hours and an unknown departure time, as well as the storage of different groups of containers in the same block and stack. The results showed that using the proposed model, the balance of the workload for the total number of containers in the planning horizon improved by 40% compared to the terminal performance and the unbalanced workloads between yard blocks are improved by 11.6% to 44.5% to the terminal performance. The proposed model did not take into account either the stack utilisation or the arrival time of containers into the storage area.

(Zhang et al. 2003) discussed a rolling horizon approach for improving the storage space allocation process of containers in a static environment with known arrival times and unknown pick up times. The total number of planning periods in a planning horizon was 18 hours. The main aim of the proposed approach was to minimise both the workload in the storage yard blocks and the total distance required to transport the container between the storage blocks and the vessel berthing location. The containers were assumed to be only one size and type in this work. Mathematical programming models were utilised to solve the problem. The experimental results confirmed that the approach could provide efficient and effective solutions to the storage space allocation problem. The duration of stay for containers or the utilisation for each stack in the yard was not considered.

(Junqueira et al. 2016) studied the static problem of storage space assignment for containers by developing a simulation-based genetic algorithm to optimise storage rules representation. The purpose of the algorithm was to reduce the amount of unnecessary movement of containers with a short duration of stay in the yard. To test various sequences and search for the best solution in terms of movements, a Genetic Algorithm model employing simulation was developed for the evaluation of its individuals. The results obtained with showed that with low computational time it was possible to obtain good sequences of feasible movements. Although containers of different types with predictable arrival times were considered in this paper they were all the same size and had random departure times. Stack height and the duration of stay of containers were not considered by the authors.

However, if containers have arrived collectively at the same time, their duration of stay starting from day 1 in the yard will be similar. This will make the storage decisions even more complex in case this factor is considered.

2.2.2 Unknown Departure of Containers with Random Arrival Times

This section discusses the most relevant approaches for providing solutions to the problem of storing or stacking containers with a random time of arrival and departure. A random arrival time means the exact time when the containers arrive at a yard is unknown (Stahlbock and Voβ 2008). One of these approaches represented by a reward-based algorithm was developed by (Ozcan and Eliiyi 2017) for solving the outbound static container stacking problem where the storage time of containers was restricted to a few days. The aim of the proposed algorithm was to minimise the number of re-handlings of containers as well as the crane travelling time in the yard. The distance between the containers and the closest gantry crane, gantry crane workload, number of stacked containers in neighbourhood bays, and the current height of the stacks at the storage area were considered in the proposed algorithm to support the storage operation. In this study, different sizes, weights, types of containers along with the Expected Departure Time (EDT), and Port of Destination (PoD) for containers were also taken into account while developing the algorithm. The results seemed to be promising when compared to the current randomised stacking strategy in the container terminal. However, the inter-arrival times between containers in the yard to estimate the length of stay for each container was not considered.

A static mathematical model was proposed by (Woo and Kim 2011) to reduce the number of reservations (i.e. clusters) for each export container group when allocating storage space in a terminal where the duration of stay for containers was less than a week. In this work, two principals were considered for the allocation of space; the first principle was that containers in the same group should be stored close together (i.e. in the same bay). By grouping containers in the same bay, the yard crane travel distance could be minimised. The second principle was that different groups of containers could not be mixed in the same stack. The results showed that the model was able to provide the best solution for determining storage space for export containers in container terminals. Although the grouping of containers considered container size, port destination, stack height and the probability distribution of container departures, allocating space according to specific criteria such as type, weight and stay time of containers was not considered.

A heuristic algorithm for remarshalling inbound and outbound containers was presented by (Ayachi et al. 2013c). This algorithm was adopted to remedy the uncertainties related to departure times for containers by decomposing the day into a specific number of timeslots. In the event of a change to the expected departure time for a container in a dynamic environment, this algorithm tried to assign the container to a different location taking into account the new departure time with a short duration of stay in the yard. The objective of this heuristic was to find an optimal container storage plan which respected the departure time and reduced the re-handling of containers. The experiments confirmed that the heuristic algorithm reduced the re-handling operations. Although considering different types of containers, the size, stack utilisation and stay time were not considered by this algorithm.

(Ku and Arthanari 2016) proposed a stochastic dynamic programming model to calculate the minimum number of expected reshuffles for containers. Relocated containers were given different departure time windows with an assumed duration of stay of only a few days. The model incorporated a search-based algorithm in a tree search space, together with an abstraction heuristic. The heuristic, called the "expected reshufflings index" (ERI), was defined as the expected number of containers that depart earlier than the container being reshuffled to the column. The ERI heuristic chose the column with the lowest ERI as the target column for the reshuffled containers. Between 30%–40% reduction in the average number of reshuffles was achieved by the proposed ERI heuristic compared to the random selection method. The model did not consider the storage of containers based on their actual duration of stay in the yard.

(Tang et al. 2015) studied the reshuffling of containers in both static and dynamic environments, where the departure time of containers were random, based on an exponential distribution with a short range of stay time in the yard. These problems were studied to minimise the number of container reshuffles to improve the yard operations. For the static reshuffling problem, an improved model was formulated and a number of effective heuristics and extensions were developed and their performance analysed. For the dynamic problem with continual arrivals and retrievals of containers, the different heuristics of the static environment were applied and tested, and a simulation model was developed with an animation function to show the stacking, retrieving and reshuffling operations. The experimental results showed that the improved model obtained feasible and optimal solutions more quickly than the existing models. Also, in both the static and dynamic cases the extended versions of the proposed heuristics were either superior to, or similar to, the best results of the existing heuristics and consumed very little time. Although containers of the same size and type were considered, reshuffling and stacking criteria taking into account either the grouping of containers or stack height was not taken into account.

(Park et al. 2011) suggested an online search algorithm for improving the container stacking operation in a dynamic environment for containers with an assumed duration of stay of one week. The algorithm included storage and retrieving operations for incoming containers in container yard blocks, where the arrival and departure times were random, based on Gaussian distributions. The algorithm was used to reduce the re-handling and retrieval time for containers, which were grouped according to size or weight. Experimental results confirmed that the proposed algorithm was used successfully to determine the stacking positions of containers. The utilisation for each stack or the actual duration of stay of containers was not considered in this algorithm.

(Casey and Kozan 2012) developed a mathematical model for the static container storage problem. The model was developed to minimise both the number of re-handling moves and the total job times by keeping the number of containers in a stack as low as possible. A number of heuristic algorithms were used to produce good initial solutions for the storage problem for containers with a short duration of stay in the yard. A meta-heuristics algorithm was also developed to improve on these solutions. Results from the various heuristics applied to a few case study problems were compared and discussed. Containers of the same size and type where the departure time was unknown was considered, however, the model did not consider the duration of stay or the grouping of containers by customer.

(Zehendner et al. 2017) introduced an algorithm with performance guarantee for the Online Container Relocation Problem (OCRP) where the retrieval sequence of containers is revealed over time. The algorithm aimed to minimise the number of relocations of containers with a short duration of stay in the yard. A so-called levelling heuristic using the perspective of worst-case competitive analysis of online algorithms and derive its competitive ratio. Some computational experiments were provided which gave insights on the actual average performance of the heuristic. Although the number of containers per stack was considered by the algorithm, however, duration of stay and similarity of containers were not considered.

In this case, duration of stay of each arrived container will be different as they arrived at random times. This will assist for easier storage decisions especially when the duration of stay is considered in such decisions.

2.3 Relevant Literature Review

The most relevant study that was used for this research was (Ries et al. 2014) who developed the fuzzy logic based rule model for the storage space allocation problem which reduces the relocation ratio for stacking containers. A set of criteria was considered in the model which included: the distance from the block to the gate, the block utilisation, the stack height, and the difference in the estimated dispatch time between the newly arrived and topmost containers in the stack. The model provided acceptability level values of stacks for container storage operation. The containers were stored in the stacks with the highest acceptability level values. These acceptability level values were provided based on the selected fuzzy rules from the rule base.

However, this research considers the selection of active rules from the fired fuzzy rules for each stack, for each container.

Saurí and Martín (2011) proposed a mathematical model based on probability distribution functions to achieve optimal storage, while minimising the unnecessary movement of containers. The arrival and departure rates for containers, storage yard characteristics, the probability of containers leaving the terminal, and the relationship of that probability to the inter-arrival time were considered during the optimisation process. The model considered containers with a short duration in the yard. However, this research considered containers with a longer duration of stay in the yard which makes the storage and retrieval operation of containers more challenging. This challenge appears when the duration of stay of containers is long, because containers with a long duration of stay have more chance of departing the yard which makes the stack allocation process for container storage complex.

Liu, Kang, and Zhou (2010) introduced a fuzzy-based optimisation model for solving the storage space allocation problem. The purpose of the model was firstly to minimise the unbalanced workloads between yard blocks and secondly to minimise the number of blocks to which the same group (similarity) of containers were split. The model considered both containers with an unknown departure time, as well as the storage of different groups of containers in the same block and stack.

In addition, fuzzy-knowledge-based and mathematical models were previously used by researchers for the storage operation of containers, such as Ries et al. (2014), who used a fuzzy logic model with the number of containers per stack factor, Saurí and Martín (2011), used a mathematical model with a number of containers per stack factor, but only for a short duration of stay; and Liu, Kang, and Zhou (2010), used a fuzzy model with the similarity of containers per stack factor.

However, this research developed a Fuzzy Knowledge Based model which takes into consideration all three factors (i.e. the number, similarity, and duration of stay for containers in a stack), collectively, but, for a longer duration of stay, to improve the management of container-yard operations.

Although fuzzy-knowledge-based models were previously used to select the fuzzy rules from the rule base (Nelles et al. 1996, Carmona et al. 2004, Cintra and de Arruda 2007, Shill et al. 2015). genetic algorithms (GA) were utilised in this research to tune and finally select the optimal/near optimal rules from the fired fuzzy rules, by removing those that might reduce system performance.

2.4 Gap in Knowledge

In this section, the research contribution is identified by comparing a wide range of tools and techniques that have been used to solve the management of different container yard operation problems. The techniques reviewed above were focussed on containers with a short duration of stay in the yard. Only a few of them considered the individual effect of storage factors in the stack allocation process in yards. See table 2.1 for a summary of the storage factors considered in the literature.

Table 2.1: The storage factors considered in previous researches

Author(s)	Factors (Criteria)			This Research
	Number of Containers in each Stack	Similarity of containers in each Stack	Duration of Stay of Containers	In this research, all three factors combined together are considered while modelling storage and retrieval operations of containers in the yard.
(Ries et al. 2014)	✓			
(Jin et al. 2004)	✓			
(Woo and Kim 2011)	✓	✓		
(Saurí and Martín 2011)			✓	
(Liu et al. 2010)		✓		
(Ozcan and Eliiyi 2017)	✓	✓		
(Casey and Kozan 2012)	✓			
(Zehendner et al. 2017)	✓			

As shown in table 2.1, (Ries et al. 2014, Jin et al. 2004, Casey and Kozan 2012, Zehendner et al. 2017) considered the number of containers per stack factor while solving the container storage space allocation problem. The main aim was to reduce the number of container re-handlings during the retrieval operation. However, other real life factors such as the similarity of containers per stack and/or the duration of stay of the topmost containers per stack have not been considered in their works. These factors could contribute in reducing the number of re-handlings (i.e. relocation) of containers, as some of these containers might be stored on top of containers for different customers and / or those that have a longer duration of stay in the yard which eventually increases the number of re-handlings. In (Ries et al. 2014, Jin et al. 2004), the number of containers in each stack factor was considered as fuzzy variables in their study. While in (Casey and Kozan 2012), the number of containers was considered as a deterministic factor.

Both the number and similarity (e.g. based on customer) of containers for the storage operation were considered by (Woo and Kim 2011, Ozcan and Eliiyi 2017), with the aim of minimising the number of re-handlings for containers. However, these works did not take into consideration the duration of stay factor that influences the number of re-handlings during the retrieval operation of containers. When the duration of stay of containers is not considered, the containers may be stored in stacks of one or more containers that have been stored for a long time but need to depart. As discussed earlier, the longer the stay time, the higher the probability that the container will need to depart. The number of containers and similarity of containers factors were considered as deterministic variables in both (Woo and Kim 2011, Ozcan and Eliiyi 2017).

The duration of stay factor for the storage space allocation problem was considered only by (Saurí and Martín 2011) to reduce the number of re-handlings of containers. However, the authors did not consider either the number or similarity of containers factors. In addition, while the duration of stay factor was considered for the storage operation of containers, the containers would still be stored on top of those that have a shorter duration of stay in the yard. These containers might also be stored in stacks that have either a high number of containers or less similarity, or both. All of these possibilities could lead to an additional number of re-handlings for containers. The duration of stay of containers was considered as a stochastic factor in the work of (Saurí and Martín 2011).

While (Liu et al. 2010) considered only the similarity of containers factor, the number of containers per stack and/or the duration of stay factor have not been taken into consideration for minimising the number of re-handlings. When the number of containers and/or duration of stay factors are not considered during the storage operation, containers might be stored in stacks that have a higher number of containers which will result in a higher number of re-handlings in the retrieval process. (Liu et al. 2010) considered the similarity of containers factor as a deterministic variable. The gap in knowledge of this research is summarized as follows:

- In previous work (Ries et al. 2014), the number of containers per stack factor was considered for container storage by using a fuzzy logic model. This model was used for the storage and retrieval operations to assess the stacks in the yard. The model provided an acceptability level value for each stack based on the number of containers per stack factor. The stack was allocated based on the highest acceptability level value for the container storage operation. However, this research, for the storage and retrieval operations, considers containers with long durations of stay and the use of a fuzzy technique to both predict the likely departure, and to assess the effect of other factors (e.g. the number and similarity of containers in each stack).
- The duration of stay factor was previously used by other researchers (Saurí and Martín 2011) for the storage operation of containers. This factor was considered even when there were only slight differences between the duration of stay of containers in the yard. This research, however, considers continuous variations in the duration of stay of containers over time and introduces an 'ON/OFF' strategy to activate/ deactivate the duration of stay factor, if the length of stay varies significantly over time.
- In a study by Shill et al (2015), rules for container storage were selected from a rule base using GAs. This study, however, for each container stack, used the GA for selecting the optimal/near optimal rules from a set of fired fuzzy rules,

for each stack where a container may be stored. This will lead to the selection of an optimised acceptability level value of a stack for the container storage operation.

- A number of scenarios were considered in a previous study by Ries et al. (2014), including: busy, moderately busy and quiet yards. This research, however, investigates in detail these scenarios but also takes into account two different periods of arrival for the container trains (i.e. rushed and slow).

Hence, this study presents an innovative fuzzy knowledge based optimisation system named ‘FKB_GA’, which was specially developed for the efficient storage and retrieval of containers taking into account a number of realistic factors, including the container duration of stay factor.

To further clarify the gap in knowledge, a number of tools and techniques that have been previously used to solve the container storage problem when the departure time for containers is unknown are analysed. See table 2.2 which shows different tools and techniques from the related previous works which include fuzzy logic models, genetic algorithms, heuristics, fuzzy logic based genetic algorithms, mathematical models, mathematical model based genetic algorithms, and mathematical model based heuristic algorithms.

Table 2.2: Summary on the techniques used in related works

Technique	Author(s)	Problem Type (Static/Dynamic)	Container Staying Time Length Short/Long Run of Container Stay in Yard
Fuzzy Logic, FKBM	(Ries et al. 2014)	Static	Short
GA (Genetic Algorithm)	(Ayachi et al. 2013a)	Static	Short
	(Ayachi et al. 2010b)	Static	Short
	(Junqueira et al. 2016)	Static	Short
Heuristics	(Huynh 2008)	Static	Short
	(LAN et al. 2001)	Static	Short
	(Ozcan and Eliiyi 2017)	Static	Short
	(Ayachi et al. 2013c)	Dynamic	Short
	(Tang et al. 2015)	Static & Dynamic	Short
	(Park et al. 2011)	Dynamic	Short
	(Zehendner et al. 2017)	Static	Short
FKBM Based GA	(Jin et al. 2004)	Static	Short
	(Liu et al. 2010)	Dynamic	Short
Mathematical Model	(Woo and Kim 2011)	Static	Short
	(de Castillo and Daganzo 1993)	Static	Short
	(Kim and Kim 1999a)	Static & Dynamic	Short
	(Kim and Kim 2002b)	Static & Dynamic	Short
	(Saurí and Martín 2011)	Static & Dynamic	Short
	(Kim 1997)	Static	Short
	(Zhang et al. 2003)	Static	Short
	(Ku and Arthanari 2016)	Dynamic	Short
Mathematical Model Based GA	(Yang et al. 2015)	Static	Short
Mathematical Model Based Heuristic Algorithm	(Casey and Kozan 2012)	Static	Short

Table 2.2 presents tools and techniques that have been used in modelling the storage operation of containers. The technique developed by (Ries et al. 2014) for the space allocation of containers in a static environment used a fuzzy logic based rules model to reduce the relocation ratio. A set of criteria was considered in the model to support the storage operation which included: the distance from the block to the gate, the block utilisation, the stack height, and the difference in the estimated dispatch time between the newly arrived and topmost containers in the stack. In the work of (Ries et al. 2014), the containers had a short duration of stay in the yard.

A Genetic Algorithm model was developed by (Ayachi et al. 2013a) for optimising the storage positions of import and export containers in a static environment, with a short duration of stay, to minimise the number of re-handling operations.

(Ayachi et al. 2010b) proposed a Genetic Algorithm model to optimise the positions of containers in a static environment whilst taking into consideration the container type. The objective of the optimisation was to reduce the number of re-handlings for containers that had a short duration of stay in the yard.

(Junqueira et al. 2016) used a Genetic Algorithm to improve static storage rules for containers in a static environment with a short duration of stay. The purpose of the algorithm was to reduce the amount of unnecessary movement of containers.

(Huynh 2008) introduced two methods including mixed and non-mixed for the storage space allocation problem in a static environment. The non-mixed method does not allow containers from different ships to be mixed, containers from each ship being located in a specific storage area or block whereas the mixed method allows containers from one ship to be stacked on top of containers from another ship. The purpose of these methods was to evaluate the number of re-handlings for containers with a short duration of stay.

(LAN et al. 2001) compared ordered and random stacking strategies for the assignment of the correct slot for containers in a static environment. The strategies were used to simulate the stacking of containers with both known and unknown departure times in both single and twin storage areas. The aim of this comparison was to establish which of the two strategies had the smallest number of re-handlings for containers with a short duration of stay in the yard.

(Ozcan and Eliiyi 2017) proposed a reward-based algorithm for solving the storage allocation of outbound containers in a static environment where the duration of stay for containers was only a few days. The distance between the containers and the closest gantry crane, gantry crane workload, number of stacked containers in neighbourhood bays, and the current height of the stacks at the storage area were considered in the proposed algorithm. Different sizes, weights, types, Expected Departure Time (EDT), and Port of Destination (PoD) for containers were also taken into account. The aim of the proposed algorithm was to minimise the number of re-handlings of containers as well as the crane travelling time in the yard.

A heuristic algorithm for the dynamic remarshalling of inbound and outbound containers was presented by (Ayachi et al. 2013c). In the event of a change to the expected departure time for a container, this algorithm tried to assign the container to a different location taking into account the new departure time. The objective of this heuristic was to find an optimal container storage plan which respected the departure time and reduced the re-handling of containers when the duration of stay was short.

(Tang et al. 2015) studied the stacking of containers in both a static and dynamic environment. These problems were studied to minimise the number of container re-handlings where the duration of stay was short. For the static environment re-handling problem, an improved model was formulated and a number of effective heuristics and extensions were developed and their performance analysed. For the dynamic environment re-handling problem with continual arrivals and retrievals of containers, the different heuristics of the static environment were applied and tested, and a simulation model was developed with an animation function to show the stacking, retrieving and re-handling operations.

(Park et al. 2011) suggested an online search algorithm for improving the container stacking operation in a static environment where the duration of stay was one week. The algorithm was used to reduce the re-handling and retrieval time for containers, which were grouped according to size or weight.

(Zehendner et al. 2017) introduced an algorithm for online container relocation problem to improve the retrieval operation in a static environment where the duration of stay was a few days. This algorithm was introduced to minimise the number of relocations of containers

(Jin et al. 2004) developed an intelligent neural network based on fuzzy logic for the scheduling of containers in a static environment to improve the storage of containers in a yard. The aim of this development was to reduce the operation time of containers with a duration of stay in the yard of only a few days.

(Liu et al. 2010) introduced a fuzzy-based optimisation model for optimising the storage space allocation process in a dynamic environment by considering the number of fuzzy variables for a container. The purpose of the model was firstly to minimise the unbalanced workloads between yard blocks, and secondly to minimise the number of blocks to which the same group of containers were split where the duration of stay was only a few hours.

A mathematical model was proposed by (Woo and Kim 2011) to reduce the number of reservations (i.e. clusters) for each export container group when allocating storage space in a container terminal. In this work, two principles were considered for the allocation of space in a static environment when the duration of stay in the yard was less than a week. The first principle was that containers in the same group should be stored close together (i.e. in the same bay). By grouping containers in the same bay, the yard crane travel distance could be minimised. The second principle was that different groups of containers could not be mixed in the same stack.

(de Castillo and Daganzo 1993) introduced non-segregation and segregation strategies. The non-segregation strategy allowed containers from one ship to be stacked on top of containers from another ship. The segregation strategy did not allow containers from different ships to be mixed. The containers from each ship were located in a specific storage area or block. The main aim was to reduce the handling effort for containers in a static environment where the duration of stay was based on the number of ship arrivals.

(Kim and Kim 1999a) further improved on the segregation strategy proposed by (de Castillo and Daganzo 1993) by using mathematical difference equations when the size and type of containers were assumed to be the same. This method took into account constant, cyclic, and dynamic arrival patterns for containers with a duration of stay of 3 to 6 days, to achieve efficient storage operation. Kim and Kim (2002b) studied further explored the segregation strategy to determine how to optimise the space needed for a given number of containers.

(Saurí and Martín 2011) studied how to mix the containers arriving in different batches (ships), where no more than two container ships were mixed in the same block or bay. The purpose of this study was to minimise the unnecessary movements of containers for both static and dynamic environments. The duration of stay for containers was assumed to be between 3 to 4 days.

(Kim 1997) suggested a number of mathematical regression equations to solve the static container stacking problem which was used to estimate the total number of re-handlings required to pick up containers with random departure times from the bay. The main variables of the formulation were: the number of containers that were removed with a short duration of stay, the number of rows and the distribution of stacking heights in the bay.

(Zhang et al. 2003) discussed a rolling horizon approach for improving the storage space allocation of containers in a static environment where the planning periods in a planning horizon was 18 hours. The main aim of the proposed approach was to minimise both the workload in the storage yard blocks and the total distance required to transport the container between the storage blocks and the vessel berthing location.

(Ku and Arthanari 2016) proposed a stochastic dynamic programming model to calculate the minimum number of expected re-handles for containers by assuming randomness for container departures were random for containers booked in the same time window where the time window was short.

(Yang et al. 2015) investigated and analysed the process for positioning inbound containers based on an advanced handling strategy which assigned quay cranes utilizing internal trucks on a “Full In & Full Out” basis, where intense loading and unloading activities were considered at the same time. A Genetic algorithm model for a static environment was developed in order to optimise the position of containers with a few days’ duration of stay.

(Casey and Kozan 2012) developed a mathematical model for a static environment to optimise the storage of containers with a short duration of stay in the yard. The model was developed to minimise both the number of re-handlings and the total job times by keeping the number of containers in a stack as few as possible.

Although the literature above has presented various optimal allocation techniques for containers with unknown departure times, especially for containers with a short duration of stay, none of them have considered containers which stay longer, or the fuzzy storage factors and their effects in the long term on the storage and retrieval plans. And hence, this study presents an innovative fuzzy knowledge based optimisation system named ‘FKB_GA’, which is specially developed for the efficient container storage and retrieval operations taking into account a number of realistic factors including the container duration of stay.

As described earlier, how containers arrive in the yards can influence the storage operation. A summary of previous works where the arrival time of containers was deterministic and the departure time was unknown is presented in table 2.3.

Table 2.3: The summary of the previous work (deterministic arrival time of containers)

Author(s)	Contribution in Container Yards
(Ries et al. 2014)	Develop a fuzzy logic model for storage space allocation of containers at a seaport terminal with different scenarios (Empty yard, 50% busy of yard, & 80%).
(Yang et al. 2015)	Apply container stacking position determination problem
(Jin et al. 2004)	Consider container yard operations problem.
(Ayachi et al. 2013a)	Produce storage space allocation problem of import and export containers.
(Ayachi et al. 2010b)	Propose the container storage space allocation problem.
(de Castillo and Daganzo 1993)	Examine storage strategies for inbound containers with already stacked ones (Segregation and non-segregation).
(Kim and Kim 1999a)	Apply a new procedure for estimating the expected total number of re-handles (Segregation).
(Kim and Kim 2002b)	Study how to optimise the space needed for the given container volume (further explored the segregation strategy)
(Huynh 2008)	Introduce methods to evaluate the effects of storage policies on import throughput and re-handling productivity (Mixed and non –mixed).
(Saurí and Martín 2011)	Study stacking strategies of containers, how to mix containers arriving in different batches.
(Kim 1997)	Discuss evaluating the re-handles number of containers in a yard.
(LAN et al. 2001)	Study slot assignment problem of containers in a yard.
(Liu et al. 2010)	Tackle storage space allocation problem.
(Zhang et al. 2003)	Study storage space allocation problem.
(Junqueira et al. 2016)	Investigate problem of storage space assignment to containers in a port yard.

Table 2.3 presents the related works when containers arrive at the yards by ship or train considering different criteria for storage and retrieval operations. For example, the arrival time of containers was assumed to be constant by (Ries et al. 2014), however, the departure time of these containers was random. In (Yang et al. 2015), containers arrived by ship at the same time but departed at uncertain times. While containers arrived at a similar time in (Jin et al. 2004), they departed in a range of

random operation days. In (Ayachi et al. 2013a) all containers arrived at a terminal at the same time, but the departure time of the containers was unknown. Although random delivery times were taken into consideration by (Ayachi et al. 2010b), the inter-arrival time of container ships were constant. In (de Castillo and Daganzo 1993), the arrival time of containers by ship was constant, while the departure time of containers was random, based on a uniform distribution. (Kim and Kim 1999) assumed that containers were arriving at the same time by ship, but the containers were assumed to depart randomly. (Huynh 2008) considered that containers arrived at the same time in vessels with a constant inter-arrival time, however, the departure time of these containers was unknown. In (Saurí and Martín 2011)Sauri and Martin (2011), the inter-arrival time of container ships was constant but the departure time was random, based on a triangular distribution function. While, in (Kim 1997), containers were unloaded from ships that arrived at known times but departed randomly. Containers arrived by a container ship at the same time in (LAN et al. 2001), however, the departure date/time was unknown. Similar inter-arrival times of container vessels was assumed by (Zhang et al. 2003), however, the departure time of the containers was unknown. Finally, (Junqueira et al. 2016) addressed the problem of containers with a similar arrival time but an unknown departure time.

When the containers arrive at the yards at deterministic and similar times, storing these containers based on duration of stay will be inefficient as all these containers will have the same duration of stay, so predicting when the containers will depart for the storage operation will be difficult.

For summary of the previous work when the arrival of containers is random and the departure time is unknown, see table 2.4.

Table 2.4: The summary of the previous work (random/unknown arrival time of containers)

Author(s)	Contribution in Container Yards
(Ozcan and Eliiyi 2017)	Propose the outbound container stacking problem.
(Woo and Kim 2011)	Present allocating storage space problem of export containers.
(Ayachi et al. 2013c)	Study remarshalling problem of container in a yard.
(Ku and Arthanari 2016)	Study container reshuffling problem during retrieval process.
(Tang et al. 2015)	Propose static reshuffling problem and the dynamic stacking problem in container yards.
(Park et al. 2011)	Consider container stacking operations in an automated container terminal.
(Casey and Kozan 2012)	Discuss the container storage problem in yards.
(Zehendner et al. 2017)	Introduce the online container relocation problem.

In table 2.4, (Ozcan and Eliiyi 2017) presented the container stacking problem, where both the arrival and departure time of containers was unknown. Again, in (Woo and Kim 2011), the inter-arrival time of container vessels was random and the departure time of containers was random. (Ayachi et al. 2013c) also addressed uncertainties related to the arrival and departure dates of containers. (Ku and Arthanari 2016) discussed both random arrival of containers and departure time for containers. (Tang et al. 2015) assumed containers were arriving and departing randomly based on an exponential distribution. (Park et al. 2011) tackled random arrival and departure times for containers based on Gaussian distributions. (Casey and Kozan 2012) considered random inter-arrival time for containers based on an exponential distribution, and random departure time for these containers obtained from a triangular distribution. Finally, (Zehendner et al. 2017) introduced the online container relocation problem where the information on arrival and departure of containers was uncertain.

When the containers arrive at the yards at different times, considering duration of stay in the storage operation will be more efficient because it will be different for each container. However, by considering both the duration of stay as well as the number and similarity of containers in a stack, as factors for the decision as to where a container will be stored, will produce more efficient storage and retrieval operations, as the duration of stay of containers varies over time. When the duration of stay of containers is either similar or the same, then only the number and similarity of containers per stack need be considered for the storage and retrieval operations.

However, when the duration of stay of containers is different, by consideration of the duration of stay factor, together with the number of containers and the similarity of containers in the stack, will lead to more efficient storage and retrieval operations and yard management.

The deterministic departure time for containers has been investigated in container yard management in previous work, see Appendix A.

For finding the best allocation in the yard for containers with known departure times (deterministic), while minimising the number of re-handlings, a number of techniques were used. (Wang et al. 2014) developed an algorithm for storage space allocation in railway container terminals to reduce the numbers of re-handling (i.e. overlapping) inbound containers. This algorithm searched for slots to store incoming containers based on their departure time, the incoming containers being stored on containers with a later departure time. (Sriphrabu et al. 2013) developed a simulation model for the container stacking problem in a terminal with a known retrieval sequence. The simulation model was developed to reduce the total lifting time and increase the service efficiency of the container terminals. A Genetic Algorithm was applied in order to assign an appropriate location for containers based on their retrieval sequence. For container yard optimisation, a mixed integer programming model was presented by (Türsel Eliiyi et al. 2013) which determined locations of export containers in the yard according to their assigned vessels and destination.

The Fuzzy Knowledge Based (FKB) model developed in this research considered collectively the duration of stay and the number and similarity of containers in a stack for the storage and retrieval operation. The requirement to involve the duration of stay factor in the stacking decision is dynamic as the duration of stay for the topmost containers in each stack varies over time. Stacks will only be identified as suitable for the newly arrived container if the differences in the durations of stay for containers in the stack are small. If not, the FKB model only uses the number and similarity of containers to make the required storage decision.

As an optimisation engine, a genetic algorithm model is developed and embedded into the fuzzy knowledge based rules model to select the optimal/near optimal rules from a set of fired fuzzy rules for each container and possible stacks in the yard for storage and retrieval operations.

From the review above, it has been clearly shown and justification has been given for the development of a new approach to solving the problems associated with the management of the container yard. A fuzzy knowledge based optimisation system for solving the stack allocation problems of containers with an unknown departure time is proposed. The developed system will be able to identify optimal/ near optimal allocation plans of containers with a long duration of stay for pre-existing containers.

An influential real life storage factor such as duration of stay together with other real-life constraints will be considered in order to optimise dynamic storage-retrieval operations.

2.5 Chapter Summary

In this chapter, storage and retrieval operations of containers in yards with unknown departure time in previous studies were presented. In addition, the literatures in this chapter were categorised based on arrival behaviour of containers including the deterministic arrivals and random arrivals. The reason behind this categorising was to find out how the containers were stored taking into consideration these arrivals behaviour. Different tools and techniques used in previous work in this area were presented as well. A gap in knowledge was identified after a comprehensive review in the area of container yard operations management. The gap was bridged by considering collectively influential fuzzy real-life factors for storage and retrieval operations of containers with long duration of stay in yards. Reviewing related work was useful for identifying tools and techniques for container storage and retrieval operations of containers.

The following chapter describes the tools and techniques used in this research in details.

CHAPTER 3

DEVELOPMENT OF THE CONTAINER YARD MANAGEMENT SYSTEM

3.1 Introduction

In this chapter, the fuzzy-knowledge-based rules technique is introduced to solve the container-stack allocation problem where the departure time is unknown.

This technique is also used to model the re-handling operation of containers during the collection (retrieval) operation. A Genetic Algorithm (GA) model is proposed and embedded into the fuzzy-knowledge-based rules model, for selection of the best fired fuzzy rules for each possible stack in the yard. Fired rules mean the rules whose conditions are verified to a degree greater than zero are firing when an input is applied to a fuzzy system (Kacprzyk and Pedrycz 2015: 226). For comparison purpose, a heuristic algorithm called 'neighbourhood' is used to model the re-handling operation of containers. In addition, a 'random' algorithm is used for modelling both storage and retrieval operations. The Discrete Event approach is also utilised to mimic the events of arrival, resource status, storage, retrieval and departure of containers.

The theoretical aspects of each component of the container yard management system are discussed. These aspects include the development of the model specification, fuzzy based rules model and optimisation model. In the next section, the theoretical framework of the Fuzzy-Knowledge-Based GA system 'FKB_GA' is presented.

3.2 Rationalise the selection of Fuzzy Approach

In this section, the rationalise of selection of fuzzy approach for storage and retrieval operations of containers is summarized as shown in table 3.1

Table 3.1: The Retional of Selection of Fuzzy Approach

Factor	**Source of Fuzziness**	**The Rational of Using Fuzzy Approach**
No. of Containers per stack.	The number of containers to be picked up (depart) from each stack over a period of time is uncertain and hence, was considered as a fuzzy variable.	In order to handle these fuzzy factors, a fuzzy knowledge based approach was used for the stack allocation process during the storage and retrieval operations for containers in the yard.
Similarity of containers per stack.	As far as the similarity of containers related to the number of containers stored in, or departing from each stack, hence it was considered as one of the fuzzy variables of the developed 'FKB_GA' system.	
Duration of stay of containers per stack.	The duration of stay of the topmost containers was considered as a fuzzy variable due to the fact that it relates directly to their locations. These locations were continuously changing in response to the rapid retrieval operations for containers that need to depart at unknown times, and hence there is no deterministic pattern for the duration of stay of the topmost containers.	

3.3 The 'FKB_GA' System Framework

This section presents the framework for the system. The system framework is comprised of the input, process and output components. The input component consists of the specification information along with storage factors, GA information and other related factors and constraints information. The process component involves a collection of integrated techniques including fuzzy-knowledge-based rules and genetic Algorithms that work together to process the inputs. Finally, the output component includes a number of key performance indicators which are categorised based on the operational criteria and yard criteria. Yard utilisation can be considered as one of the yard criteria for terminal throughput evaluation. See figure 3.1 which shows a diagram of the system framework

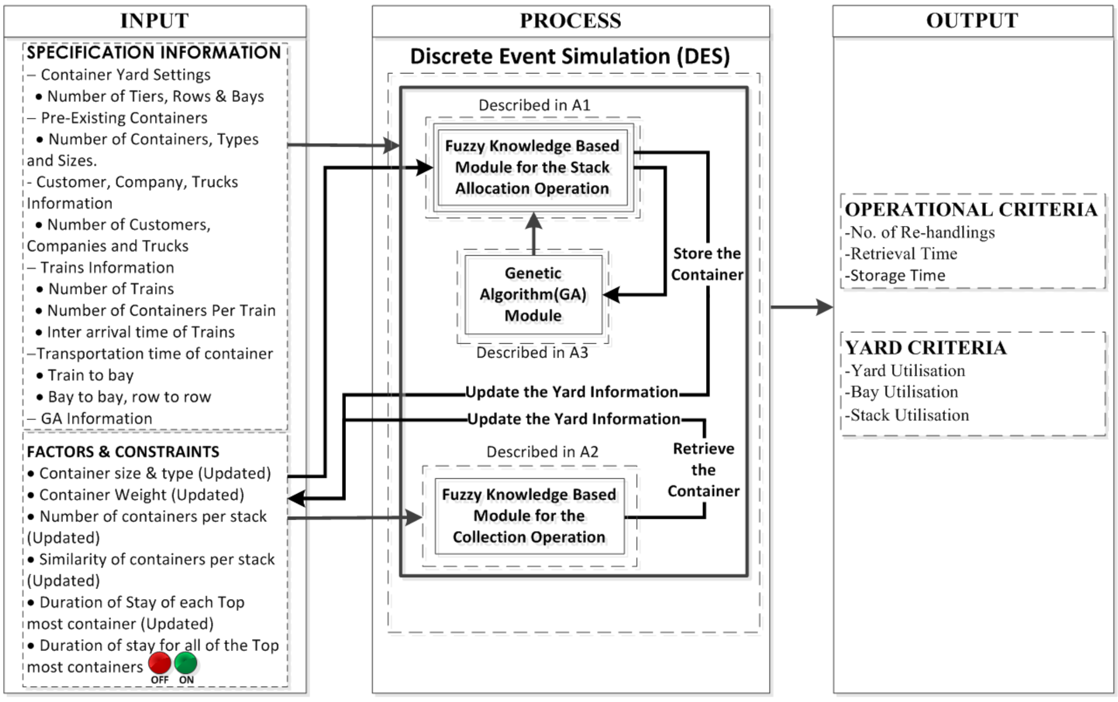


Figure 3.1: The 'FKB_GA' System Framework

The inputs, process, and output presented in the system framework (figure 3.1) are introduced and discussed as follow:

- **Inputs**

In the input component, the specification information including the container yard definition, together with the details for pre-existing containers; trains; transportation time (content shown in the framework above), and finally, the GA model information, are fed into the system. The GA information consists of population size, probabilities of crossover and mutation and stopping criterion.

The storage factors and constraints information are fed into both the storage and retrieval/ collection modules.

These factors include the number of containers in each stack, the duration of stay (i.e length of time the topmost container has stayed in each stack) and the similarity of containers in each stack (i.e. containers which belong to the same customer). The constraints are container size, type, and weight (empty or full). Information regarding the duration of stay for all the topmost containers collectively is also fed into the storage and retrieval modules. As the duration of stay of containers changes dynamically over time, it becomes less influential in the storage process when the duration of stay for all the topmost containers of stacks are similar. Hence, a decision is required on whether or not to consider this factor in subsequent processing.

- **Process**

The process component is comprised of two modules which include the fuzzy-knowledge-based (FKB) module and the genetic algorithm (GA) module. The process starts when the specific information is fed into the FKB module and the factors and constraints information is fed into both the storage and retrieval modules to be processed. Using the input above, the module determines (i.e. allocates) the stack in which to store the container. It achieves this by firing a number of fuzzy rules per stack then calculating an acceptability level (α_i) for each stack. The GA is then introduced to temporarily select some of the active rules out of the fired fuzzy rules for each incoming container and the possible stacks in which they could be stored, providing the activated rules for de-fuzzification to re-calculate the acceptability level values of the stacks (α_i). The stack that has the highest acceptability level value is the optimal stack and is then allocated to store the incoming container. The container is stored in the allocated stack and the yard information including factors and constraints will be updated accordingly. In addition, this update takes place when a retrieval operation is complete and the required container is despatched.

The output module includes operational criteria and yard criteria as can be seen in figure 3.1.

In the next section, each of the techniques defined in the 'FKB_GA' system are explained in more detail.

3.4 The FKB_GA' Conceptual Model

The conceptual model consists of what is to be modelled and how it will be modelled, providing information on tools and techniques, together with the relationship between them and the data to be processed (Pace 2000). The conceptual model is used here to provide a clear understanding of the target domain and the problem faced by container yard operations management, representing both the structural and behavioural features of the 'FKB_GA' system. This section provides a detailed discussion on the techniques used in developing the system. The FKB model together with Genetic Algorithm is used to identify the optimal/near optimal storage strategy for containers with an unknown departure time.

In the 'FKB_GA' system, the FKB module assesses the location to store the incoming container by using fuzzy reasoning taking into account certain factors and constraints, and subsequently assigns an acceptability level of storage value (α_i) to each stack. The acceptability level of storage (α) is the output from the module, which is an arbitrary value that reflects the value of the current stack in the decision process. This arbitrary value is defined as the acceptability level of an incoming container to the stack i (α_i). For every stack i available in the container yard, a value (α) is generated based on the input factors and constraints. The acceptability level allows for the assessment of the most suitable stack location for the incoming container. The stack that has the highest acceptability level value will be allocated to store the new container.

Inputs from the container yard operation are regarded as crisp inputs, which need to be fuzzified using fuzzy sets, represented by their respective membership functions, in order to apply the FKB module. The fuzzy inference component which includes aggregation, will manipulate the given information in fuzzy format according to fuzzy rules. The fuzzy output will then be de-fuzzified using one of the methods (Zadeh 1965, Zimmermann 2001) to calculate the acceptability level value (i.e. crisp value) of each stack (α_i), to be used for the allocation of incoming containers. The stack with the highest acceptability level value will then be used for container storage, while simultaneously satisfying all inputs and conditions. Once the container is stored, the system updates the yard information for the next incoming container. After the fuzzy rules have been assigned for all, the storage operation is then optimised using the GA module. This module holds all the fired fuzzy rules for each incoming container for all the possible stacks on which it can be stored, then releases them for the optimisation process. The GA will then temporarily select some of the rules out of all the possible fired fuzzy rules for each stack, providing the selected rules for de-fuzzification to re-calculate the acceptability level values of the stacks (α_i). The stack that has the highest acceptability level value is the optimal stack and will be allocated to store the incoming container.

The proposed GA model selects the optimal/near optimal fuzzy rules from all the fired rules per stack to achieve the minimum number of container re-handlings. This reflects the learning process of the system to achieve its total number of re-handlings objective. See figure 3.2 for the 'FKB_GA' system core components.

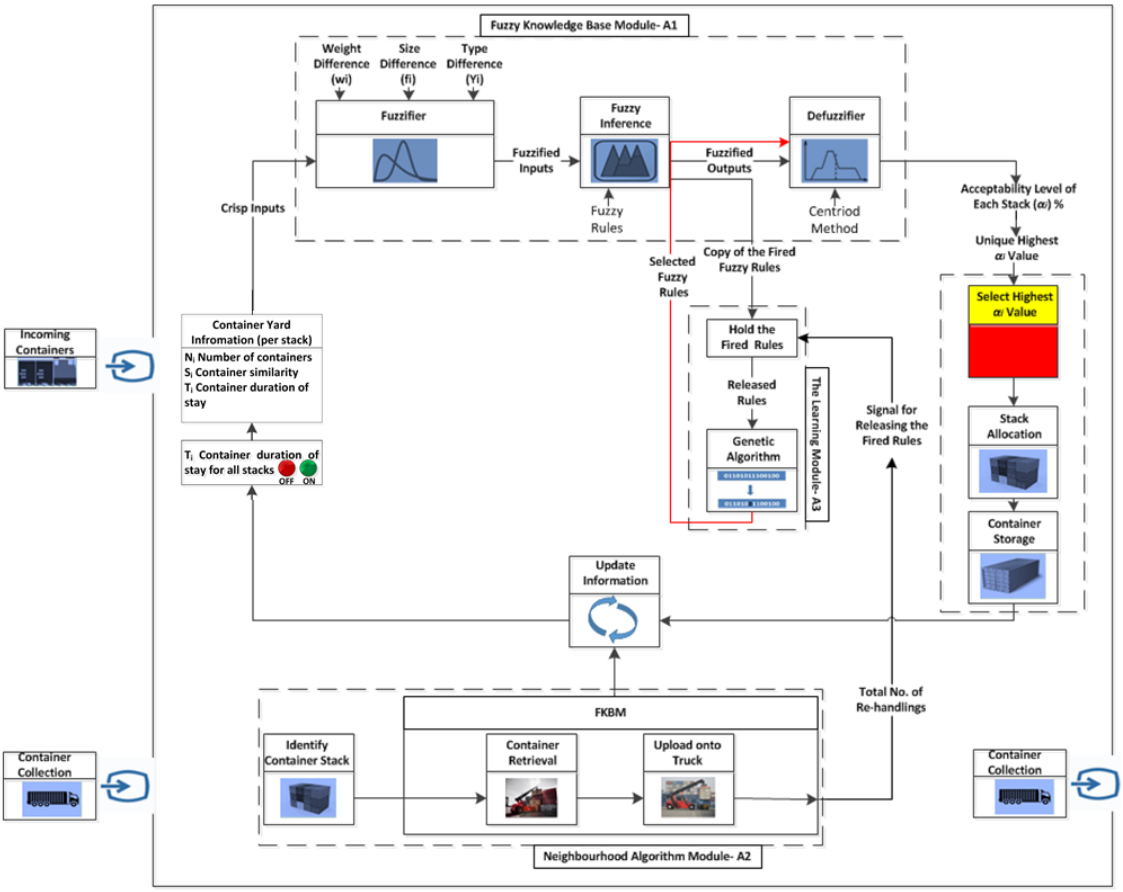


Figure 3.2: Core of the ‘FKB_GA’ Conceptual Diagram

In figure 3.2, the container collection operation occurs when a truck arrives for collection and the required container stack has been identified for retrieval. The collection operation is carried out using the FKB model. The container retrieval process initiates the re-handling operation if any container is on top of the required one. In order to re-handle containers during the retrieval operation, the module is applied using the same steps adopted in the storage operation. In the collection operation, containers are retrieved and re-handled to other stacks. These stacks are allocated for the re-handled containers by using the FKB module applying the same steps used in the stack allocation operation for the container storage.

The collection process might happen during the storage process (i.e. the allocation operation). When these two operations are required as the same time, the allocation operation will then be stopped (i.e. terminated) and the retrieval operation will be carried out, because the collection process has priority over the allocation process. Once the collection process is completed, then the allocation process will be resumed.

In the following sections, the development of the KKB_GA system will be explained in more detail.

3.5 Development of the 'FKB_GA' System

This section presents the development stages of the FKB_GA system for the management of container yard operations which combines both the FKB and GA modules. Both the FKB and GA models are now explained in detail.

3.5.1 Fuzzy Modelling of the 'FKB' Model

The FKB model consists of a number of membership functions including fuzzy sets for each of the input factors and output of the model, together with the crisp constraints. Each input factor is modelled using three fuzzy sets. The reason for this (Ries et al. 2014) is that the use of triangular membership functions with three linguistic variables assigned is more effective for a similar input variable when deciding output values. While, it is found (Menaka et al. 2016) that using six fuzzy sets (i.e. linguistic variables) is more effective to represent the output of a fuzzy system.

In the development of the FKB model, three stages of development are performed to identify an appropriate level of container storage, which are described in the following sections.

3.5.1.1 Fuzzy Sets and Membership Functions of the 'FKB' Model

Certain combined factors and constraints are considered as the main inputs of the model. Membership functions including their fuzzy sets are assigned to each factor along with linguistic definitions to capture the effects of the uncertainty in each stack in the yard in relation to each factor, and to model the unknown behaviour of the departure time of containers. In order to define a fuzzy set, the membership function has to be introduced by definitions, see Appendix B for the fuzzy set definitions and membership functions. Although there are various types of fuzzy numbers defined in the literature, triangular fuzzy numbers are one of the simplistic and widely used types of fuzzy numbers that may be used to define uncertainty (Amendola et al. 2005, Yanagi Junior et al. 2006), and hence they are selected to define uncertainties inherited in the storage factors. These three factors include:

- **Factor 1 (N): Number of Containers in the Stack**

The first input (N) considered in this module is the number of containers in stack i (N_i). The effect of N_i on the output (the possibility percentage for container storage) is that the more containers currently in the stack, then a lower acceptability level for the new incoming container to the stack i (α_i) will be obtained. If the truck arrival time for collection of a container is unknown, then the probability for the service time being longer, (i.e. owing to the number of re-handlings that would need to happen for a condensed container stack), would be high. Equally, when the number of containers in a stack is high, the number of re-handlings will be high in that stack. Therefore, input N_i is implemented to consider the number of containers for every stack i. It is worth mentioning that number of containers to be picked up (depart) from each stack i in period of time is uncertain and hence, it's considered as a fuzzy variable

- **Factor 2 (S): Similarity of Containers**

The second input (S) to be implemented in this module is the similarity of the incoming container to the containers that are already stored in the stack i (S_i). The effect of S_i on the output is that the higher the similarity within the containers of the stack then a higher acceptability for a new incoming container for the stack i (α_i) will be obtained. The attribute included in determining the similarity of containers is the customer (i.e. containers that belong to the same customer).

As far as the similarity of containers relates with number of containers stored in/departed from each stack i, hence it's considered as one of the fuzzy variables of the developed 'FKB_GA' system.

- **Factor 3 (T): Duration of Stay (DoS) of Containers**

The third input (T) is the duration of stay of the top most containers in each stack i (T_i). The effect of T_i on the output is that the longer the duration of stay of the topmost stored containers in the stack, then a lower acceptability for a new incoming container for the stack i (α_i) will be obtained. Based on work discussed by (Saurí and Martín 2011), it can be shown that a longer duration of stay correlates directly with a higher probability of departure on the next time unit. It is assumed that as time passes, when a container is not collected, the probability of departing in the future is increased, since the duration of stay of the containers will be updated. If there is no significant difference between the durations of stay of containers, then an 'ON/OFF' strategy is introduced in section 3.4.1.4 to deactivate and reactivate this factor as appropriate. However, the duration of stay of the top most containers is considered as a fuzzy variable due to the fact that it relates directly to their locations. These locations are continuously changing in response to the rapid retrieval operations of containers that need to depart at unknown times, and hence there is no deterministic pattern of duration of stay of the top most containers.

- **Other Storage Constraints**

In addition to the above factors, three constraints (W, F & Y) are considered by the model. These include the difference in weight (W_i), size (F_i) and type (Y_i) between the incoming container and the topmost container in the considered stack i. W_i is determined by subtracting the weight of the incoming container from the weight of the container in the topmost location of stack i. Similarly, F_i & Y_i are determined by comparing the size and type of the incoming container with the size and type of the container in the topmost location of stack i. In this study, three sizes of containers are included which are 20ft (Small), 30ft (Medium) and 40ft (Large) with different types for each size.

The factors and constraints explained above, together with their fuzzy sets are shown in table 3.2 and explained below.

Table 3.2: Fuzzy input factors, constraints and the output factor

Inputs	Fuzzy Sets					
N	Low	Medium	High			
S	Low	Medium	High			
T	Low	Medium	High			
Constraints	Crisp Sets					
W	Accept					
F	Accept					
Y	Accept					
Output	Fuzzy Sets					
Acceptibility Value (α)	Very Low	Low	Medium Low	Medium	Medium High	High

The output variable (α_i) is assigned a triangular membership function with six fuzzy sets (linguistic variables) as recommended by (Menaka et al. 2016). The idea of fuzzy sets (linguistic variables) was introduced by (Zadeh 1965) in order to mimic human thinking in systems rather than using crisp representation. The linguistic variables are subjectively decided based on expert opinions and experience, which in this case, based on the literature. The triangular membership function of the output variable (α_i) is defined with six linguistic variables, and there are six fuzzy sets with their respective membership functions as shown in figure 3.3(a). These fuzzy sets include 'Very Low', 'Low', 'Medium Low', 'Medium', 'Medium High', and 'High'.

For the first input variable (N_i), there are three linguistic variables with assigned triangular membership functions as in (Ries et al. 2014). The triangular membership function is defined, the three fuzzy sets (linguistic variables) decided for n_i are 'Low', 'Medium', and 'High'. In figure 3.3(b), the membership function of input (N_i) is presented.

The second input variable is S_i. Similar to T_i and N_i, S_i has triangular shaped membership functions. The linguistic variables (levels) determined for s_i are 'Low', 'Medium', and 'High' as in (Ries et al. 2014). Figure 3.3(c) represents the membership function of S_i.

The third input variable is (T_i). Fuzzy sets have triangular membership functions, there are three linguistic variables as in (Ries et al. 2014a) that are selected for T_i; 'Low', 'Medium' and 'High' as shown in figure 3.3(d).

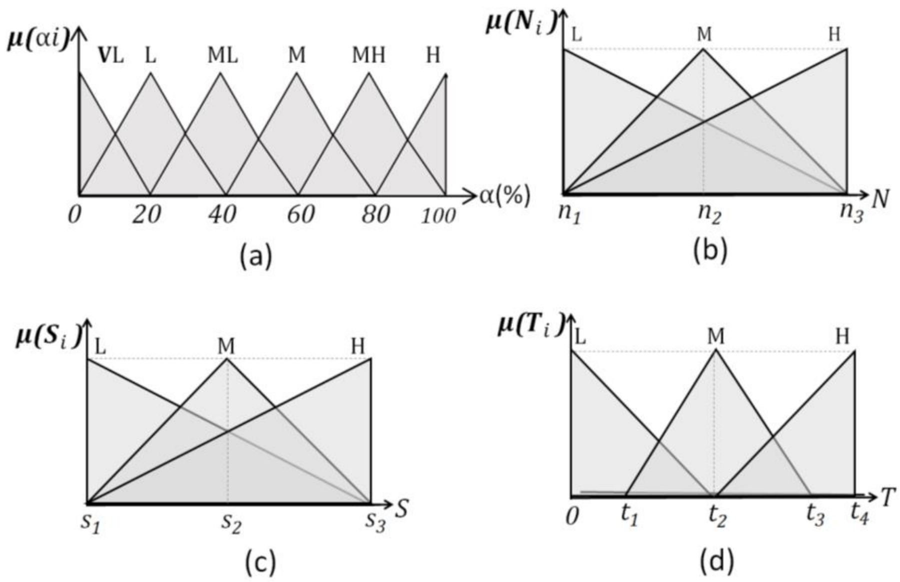


Figure 3.3: The fuzzy membership functions

(a) The fuzzy membership function of the output, (b) The fuzzy membership function of the input factor (N), (c) The fuzzy membership function of the input factor (S) and (d) The fuzzy membership function of the input factor (T)

In figure 3.3, membership functions depend on the interval value of variables considered (Wang and Kao 2008). For the number of containers per stack, the interval value is set to (0 to 5) containers in each stack i. Because of the interval value is small all three fuzzy sets (L, M, H) are considered to range from 0 to 5. The similarity of containers is related to the number of containers and hence the fuzzy sets are considered to be the same as the number of containers. As far as long period of time is considered for the duration of stay of containers, hence, different interval value is set for each fuzzy set (L, M, H).

The three considered constraints W_i, F_i and Y_i have only one set called 'Accept' or crisp membership functions. The graphical representation of their membership functions is presented in: figure 3.4(a) for W_i, figure 3.4(b) for F_i and figure 3.4(c) for Y_i.

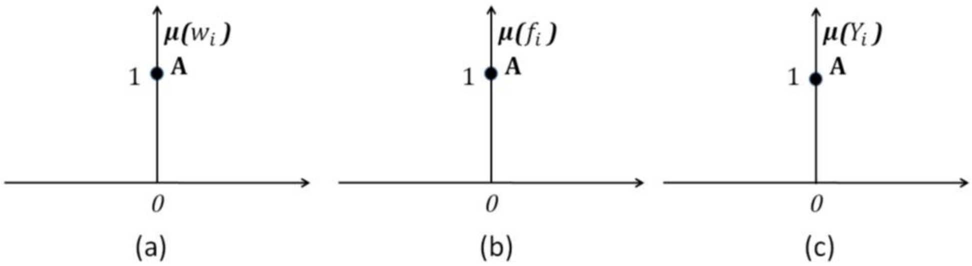


Figure 3.4: The defined crisp membership functions of the constraints, (a) The membership function of the weight (W_i,), (b) The membership function of the size (F_i) and (c) The membership function of the type (Y_i)

3.5.1.2 Fuzzy Rules of the 'FKB' Model

To define the relationship between the inputs and the output, fuzzy rules have been determined. These rules define the outcome of the interaction of each input variable on the output (Zadeh 1979). For this purpose, the selected input variables (N_i, T_i, and S_i) and their interactions are analysed, and the rules are determined. A total of 27 different rules are identified with respective levels for each input factor. The rules follow the 'If-Then' structure. The rules are subjectively decided based on expert opinions and experience, which in this case, based on the literature, observation and logic, regarding the effect of each input variable on the output. In addition, the rules are set to reflect the location availability for the incoming container in order to minimise the number of re-handlings of containers during the retrieval operation. Table 3.3 provides all the fuzzy rules defined in this study.

Table 3.3: The defined fuzzy rules

Rule No.	N_i	S_i	T_i	α_i	Rule No.	N_i	S_i	T_i	α_i
1	L	L	L	H	15	M	H	M	M
2	L	M	L	H	16	M	L	H	ML
3	L	H	L	H	17	M	M	H	ML
4	L	L	M	H	18	M	H	H	ML
5	L	M	M	H	19	H	L	L	L
6	L	H	M	H	20	H	M	L	L
7	L	L	H	MH	21`	H	H	L	ML
8	L	M	H	H	22	H	L	M	L
9	L	H	H	H	23	H	M	M	L
10	M	L	L	M	24	H	H	M	L
11	M	M	L	M	25	H	L	H	VL
12	M	H	L	MH	26	H	M	H	VL
13	M	L	M	ML	27	H	H	H	VL
14	M	M	M	M					

In this stage, an aggregation process is applied. The aggregation includes manipulating the given information in fuzzy format within the defined rules.

Upon completing the rules, the aggregation is implemented with the minimum operator (Zimmermann 2001). Equation (3.1) is introduced for the container stack allocation process to implement the aggregation with the minimum operator. For each rule j, a truncated value (T_j) is calculated.

$$T_j = min\{\mu_{(\widetilde{N})}\, n_i, \mu_{(\widetilde{S})} s_i, \mu_{(\widetilde{T})} t_i, \mu_{(\widetilde{W})} w_i,\ \mu_{(\widetilde{F}\ )} f_i,\ \mu_{(\widetilde{Y})} y_i\} \tag{3.1}$$

When any or all of constraints (W_i, F_i and Y_i) of a newly arrived or a re-handled container do not match the topmost containers W_i, F_i and Y_i in each stack, then the acceptability level values of that stacks will be 0. As the aggregation operator is minimum (as stated in equation (3.1) in any rule because of the considered constraints), if the degree of membership of a given value for W_i, F_i and Y_i is computed to be 0, the final output for all T_j will also be 0. For example, when $\mu_{(\widetilde{W})} w_i = 0$, $\mu_{(\widetilde{F})} f_i = 0$, $\mu_{(\widetilde{Y})} y_i = 0$, then the T_j value will be 0 using equation (1) and the acceptability level values will be 0.

3.5.1.3 Output of the 'FKB' Model

The de-fuzzification stage involves the operations required to transform the fuzzy output set into a crisp output. There are various methods for de-fuzzification including Centre of Gravity, Mean of Maximum and Centre Average (Zadeh 1965, Zimmermann 1991). In this study, the Centroid Method which is a specific implementation of the Centre strategy of Gravity method is selected for the de-fuzzification process due to the fact that it is the most prevalent and physically appealing of all the other methods and the most common method used in most applications (Castro 1995, Lee 1990, Morim et al. 2017).

This strategy finds the centre value (y_j) for each rule by using the truncated value reflected on the output fuzzy sets, then the overall centre of gravity is computed. Consider the truncated value T_j and the output $\widetilde{\alpha}$ where the rule defines the outcome to be the level-p. The centre value is given by the equations (3.2 to 3.5), as shown in figure 3.5 below. Upon finding the corresponding centre values for each of the rules j (y_j) as defined, the crisp output value defined as (y^*) is computed with the centre of gravity method as shown in equation (3.6).

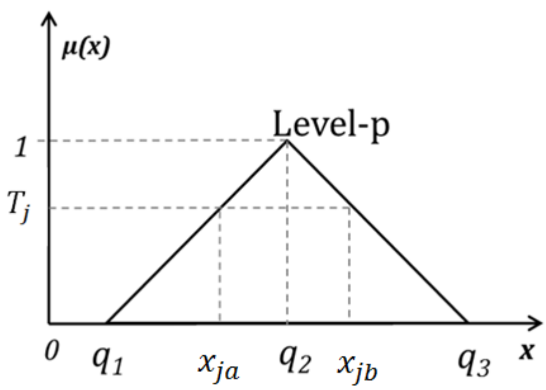


Figure 3.5: Truncated value on the output fuzzy set

$$y_j = \frac{x_{ja} + x_{jb}}{2}, \qquad where; \tag{3.2}$$

$$T_j = \frac{x_{ja} - q_1}{q_2 - q_1} = \frac{q_3 - x_{jb}}{q_3 - q_2}, \qquad where; \tag{3.3}$$

$$x_{ja} = q_1 + T_j(q_2 - q_1) \qquad and \qquad x_{jb} = q_3 - T_j(q_3 - q_2) \tag{3.4}$$

$$\therefore \qquad y_j = \frac{x_{ja} + x_{jb}}{2} = \frac{q_1 + q_3 + T_j(2q_2 - q_1 - q_3)}{2} \tag{3.5}$$

$$y^* = \frac{\sum_{j=1}^{l} y_j T_j}{\sum_{j=1}^{l} T_j} \tag{3.6}$$

Equation (3.2) is used to find the centre value of the output fuzzy set (y_j) from the boundary values (x_{ja}, x_{jb}). Equations (3.3) and (3.4) are used to find boundary values (x_{ja}, x_{jb}) of the centre value in any rules *j*. Equation (3.5) is used to find the centre value (y_j) of any rules *j*, and equation (3.6) is used to calculate the acceptability level values of stacks (i.e. crisp outputs).

Numerical Example

A numerical example is presented demonstrating how fuzzy-knowledge-based rules are used to select one out of three possible stacks for storing one incoming container. In this example, the case of three stacks in a yard where each stack contains a different number of containers is explained.

To start with the allocation of the stack for the incoming container, the fired rules that have been identified by the system are: Stack 1; rules 1, 10 and 19, Stack 2; rules 7, 8, 9, 16, 17, 18, 25, 26, and 27, while Stack 3; rules 4, 7, 13, 16, 22, and 25.

Stack 1 has 4 containers with similarity of containers equal to 0% (i.e. none of these containers belong to the same customer), with the duration of stay of the topmost container equal to 1 day. There are 2 containers in stack 2 with a similarity of containers equal to 20% and the duration of stay of the topmost container equal to 24 days. Stack 3 has 2 containers with a similarity equal to 0%, and duration of stay equal to 19 days.

For the given inputs above, the matched degrees of the input factors in rule 1 are 0.2, 1, and 0.917 as shown in figures 3.6(a), 3.6(b), and 3.6(c) respectively. The matched degrees of three corresponding factors are determined by the given inputs of one fuzzy rule. The matched degree of consequence in the one rule will be the minimised value of the matched degrees of three corresponding factors (Wang and Kao 2008). The truncated value T_1 is calculated by using equation (3.1) and equal to 0.2, see figure 3.6(d).

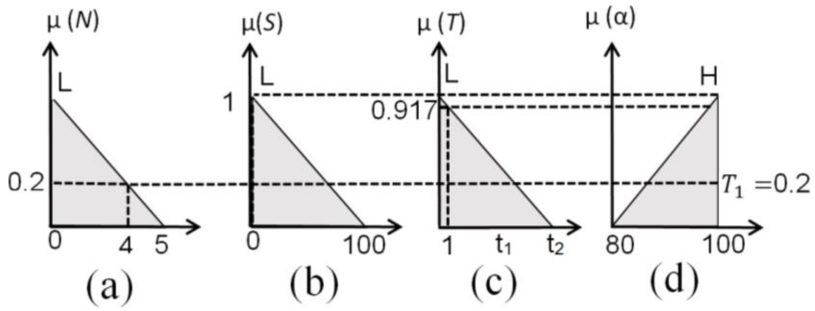


Figure 3.6: The matched degrees and truncated value in rule 1

*(*a) The matched degrees of the number of containers factor in the stack, *(*b) the matched degrees of the similarity of containers factor in the stack, *(*c) The matched degrees of the duration of stay factor of containers in the stack, and *(*d) The truncated value.

In order to calculate the acceptability level value of stack 1, the centre values y_1 of all rules of stacks need first to be calculated. Starting with the calculation of the centre value y_1 of rule 1, referring to figure 3.5, the boundary (x_{1a},x_{1b}) is constructed first from the truncated value T_1 in rule 1 as shown in figure 3.7. The values of q_1 and q_2 in figure 3.7 are 80 and 100 respectively (for the high fuzzy set of the output membership function, see figure 3.3(a).

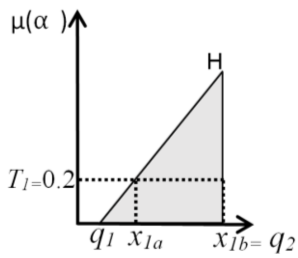


Figure 3.7: The boundary of centre value for rule 1

Now the centroid *method* is applied to calculate the values of x_{1a} and x_{1b} using Equation (3.4);

$x_{1a} = 80 + 0.2(100 - 80) = 84$ and

$x_{1b} = q_2 = 100.$

By using Equation (3.5), the centre value y_1 is then calculated as shown below:

$$y_1 = \frac{84\% + 100\%}{2} = 92\%$$

Similarly, the matched degrees of the input factors, truncated T_j and centre values y_j can be obtained by adapting the other rules.

The matched degrees of the input factors in rule 10 are 0.4, 1, and 0.917. The matched degrees of the input factors in rule 19 are 0.8, 1, and 0.917. The truncated values of rules 10 and 19 are 0.4 and 0.8 respectively. The centre values of rules 10 and 19 are 60% and 20% respectively. Then the acceptability level value of stack 1 is calculated using equation (3.6).

$$y^* = \frac{\sum_{j=1}^{l} y_j T_j}{\sum_{j=1}^{l} T_j} = \frac{(0.92*0.2)+(0.60*0.4)+(0.20*0.8)}{0.2+0.4+0.8} = 0.417$$

However, the above steps can be applied for calculation acceptability level values of stacks 2 and 3. Acceptability level values of stacks 2 and 3 are 0.307 and 0.362 respectively.

Since the acceptability level value of stack 1 is the highest one, stack 1 has been allocated for accommodating the incoming container.

3.5.1.4 The Proposed 'ON/OFF' Strategy

As mentioned earlier, the FKB model has three input factors. Based on these factors and other related constraints the acceptability level value of each stack is computed. The stack with the highest acceptability level value is selected/ allocated to store the container.

To provide realistic acceptability level values for the stacks, one of the input factors (i.e. duration of stay) provided to the system changes dynamically over time. This is due to the fact that in the passing of time, the new containers will become pre-existing and the duration of stay for these containers will be updated and each could have a different duration of stay. In addition, the retrieval operation could lead to different durations of stay of the topmost containers in the selected stacks and hence, this factor has to be carefully investigated for a more effective stack allocation decision.

As the duration of stay for containers can vary over time, an 'ON/OFF' strategy is proposed to activate/deactivate the duration of stay factor in the system if there is a significant difference in the durations of stay for the topmost containers in all the stacks. See figure 3.8 for the 'ON/OFF' strategy for the duration of stay factor.

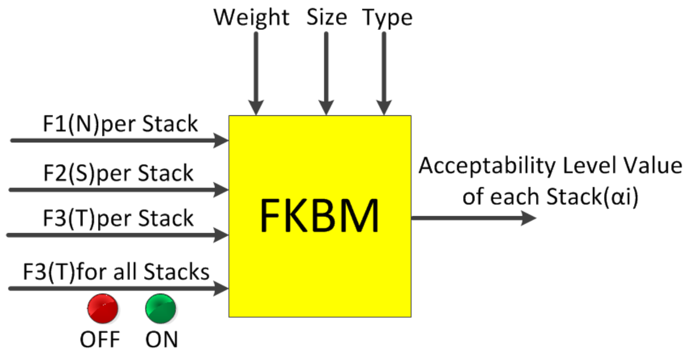


Figure 3.8: The 'ON/OFF' strategy of Duration of Stay factor

When the duration of stay factor as a system input, is activated (i.e. set to "ON"), all factors (N, S, and T) are collectively used to calculate the acceptability level values for the container storage operation. While, when the duration of stay factor is temporarily deactivated (i.e. set to "OFF"), only the two factors (N and S) are used to calculate the acceptability level values for the container storage operation (i.e. for stack allocation).

The decision of how the combination of different linguistic variables for each input factor affect the output (i.e. acceptability level values) is determined by the defined fuzzy rules. For this purpose, 27 fuzzy rules are identified as described in Table 3.2 above, which define the outcome of the interaction of each input factor on the output. When the duration of stay factor is activated together with the other two factors to the system, all defined rules (27 rules) are fed to the fuzzy inference engine to calculate the output (i.e. acceptability level values for each stack) for the container storage operation.

However, when the duration of stay factor is deactivated, the other two factors (N and S) are only utilised to calculate the acceptability level values for the stacks. In this case, the number of the defined fuzzy rules is reduced to 9 and the acceptability fuzzy sets are updated as shown in Table 3.4.

Table 3.4: The reduced fuzzy rules

Rule No.	N_i	S_i	T_i /OFF	α_i
1	L	L	L	~~H~~ M
2	L	M	L	~~H~~ MH
3	L	H	L	H
4	L	L	M	H
5	L	M	M	H
6	L	H	M	H
7	L	L	H	MH
8	L	M	H	H
9	L	H	H	H
10	M	L	L	~~M~~ ML
11	M	M	L	M
12	M	H	L	MH
13	M	L	M	ML
14	M	M	M	M
15	M	H	M	M
16	M	L	H	ML
17	M	M	H	ML
18	M	H	H	ML
19	H	L	L	~~L~~ VL
20	H	M	L	L
21`	H	H	L	~~ML~~ M
22	H	L	M	L
23	H	M	M	L
24	H	H	M	L
25	H	L	H	VL
26	H	M	H	VL
27	H	H	H	VL

In Table 3.4, when the duration of stay factor is deactivated, only the rules highlighted in green will be used by the system. In this case only the number of containers and the container similarity factors will be used to calculate the acceptability level values for the stacks in the container storage operation. The highlighted column in red displays the linguistic variables for the duration of stay factors. The highlighted rows in green displayed in the second and third columns are the linguistic variables for the number of containers and container similarity factors. The rows highlighted in green in the last column are the linguistic variables for the output (i.e. acceptability levels). The linguistic variables of the output are updated based on the linguistic variables for the two input factors as shown in the above table. In the following section, the 'ON/OFF' strategy will be explained further based on an approximation algorithm that is introduced to handle time progression/increments for each container duration/length of stay.

3.5.1.5 The 'ON/OFF' Strategy Based Approximation Algorithm of Time Incremental of Container Duration of Stay

The approximation algorithm for the 'ON/OFF' strategy is developed to approximate and convert the duration of stay of containers from an hour unit into a day unit. This is in response to the condition which states that no more than 30 days are allowed for a container to stay in the yard. This algorithm is related to the 'ON/OFF' strategy by owing to different durations of stay of the topmost containers stored in each stack and explained below in more detail.

Once a container is stored in the yard, the duration/ length of stay of a container is incremented continually from the time a container is stored in the yard until it departs. This assists the decision of when and where to store newly-arrived containers when there are pre-existing containers in the yard. After a period of time, each of the containers in the yard will have different lengths of stay. See figure 3.9, which illustrates the progression for the incrementing of the duration of stay for containers over time.

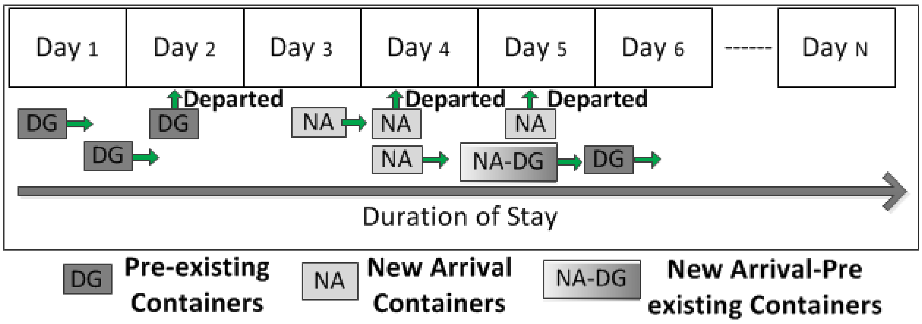


Figure 3.9: The Time Incremental for the Container Duration of Stay

In figure 3.9, there can be seen a number of pre-existing containers (DG) which have been stored in the yard for a period of time (i.e. containers in dark grey). When a newly-arrived container (NA) needs to be stored with pre-existing ones, it will be stored according to the acceptability level values obtained from the FKB model as explained in section 3.4.1.1. While the new arrivals are being stored (i.e. containers in light grey), some of the pre-existing containers may depart. In the passage of time, those new containers (i.e. containers in half light grey and half dark grey) will become pre-existing (NA-DG), the duration of stay for the containers will be updated and each could then have a different duration of stay.

Because the durations of stay of the topmost containers stored in each stack can be different, the 'ON/OFF' strategy will need to activate/de-activate the duration of stay factor in the processing accordingly (e.g. as shown in the following algorithm).
The notations used by the approximation algorithm are defined below.

- DoS: Duration of Stay of topmost container in each stack
- t_o: Minimum DoS in hours
- t_{Max}: Maximum DoS in hours
- d: DoS in days unit
- d_o: Minimum DoS in days unit
- d_{Max}: Maximum DoS in days unit
- t_n: DoS between t_o and t_{Max}
- d_n: DoS between d_o and d_{Max}

The steps of the 'ON/OFF' strategy based the Approximation algorithm are:

Step 1: Obtain durations of stay of the topmost container for all stacks

Step 2: Calculate the possibility percentage for container storage (storage success)

Step 2.1: Approximate the duration of stay (DoS) of container

Step 2.1.1: If t_o< *DoS* $\leq t_1$*, then approximate the DoS to* d_o

Step 2.1.2: If t_1< *DoS* $\leq t_n$*, then approximate the DOS to* d_n

Step 2.1.3: If t_n< *DoS* $\leq t_{n+1}$*, then approximate the DoS to* d_{n+1}

Step 2.1.4: If t_{n+1}< *DoS* $\leq t_{Max}$*, then approximate the DoS to* d_{Max}

Step 3: Check the approximated durations of stay in days unit

Step 3.1: Consider the stacks that have the same approximated duration of stay values as possible (success) stacks for storage

Step 3.2: Calculate the number of different durations of stays

Step3.3: Calculate the possibility percentage for container storage (number of different durations of stay / total number of stacks in the yard)

Step 3.4: If the possibility percentage for container storage (success) is ≥a specific percentage, then go to Step 4

Step 3.5: If the possibility percentage for container storage (success) is < a specific percentage, then go to Step 5

Step 4: Activate the duration of stay factor (ON).

Step 5: Deactivate the duration of stay factor (OFF).

To explain further the above algorithm, the process starts by obtaining the duration of stay for the topmost container in each stack. Then, the possibility percentage for container storage (i.e. the chance of the container being successfully stored in a stack) is calculated. To calculate the possibility percentage for container storage, an approximation of the duration of stay for containers in days was necessary.

Figure 3.10 shows the duration of stay approximation process.

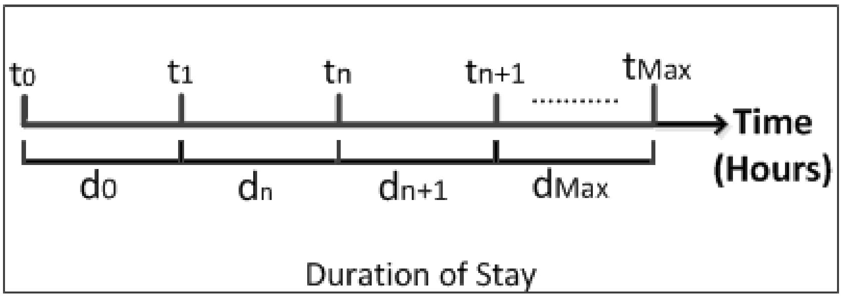


Figure 3.10: The Duration of Stay Progression Approximation

In figure 3.10, when the duration of stay is t_1 hours or less, then the DoS is approximated to d_o days. However, when the duration of stay is greater than t_1 hours and less than or equal to t_n hours, then the duration of stay is approximated to d_n days. If the duration of stay is more than t_n hours and less than or equal to t_{n+1} hours, then the duration of stay is approximated to d_{n+1} days. But, when the duration of stay is more than t_{n+1} hours or equal to t_{Max} hours, then the duration of stay is approximated to d_{Max} days.

The next step was to check the approximated duration of stay for the topmost container of all stacks, and to consider the stacks that have the same approximated duration of stay values, as possible stacks for storage. This checking was necessary to calculate the number of different durations of stay for containers in the yard. The possibility percentage for container storage was calculated as the number of different durations of stay, divided by the total number of stacks in the yard. If the possibility percentage for the container storage) is greater than or equal to a predefined percentage provided by the user, then the DoS factor is activated (i.e. ON) to the system. However, if the possibility percentage for container storage is less than the same percentage provided by the user, then the DoS factor is deactivated (i.e. OFF) to be temporarily suspended as an input factor to the system.

Numerical Example

This section presents a numerical example of how the approximation algorithm works together with the use of the ON/OFF' strategy. The maximum duration of stay in this example is assumed to be two days (48 hours).

The first step of the algorithm starts by obtaining the duration of stay of the topmost container for each stack in the yard (i.e. time spent in the yard), then the calculation of the possibility percentage for container storage takes place. To calculate this, an approximation process for the duration of stay of containers was necessary. See figure 3.11 which demonstrates the DoS approximation process.

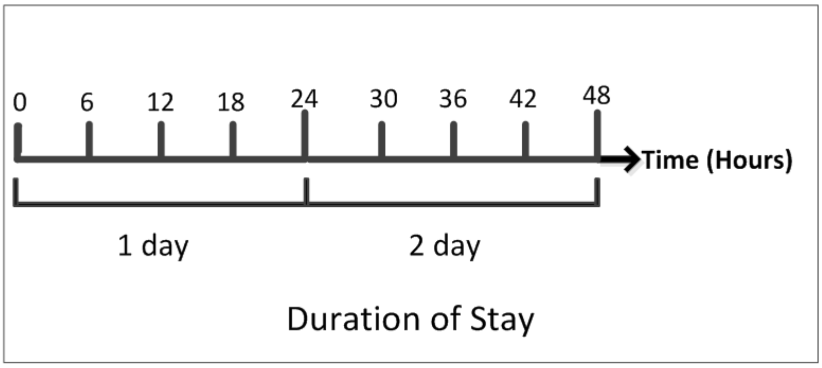


Figure 3.11: Example of DoS approximation process

In stacks where the duration of stay of the topmost containers is equal to or less than 6 hours, then the duration of stay is approximated to 0 day. But, if the duration of stay is greater than 6 hours and less than or equal to 12 hours, then the duration of stay is approximated to ½ day. When the duration of stay of containers is greater than 12 hours and less than or equal to 18 hours, then the duration of stay is approximated to ½ day. If the duration of stay is greater than 18 hours and less than or equal to 24 hours, then the duration of stay is approximated to 1 day. If the duration of stay is greater than 24 and less than or equal to 30, then the duration of stay is approximated to 1 day. If the DoS is greater than 30 hours and less than or equal to 36 hours, the DoS is approximated to 1 & ½ day. If the DoS is greater than 36 hours and less than or equal to 42 hours, then the DoS is approximated to 1 & ½ day. If the DoS is greater than 42 hours and less than or equal to 48 hours, then DoS is approximated to 2 days.

After the approximation process is completed, the approximated duration of stay of the topmost containers of stacks that have the same values are considered as one possible location for storage.

In order to calculate the possibility percentage for container storage, it is necessary to calculate the durations of stay for the topmost containers of all the stacks in the yard. The possibility percentage for container storage is obtained by dividing the number of containers with a different duration of stay by the total number of stacks in the yard and expressing that as a percentage. When the possibility percentage for container storage is greater than or equal to a percentage predefined by the user then the duration of stay factor is activated (ON). When the possibility percentage for container storage is less than the predefined percentage, then the duration of stay factor is deactivated (OFF).

The development of the GA optimisation model is presented in detail in the next section.

3.5.2 Development of the GA Model

In this section, a GA model is developed to be integrated with the FKB model, for optimising the stack allocation of the container storage and retrieval operations. The genetic algorithm is a probabilistic search method that employs a search technique based on ideas from natural genetics and evolutionary principles (Hassanein et al. 2004). Using an optimised FKB model can be defined as the process of selecting the best set of fuzzy rules (Pawlukowicz 2012).

The purpose of using GA and integrating it with the FKB model is explained in the next section.

3.5.2.1 The Purpose of Using GA

Although fuzzy-knowledge-based models were previously used to select the fuzzy rules from the rule base (Nelles et al. 1996, Carmona et al. 2004, Cintra and de Arruda 2007, Shill et al. 2015). However, the genetic algorithms (GA) was utilised to tune and finally select the optimal/near optimal rules from the fired fuzzy rules, by removing those that might reduce system performance. This is due to the fact that the definition of fuzzy rules and membership functions is actually affected by subjective decisions, and some of the fired rules would be redundant which reduces the overall performance of the fuzzy-knowledge-based system.

In the storage problem being investigated, a set of rules are fired for each possible stack taking into account the input factors and constraints. The GA model is then used to tune a set of the fired fuzzy rules per stack and then to optimise these rules by selecting the most effective rules in each set that leads to the minimum number of re-handlings for containers.

The GA model starts by selecting only the fired rules per stack which are to be included in the calculation of the acceptability level values for stack allocation. The rest of the rules will be temporarily unselected. The learning process enables the GA model to keep continuously evolving the selection process for rules until a solution with the minimum number of re-handlings of containers is achieved. See figure 3.12 for an explanation of the GA module rules selection per stack.

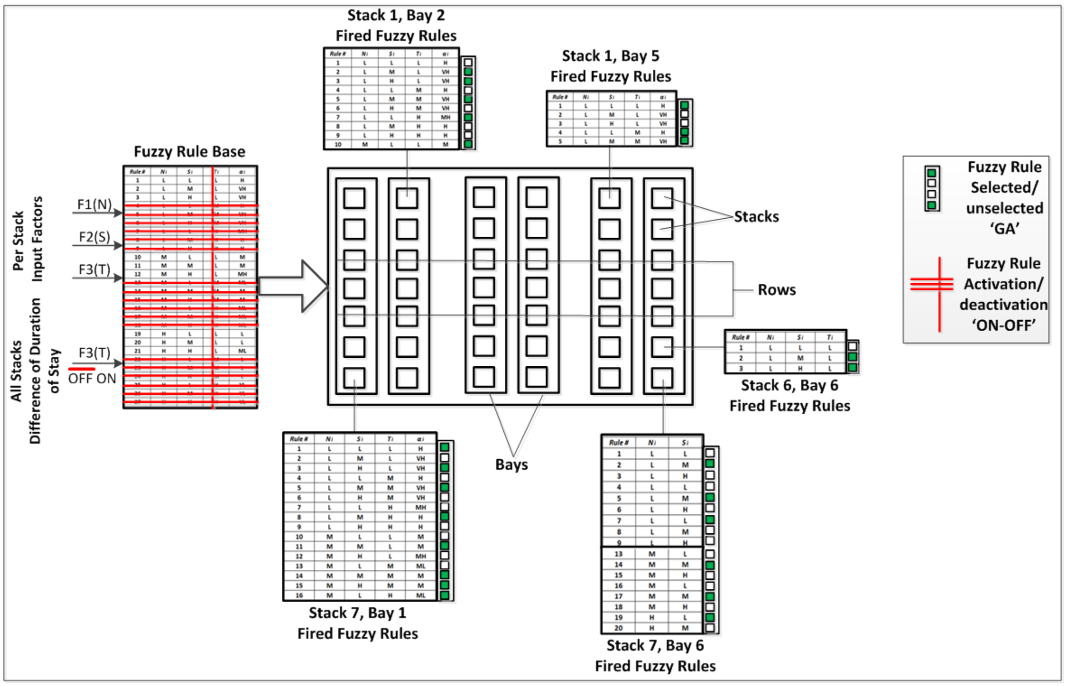


Figure 3.12: The proposed GA for rules selection per stack

In figure 3.12, a number of fuzzy rules for each possible storage stack in the yard fuzzy rule base are fired by the FKB model. The selection of some of these rules for each stack, illustrated by the green boxes is then made by using the GA model, while the white boxes illustrate the temporarily unselected rules for each stack.

To further explain the mechanism of GA in rules tuning and selection, consider the 5 fuzzy rules that are fired in stack 1, bay 5. Rules number 2 and 3 are unselected as represented in white boxes. While rules 1, 4 and 5 are selected by using GA, represented by green boxes. Based on the selected rules 1,4, and 5, the acceptability level value of storage in stack 1, bay 5 is calculated rather than the one obtained by using all the 1-5 fired rules.

From the process of selecting rules per stack for the container storage and retrieval operations explained above, the fired fuzzy rules for each container and each possible stack can be represented as a three-dimensional vector as shown in figure 3.13(a). The fired fuzzy rules for one container per possible stack allocation can be represented by a uni-layer chromosome as shown in figure 3.13(b).

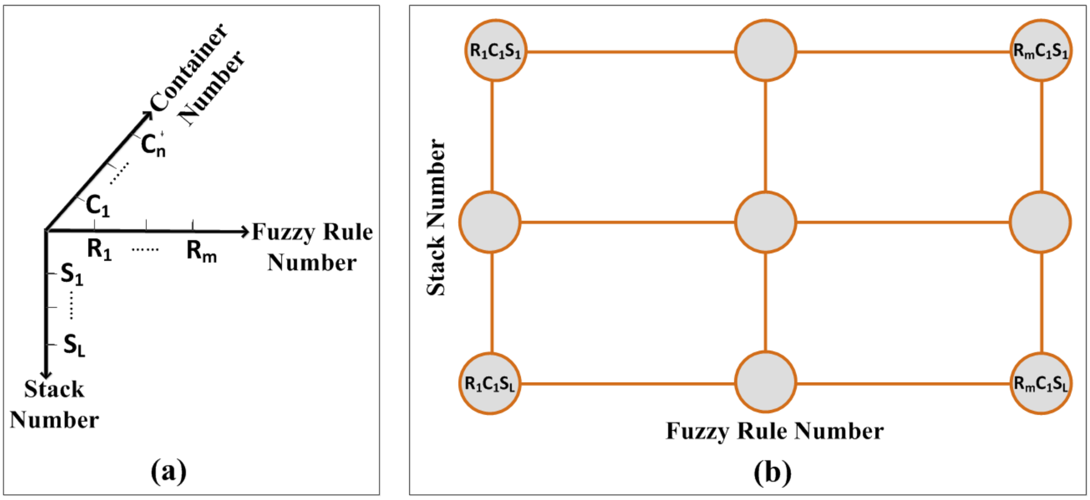


Figure 3.13: The fired fuzzy rules representation

(a) The three-dimensional vector representation of n containers, L possible stacks and m fired rules, and (b) The uni-layer chromosome structure of 1 container, L stacks and m fired rules

In figure 3.13(a), a maximum of m number of rules fired from the fuzzy rule base is allocated for each possible stack to store one container. This number varies from one stack and/or container to another. This depends on the number of fired rules obtained for each possible stack and container.

These rules can be arranged in the form of a uni-layer chromosome as presented in figure 3.13(b) if only one container (C_1) is considered. For more than one container and possible stack, a maximum of *m* fired fuzzy rules will be stored in a multi-layer chromosome structure which will be further explained in the next section.

3.5.2.2 Multi–Layer Chromosome Structure

The design and structure of a chromosome depends on the problem requirements. In a multi-layer chromosome, each layer can be used to represent a set of information. To deal with a set of information with multi attributes, the multi-layer chromosome is an efficient structure (Al-Bazi et al. 2010).

In the proposed chromosome, the content of each gene was represented by a fired fuzzy rule for a specific container and the possible stack(s) in which it can be stored. The number of genes was equal to the number of fired (i.e. used) fuzzy rules for a specific container and possible stacks, and the number of layers was equal to the number of containers.

The height dimension of the chromosome, for the possible stacks for storing each container, was attached with each gene. The fired fuzzy rules were placed in the length dimension, which was a chromosome. This chromosome included a number of genes that represented the fired fuzzy rules per container per possible stack(s). The container number was placed in the width dimension; each container being represented in one layer with its fired fuzzy rules and possible stacks. The number of fired rules from the fuzzy rule base was different for each possible stack and each layer and/or container number and hence, there was a different number of genes defined for each possible stack and container number. However, the maximum number of the fired number of rules per stack per container must not exceed the total number of rules of the fuzzy rule base which was m.

The multi-layer chromosome structure was proposed to provide more flexibility to deal with such sets of information to select the best fuzzy rule(s) from the fired rules for each container and possible stacks. Figure 3.14 shows the multi-layer chromosome structure.

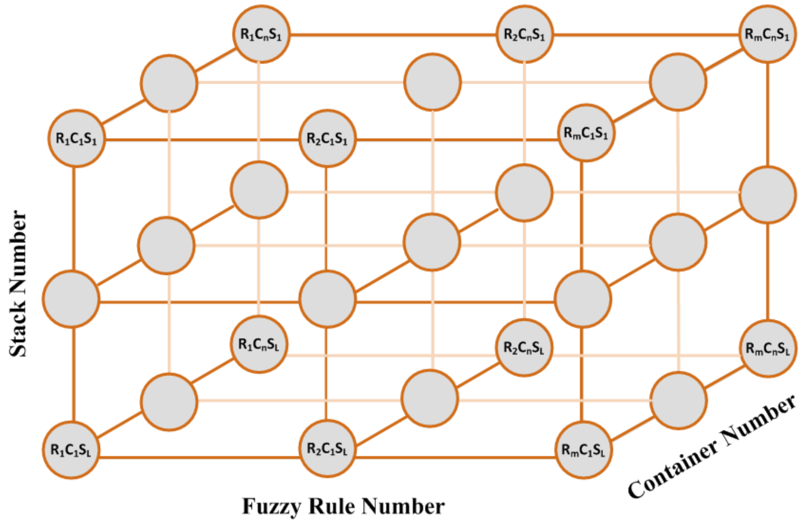


Figure 3.14: The Multi-Layer chromosome structure for fuzzy rule representation of n containers

The reason behind this multi-layer chromosome structure was to accommodate different sets of information that can be represented in a chromosome structure. For each container and possible stack(s), a number of fuzzy fired rules were stored in genes of a chromosome. The front (i.e. first) layer of the chromosome involved the first container with its fired fuzzy rules and possible stack(s). The second layer of the chromosome involved the second container with its fired fuzzy rules and possible stack(s). The number of layers depended on the total number of containers. Each gene of each layer was used to select or not to select rule(s) from the fired fuzzy rules using binary coding. All fired fuzzy rules per container and possible stack(s) were then stored in multiple layers.

3.5.2.3 Genetic Algorithm Steps

In previous studies, multi-layer genetic algorithm models were considered in terms of multi-level GAs, where each level represented a separate traditional GA model (Kelareva and Negnevitsky 2002, Negnevitsky and Kelareva 2008, Abdulhalim and Attea 2015).

In this study, each chromosome layer was assigned per container to include all its possible stacks along with their fired rules taking into account the reduced number of rules obtained by applying the ‘ON/OFF’ strategy. A multi-layer chromosome structure was proposed to accommodate more than one container including the possible stacks to which each can be stored, together with the set of fired fuzzy rules assigned for each stack. A gene with no fuzzy rule included was marked as a vacant gene. This structure provided the GA with more flexibility for dealing with large sets of information and the capability to solve the problem of selecting the optimal/near optimal rule(s) out of a set of fired fuzzy rules per container.

The Genetic Algorithm steps for the proposed multi-layer concept including objective function, generation of an initial population, selection method, cross-over and mutation operators and finally the termination condition are presented in figure 3.15.

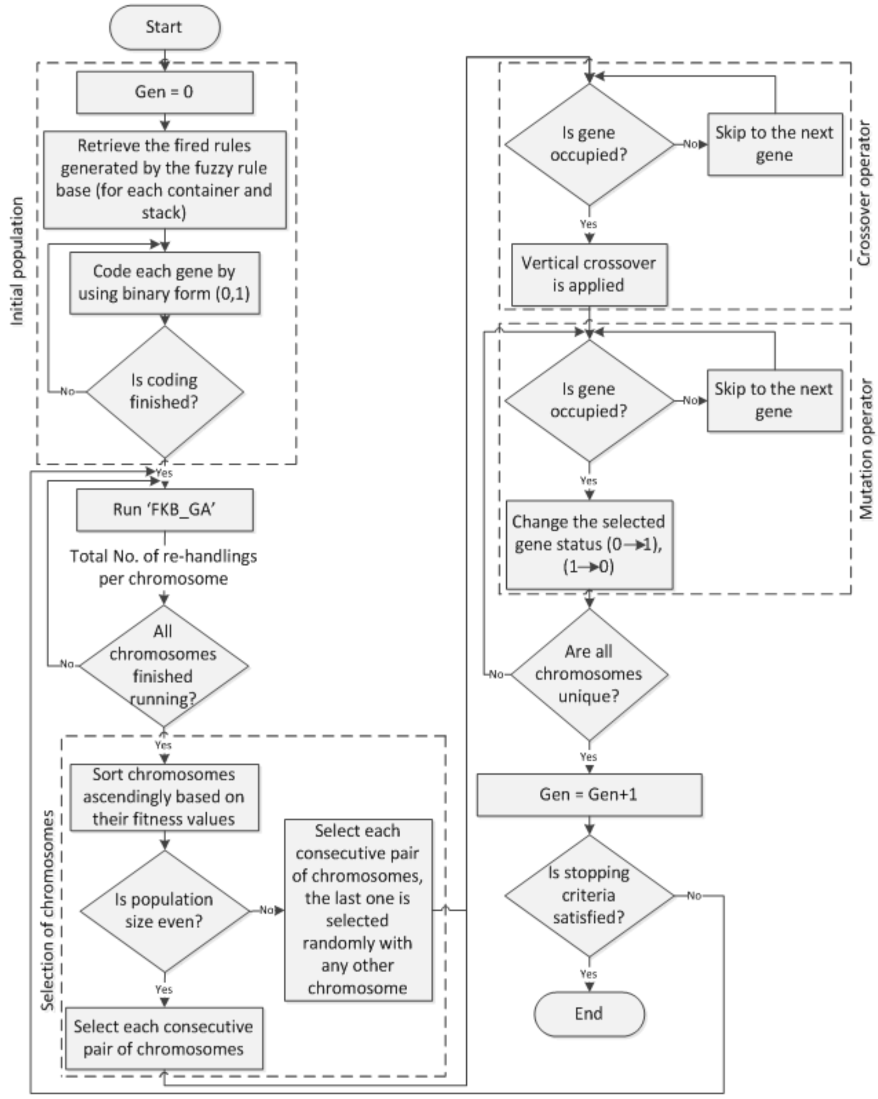


Figure 3.15: The flow chart of the proposed Multi-Layer GA

In the Multi-Layer GA model, an initial population of selected rules, out of each set of rules per stack, was randomly identified. Binary coding was applied on each chromosome layer by coding the selected rules to 1 as shown in green boxes in figure 3.12, and 0 for any other temporarily unselected rules, as shown in white boxes. Based on the selected fuzzy rule(s), the acceptability level values for the possible stacks were calculated then a stack was allocated to store the container.

The GA starts by repeating the genetic cycle, manipulating the chromosomes, from the initial random population, to generate new offspring chromosomes (i.e. strings). Each chromosome was evaluated based on its fitness function value. At the end of each generation, all fitness function values were sorted in ascending order, those chromosomes with the minimum number of re-handlings of containers being kept on the top of the selection list for further investigation. Crossover and mutation genetic operators were then applied to create next generations. The steps repeat until the termination condition was satisfied.

The Objective function

The objective function was formulated to evaluate the performance of the 'FKB_GA' system in terms of the total number of re-handlings for containers. The total number of re-handlings obtained by executing each chromosome was used to develop the objective function below:

$$\min \sum_{i=1}^{n} \gamma_n \tag{3.7}$$

Where i represented the container number, n was the total number of stored containers in the yard. The variable γ_n was the number of re-handlings of all n containers. The formulated objective function guaranteed a minimum total number of re-handlings for containers. This total number of re-handlings was the sum of the number of re-handlings to retrieve all containers in the yard.

Initial Population of Selected Fuzzy Rules

As a starting point, an initial basic feasible solution was required. After the set of fired fuzzy rule(s) for each possible stack and container was stored, binary coding was applied randomly to select some of these fuzzy rules and set them to 1 and temporarily unselect the rest and set them to 0. Based on the selected fuzzy rules, the acceptability level values for the stacks were calculated, then, a stack was allocated to store each container. The binary coding process avoided generating 0s for all the genes at each layer of the chromosomes. In addition, it skipped the vacant genes, i.e. genes which have not included fuzzy rules.

The Selection Method

After the chromosomes were sorted in ascending order based on their fitness values (i.e. total number of re-handlings of containers), each pair of chromosomes with minimum fitness function values in the population list were selected to generate further chromosomes (i.e. offspring) using GA operators. This applies in case the population size was even. If the population size was odd, each pair of chromosomes with minimum fitness values was selected for further investigation. However, the last left chromosome was coupled with any of the randomly selected chromosome from the population for further offspring generation. The GA operators are explained in detail in the section below.

Multi-Layer Genetic Algorithm Operators

The Crossover Operator

The crossover operator is a genetic operator that combines (i.e. mates) two chromosomes (i.e. parents) to produce a new chromosome (i.e. offspring) with crossover probability. The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents (KAYA et al. 2011).

The crossover operator is based on the exchange of genes between two chromosomes when they are selected. With the crossover operator, selection of genes to be exchanged depends on the probability of crossover (i.e. a specific percentage). The probability of crossing over genes determines how many genes will be selected for exchanging. If a gene does not contain a fuzzy rule (i.e. the rule of selecting a stack for a container is not fired from the fuzzy rule base), then the crossover process skips to the next gene.

A vertical crossover type is used to change the status of the selected genes of the first selected chromosome in the selected layer with the opposite gene of the second selected chromosome in the same selected layer. The opposite gene means the gene that is selected based on probability of crossover in a chromosome to be exchanged with its opposite gene in another selected chromosome (Tang et al. 2007). The status of each selected gene is changed from selected (1) to temporarily unselected (0) or vice versa. Two genes status in corresponding chromosomes can be changed from being selected (1) to temporarily unselected (0) or vice versa, given that there should be at least one selected gene (active rule) in the related chromosome.

This crossover operator was applied to achieve the best random exchanging of genes between each pairs of chromosomes. See figure 3.16 for an illustration of the crossover of two selected chromosomes.

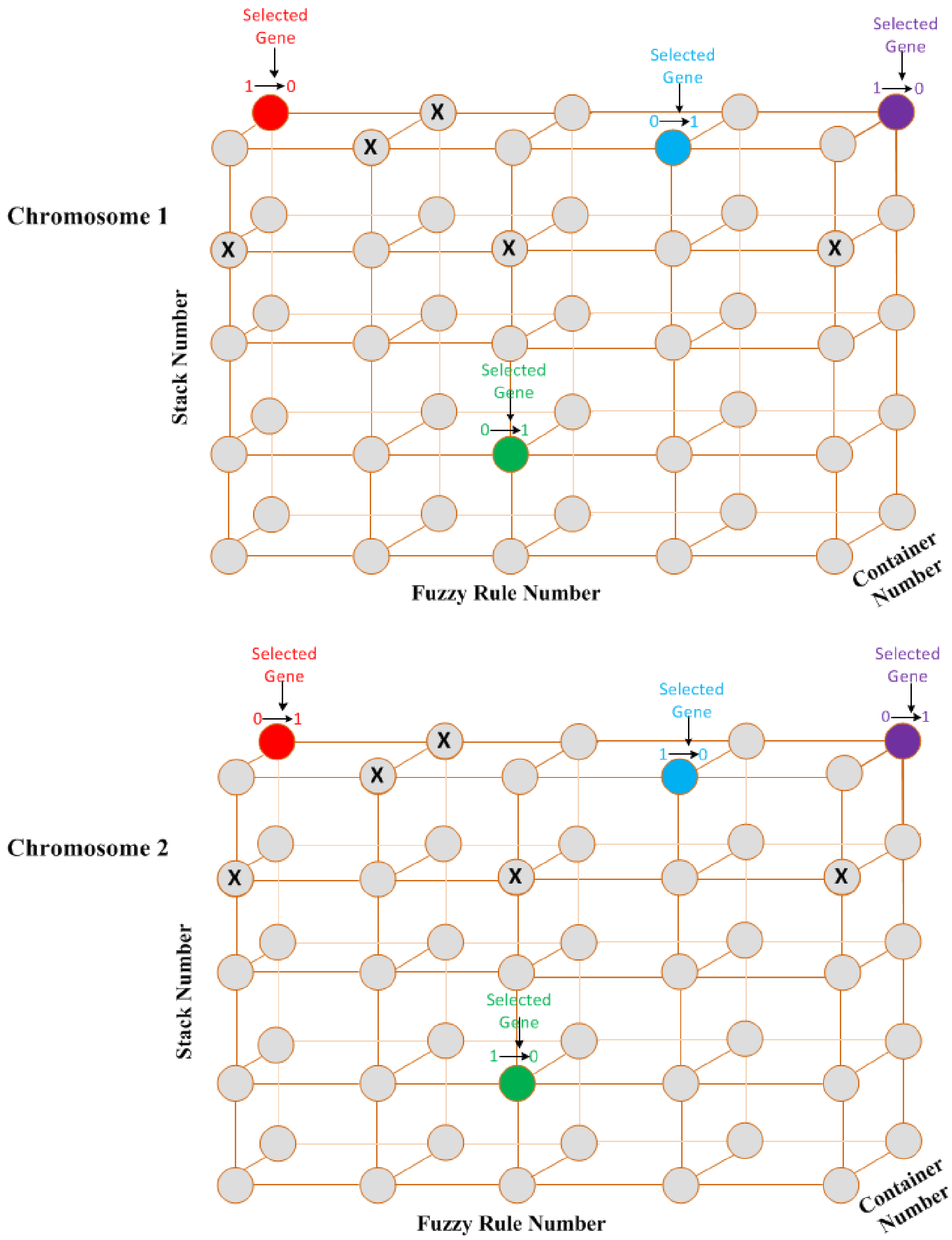


Figure 3.16: The Crossing-over of genes in Multi-Layer chromosome

The probability of crossover value decides the number of genes to be exchanged at each chromosome. The crossover is skipped when genes contain no fired rule. This type of crossover operator provides an equal chance for all genes in a layer to be selected for swapping with the opposite chromosomes genes by changing the status of the fuzzy rule stored in a gene from being selected (1) to temporarily unselected (0) and vice versa.

The Mutation Operator

This operator is used to modify the genes of a chromosome selected with the aid of a mutation probability to achieve more randomness (ABDOUN et al. 2012).

A mutation operation is applied on new chromosomes that are generated from the crossover operation. This operator changes the status of fuzzy rules being stored in genes of each layer from selected status (1) to temporarily unselected status (0). Based on the probability of mutation, the number of genes is selected randomly. The multi-layer GA was used to test only unique (i.e. non-repeated) chromosomes. Any repeated chromosomes will be discarded as there is no point to test these chromosomes again. This repetition increases wastes time and lead to long computational efforts. see figure 3.17 for the mutation operator.

Chromosome

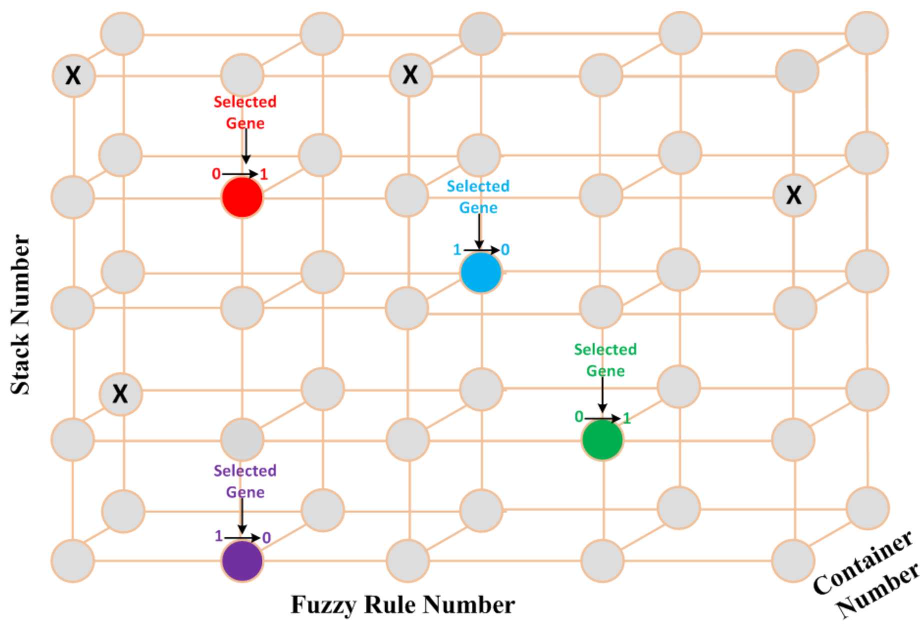


Figure 3.17: The Mutation of genes in Multi-layer chromosome

The crossed genes in figure 3.17 are empty genes that do not include fired fuzzy rules. This operator excludes any crossed genes from the mutation operation and considers only genes with fired fuzzy rules. . In each chromosome, the equipped genes are randomly selected across all layers with an equal chance to change their status from selected (1) to temporally unselected (0) and vice versa.

Stopping Criterion

There are a number of methods that can be used to stop the Genetic Algorithm loops for population generations such the need to produce a desired solution or to limit the execution time of the algorithm. The popular one is to decide beforehand the number of iterations to be executed. Another criterion to stop the Genetic Algorithm process is when the best fitness values are obtained over the generations. In this study, the stopping criterion used to stop the algorithm was the number of iterations to be executed and was decided beforehand.

3.6 Chapter Summary

In this chapter, the framework was developed to identify the inputs, processing, and outputs of the proposed container stack allocation system. The development of the 'FKB_GA' system was illustrated by the use of a conceptual model diagram. The Fuzzy Knowledge Based model was explained taken into consideration real life factors in details. An example using the FKB model was presented in this chapter. An 'ON/OFF' strategy using an approximation algorithm of time incremental of container duration of stay was proposed. An example has been introduced using approximation algorithm of time incremental of container duration of stay. The development of the multi layers Genetic Algorithm model was discussed and the modification of the chromosome structure, selection method and other operators were addressed.

CHAPTER 4

CONTAINER YARD OPERATIONS SPECIFICATION MODELLING AND INTERFACE DESIGN

4.1 Introduction

This chapter is an extension to the previous chapter in terms of the tools and techniques used to develop the 'FKB_GA' system. In the previous chapter, Fuzzy Knowledge Based Rule Modelling (FKB), Discrete Event Simulation (DES), and Genetic Algorithm (GA) have been used to develop the 'FKB_GA' system. The FKB modelling is used in order to solve the container-stack allocation problem where the departure time is unknown. This technique is also used to model the re-handling operation of containers during the collection (retrieval) operation. Discrete Event Simulation is used to mimic the overall yard operations including the arrival of trains. Finally, a Genetic Algorithm is used to optimise rules from the fired fuzzy rules for each container in the stack, which will assess the stacks and obtain optimised acceptability levels for the container storage operation.

In addition, a number of system designed techniques including UML diagrams, Activity diagrams, Class diagrams and Sequence diagrams are developed in this chapter to present the specification of the 'FKB_GA' system.

A user interface is designed to accommodate large number of inputs that are required to produce the storage allocation performance. In addition, MS Office Excel software is used in order to develop a platform for the solution of the system. Within the software, the activity within the virtual container yard operation is simulated with Visual Basic for Applications (VBA).

4.2 Specification Diagrams

UML is the standard language for object-oriented modelling of software applications (Booch et al. 1999). UML provides system architects with a visual language for specifying, constructing and documenting the artefacts of software systems. In particular, user interfaces should be visually modelled in order to describe the behaviour of the system in response to user interactions (Lodderstedt et al. 2002). A UML Use Case Diagram is used for extracting the main user interfaces. The use case diagram is described by means special kinds of UML activity diagrams, called user-interaction diagrams, whose state represents data output actions and transitions represent data input events. This perspective allows the designer to model the user interaction (i.e. input-output interaction) in each main user interface for container yard operations. Each input and output interaction of the user-interaction diagrams allows the designer to extract a class diagram for container yard operations management. Finally, a sequence diagram is described in terms of the behaviour of various objects within a use case to show the collaborations among the model objects.

4.2.1 Use Case Diagram

The UML Use Case Diagram is used to depict the behaviour of the user (i.e. actor) in relation to the system entities and activities (i.e. subjects). See figure 4.1 for the Use Case Diagram for container yard operations.

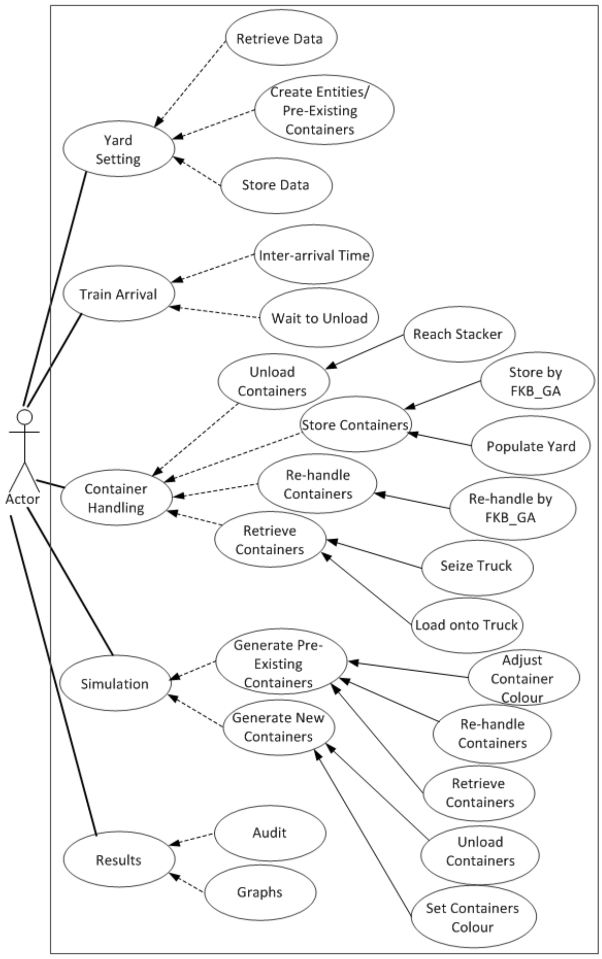


Figure 4.1: Use Case Diagram

In the Use Case Diagram presented in figure 4.1, the user/actor is having access to and is able to control: yard setting, train arrival, container handling, simulation and results. Through manipulation of the *yard setting*, the user is able to retrieve data, check integrity, and create entities or pre-existing containers and store data, according to the requirement for the yard setting. Another behaviour of the user is to *update train arrival*. The "inter-arrival" and "wait to unload" form the basis of this update.

The *container handling* enables the user to unload, store, move (re-handle), and retrieve containers. However, Fuzzy Knowledge Based Genetic Algorithm ('FKB_GA') is the techniques that enable respectively the correct storage and moving of the containers. The system populates the container yard area in which these containers can be stored. It seizes an available truck to retrieve the containers ready for delivery.

The user performs the system *simulation* through its interface by retrieving existing containers, adjust colour and check re-handling. Before applying the simulation, the user can also add new containers, unload counter and set new container colour.
After simulation run and results are generated, the user can have access to the *results* to audit and obtain graphical representation.

4.2.2 Activity Diagram

Another useful UML diagram is the activity diagram. This is used to describe the dynamic characteristics of the container yard operations management system. The activity diagram is usually in the form of a flow chart that represents the flow of one activity to another. In this description, the activity is referred to as system operations, and therefore the flow is between the actor, system user and the system. There are 7 activity diagrams presented for the system;

- Activity diagram for yard configuration.
- Activity diagram for train arrival.
- Activity diagram for container handling (unloading).
- Activity diagram for container handling (storage).
- Activity diagram for container handling (retrieve).
- Activity diagram for container handling (re-handling).
- Activity diagram for simulation and result.

4.2.2.1 Activity Diagram for Yard Setting

The activity diagram of yard configuration shows the actor as the system user as illustrated in figure 4.2. The system user is able to open the user form to retrieve previously used Data. From the retrieved data page, the user is able to inform and apply the yard levels, rows and columns. The same activities apply for specifying the number of replications and pre-existing containers, as well as assigning the owner, company and truck. The user is also able to define the train and container as well as apply reach stacker rules. However, all the parameters input by the user will be checked by the system. If they all okay, the system creates system entities and then the data is stored for future retrieval and usage, otherwise, the system returns an "incomplete data" or "system error" for the user to re-enter the correct data.

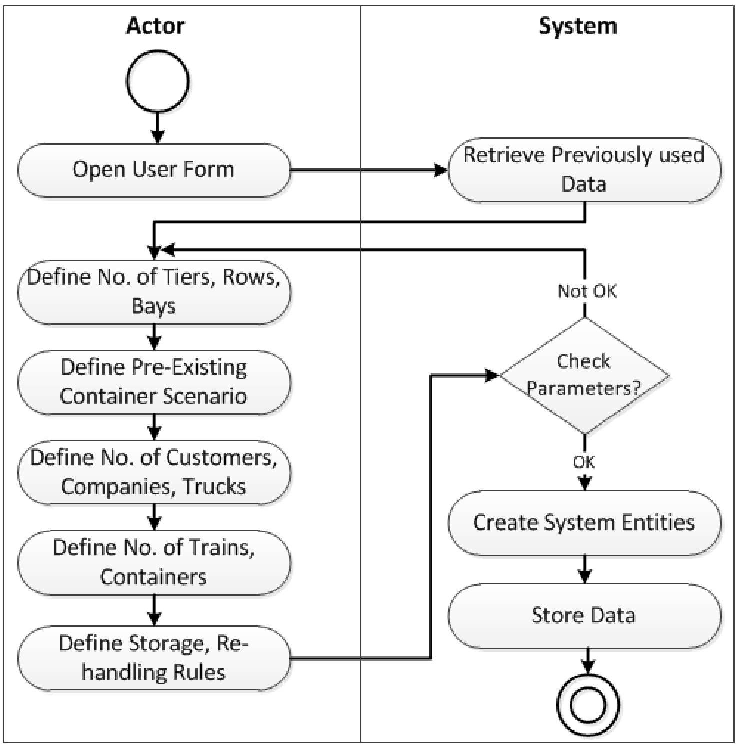


Figure 4.2: Activity Diagram for Yard Setting

4.2.2.2 Activity Diagram for Train Arrival

The activity diagram for train arrival shows the activities that occur between the *system* and the *train control* entity as shown in figure 4.3. The *system* uses the information it contains to check for pre-existing containers. If there are pre-existing containers, then the *train control* places the pre-existing containers randomly and returns to the system. If not, the *system* then checks the inter-arrival time for next train. Through the *train control*, the *system* determines if the next train has arrived or not. If the next train has arrived, the *train control* adds the train to the queue to be unloaded. The *train control* follows through by checking if the previous train has been fully unloaded after which the next train can be unloaded. The system repeats the necessary actions until all trains are unloaded.

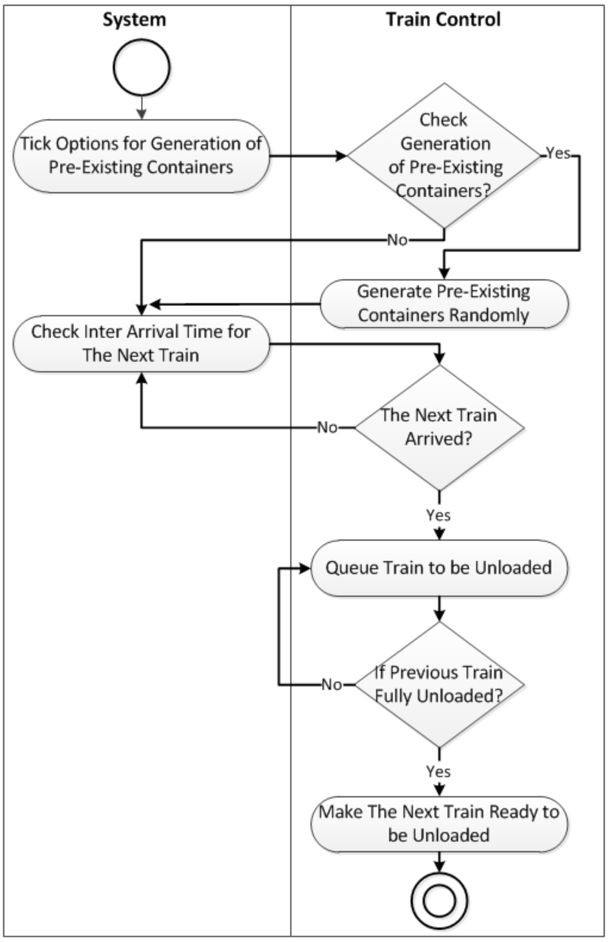


Figure 4.3: Activity diagram for train arrival

4.2.2.3 Activity Diagram for Container Handling (Unloading)

The activity diagram for container handling for the unloading process presents activities between the *system* and the *unload container* as shown in figure 4.4. The *system* finds a train ready to unload container and checks with the *unload container*, if the current train has been fully unloaded, the condition ends the activity. However, if the current train has not been fully unloaded, the *unload container* checks if there is space in the yard to complete this unload and loops until there is space available.

When space becomes available, the *unload container* also checks if the reach stacker is available and loops until it is available when the container is unloaded. The same activities are repeated by the *unload container* until all trains are fully unloaded.

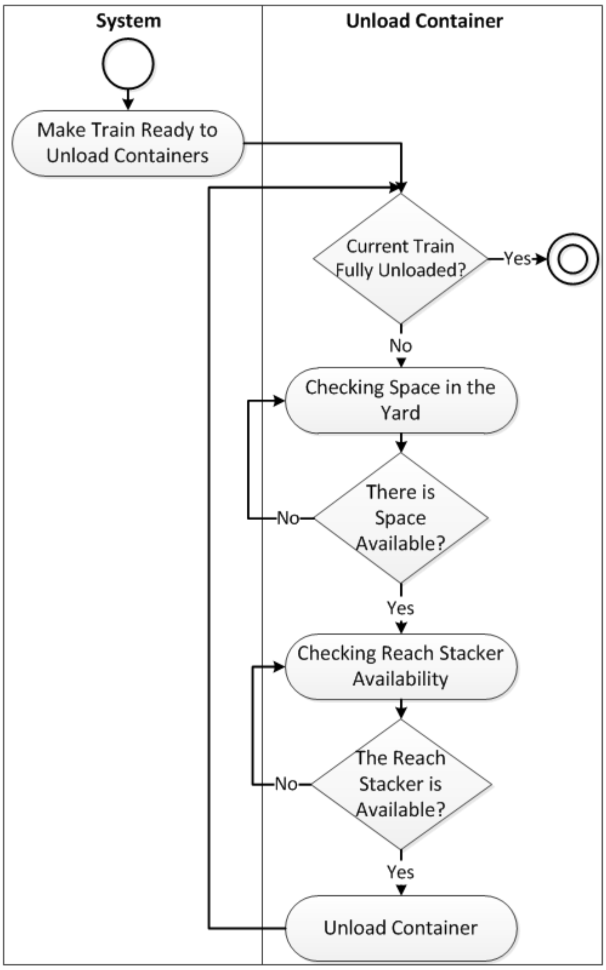


Figure 4.4: Activity diagram for container handling (unloading)

4.2.2.4 Activity Diagram for Container Handling (Storage)

The activity diagram for container handling for the storage operation shows the various activities between the *system* and the *store container* as shown in figure 4.5. The system accesses the container handling for the unloading through applying the fuzzy knowledge based genetic algorithm container storage. The fuzzy knowledge based genetic algorithm calculates the acceptability level for storage the container taking into account similarity and number of containers per stack, as well as duration of stay, weight and size of the topmost container in the stack. Then the fuzzy membership function is applied according to the information obtained before the container is stored.

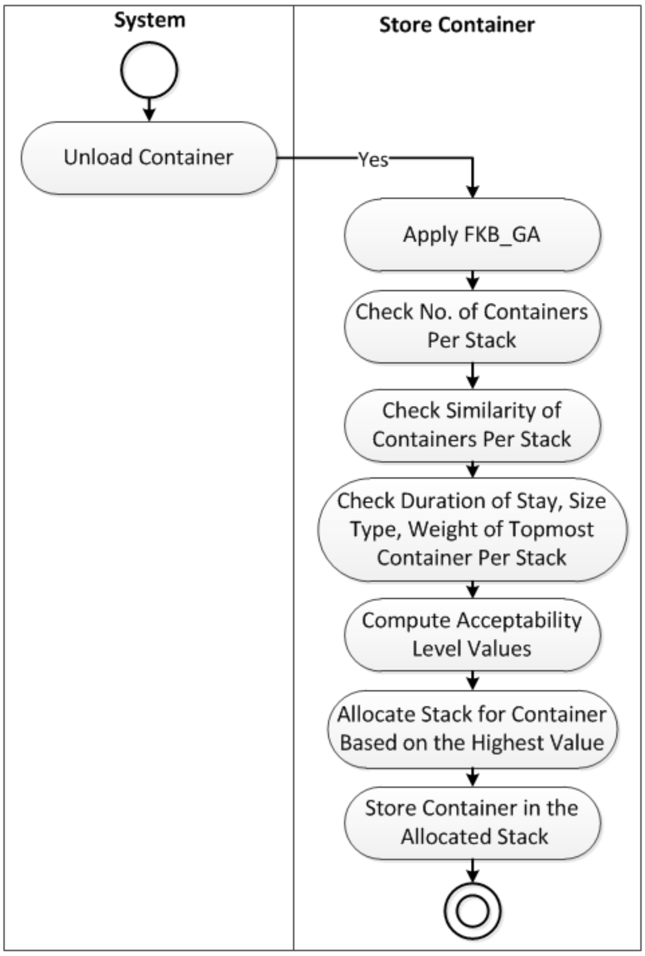


Figure 4.5: Activity diagram for container handling (storage)

4.2.2.5 Activity Diagram for Container Handling (Retrieval)

The activity diagram for container handling for the retrieving process shows the activities between the *system* and the *retrieve container* as shown in figure 4.6. The *system* identifies the container to be retrieved and checks if the reach stacker is available. If the reach stacker is not available, the *system* terminates. If the reach stacker is available, the *system* goes through the *retrieve container* to further determine truck availability. If there is no available truck, the *system* terminates. If a truck is available, the system moves the container on top of the stack unless the top container is to be moved when the container handling activity is applied for the move.

In the case where there is no top container to be moved, containers are simply retrieved and delivered to the truck, then the number of containers per stack is adjusted and the length, weight, and size of the container at the top of the stack updated before applying container handling for the container return. A successful move of all relevant containers will terminate the activity.

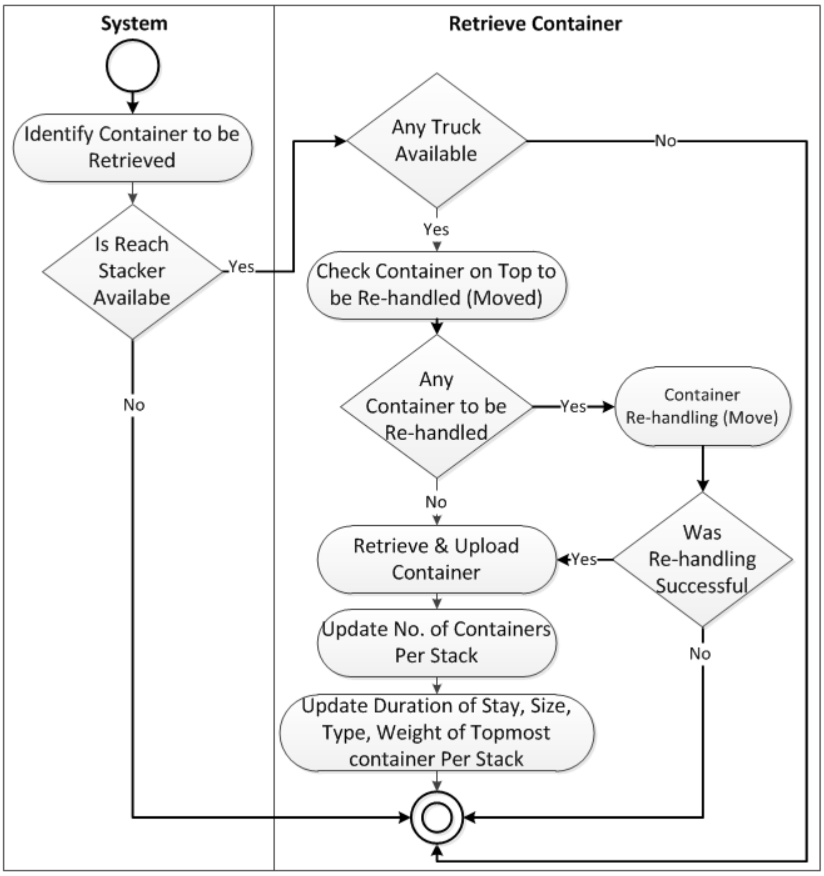


Figure 4.6: Activity diagram for container handling (retrieve)

4.2.2.6 Activity Diagram for Container Re-handling

The activity diagram for container handling for moving the containers represents the activity between the *system* and the *retrieve container* as shown in figure 4.7. The *system* checks with the *retrieve container* to see if there is an available stack for the container to be received. In the *retrieve container*, if a stack is not available re-handling will be unsuccessful and the movement of the container will terminate. If a stack is available the decision is to apply FKB_GA approach will be used to successfully move container.

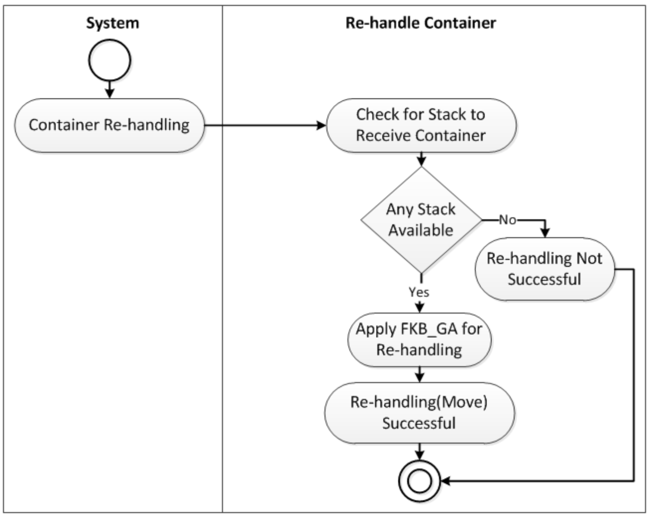


Figure 4.7: Activity diagram for container handling (re-handling)

4.2.2.7 Activity Diagram for Simulation and Results

The activity diagram for simulation and result depicts the system simulation and result related activities as shown in figure 4.8. Through the *system*, the yard map can be shown which uses *simulation/results* to find a pre-existing container. If there are no pre-existing containers, the fuzzy membership function is applied, in order to show new container unload and handling for storage the container. The simulation will also show the container handling activities such as move retrieve and return. This is done before the simulation results are computed to show the results for the container yard, and the creation of graphs.

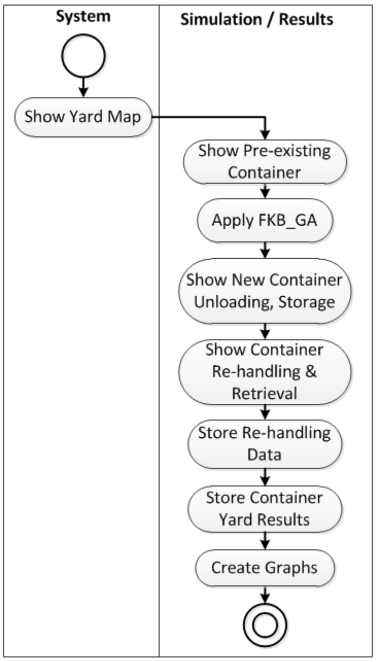


Figure 4.8: Activity diagram for simulation and result

4.2.3 Class Diagram

The class diagram is considered as an important modelling technique for object oriented systems because of its versatility in mapping other object oriented languages. See figure 4.9 for the class diagrams of the container handling processes.

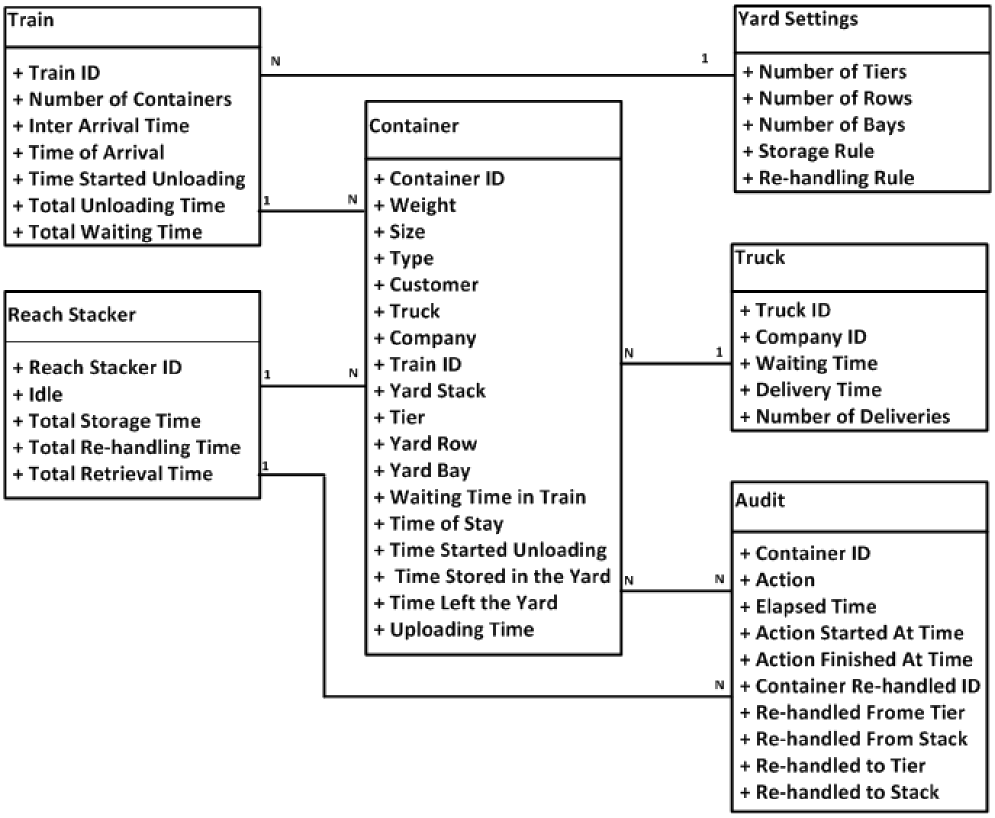


Figure 4.9: Class Diagram

The class diagram shown in the figure 4.9 is a collection of classes, associations, and constraints involved with container handling processes. There are five major classes identified such as the train, reach stacker, container, yard configuration, truck and audit. The class of “train” includes the primary identifier train ID and other properties such as number of containers, inter arrival time, time of arrival, time started unloading, total unloading time and total waiting time. The class “train” has a many-to-one relation with the class “yard configuration”, indicating that the yard configuration can have many trains. The class “train also has a one to many relationships with the class “container”, which implies that a train can carry many containers.

The class “yard configuration” is identified by the number of container level together with other properties such as the number of rows, number of trains, number of columns, number of owners, number of companies, placement rules and relocation rules.

The class “reach stacker” is identified by reach stacker ID and other properties which include idle time, total placing time, total relocating time, total retrieving time, and total retuning time. The “reach stacker” class has two relationships associated with it; a one to many relationships with class “container” and another one to many relationships with class “audit”. The relationship with class container means that a reach stacker can be used for more than one container and can be audited multiple times.

The class “container” has container ID as the primary identification, with serial number, size, weight, owner, truck, customer, train ID, yard stack, yard level, yard row, yard column, waiting time in train, time of stay, time stated unloading, time placed in the yard, and delivery time as properties. Class container has a many to one relationship with class “train”; class “reach stacker”; and class “truck” and has a many to many relationships with audit. 2 or more containers can be carried by a train, reach stacker, and truck. However, many containers can have many audits.

The class “truck” is identified by the truck ID and has company ID, waiting time, delivery time and number of delivery as properties. Container ID is used to identify class “audit”, which has entities such as action, container ID, elapse start time action start time, action finished time, container re-handling ID re-handling from level, re-handling Stack re-handling to level and re-handling to stack.

The class diagram generally shows the relationship among the various classes and their properties in order to match relevant classes together when performing customer container handling and re-handling operations.

4.2.4 Sequence Diagram

A sequence diagram is used to examine various objects behaviour within a use case. The sequence diagram is presented to show collaborations among the model objects. In the sequence diagram for container handling presented in figure 4.10, the interaction between the train, stacker, container and truck are shown.

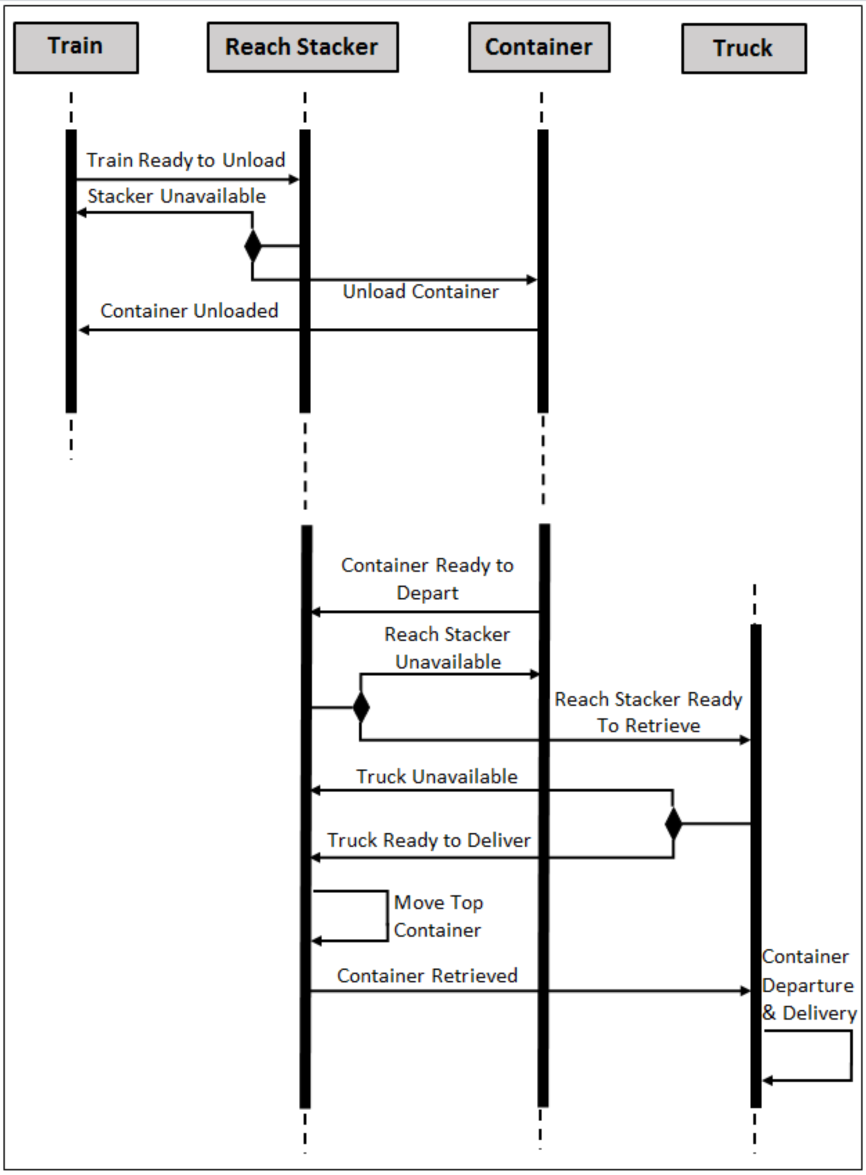


Figure 4.10: Sequence Diagram

When the train arrives and is ready to be unloaded, the stacker is checked for availability. If the stacker is not available, the train with containers on it will continue to wait until the stacker become available. The stacker then unloads containers from the train into the container yard. Containers remain in the container yard until they are ready to be delivered. At any given point of time, any type of container can be requested for delivery. In which case, retrieving containers cannot be sequential as they have been loaded unto the container yard. The truck is used to move and replace containers into stacks within the container yard so that specific containers that have been requested for delivery can be retrieved. When the type of container requested for delivery is at the top of the stack, they are ready to be moved. Also, when the type of container requested for delivery needs to be retrieved, the stacker is required. If the stacker is not available for the particular retrieval, there will be a delay until the stacker is available. When available the stacker moves the top containers to retrieve the requested container for delivery and then when this action is complete collects and returns the containers back to the stack in the container yard. These are the types of sequential activities that take place in container handling, unloading and retrieval and the reason for adopting the sequence diagram to explain this behaviour.

4.3 Data Flow Specification Modelling

In order to develop the 'FKB_GA' system, Microsoft Office Excel together with a VBA add-in was used. The Excel file consisted of different profiles including: *'Container Location Map', 'FKB model for Storage and Retrieval Operations', 'Pre-existing containers', 'Arrival of Container Trains', 'ON/OFF of DoS', 'New Containers Storage', 'Containers Retrieval,', 'Fuzzy Fired Rules', 'Coding of Genes', 'Sorting & Selection of Chromosomes', 'Results', and 'Results-Graphs'.*

The system was comprised of two main models which included the Fuzzy-Knowledge Based-model (FKB) and the Genetic Algorithm (GA) model. Each of these models is explained in detail in Chapter 3, section 3.4. These models consist of data that flows between different processes of the storage and retrieval operations for containers.
flow diagram for the 'FKB_GA' system.

4.3.1 Data Flow Specification Model of 'FKB_GA'

The data flow of the FKB Model integrated with the GA Model is discussed. The integration of these two models produces the FKB_GA system for the stack allocation process in the store and retrieve operations for containers. Figure 4.11 shows the data flow diagram for the 'FKB_GA' system.

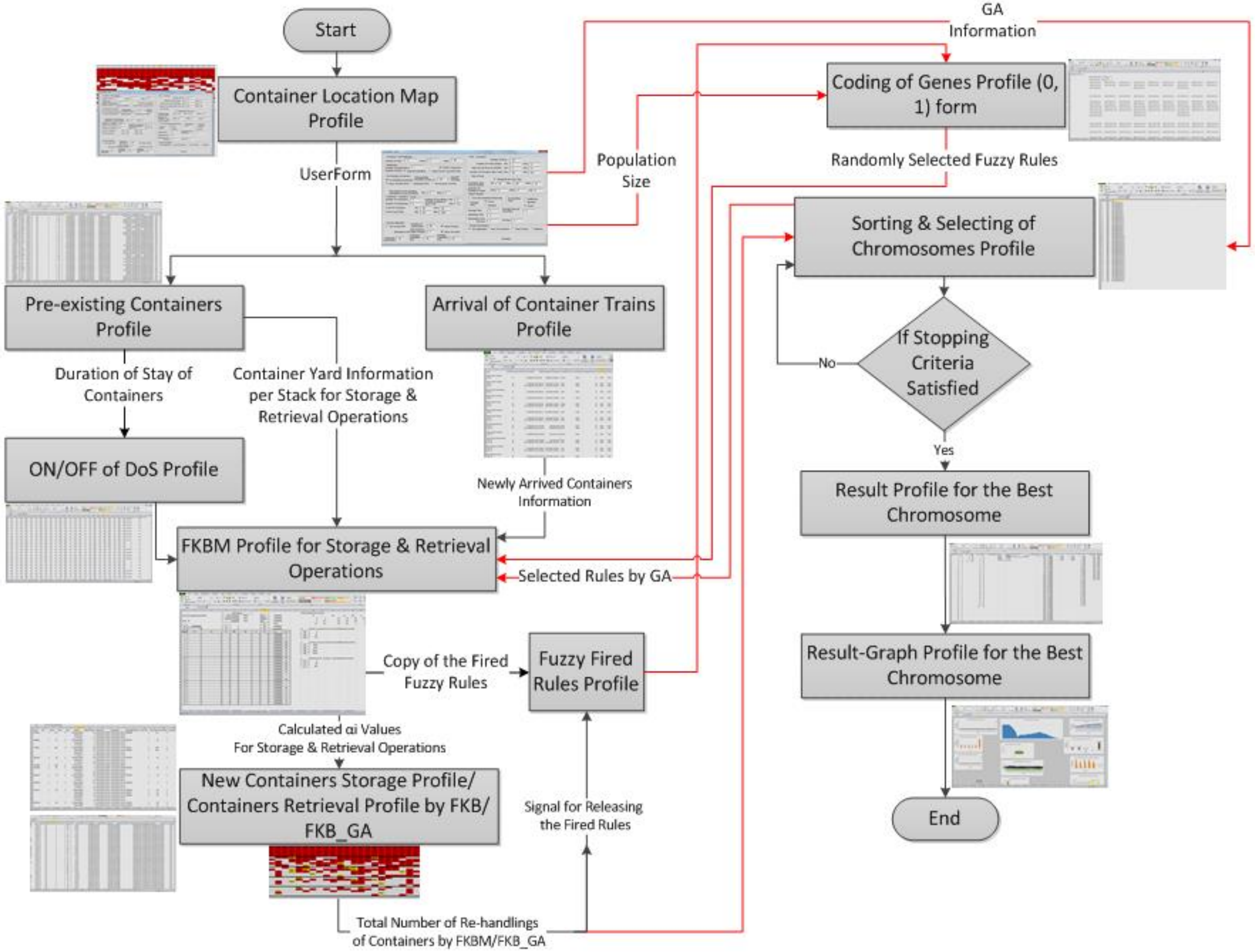


Figure 4.11: Data flow diagram for the ‘FKB_GA’ system

The flow of data starts by interrogating the *'Container Location Map profile'*. This profile includes a UserForm interface and storage and container yard operations. The required inputs are provided by the user using UserForm. In order to generate pre-existing containers, some data is fed from the UserForm to the *'Pre-existing Containers Profile'*. In the *'Arrival of Container Trains Profile'*, the containers which have newly arrived by trains are generated based on information which comes from the UserForm. Regarding the *'ON/OFF of DoS Profile'*, the Duration of Stay of containers is passed to this profile by the *'Pre-existing Containers Profile'* to activate/deactivate the DoS factor. The information from *'Arrival of Container Trains Profile'*, *'ON/OFF of DoS Profile'*, and *'Pre-existing Containers profile'* is fed to the *'FKBM Profile'*. Based on the fired fuzzy rules per stack and container, the *'FKB model Profile'* calculates the α_i values for the storage and retrieval operation for containers.

The α_i values calculated either by the 'FKB' Model or by the 'FKB_GA' System are passed to the *'New Containers Storage Profile'/'Container Retrieval Profile'*. A copy of the fired fuzzy rules is stored in the *'Fuzzy Fired Rules Profile'*, and these rules will be on hold until the storage and retrieval operation of containers by the FKB model are complete. Then the total number of re-handlings of containers is obtained and the stored fuzzy rules in the *'Fuzzy Fired Rules Profile'* are passed to the '*Coding of Genes Profile'* to start the optimisation process. The *'Coding of Genes Profile'* is fed with the population size from the UserForm to generate the initial population randomly (i.e. using randomly selected fuzzy rules). These selected rules are provided to the *'FKB model Profile'* for both storage and retrieval operations to recalculate the α_i values for the stack allocation process. After the storage and retrieval operations are completed by the 'FKB_GA' system, the total number of re-handlings per chromosome is obtained. For further improvements in the number of re-handlings, the chromosomes with their number of re-handlings are provided to the *'Sorting and Selecting of Chromosomes Profile'*. In this profile, the chromosomes are sorted in ascending order based on their number of re-handlings. The '*Sorting and Selecting of Chromosomes Profile'* is fed with GA information (i.e. the probabilities for crossover over and mutation of genes). The selected fuzzy rules (new generation) by the GA are passed to the *'FKB model Profile'* to recalculate the optimised α_i values for the storage and retrieval operation for containers. The GA loop continues until the stopping criteria is satisfied, if it is satisfied then the GA loop ends. Then the results for the best chromosome will be generated in the '*Results Profile'*, and the result graphs for the best chromosome will be generated in the *'Results-Graph profile'*.

The data flow of each of the models will be discussed in more detail in the next section.

4.3.2 Data Flow of the 'FKB' Model

In this section, the data flow of the 'FKB' Model is explained and presented. The 'FKB' Model uses different data in order to allocate stacks for the container storage operation. Figure 4.12 shows the data flow diagram for the 'FKB' Model.

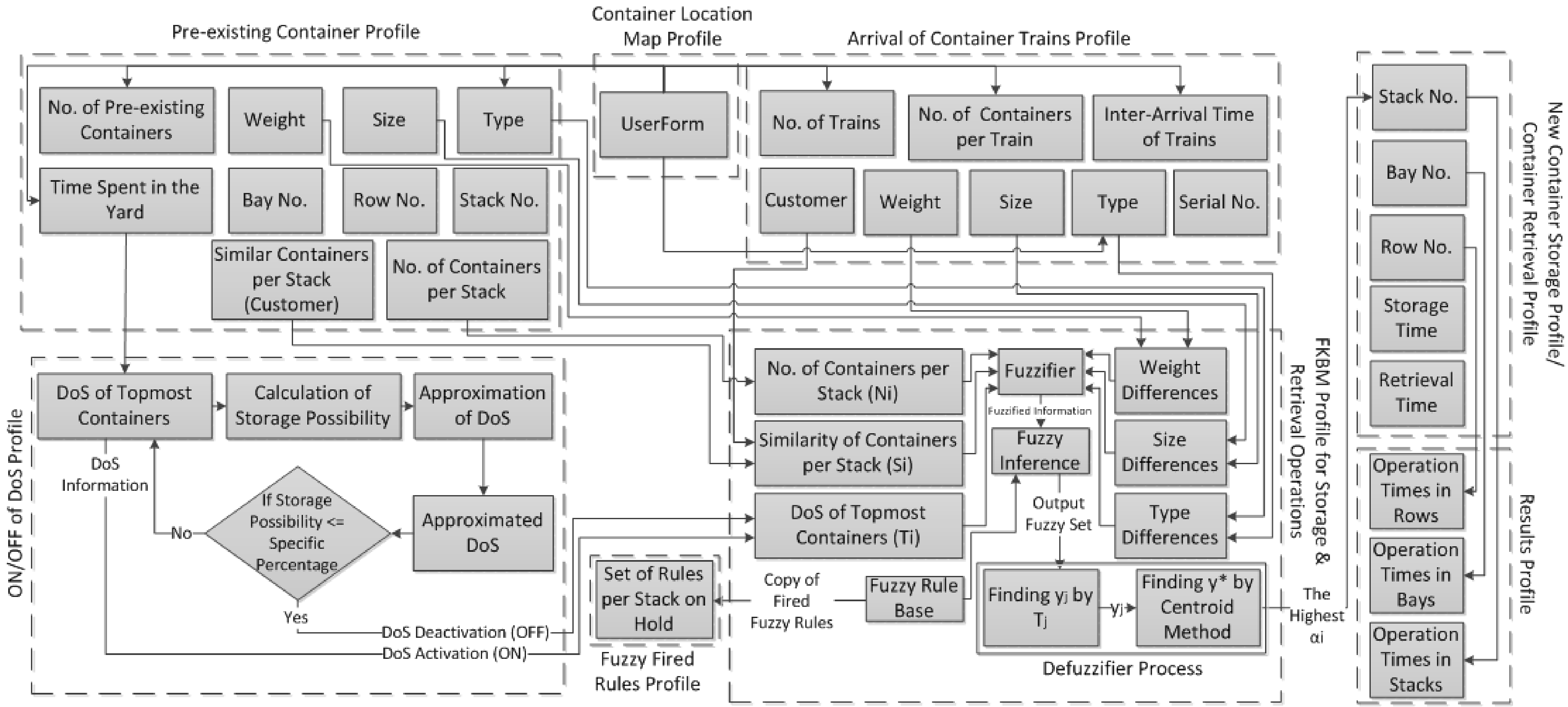


Figure 4.12: Data flow diagram for the FKB model

Referring to figure 4.12, the *'Container Location Map Profile'* includes two main parts; the UserForm interface and container yard operations. This interface involves the simulation of container yard operations using the discrete event approach.

Through this profile, a user interface can be initiated to define the inputs. After identifying the inputs, the *'Pre-existing Containers Profile'*, *'Arrival of Container Trains Profile'* and '*FKB model for Storage & Retrieval Operations Profile'* take information from the UserForm. The information for the *'Pre-existing Containers Profile'* includes the number, weight, size, types, similarity, and time of containers spent in the yard. Based on this information, the pre-existing containers are generated in the *'Pre-existing Containers Profile'*. For the *'Arrival of Container Trains Profile'*, the information consists of the number, size, type and similarity of newly arrived containers in each train. The information for the *'FKB model for Storage & Retrieval Operations Profile'* including the three input factors and constraints for storage and retrieval operations of containers are presented in section 3.4.1.1. The Duration of Stay of the topmost containers in each stack is provided from the 'Pre-existing Containers Profile' to the *'ON/OFF' of DoS Profile'* in order to activate (ON)/deactivate (OFF) the Duration of Stay factor by the *'FKB model for Storage & Retrieval Operations Profile'*. The ON/OFF strategy is explained in detail in section 3.4.1.4. The *'FKBM for the Storage & Retrieval Operations Profile'* is also fed by the newly arrived container information from the *'Arrival of Container Trains Profile'*.

After all the required data including the input factors and constraints are provided to the *'FKB model for Storage & Retrieval Operations Profile'*, the FKB model calculates the acceptability level values (α_i) for each stack based on the fired fuzzy rules in both the operation storage and retrieval of containers. Then, the calculated acceptability level values by the FKB Model are fed to the *'New Containers Storage Profile'* and *'Container Retrieval Profile'* for stack allocation. The stack with the highest α_i is allocated for container storage. A copy of the fired fuzzy rules per stack and containers is stored in the *'Fuzzy Fired Rules Profile'*. Once the storage and retrieval operations for all containers are completed, then the results are provided for the user in the '*Results Profile'*.

4.3.3 Data Flow of the 'GA' Model

As an optimisation engine, a GA Model is developed to be integrated with the FKB Model for optimal storage and retrieval operations of containers. The GA model consists of the different data that is required for the optimisation of the storage and retrieval operations for containers. Figure 4.13 illustrates the data flow diagram for the 'GA' Model.

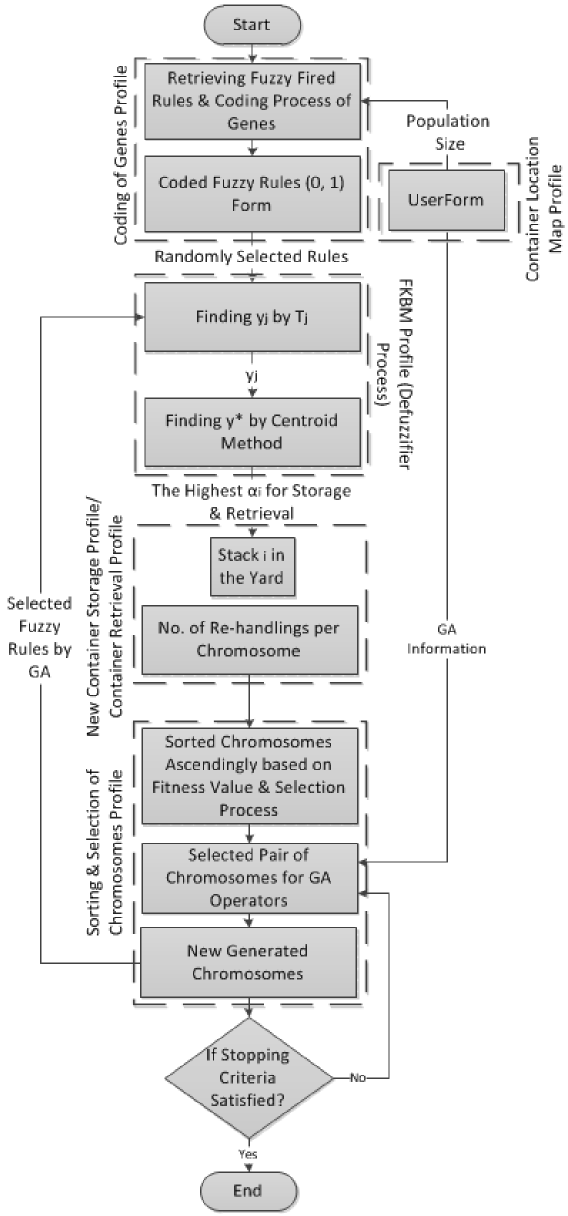


Figure 4.13: Data flow diagram for the 'GA' model

In figure 3.20, the data flow starts with *'Coding of Genes Profile'*. In this Profile, the fuzzy fired rules are retrieved and coded in (0, 1) form randomly. The population size of each generation is provided from the *'Container Location Map Profile'* to the *'Coding of Genes Profile'*. The randomly selected fuzzy rules are provided to the defuzzifier process in the *'FKB model Profile'* to calculate the α_i values. The calculated α_i values are fed to the *'New Container Storage Profile'* and '*Container Retrieval Profile'*, the highest α_i is selected to store containers in stacks. Once the storage and retrieval operations are complete, the number of re-handling of containers per chromosome is obtained. The number of re-handlings of containers is passed to the *'Sorting & Selection of Chromosomes Profile'*. GA information including the probability of crossover and mutation of genes, together with the number of generations are fed to the *'Sorting & Selection of Chromosomes Profile'*. In this profile, the chromosomes are sorted in ascending order based on the number of re-handlings, and each pair of chromosomes is selected and passed to the GA operations to produce new chromosomes (i.e. fuzzy rules selected by the GA). The fuzzy rules selected by the GA are sent to the de-fuzzifier process in the *'FKB model Profile'* for the α_i values calculation for stack allocation. When the stopping criterion is satisfied, then the GA model loop is completed.

4.4 The 'FKB_GA' System User Interface

In order to develop a platform for the solution of the model, Microsoft Office Excel is used. Within the software, a user interface is programmed in Visual Basic for Applications (VBA) Figure 4.14 shows how the system is developed and shows how the input parameters are fed through the UserForm by the user. The system consists of three main parts including the UserForm, arrival of new containers by trains, and simulation of container yard operations.

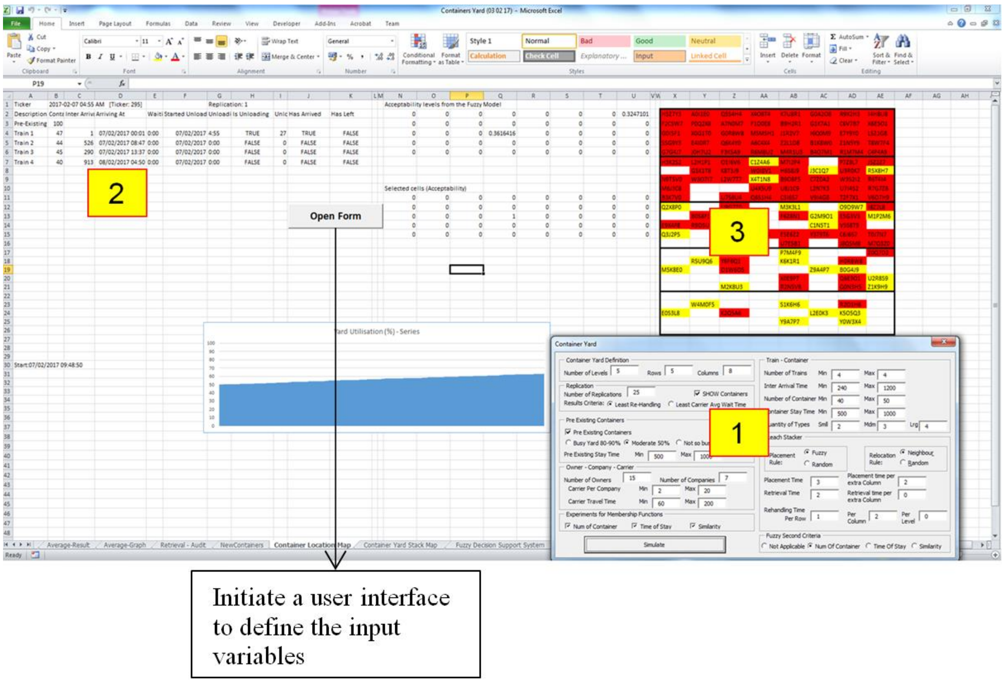


Figure 4.14: The Developed System

The UserForm is the main part of the system, all parameters are inserted by the user through this form in order to start arrival of new container by train and yard operations. Figure 4.15 illustrates the UserForm of the system.

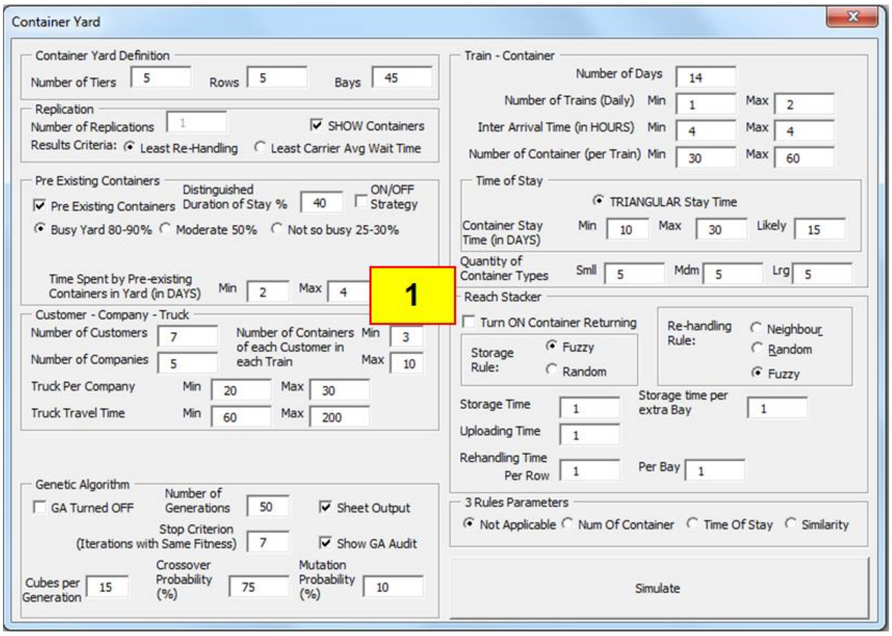


Figure 4.15: UserForm of the developed system

This form includes container yard definition which means the yard size. The yard size can be defined by the number of: bays, rows, and levels. The number of pre-existing containers can be defined in this form by ticking one of the options including busy yard, moderately busy yard, and not so busy yard. The stay time of the containers can be specified by the user by specifying minimum and maximum values. The containers stay in the yard according to these values and depart randomly. The new containers arrive by trains and are stored with pre-existing containers. The number of these trains can be specified by the user which is specified as minimum and maximum values.

The inter-arrival time of these trains is random and can be also be input by the user. The containers have different size and types which can be provided by the user through the UserForm. The form contains the storage and re-handling strategy, this can be indicated by choosing either fuzzy or random as storage strategies, and neighbourhood and random as re-handling strategies. Storage time, retrieval time and re-handling time for containers can be provided to the system using this form.

The second part of the system is the arrival of new containers by train. This part was simulated using the discrete event simulation model, where a number of trains arrive with different a number, size, and type of containers. Figure 4.16 shows the arrival of containers by trains.

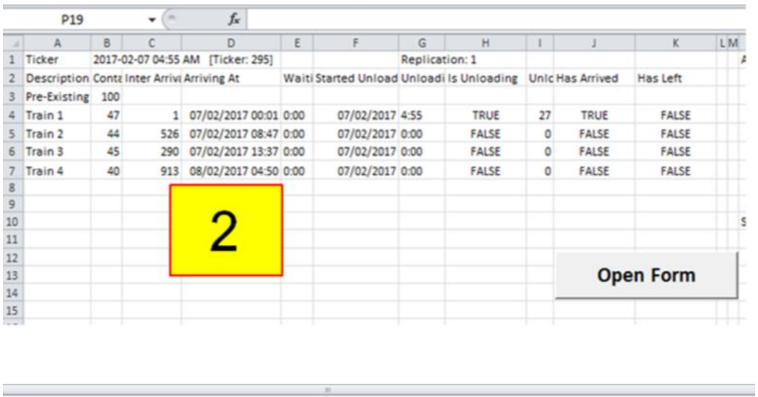


Figure 4.16: Arrival of Containers by Trains

As shown in figure above, a number of trains arrived with a different number of containers. There is an inter-arrival time and arrival date and time for each train. Once the train arrives, the containers from that train will be unloaded one by one. If there is a train already unloading containers, then the train has to wait until the first train finishes unloading containers. Once all containers of the train are unloaded, then the train will depart. The next train will then start unloading containers.

The third part of the system is the simulation of container yard operations. This part includes the storage, re-handling and retrieval operations for pre-existing and newly arrived containers. See Figure 4.17 for container yard operations of containers. The red cells are representing the pre-existing containers, and the yellow cells represent the new containers.

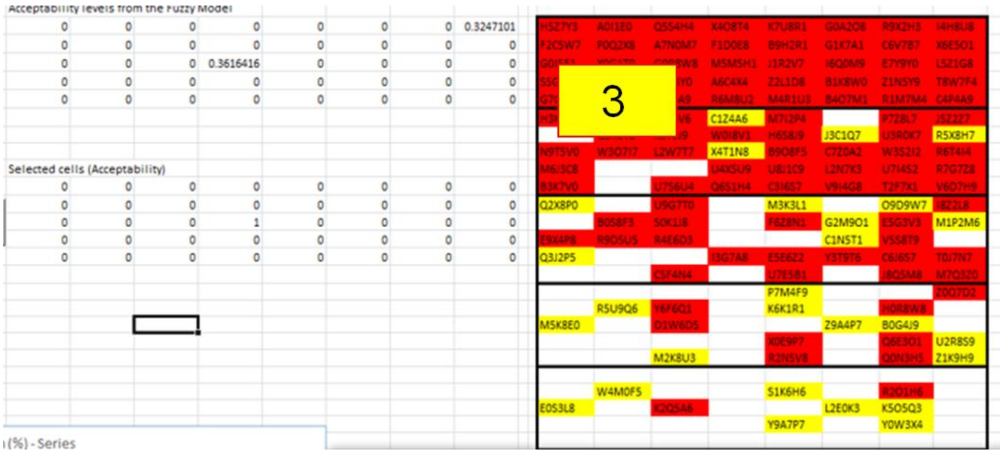


Figure 4.17: Simulation of Container Yard Operations

The Fuzzy Knowledge Based Model (FKBM) was formulated in EXCEL, this model was called through VBA code to calculate the acceptability level values of stacks. See figure 4.18 for the form of the FKBM sheet.

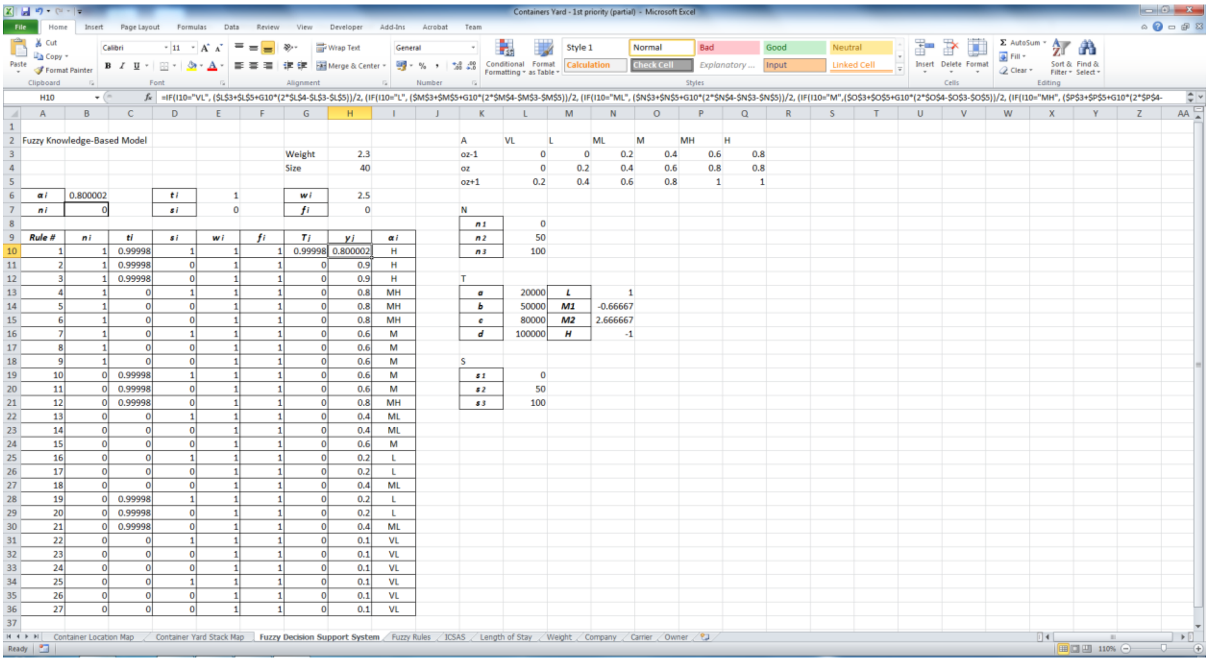


Figure 4.18: Formulated FKBM Sheet (FDSS)

As shown in figure 4.18, the fuzzy rules are defined using the linguistic variables of the input factors including number of containers (ni), duration of stay of containers (ti) and similarity of containers (si). The constraints are embedded in this sheet in order to decide to whether or not to accept containers in stacks. The linguistic variables of the acceptability level are embedded in the sheet to calculate the acceptability level values of stacks for container storage.

4.5 Chapter Summary

This chapter discussed the development of 'FKB_GA' system for management of container yard operations. The specifications of the developed system were presented in terms of UML diagrams in details. The UML diagrams included use case diagram, activity diagrams, class diagram and sequence diagram. The activity diagrams consist yard configuration diagram, train arrival diagram, container unloading diagram, container storage diagram, container retrieval diagram, container re-handling diagram, and simulation and result diagram. Data flow modelling of the 'FKB_GA' system was demonstrated for developing a prototype for a stack allocation system for the storage and retrieval operations of containers. A user interface was programmed including a UserForm to enable the user to feed inputs to the developed system.

CHAPTER 5

CASE STUDY, DATA COLLECTION AND MODEL VERIFICATION AND VALIDATION

5.1 Introduction

In this chapter, the container yard management at Maritime Company is considered as a case study. An adequate amount of information was collected and the logic of processes was captured. In this case study, the profile of Maritime Company which operates a number of depots is presented. The container yard operations of Tilbury Railport which is operated by Maritime Company is explained in detail. These operations include storage and retrieval of containers. Flowcharts of each operation and the conceptual site model of Tilbury Rail port container yard are introduced. A fishbone delay analysis diagram of the rail port is presented. This diagram is used in order to evaluate the delays that can occur during operations. All the technical information for this rail port is presented as well as the sizes and types of containers that this port operates, the capacity of the container yard, resources etc. A logical container flow diagram is also provided in this chapter to illustrate the movement of containers within the yard.

5.2 Maritime Company Profile

Maritime is one of the UK's leading multimodal transport and container service specialists, combining road, rail & storage modes to become an integral element of the customer supply chain. The company also provides highly effective UK container transport and services.

This company operates a number of depots (i.e. terminals) for containers including Tilbury Railport and Birmingham Rail. Maritime has a distribution (i.e. transportation) fleet that is spread across a number of depots including Felixstowe, Southampton, Liverpool, Manchester Bristol, Doncaster, Immingham, Leeds, London

Gateway, Milton Keynes, Northampton, Peterborough, Reading, South Shields, and Thame sport.

All relevant data for this case study is collected from the Tilbury Railport terminal. The Tilbury depot has two container yards, one for import containers and one for export containers. The warehouse offers full devan/storage/distribution of bulk steel/pallet/groupage cargo, and container re-works/transhipments. A number of basic functions are provided including empty/full container storage, container repair/cleaning, reefer container monitoring/preparation, cargo devanning/handing (in conjunction with the warehouse), car loading/unloading into containers, cross docking/transhipments (in conjunction with the warehouse), and cargo fumigation services. A number of trains are operated per day including 1 to 2 container trains and 3 to 4 steel trains.

5.2.1 Maritime Key Services

The Maritime depots are connected to the rail network providing cost effective container transport to major industrial and commercial areas. They deliver reliable, on time, independent services to shipping lines, freight forwarders as well as direct to retailers.

Maritime provides a unique framework for the transfer of containerised traffic from the principle UK ports to their final destination. Using modern handling equipment, Maritime can offer full and empty container handling, storage and repairs. It also offers office and truck accommodation, creating a unique platform for the supply chain.

The company is capable of transporting and storing containers across the United Kingdom and performs a number of operations for containers, including storage, retrieval, uploading and transportation. Many of the operations are performed using a piece of equipment called a reach stacker which unloads containers from the train and move them to a yard for storage. It is also used to retrieve containers from the yard and upload them onto trucks for delivery. The transportation operation, performed by trucks, is used for delivering containers from the depot to their destination (i.e.

customer) and vice versa. Tilbury terminal is one of the most efficient railheads in the UK. The terminal has a number of resources to handle different sizes and types of containers. The case study uses a specific number of resources and containers. The resources are a reach stacker, trucks and a container yard. The containers arrive at this terminal by trains. This terminal serves one to two container trains a day over one railway. Each train has 15 platforms which can carry 3 x 20ft containers of mixed contents (30 to 60 mixed containers), each platform can be loaded to a maximum of 60 tonnes.

5.3 Description of Operations

The description of each operation including storage, retrieval and transportation of containers are discussed. The storage operation for containers includes where and how containers can be stored in the yard. The retrieval and transportation operation for containers identifies the location where they are stored, performs re-handling if any containers on the stack are on top of the required one, then uploads them onto trucks and transports them to customers.

5.3.1 Storage Operation

The storage operation for both import and export containers at Tilbury Railport is carried out by a reach stacker. All containers are stored and stacked in an area called a container yard before they leave the terminal. The use of the container yard avoids delay of the container trains, reduces distraction of the work and decreases the handling operation time (Misliah et al. 2012). The containers are stored and stacked in this area by a reach stacker, which is used to transport containers from the train to the container yard (Skinner et al. 2013). The containers stay in this area until a delivery request for collection is placed. Some of these containers do not need to be stored in the yard and can be directly transported to the required customer destination. Figure 5.1 shows a reach stacker resource.

The container yard in this depot (i.e. terminal) is divided into two areas, one of them is to store the import containers, and the second one is to store the export containers. Both of them are the same capacity. Each yard has 45 bays and each of these bays consists of 2 to 5 rows. Different types of containers (e.g. dry van, open top, reefers, flat racks and tanks) of 20ft, 30ft and 45ft sizes can be stored in the yard.

The containers are first stored on the ground, side by side in the container yard. Then they are stacked one on top of the other. Up to four loaded or five empty containers can be stacked in this way. After the arrival of a train loaded with containers to the terminal site, the import containers are unloaded one by one from the train by the reach stacker and transported to the import container yard area to be stored in specific stacks (i.e. location). The storage time for containers from the train to the first bay in the yard is one minute. The imported containers are stored and stacked by customer.

In the export yard, containers are brought to the terminal by trucks and then unloaded by the reach stacker to be stored in specific stacks in the export yard. The reach stacker spends a minute to unload a container from a truck and store it in the container yard and one minute to upload a container from the container yard onto a truck. Figures 5.1 and 5.2 show the import yard at Tilbury Railport. Figure 5.3 shows how the four loaded containers and the five empty containers are stacked on top of each other.

Figure 5.1: a Reach Stacker Resource

Some materials have been removed due to 3rd party copyright. The unabridged version can be viewed in Lancester Library - Coventry University.

Figure 5.2: The Container Yard of Tilbury

Some materials have been removed due to 3rd party copyright. The unabridged version can be viewed in Lancester Library - Coventry University.

Figure 5.3: Containers are Stacked on top of each other

The most important operation in container terminal sites is the stack allocation for container storage in the yard. This process is dealing with how to allocate storage locations (stacks) for containers efficiently. An efficient storage allocation process reduces the storage, retrieval and re-handling time of containers in the yard. The efficiency of the container yard operation depends on the availability of information for the arrival and departure times of containers, which, for this case study is uncertain, requiring a fuzzy knowledge based model for the container storage operation.

This information is uncertain, because, the departure time of containers will not be known until the customers come to collect their containers. Additionally, the arrival time of containers being transported to the terminal by trucks is unknown. Both of these situations affect the containers yard operation time for retrieving the required containers. For example, consider the case where three containers are stacked on top of each other and the required container is the bottom one. In this case, to access the required container, two containers need to be moved (i.e. re-handled to other stacks) in order to reach the required container. When re-handling operations for containers happens frequently, this will waste time. If the containers are stacked effectively in the storage yard, the amount of container re-handling operations will be reduced. By reducing the re-handling and retrieval time the overall container yard operation time will also be reduced. All these lead to increasing the container terminal efficiency and the customer service level.

5.3.2 Retrieval and Transportation Operation

The transportation operation is one of the most important operations in the container terminal. This operation is performed by a reach stacker and trucks. The trucks are used to bring the containers in and out of the terminal. The containers prepared for import that are already stored in the storage yard can be transported by these trucks out of the terminal to customers. The trucks are also used to transport the export containers from the customers to the terminal, to be stored in the yard and then loaded onto the train. The trucks are able to transport different types and sizes of containers one container at a time. The containers on the trucks are loaded and unloaded by the reach stacker. The import containers are unloaded from the trains by the reach stacker and moved to the yard to be stacked and stored in specific bays and rows in the yard.

Then they are uploaded onto trucks to be transported to the destination required by the customer according to a schedule as shown in figures 5.4 and 5.5. Some of these containers are not stored in the yard, but are moved directly from the train and uploaded onto the vehicles by the reach stacker to be transported out of the terminal to the required customer. The export containers are brought into the terminal on trucks, then unloaded by the reach stacker and stored in the yard in specific bays and rows. Then they are moved onto the train by the reach stacker. Some of these containers are not stacked in the storage yard, but, are unloaded directly from the trucks and loaded onto the train according to a specific schedule.

Some materials have been removed due to 3rd party copyright. The unabridged version can be viewed in Lancester Library - Coventry University.

Figure 5.4: a Reach Stacker with a Container

Some materials have been removed due to 3rd party copyright. The unabridged version can be viewed in Lancester Library - Coventry University.

Figure 5.5: a Truck Resource

When a truck arrives at the terminal with a container, the driver is told by the control function the stack on which the container is to be stored. The container is then unloaded from the truck and stored by the reach stacker in the export yard. When an empty truck arrives at the terminal, the same process happens. The driver is given the by the control function and then required container is retrieved and loaded onto the truck by the reach stacker. If the required container is at the top of the stack, then the container is retrieved and uploaded onto the truck straightaway. If there are containers on top, then they are re-handled (i.e. moved) to other stacks, after which the required container is retrieved and uploaded onto the truck. The re-handling time of containers between neighbourhood bays is 30 seconds. The export containers depart the terminal by trains, but, the import containers depart the terminal by trucks.

5.4 Logical Workflow Diagrams

In order to clarify the sequence of operations and logical movement of containers in the terminal, a number of flowcharts have been developed for the storage, retrieval and transportation operations.

5.4.1 Storage Operation Flowchart

The storage operation is responsible for making decisions on where to store containers in the yard based on specific criteria. See figure 5.6 for the storage operation flowchart.

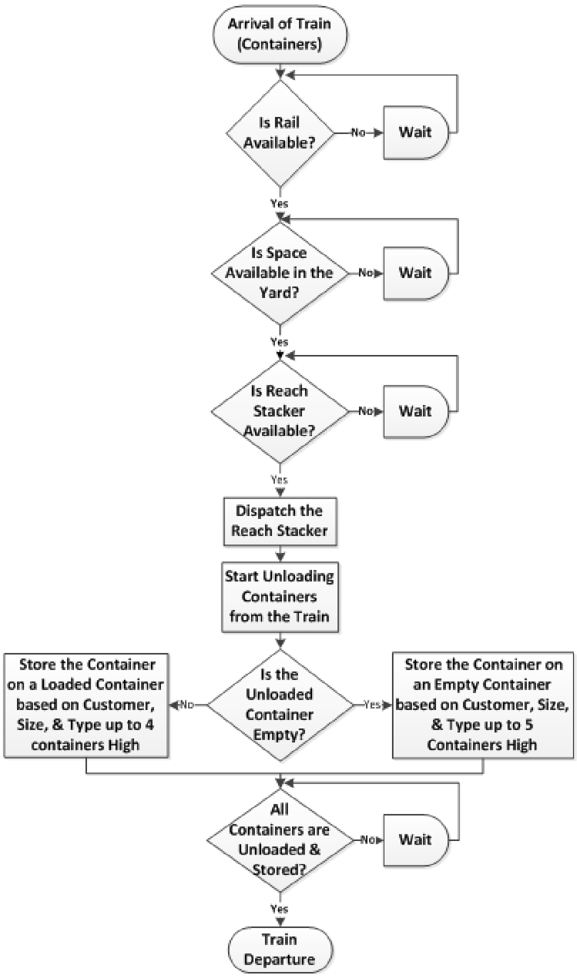


Figure 5.6: Storage Operation Flowchart

The flow for the operation starts with arrival of containers by train then is followed by checking the rail area (i.e. unloading area) availability. If the rail area is unavailable then the train waits until the unloading area is available. The space availability in the yard is checked before moving the containers from the train to the yard for storage. If the reach stacker is unavailable, then the containers wait until the reach stacker is available before moving the containers. After all import containers are unloaded and stored in the yard in stacks by customer, size and type, the stack height being limited to either 4 loaded or 5 empty containers. Containers that belong to the same customer need to be stored in the same stack, and containers that are being unloaded need to be stored on top of those of the same size and type, then the train departs.

5.4.2 Retrieval and Transportation Operations Flowchart

After containers have been in the yard for a period of time, the containers are retrieved by a reach stacker and uploaded onto trucks and transported to the customer destination. See figure 5.7 for the retrieval and transportation operations flowchart.

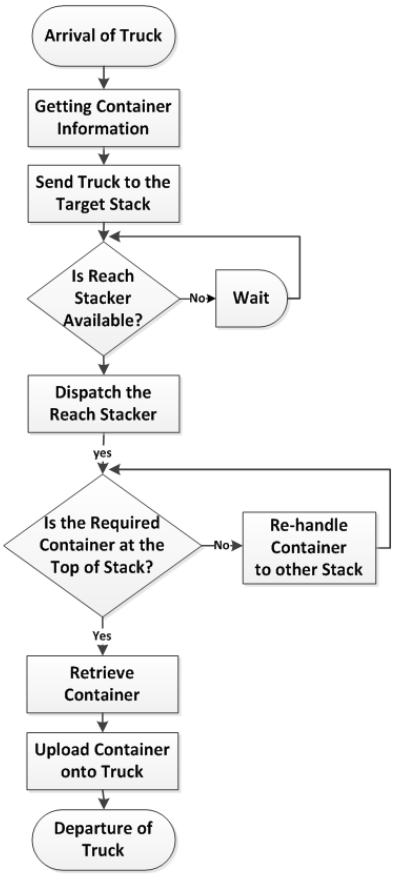


Figure 5.7: Retrieval and Transportation Operation Flowchart

The flow of this operation starts with the arrival of a truck which, after getting the container information, will be sent to the target stack to be stored. The truck will wait for the reach stacker to become available when it will begin to retrieve the required container. If the required container is not at the top of the stack, then the containers on top will be re-handled to other stacks. The required container will then be retrieved and uploaded onto the truck for departure and transported to the customer.

5.4.3 Container Yard Operations Flowchart

A flowchart was developed for the container yard operations, which included the storage and retrieval of containers. This can be seen in figure 5.8 showing the sequence of these operations.

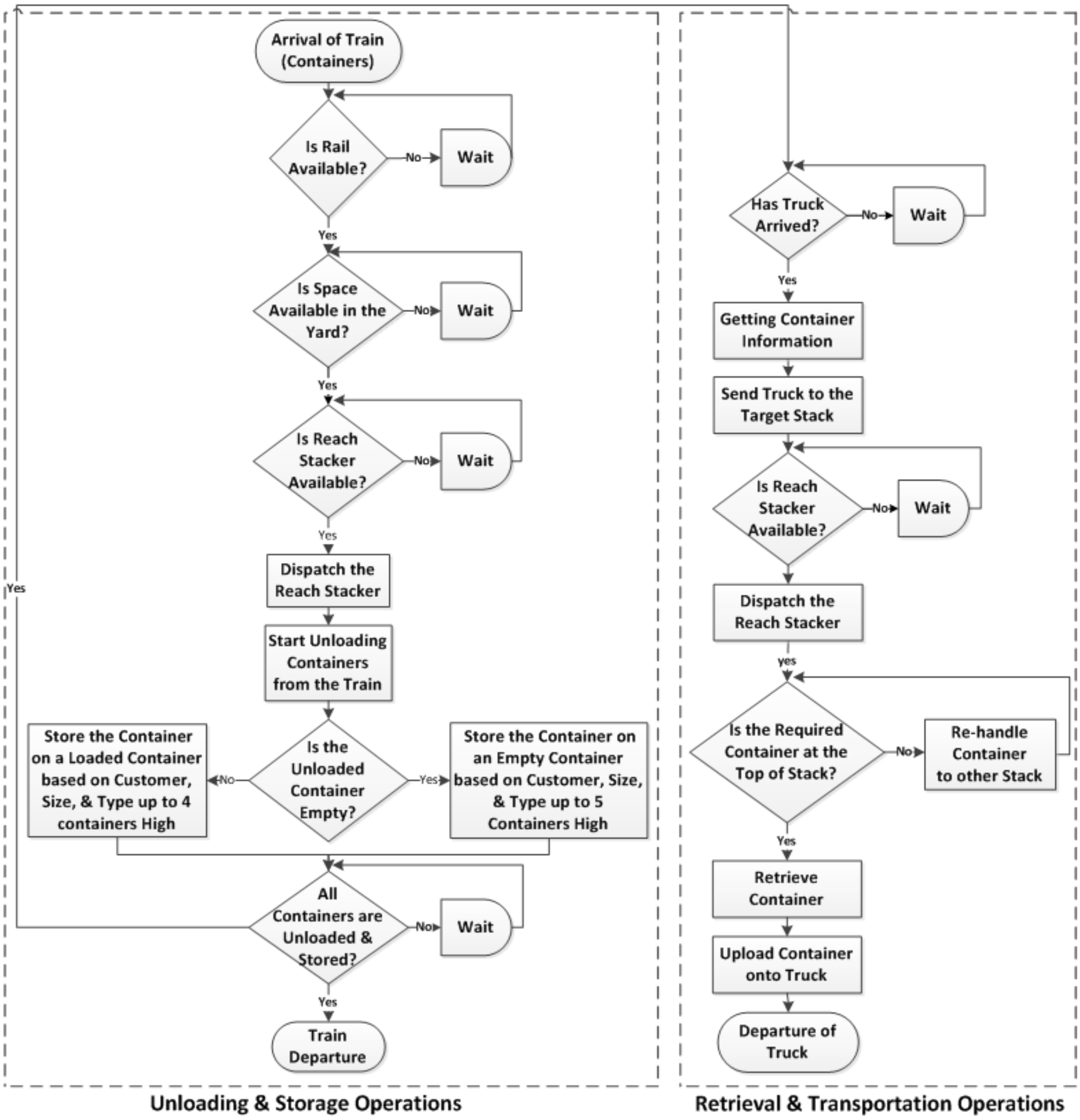


Figure 5.8: Container Management Operation Flowchart

This diagram starts with the container train arrival which waits until the rail area is available. When there is an available location for storage, the container waits until the reach stacker is available. When the reach stacker is available, the container is unloaded from the train and stored according to size and customer. Containers that belong to the same customer are stored in the same stack. Only containers that have the same size and type are stored on top of each other. The stack height is limited to four loaded containers and up to five empty containers.

After unloading and storing all containers in the yard, the train will depart the terminal. The next step will be the arrival of trucks for container departure (i.e. collection). When the truck arrives to collect a container, the required container information is handed to the control, then the truck will be sent to the target stack where the container is stored. The truck waits for the availability of the reach stacker and then retrieves the required container. The containers that are on top of the required one have to be re-handled to other stacks, and the required container is then retrieved and uploaded onto the truck to be transported to the customer.

5.5 Tilbury Railport Container Yard

A conceptual model is a technique that can be used to understand, simulate, and develop the processes of a system (Khatri et al. 2014). It is also used to illustrate the relationship between the processes that can be executed in the system (Khatri et al. 2014). See figure 5.9 for the Tilbury Railport container yard conceptual model.

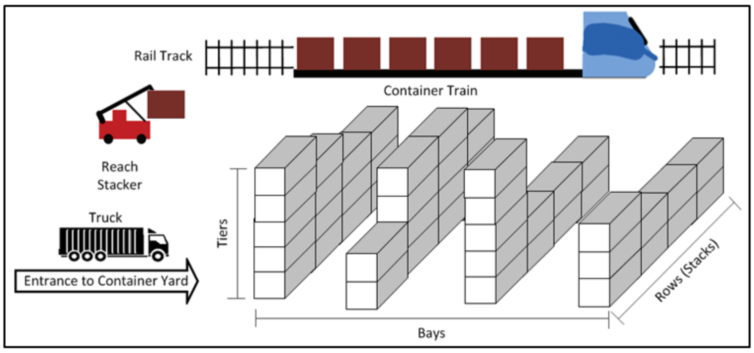


Figure 5.9: Conceptual Site Model Diagram of Tilbury Rail port Container Yard

This model illustrates the Tilbury Railport equipment as well the container yard, trucks flow and other fixed resources such as the reach stacker.

This terminal has an import container yard for stacking the import containers. The yard holds three sizes of containers which are 20ft, and 30ft, and 45ft. The capacity of this yard is 45 bays. Each bay consists of 2 to 5 rows (stacks). In these stacks, the stack height is limited to 4 loaded or 5 empty containers. To store or retrieve a container, it has its own location (i.e. stack) in the yard. This location is identified by the bay the container is stored in (i.e. bay number), the position the container is stacked in (i.e. row number), and the tier where the container is located (i.e. tier number).

There is a reach stacker which is used for both the unloading of containers from trains as well as the storage operation. The reach stacker is movable between the rail track side and the container yard side. There is a control function that provides information about the required container and sends the drivers (Trucks) to target stacks where containers are stacked for picking up. After the arrival at the terminal of a train loaded with different types and sizes of import containers, the reach stacker starts unloading import containers one at a time and moves them to the container yard side to be stored in a specific bay, stack and tier before they depart the yard on trucks. After unloading the import containers, the container train will leave the terminal, some of the import containers being loaded onto trucks directly without being stored in the yard.

When an empty truck arrives at the terminal, the truck driver provides the control function with the necessary information and is sent to the target stack where the container is stored. If the required container is on the top of the stack the container is retrieved by the reach stacker and loaded onto the truck. If the required container is not at the top of the stack, then the containers on top are moved (i.e. re-handled) to other stacks. After uploading the truck will leave the terminal with the uploaded container. If the required container is not stored in the yard, but is still on the train, then the reach stacker unloads the container from the train and uploads it onto the truck to be transported to the customer.

5.6 Fish Bone Delay Analysis of Tilbury Railport Operation

A fishbone delay diagram is a technique used to understand and evaluate the delays that can occur in a container terminal operation (Yousefi et al. 2012). In order to improve the terminal efficiency, these delays have to be reduced. When these delays are reduced, the terminal operation time will also be reduced and the customer service level increased. Figure 5.10 shows the fishbone analysis of the Tilbury rail port operations.

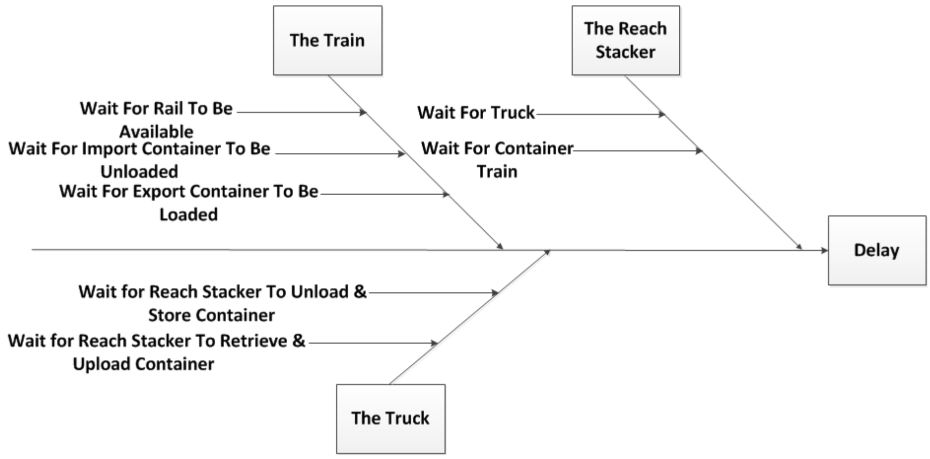


Figure 5.10: Fish Bone Delay Analysis of Tilbury Railport Operation

This diagram illustrates the source and cause of delays that can occur during operations in a container terminal.

1- The train might be delayed to the following reasons:

- Waiting for rail to be available.
- Waiting for the import containers to be unloaded.
- Waiting for the export containers to be loaded.

2- The reach stacker might be delayed due to:

- The uncertain arrival time of trucks for collection.
- The uncertain time of containers arrival/availability by trains.

3- The vehicles might be delayed due to the reasons below:

- Waiting for the reach stacker to unload and store container.
- Waiting for the reach stacker to retrieve and upload containers.

5.7 The Logical Containers Flow Diagram

This diagram shows the flow of containers within the terminal site, the operations (i.e. processes) carried out and the sequence in which they are performed. Figure 5.11 illustrates the logical flow of containers at Tilbury Railport.

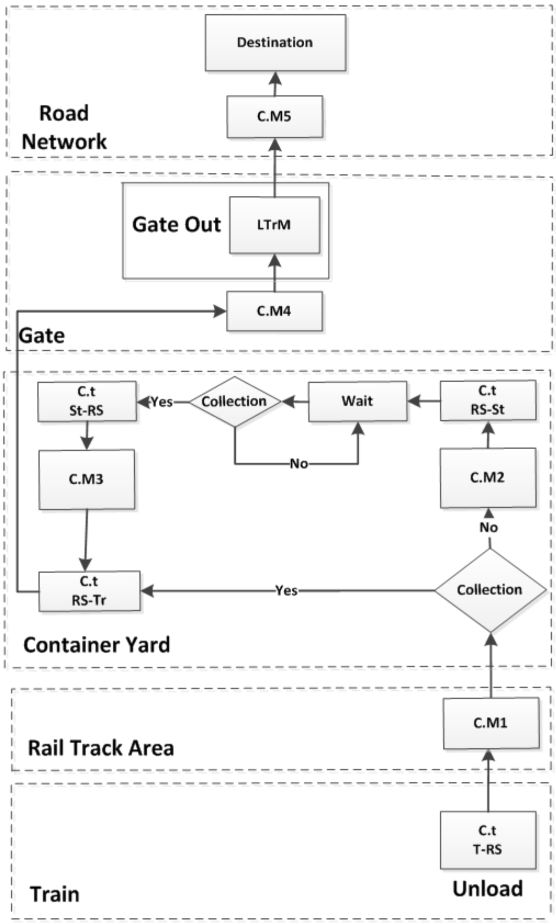


Figure 5.11: Logical Containers Flow Diagram

When a train arrives at the terminal, the import containers will be transferred by the reach stacker (C.t T-RS) to the yard area (C.M1). If the containers are ready for collection, they will be moved (C.M2), transferred to trucks (C.t RS-Tr), and then moved to the Gate Out (C.M4). After that, the loaded trucks will be moved to the road (LTrM). If the containers are not ready to be transported (i.e. to depart), then they will be moved to the container yard area (C.M2) and transferred by the reach stacker to the stacks (C.t RS-St) where they will wait until the trucks come to collect them. When the trucks come, the containers will be transferred from the stacks by the reach stacker (C.t St-RS, moved (C.M3) and transferred to trucks (C.t RS-Tr), when they will be moved through the Gate Out to the road and to their destination.

<u>Abbreviation</u>

- C.t T-RS (Container transfer from train to reach stacker).
- C.M (Container movement).
- C.t RS-Tr (Container transfer from reach stacker to trucks).

- St (stacks).
- C.T St-RS (Container transfer from stacks to reach stacker).
- LTrM (Loaded truck movement).
- C.t RS-St (Container transfer from reach stacker to stacks).
- Wait (Waiting time for container at yards).

5.8 'FKB_GA' Modelling Assumptions

A number of assumptions are made in order to simplify the modelling process for the container yard operations system. These assumptions are:

- The transportation time of a container between two neighbour bays is 1 minute.
- The re-handling time of a container between two neighbour rows is 1 minute.
- The re-handling time between two neighbour bays is 1 minute.
- The uploading time of a container from the top of stacks onto trucks is 1 minute.
- Any breakdown in the reach stacker is not considered in this model.
- In order to activate/deactivate the duration of stay factor in the system, the variation percentage of the duration of stay is set to 40%. When the difference in length of stay of the containers is 40% or above, then the duration of stay factor is activated (ON) otherwise it is deactivated (OFF).
- Trucks are considered to be available when containers need to depart.

5.9 Data Collection

The following tools and techniques were selected and used as the methodology for collecting data:

- **Site Visits**

For the collection of the process logic and operations time for the container stack allocation system, two visit days were needed to collect the required amount of information for this study. In these two visits, all the logical container yard operations including storage and retrieval of containers were identified and captured. Data from historical records that include storage and retrieval information in the yard has been collected in these site visits. See Appendix E for the sample of collected data.

- **Structured Interview**

The Structured interview was used to collect the required data, which included: the container yard size, the stay time of pre-existing containers, the number of trains that arrived each day, etc. A number of questions were carefully formulated in order to collect the required information (see Appendix F). The structured interview took taken place in two visits, the first on 09th October 2015, from 10:30 am until 01:00 pm, and the second on 5th July 2017 from 01:30 pm until 03:45 pm. These questions were carefully developed, before meeting the yard operators, to extract the storage and retrieval operation times for containers in the yard. The questions were developed and prepared to identify the problem and collect the relevant data in order to develop a new system for the management of the container yard operations. During the onsite visit, enough support from the yard operators was provided to identify the true logic for the storage and retrieval operations of containers in the yard and to collect the required amount of data. For a sample of the collected data, see the snapshot in Appendix E.

The interviews were conducted with the Business Development Manager and the Operations Manager at Maritime Transport Limited, both senior management staff at the company. The Business Development Manager and Operations Manager at Maritime Transport Limited, were selected as participants. These participants had all the data that was needed for this research and enough experience to respond to all the questions related to this study. During these interviews, all the logic and flow operations of the container yard were understood and captured. The required level of information and all logical relationships between the operations in the yard were also obtained from these participants.

Maritime is one of the UK's leading multimodal transport and container service specialists, combining road, rail & storage modes to become an integral element of the customer supply chain. The company also provides highly effective UK container transport and services.

This company operates a number of depots (i.e. terminals) for containers including Tilbury Railport and Birmingham Rail. Maritime has a distribution (i.e.

transportation) fleet that is spread across a number of depots including Felixstowe, Southampton, Liverpool, Manchester Bristol, Doncaster, Immingham, Leeds, London Gateway, Milton Keynes, Northampton, Peterborough, Reading, South Shields, and Thame sport.

- **Stop Watch**

A stop watch was used to identify all the container yard operation times including the storage and retrieval for containers as well as the transportation times of containers between bays and rows in the yard.

The methodology used to collect the required data for this research is summarised in table 5.1.

Table 5.1: The methodology used for data collection

Data Type	Tool/Technique	Purpose
Number of tiers	Structured Interview	To find out how many tiers were possible per stack.
Number of rows	Structured Interview	To identify the number of rows in the current yard.
Number of bays	Structured Interview	To find out how many storage bays were available in the yard.
Time spent by the pre-existing containers in the yard	Structured Interview	To estimate how long each pre-existing container could stay in the yard before departure.
Number of trains per day	Structured Interview	To understand how frequently trains arrived over a specific period of time.
Number of containers per train	Structured Interview	To identify the maximum number of containers that could be transported by each train.
The inter-arrival time of trains	Structured Interview	To estimate the duration between two consecutive train arrivals.
Types and sizes of containers	Structured Interview	To identify different types and sizes of containers.
The storage time per container in the first bay	Stopwatch	To estimate the time required to transport a container from the train to the first bay in the yard.
The storage time/container per extra bay	Stopwatch	To estimate the time required to transport a container between two neighbouring bays in the yard.
Uploading time onto truck	Stopwatch	To identify the time required to load a container onto a truck.

Table 5.1: The methodology used for data collection

Data Type	Tool/Technique	Purpose
Re-handling time of containers per row	Stopwatch	To measure the time required to transport a container between two neighbouring rows during the re-handling process.
Re-handling time of containers per bay	Stopwatch	To estimate the time required to transport a container between two neighbouring bays during the re-handling process.
Number of companies	Structured Interview	To identify how many 3PL companies were dealing with the yard operators.
Number of trucks per company	Structured Interview	To identify the number of trucks owned by each 3PL company.
Number of customers	Structured Interview	To identify the number of customers dealing with each 3PL company.
Travel time per truck	Structured Interview	To estimate the time taken for each truck to transport the containers to customers.
Number of containers for each customer per train	Structured Interview	To identify, for each customer, the number of containers in a train.
Departure sequence for containers	Structured Interview	To investigate the behaviour of containers departing from the yard.

In the table shown above, all the required data types, tools and techniques that were used to collect such data are presented to investigate the storage and retrieval operations for containers in the yard.

The tools and techniques that were used to collect data to develop the system for the container-yard operations management including the structured interview and stopwatch.

The data that was collected for the structured interview technique including:

the number of: tiers, number of rows, number of bays, number of days for train arrivals, number of trains per day, number of containers per train, the type and size for each container, number of companies, number of trucks per company, and the number of containers for each customer per train.

The inter-arrival time for the trains, the departure sequence of containers and the time spent by existing containers in the yard were also collected by the structured interview technique.

The stopwatch tool collected the time taken to: remove a container from the train and store it in the first bay in the yard, store a container between two neighbouring bays, upload a container onto a truck, re-handle a container between two neighbouring rows, and re-handle a container between two neighbouring bays.

5.9.1 The Container Yard Information

In this section, the inputs of the developed system are described as shown in table 5.2. The inputs described in this table are applied for all the three scenarios.

Table 5.2: The inputs of the system

Inputs	Values	Units
No. of tiers	5	Tiers
No. of rows	5	Rows
No. of bays	45	Bays
Time spent by the pre-existing containers in the yard	(2-4)	Days
The differences of duration of stay	40	%
No. of days for train arrivals	14	Days
No. of trains/day	(1-2)	Trains
No. of containers/train	(30-60)	Containers
The inter-arrival time of trains	12	Hours
No. of type of each container size	5 small, 5 Medium, of large	Types
The storage time/container in the first bay	2	Minute
The storage time/container per extra bay	1	Minute
Uploading time onto truck	1	Minute
Re-handling time of containers per row	1	Minute
Re-handling time of containers per bay	1	Minute
No. of companies	7	Companies
No. of Truck/company	(20-30)	Trucks
No. of customers	5	Customers
Travel time/truck	(60-200)	Minutes
No. of containers of each customer/train	(3-10)	Containers
Departure sequence of containers	minimum 10, maximum 30, most likely 15	Days

Some of the input values in the table above were normalised as follows:

- The actual value of the storage time of a container per extra bay was 0.78 minute. The normalised value of this storage time was 1 minute.
- The actual value of the uploading time of a container onto a truck was 0.70 minute. The normalised value of this uploading time was 1 minute.

- The actual value of the re-handling time of a container per row was 0.76 minute, while the normalised value of this re-handling time was 1 minute.
- The actual value of the re-handling time of a container per bay was 0.80 minute. The normalised value of this re-handling was 1 minute.

The fuzzy values were represented using fuzzy sets and membership functions as discussed in chapter 3, section 3.4.1.1.

The number of containers considered varies based on the occupancy of the yard being investigated that includes: 45 bays, 5 rows per bay, 225 stacks, and up to 5 containers (tiers) can be stored per stack. Three different volumes of containers are considered as follows:

1- 80%-90% of the total yard area is occupied by pre-existing containers (roughly 900 to 1012 containers) which can be considered a busy yard.
2- 50% of the yard area is occupied (around 562 containers).
3- An occupation level varies between 20%-30% of the total yard area (225 to 337 containers).

Through the analysis of the collected historical data of the departure time of containers using excel Microsoft, it was found that the sample data followed a triangular distribution with the following parameters $(a = 10, b = 15, c = 30)$, and the maximum value of probability distribution function was given by $2(c - a)$. The length of stay for containers in the yard was between 10 days to 30 days, and most of the containers departed on day 15. The probability density function for a triangular distribution was defined as follows:

$$f(x \backslash a, b, c) = \begin{cases} 0, & t < a, \\ \dfrac{2(t-a)}{(c-a)(b-a)}, & \mathrm{a} \le t \le b, \\ \dfrac{2(c-t)}{(c-a)(c-b)}, & b < x \le c, \\ 0, & c < \mathrm{t}, \end{cases}$$

The inputs given to the system was: minimum duration of stay of containers (i.e. 10 days, maximum duration of stay (i.e. 30 days) and most of the containers departed on day 15.

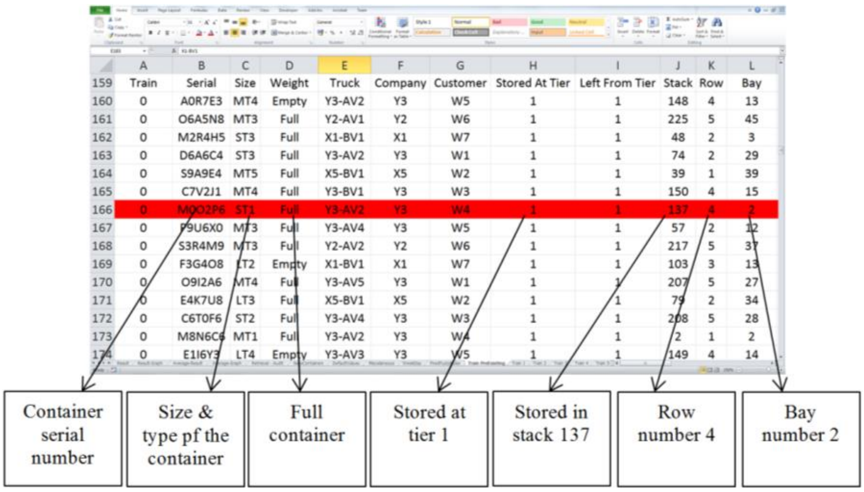
5.10 Verification and Validation of the Developed System

A number of techniques are used to verify and validate the developed system. In this study, simulation procedures were printed as described in the following sections.

5.10.1 Verification of the Developed System

Before constructing the system, all operations involving the overall logic of the container yard and other relevant flowcharts were reviewed and confirmed by the Business Development Manager and operations manager at Maritime Company.

For system verification, simulation procedures for the container storage and retrieval operations were printed in profiles including pre-existing container generation in the yard, storage of new incoming containers with the pre-existing ones, and calculation of acceptability level values for stacks for every new container. Figure 5.12 shows the generation of pre-existing containers in the yard.



	A	B	C	D	E	F	G	H	I	J	K	L
159	Train	Serial	Size	Weight	Truck	Company	Customer	Stored At Tier	Left From Tier	Stack	Row	Bay
160	0	A0R7E3	MT4	Empty	Y3-AV2	Y3	W5	1	1	148	4	13
161	0	O6A5N8	MT3	Full	Y2-AV1	Y2	W6	1	1	225	5	45
162	0	M2R4H5	ST3	Full	X1-BV1	X1	W7	1	1	48	2	3
163	0	D6A6C4	ST3	Full	Y3-AV2	Y3	W1	1	1	74	2	29
164	0	S9A9E4	MT5	Full	X5-BV1	X5	W2	1	1	39	1	39
165	0	C7V2J1	MT4	Full	Y3-BV1	Y3	W3	1	1	150	4	15
166	0	M0O2P6	ST1	Full	Y3-AV2	Y3	W4	1	1	137	4	2
167	0	P9U6X0	MT3	Full	Y3-AV4	Y3	W5	1	1	57	2	12
168	0	S3R4M9	MT3	Full	Y2-AV2	Y2	W6	1	1	217	5	37
169	0	F3G4O8	LT2	Empty	X1-BV1	X1	W7	1	1	103	3	13
170	0	O9I2A6	MT4	Full	Y3-AV5	Y3	W1	1	1	207	5	27
171	0	E4K7U8	LT3	Full	X5-BV1	X5	W2	1	1	79	2	34
172	0	C6T0F6	ST2	Full	Y3-AV4	Y3	W3	1	1	208	5	28
173	0	M8N6C6	MT1	Full	Y3-AV2	Y3	W4	1	1	2	1	2
174	0	E1I6Y3	LT4	Empty	Y3-AV3	Y3	W5	1	1	149	4	14

Figure 5.12: Generation of Pre-existing Containers in the Yard

The figure above illustrates how the pre-existing containers are stored in each stack in the yard. The example highlighted shows one of the pre-existing containers with serial number (M0O2P6), size and type (ST1) and status (full). The container was stored in tier 1 at stack 137 in row 4 and bay 2. The new containers are stored with the pre-existing ones. Figure 5.13 following illustrates the storage of new containers in the yard.

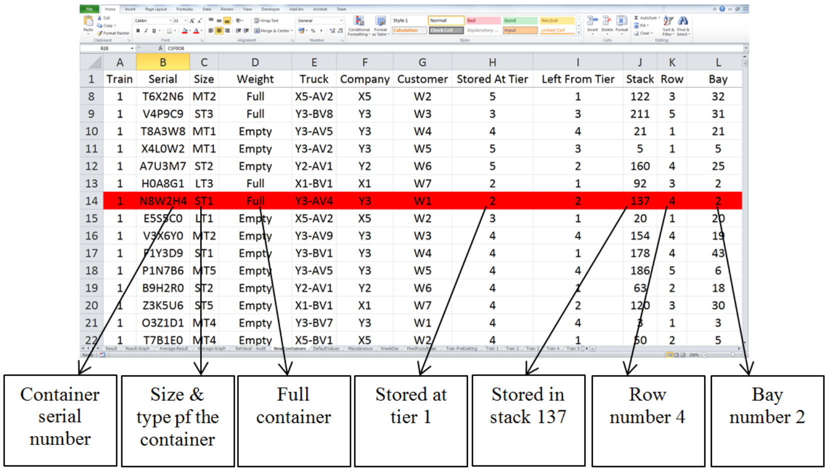


Figure 5.13: Storage of New Containers in the Yard

In the figure above, the container serial number (N8W2H4), size and type (ST1) and status (full) are shown. The container was stored at tier 2 in stack 137 in row 4 and bay 2. This means this container was stored in tier 2, in the same stack, row and bay as the pre-existing container with serial number (M0O2P6) with both having the same type and size (ST1) and status (full).

The containers are stored based on the acceptability level value for each stack. This acceptability level is calculated depending on certain input factors and constraints. the stack with the highest acceptability value being allocated for container storage. Figure 5.14 shows the acceptability level value calculation.

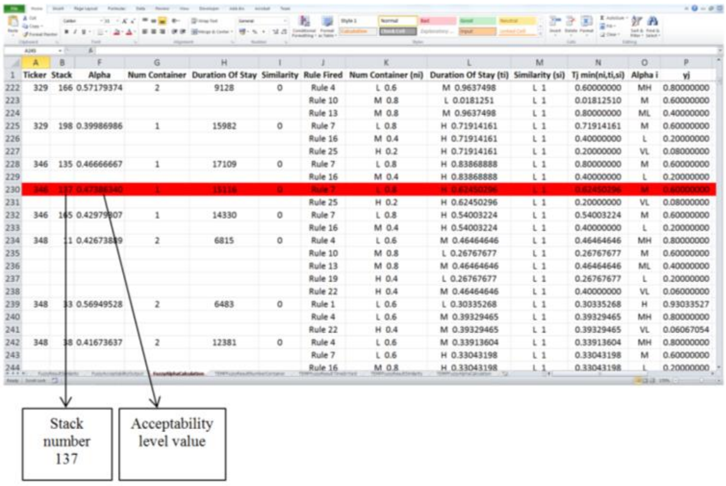


Figure 5.14: Calculation of Acceptability Level Values

In this figure, the acceptability level value (α_i) was calculated for the same stack (137) as the container with serial number (N8W2H4). This means that the system was storing the containers in the right location (i.e. stack). The system stores the containers taking into consideration the size and type. In a particular stack, the status (i.e. size and type) has to be the same. As shown in the figure, a number of fuzzy rules were fired to calculate the acceptability level values. To calculate the acceptability level value for stack 137, fuzzy rules 7 and 25 were fired.

5.10.2 Validation of the Developed System

After running the 'current' approach for the system, it was important to determine if the simulation outputs were close to reality. The validation procedure used checked the storage time for a specific container to determine the convergence of results with the 'current' outputs.

Before running the system, a sample of collected data types and their time recorded using tool/technique are summarized in table 5.3.

Table 5.3: a sample of data types and time recorded for the data types

Data Type	Time Recorded Using each Tool/Technique
The storage of a container in the first bay	2 Minutes
The storage of a container in a neighbouring bay	0.78 Minute
The uploading of a container onto a truck	0.70 Minute
The re-handling of containers to a neighbouring row	0.76 Minute
The re-handling of a container to a neighbouring bay	0.80 Minute

This table presents a sample of data types and time recorded for the data types by the stop watch tool. The time recorded for the storage of the container (ACLU4738932) in the first bay was 2 minutes, while the time recorded the storage of a container between two neighbouring bays was 0.78 minutes. 0.70 minutes was recorded for uploading a container onto a truck. Finally, the time recorded for re-handling a container (GCNU6954487) between two neighbour rows and bays was 0.76 minutes and 0.80 minutes respectively.

After running the system, the results were as follows:

The system was presented to Maritime Company operators. The real-world scenario was simulated by the developed system using the ‘Current’ approach. The company stored containers based on the similarity of containers in the stack. The system was validated as follows:

A container (GCNU1243219) was stored in row 1, stack 2 at bay 5, and the simulated time to store the container was 6 minutes. The actual storage time of the container was 5.93 minutes. The same stored container was retrieved, but before retrieving this container, two containers (GCNU2006572 & GCNU1229118) were stored on it. These containers were re-handled to other stacks, stack 3 at bay 4 and stack 5 at bay 6. The simulated time to retrieve the container was 3 minutes. The actual retrieval time of the container was 2.9 minutes. The difference in the simulation times for both storage and retrieval operations of the container were due to the approximation of the transportation time of the container by the reach stacker. After showing them the simulated and actual times for container storage and retrieval, Maritime accepted the difference, thus, the system was validated.

5.11 Human factors in FKB_GA System

Human factors are the primary cause of error-induced accidents in container yard operations (Fancello, Errico, and Fadda 2008).

The knowledge of human factors that was built in the developed model as follows:

1. Complying with the regulations of the yard operations (e.g. speed). This factor was considered in the model to avoid errors or accidents that can be caused by yard operators (e.g. reach stacker operator).
2. Obtaining container train position. This factor was taken into consideration by the model to prevent any errors that were caused by workers at the train tracks.
3. Proper watch keeping in the yard. This factor was built in the model to monitor resources and container locations to avoid errors and accidents in yards.

4. Negligence was considered in the model during the storage and retrieval operations in the yard, the arrival, departure, storage and retrieval of containers being monitored by yard operators.

5. The Size, type and weight constraints of containers was built in the model to prevent containers getting damaged in the yard. Only containers of the same size and type were stacked on top of each other. Lighter containers were stacked on heavier ones in the yard.

5.12 Chapter Summary

In this chapter, the container yard operations of the company being investigated were described and explained in details. The Maritime company profile and the key services of this company were demonstrated. Data collection methodology was discussed in order to collect the required information about the container yard operations including storage and retrieval operations. These operations were described through flowcharts and the conceptual model diagram of the company yard being investigated was explained. The inputs of the developed system were discussed in this chapter. The assumption modelling of the FKB_GA was presented. A fishbone analysis of the Tilbury rail port operations has been explained. The logical flow diagram of containers in the yard was discussed. The verification and validation of the developed system was discussed in details as well. This chapter was used in order to provide the required information about the container yard operations for further modelling. Human factors that can be considered in container yards' management were built in the developed system and presented in this chapter.

CHAPTER 6

EXPERIMENTATION, COMPARISION, RESULTS ANALYSIS AND DISCUSSION

6.1 Introduction

In order to test the behaviour of the ‘FKB_GA’ system, three real life scenarios were designed including a ‘busy’, ‘moderately busy’ and ‘quiet’ yard, with a significant, moderate and small number of pre-existing containers respectively. For all these three scenarios, the proposed Fuzzy Knowledge Based Genetic Algorithm (GA) system was adopted for the modelling and then optimising of the storage and retrieval operations. The effect of the ‘ON/OFF’ strategy on the storage and retrieval operations was also tested by the FKB model.

In addition, comparison between the ‘Current’, ‘FKB’, and ‘FKB_GA’ storage and retrieval approaches results was conducted. For each stack allocation approach, the container storage operation results in a number of re-handlings after the retrieval operation has been completed. The ‘Current’ approach applied by the companies were summarised as follows: the yard operators allocated stacks for container storage and retrieval based on the similarity of the containers in the stacks (e.g. containers belonging to the same customers). This approach also took into consideration the storage constraints (e.g. the weight, size and type of containers) during the stack allocation process in the yard. In addition to the above approaches, the most common stack allocation approaches for container storage and retrieval together with various numbers of re–handlings are compared with the system for further analysis. These include ‘Constrained-Probabilistic’ Stack Allocation and ‘Constrained-Neighbourhood’ Stack Allocation approaches. The ‘Constrained-Probabilistic’ Stack Allocation approach allocates stacks for container storage and retrieval taking into consideration the constraints mentioned above. While ‘Constrained- Neighbourhood’ Stack Allocation approach also considers the above constraints but allocates the nearest stacks to the target stack for container retrieval. See Appendices C and D for more detail on these approaches. The results of testing the system for container yard operations in conditions of slow and rush container trains are also presented.

6.2 The Experimental Part

In this section, the experiments of the FKB model with the GA model and the 'ON/OFF' strategy for storage and retrieval operations of containers are presented in detail. The model is implemented on a computer with i5 CPU 3.10 GH, 8 GB RAM and a Windows 7 operating system.

6.2.1 Experimentations of the 'FKB_GA' System- Effect of the GA Model

To test the performance of the system, inputs of the system are described in chapter 5, section 5.9.1 in table 5.1, and three real life scenarios of busy, moderately busy and quiet yards were identified as discussed in chapter 5, section 5.9.1.

The results of these scenarios were then selected to judge the behaviour of the system and to check how efficiently the system allocates stacks for the newly arrived containers and other retrieval operations.

In order to optimise the fuzzy knowledge based model, a set of GA parameters were used as shown in Table 6.1.

Table 6.1: The GA parameters

No. of chromosomes (population size)	GA Operators	
	Crossover Probability (P_c)	Mutation Probability (P_m)
5	0.45, 0.75, 0.90	0.05, 0.10, 0.20
10	0.45, 0.75, 0.90	0.05, 0.10, 0.20
15	0.45, 0.75, 0.90	0.05, 0.10, 0.20

Each set of parameters were then used in each of the scenarios above to tune/calibrate the GA parameters. The best set of parameters including population size, probability of crossover and probability of mutation was selected based on the minimum achieved total number of re-handlings of containers.

As can be seen in the table, three population sizes were tested against different crossover and mutation probabilities. Each population size consisted of a predefined number of chromosomes and each chromosome covers the fuzzy rules for all the yard stacks. Because a large yard size of 225 stacks was selected for this case study, a maximum of 15 chromosomes was decided as the population and 50 generations were run to explore more promising solutions under these restricted population sizes. As each run with this large yard size has taken long computational time which led to select these population sizes.

The computational time depends on the number of arriving containers along with yard size (number of stacks) taking into consideration the storage constraints. Each arriving container has a possible of 225 fuzzy rule sets allocated across all stacks by taking into account the storage constrains. Hence, the computational time increases based on the number of arriving containers and 225 or less fuzzy rule sets allocated per arriving container across all stacks.

The system was executed with each of these sets of inputs in order to obtain the results required for testing. In order to advance the searching process for optimised stack allocation approaches for container storage, the GA parameters were tuned after a number of experiments under 3 different sets of GA parameter settings for the three scenarios (see Appendixes G, H, and I for the tuning diagrams of these parameters). These parameters included population size and probabilities of crossover and mutation. The results of each scenario are explained in the next sections in details.

6.2.1.1 Busy Yard Scenario

In order to test the response of the container yard management system (Fuzzy Knowledge Based GA system) with a large number of pre-existing containers, the required inputs are presented in chapter 5, table 5.2 and the results are analysed and discussed in this section. This scenario assumes that there are pre-existing containers in the yard, based on an 80%-90% of occupation (roughly 900 to 1012 containers) which was provided by the Maritime yard operator, and can be considered a busy yard. There should be differences in the durations of stay for the topmost containers over time in order to activate the duration of stay factor. In order to activate/deactivate this factor, the difference between the durations of stay is assumed to be 40% which is applied for all scenarios (note: the optimisation of this parameter is beyond the purpose of this study). When the difference in the durations of stay is 40% or above, the factor is activated (ON) otherwise it is deactivated (OFF). The results of the busy yard scenario are presented and analysed in the next section.

6.2.1.1.1 Results Analysis of the Busy Yard Scenario

Three stack allocation approaches for container storage and retrieval operations were selected for analysis including the 'Current', 'FKB', and 'FKB_GA'.

As mentioned earlier, the GA parameters were calibrated after some of the experiments, several sets of population sizes together with different crossover and mutation probabilities were attempted without any significant effect. The parameters tuned by the GA along with their optimal/ near optimal total number of re-handlings are presented in Appendix J.

A population size of 15 chromosomes, a probability of crossover of 0.90 and a probability of mutation of 0.10 which were selected as the most appropriate tuned parameters for the GA. The stopping condition was 50 generations and 750 solutions.

The 'Current' approach resulted in the maximum number of re-handlings and was adopted by the company. The 'FKB' provided an improved number of re-handlings when compared with the 'Current' storage approach. The 'FKB_GA' approach, however achieved a salient reduction in the number of re-handlings when compared with the 'Current', and 'FKB' approaches. Figure 6.1 displays the busy yard scenario results for all the approaches.

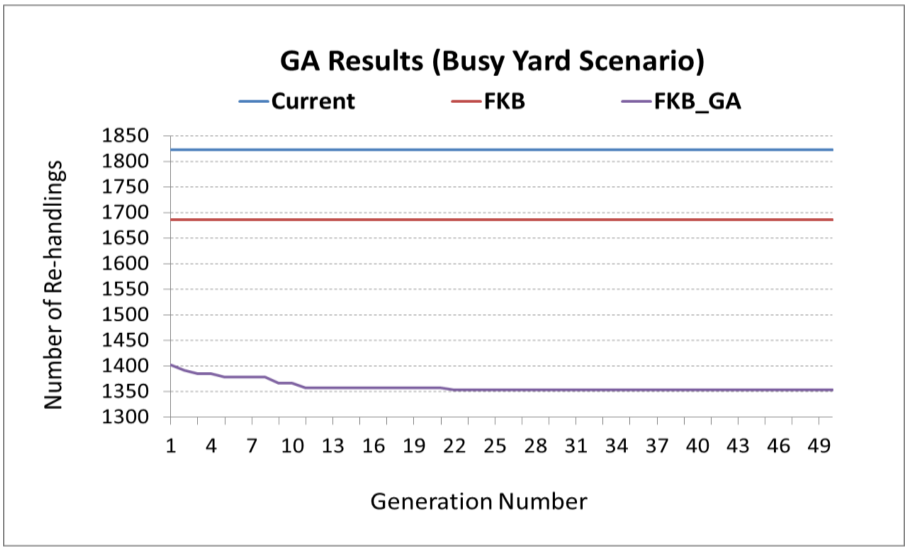


Figure 6.1: The total number of re-handlings (Busy yard scenario)

In figure 6.1, the ‘Current’ approach results in 1822 re-handlings to store then deliver all containers, which was the highest of the three approaches. The ‘Current’ approach grouped the containers using customer as a storage factor. The ‘FKB’ approach, obtained 1686 re-handlings which utilised all the fuzzy rules from the rule base for containers rather than selecting just the active rules. In the ‘FKB_GA’ approach, an early reduction of the number of re-handlings was obtained from the initial population of 1402. This was because in this run, the initial population randomly selected more promising rules from the fired fuzzy rules. Noticeable reductions in the number of re-handlings were obtained at the 2nd and 11th generations numbers 2 and 11th due to the fact that the best set of GA parameters led to the selection of a small number of effective rules from the previous rules obtained. This led to investigating more promising solutions in order to achieve the required randomness in the search process.

Slight reductions in the number of re-handlings were obtained at the 3rd and 5th generations. It can be seen that after the 21st generation, the minimum number of 1353 re-handlings was obtained. Although repeated chromosomes were not allowed as discussed in chapter 3, section 3.4.2.3, the total number of re-handlings did not improve further after generation 22. This was due to the fact that a binary coding mechanism of genes was applied where each gene represented a rule and hence, the selection of good genes might be affected by other activated weak ones, and hence, the resultant outcome of the number of re-handlings of all containers could be similar. The allocated stacks were the best stacks for the container storage operation which yielded the minimum number of re-handlings for containers after the retrieval operation was complete.

In order to store containers in the yard, a specific stack (location) needed to be allocated for each container, see figure 6.2 for the average stack utilisation. Figure 6.3 shows the number of re-handlings per stack for all storage approaches.

The highest utilised stack at 64.85% was 140 which was obtained by running the ‘Current’ approach since the number of containers stored in that stack was high as shown in figure 6.2.

However, the same average stack utilisation was obtained in most of the stacks by applying the ’FKB_GA’ approach, which achieved the best distribution of containers to stacks in order to reduce the number of containers per stack and subsequently easier retrievals with less re-handlings. The FKB_GA approach also achieved the highest stack utilisation of 60.34% which was obtained at stack 215. By applying this approach, the stored containers in stack number 216 was also high. This was because the number of stored containers over time was high. The lowest stack utilisation of 20.88% was obtained at stack number 219 using the ‘FKB_GA’ approach, as the number of containers stored in stack 216 was low. The lowest stack utilisation of 21.72% was obtained at stack number 157 by running the ‘FKB’ approach. The number of stored containers in stack number 157 was low which justified the low stack utilisation. However, the highest stack utilisation of 57.97 was achieved at stack number 41 by applying the ‘FKB’ approach due to the high number of containers stored at that stack.

The average stack utilisation is the lowest by applying the FKB_GA as this system has succeeded in spreading the containers equally across the yard for easier and faster retrieval. This is because this system has taken the number of containers per stack, duration of stay of the top most containers per stack into consideration while selecting a stack for container storage and retrieval.

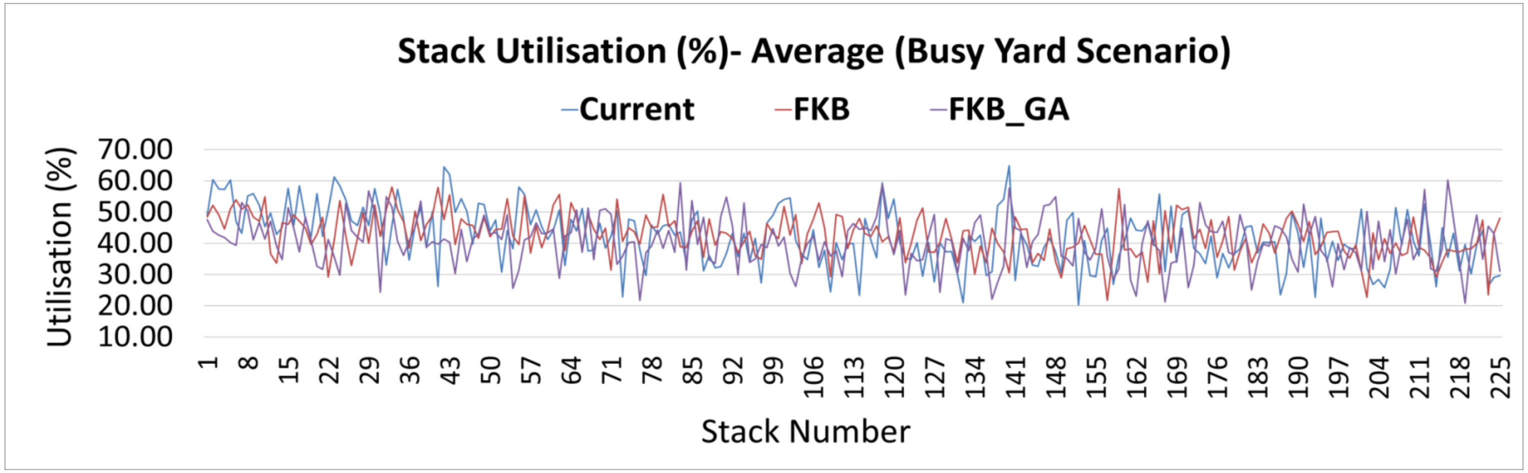


Figure 6.2: The average stack utilisation (Busy yard scenario)

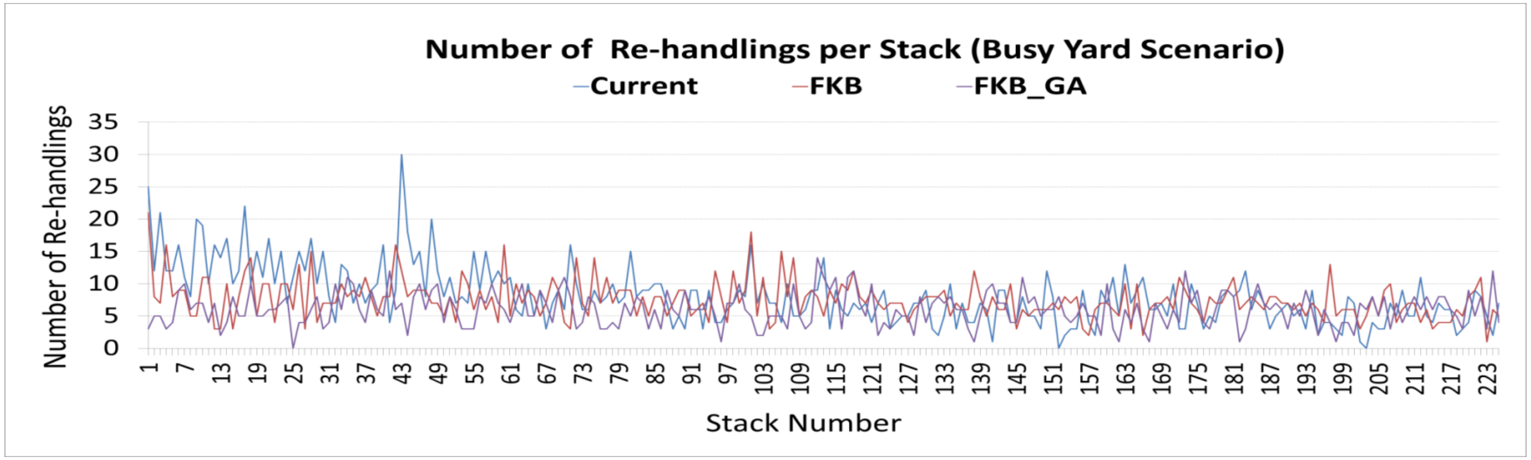


Figure 6.3: The number of re-handlings per stack (Busy yard scenario)

In figure 6.3, when comparing the number of re-handlings obtained in stacks for the 'Current', and 'FKB' and 'FKB_GA' approaches, it can be concluded that the highest number of re-handlings was achieved in most of stacks by the 'Current', and 'FKB' approaches, while the lowest number of re-handlings was achieved in most of the stacks by the 'FKB_GA' storage approach. As mentioned in Figure 6.1, the number of re-handlings in most bays achieved by the 'FKB_GA' storage approach was the lowest of all the approaches, therefore this approach also achieved the lowest number of re-handlings in most of stacks. For example, the highest number of 30 re-handlings was achieved at stack number 43 by applying the 'Current' approach. In stack 43, the average stack utilisation was high which resulted in a high number of re-handlings (discussed in figure 6.2). The highest number of 21 re-handlings can be seen in stack number 1 when applying the 'FKB' approach. No re-handlings were necessary in stack number 25 after applying the 'FKB_GA' approach. However, the average number of stored containers in stacks 116 and 197 was low which resulted in less re-handling as discussed in figure 6.2. However, the lowest number of re-handlings was obtained in most of the stacks by the 'FKB_GA' approach.

The number of re-handlings per stack is the lowest by applying the FKB_GA is the lowest because the containers have been spread equally across the yard for easier and faster retrieval. This is because this system has taken the number of containers per stack, duration of stay of the top most containers per stack factors into consideration while selecting a stack for container storage and retrieval.

With the 'FKB_GA' approach, the total retrieval time for all containers was also reduced as shown by figure 6.4.

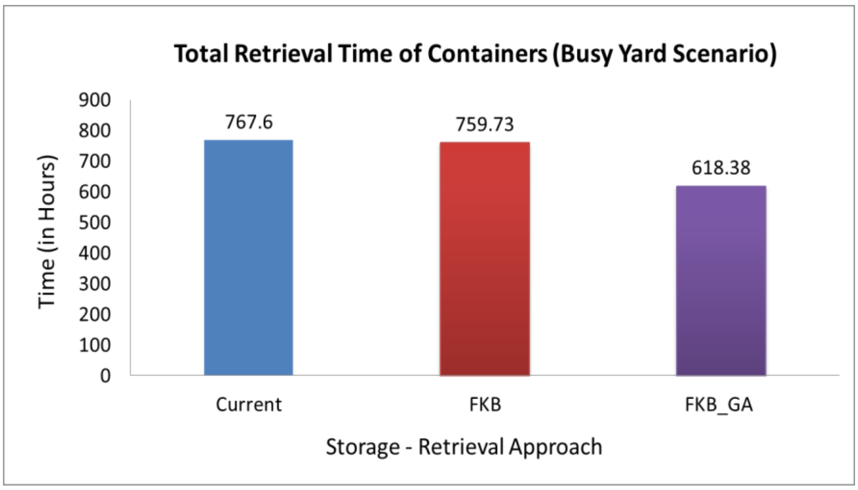


Figure 6.4: The total retrieval time of containers (Busy yard scenario)

In figure 6.4, the total retrieval time achieved by the ‘FKB_GA’ approach is 18.6% lower than the ‘FKB approach. As mentioned in figure 6.1, the number of re-handlings achieved by the ‘FKB_GA’ storage approach was the lowest of all the approaches which was why there was a corresponding reduction in the total retrieval time here. The total number of re-handlings achieved by the ‘Current’ approach was also high in figure 6.1, which is why the total retrieval time was high here.

6.2.1.2 Moderately Busy Yard Scenario

To test the system under different scenarios, the moderately busy yard scenario was investigated. The inputs of the moderately busy yard scenario described in Table 6.1 were provided to the ‘FKB_GA’ system. This scenario assumes that there existed an initial number of 562 containers which amounted to 50% occupation of the yard, this information was provided by the Maritime yard operator.

The results of the moderately busy yard scenario are introduced and analysed in the next section.

6.2.1.2.1 Results Analysis of the Moderately Busy Yard Scenario

In this section, the moderately busy yard scenario results analysis is presented for three storage and retrieval approaches including ‘Current’, ‘FKB’, and ‘FKB_GA’. The parameters tuned by the GA along with their optimal/ near optimal total number of re-handlings are presented in Appendix K.

For the ‘FKB_GA’ storage approach, the optimal settings were found to be a population size of 5, a probability of crossover genes of 0.45, and a mutation rate of genes of 0.10. The stopping condition for the FKB_GA approach was satisfied when the number of generations was 50, 250 solutions.

The ‘Current’ approach showed the maximum number of container re-handlings. A reduced number of re-handlings was achieved by the ‘FKB’ approach. However, the ‘FKB_GA’ obtained the minimum number of re-handlings of containers during the retrieval operation. See figure 6.5 for the ‘FKB_GA, ‘FKB’ and ‘Current’ results in the moderately busy yard scenario.

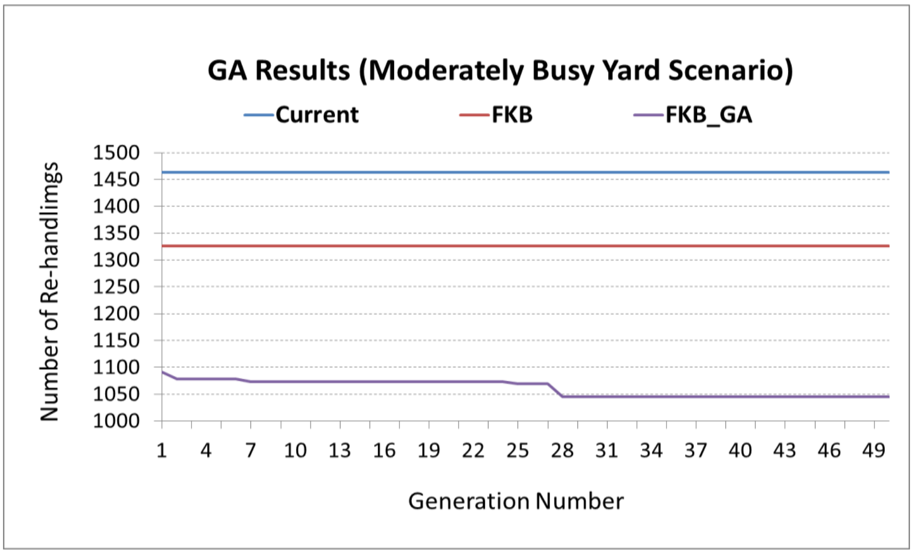


Figure 6.5: The total number of re-handlings (moderately busy scenario)

In the comparison with the other approaches the 'FKB_GA' approach obtained the minimum number of 1045 re-handlings, while, the 'Current' approach obtained the maximum number of 1464 re-handlings. This approach utilised an attribute to group containers in stacks by customer and achieved 1326 re-handlings. The 'FKB' selected the redundant fuzzy rules from the rule base which resulted in a high number of container re-handlings.

Compared to the other approaches it can be seen that the 'FKB_GA' starts from 1091 re-handlings. The reason for this is that an influential set of fuzzy rules were randomly selected as an initial population which led to an appropriate allocation of stacks to containers. Further slight improvements were made on the total number of re-handlings at generation numbers 2, 7, and 25 respectively due to the diversity of the exploration of the solution space that was achieved by the GA operators. A significant reduction was achieved at the 28th generation which was the minimum number of 1045 re-handlings. This reduction was because the GA operators had succeeded in bouncing the current search away thus exploring a more promising part of the solution space.

The average stack utilisation was also obtained for all the approaches as shown in figure 6.6. Figure 6.7 shows the number of re-handlings per stack for all storage approaches.

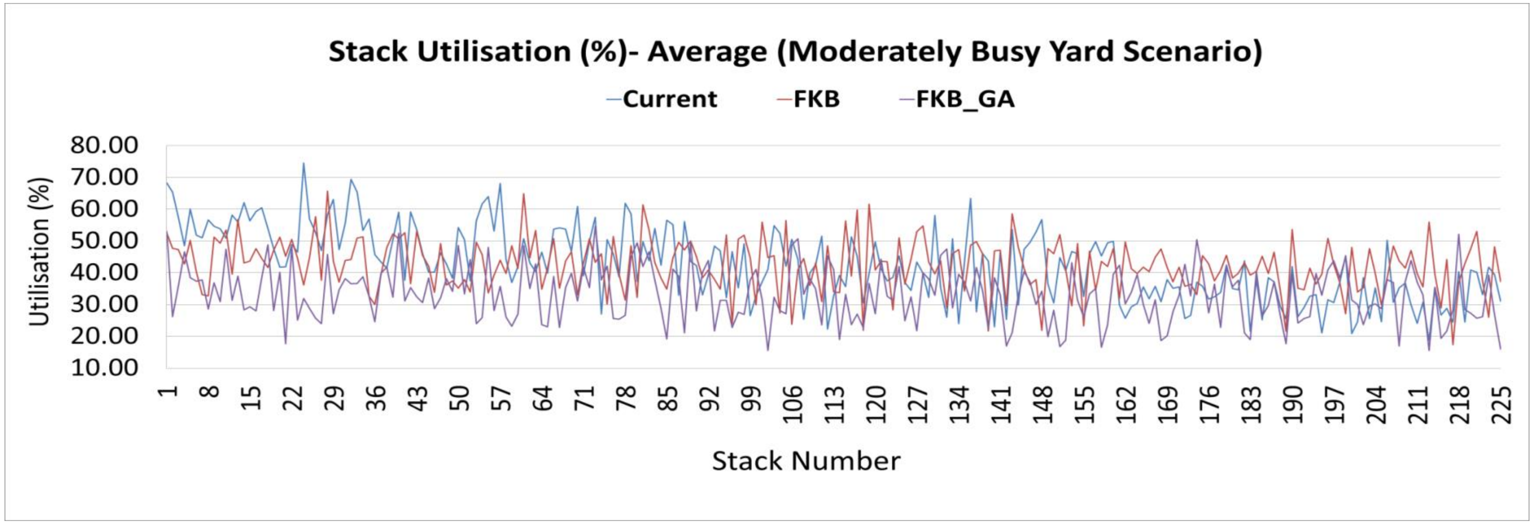


Figure 6.6: The average stack utilisation (Moderately busy yard scenario)

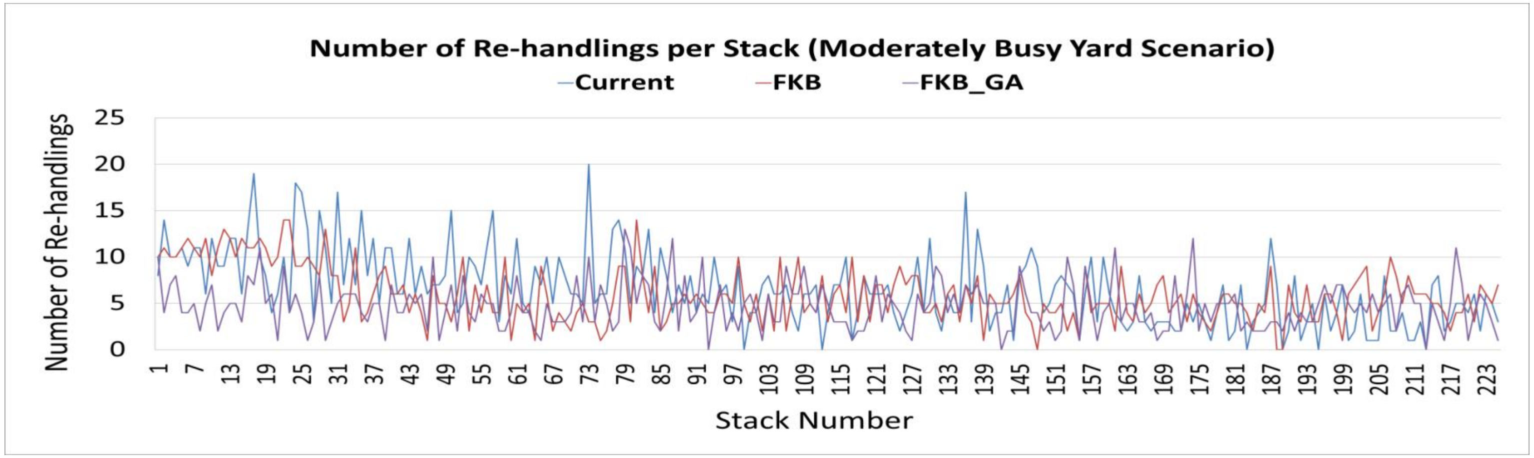


Figure 6.7: The number of re-handlings per stack (Moderately busy yard scenario)

As shown in figure 6.6, the highest utilisation at 74.49% was achieved at stack 24 by using the 'Current' approach, while 65.68% was achieved at stack 28 by running the 'FKB' approach. The number of stored containers in stack 24 for the 'Current' approach and 28 for the 'FKB' is high. The lowest utilisation of 17.36% was achieved at stack 217 by the 'FKB' storage approach, the number of stored containers being low leading to the low stack utilisation. However, the average utilisation by the 'FKB_GA' storage approach in most stacks is almost the same and the lowest of all the approaches. The lowest stack utilisation achieved by the 'FKB_GA' approach was in stack 215 at 19.31% and the highest was at stack 73 at.54.58%. In general, the reason for the high stack utilisation was that the average number of stored containers was high, while the low stack utilisation occurred because the average number of stored containers in the stack over time was low. However, the number of containers stored in stacks was similar when applying both the 'Current' and 'FKB' approaches which resulted in a similar average stack utilisation.

In Figure 6.7, the highest number of re-handlings in most of the stacks was achieved by the 'Current' approach, while the lowest number was achieved by the 'FKB_GA' storage approach. This compared with the lowest number of re-handlings in most of the bays also being achieved by the 'FKB_GA' approach as shown in Figure 6.5.
The 'Current' approach achieved the highest number of 20 re-handlings as seen at stack 73. The number of stored containers in stack 73 during the storage operation was high, which led to a high number of re-handlings during the retrieval operation. The highest number of 14 re-handlings achieved by the 'FKB' storage approach can be seen in stacks 22, 23, and 81. After applying the 'FKB_GA' storage approach, no re-handlings were required in stacks 93, 142, and 213. However, the 'FKB' approach achieved the lowest number of re-handlings in stacks 148, 188 and 189, as in these stacks the number of stored containers was low (explained in figure 6.6) which led to a lower number of re-handlings.

The number of re-handlings per stack is the lowest by applying the FKB_GA, the reason is discussed in section 6.2.1.1.1, figure 6.3.
All the stored containers during the storage operation are retrieved when they are due to depart. Figure 6.8 shows the total retrieval time of containers for all storage

approaches. The ‘FKB_GA’ approach reduced the total retrieval time for all containers which can be observed in figure 6.8.

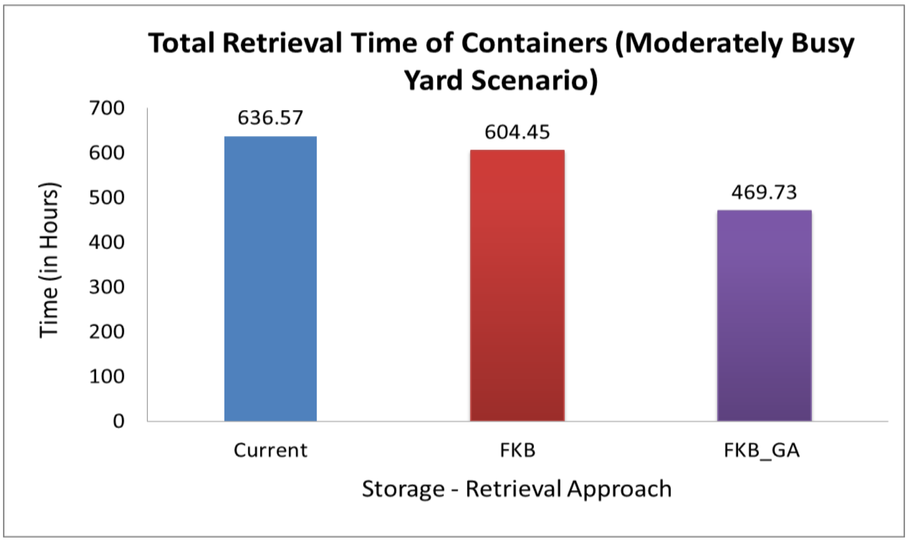


Figure 6.8: The total retrieval time of containers (Moderately busy yard scenario)

In figure 6.8, when comparing to the ‘Current’ and ‘FKB’ approaches, the ‘FKB_GA’ approach reduced the total retrieval time for containers by 25.2%, and 22.2% respectively. This compares with figure 6.5 which shows the approach also achieved the lowest total number of re-handlings for containers leading to a reduced total retrieval time.

6.2.1.3 Quiet Yard Scenario

In this section, the Fuzzy Knowledge Based GA (‘FKB_GA’) system is tested under the quiet yard scenario. The inputs of the system in this scenario are the same inputs as were provided in both the busy yard and moderately busy yard scenarios as discussed in Chapter 5. With this scenario the number of pre-existing containers is based on 20% to 30% of yard capacity (225 to 337 containers) which was provided by the Maritime yard operator. The results of the quiet yard scenario are introduced and analysed in the next section.

6.2.1.3.1 Results Analysis of the Quiet Yard Scenario

The quiet yard scenario results analysis is presented in this section for different storage and retrieval approaches, these include: ‘Current’, ‘FKB’, and ‘FKB_GA’ approaches.

After a number of experiments had been made, the optimal GA parameters were obtained in terms of population size, crossover and mutation probabilities. The parameters tuned by the GA along with their optimal/ near optimal total number of re-handlings are presented in Appendix L.

For the ‘FKB_GA’ storage approach, the optimal parameters were a population size of 15 chromosomes, crossover and mutation probabilities for genes of 0.75 and 0.20 respectively. This is due to the consistency of the behaviour of the reduction curve over the generations. The stopping criterion for the approach was 50 generations, and 750 solutions.

The maximum number of re-handlings was obtained by applying the ‘Current’ storage approach which was approach currently used by the company. The ‘FKB’ storage approach minimised the number of re-handlings of containers when compared with the ‘Current’ approach. A dramatic reduction in the number of re-handlings was achieved by the ‘FKB_GA’ storage approach. Figure 6.9 shows the results for the ‘FKB_GA’, ‘Current’ and ‘FKB’ approaches in the quiet yard scenario.

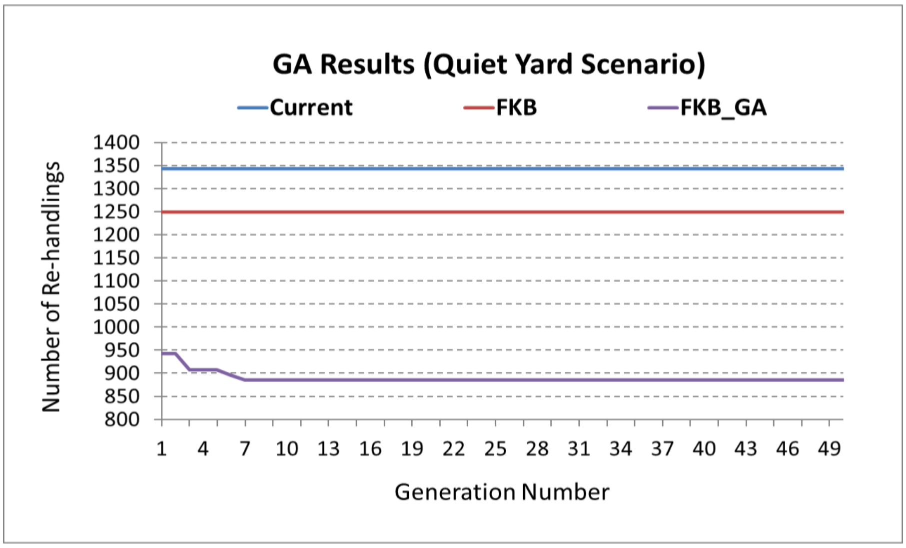


Figure 6.9: The quiet yard scenario experiment results

In figure 6.9, it has been noticed that the ‘Current’ storage approach obtained 1343 re-handlings for containers which was the highest number of re-handlings when compared with the other storage approaches. The ‘Current’ approach allocated stacks for containers based on a specific attribute which ensured that containers for a particular customer were stored in the same stack. The ‘FKB’ storage approach achieved 1249 re-handlings which was less than the number of re-handlings obtained by the ‘Current’ approach, but was still high when compared with the ‘FKB_GA’ approach.

By applying the ‘FKB’ storage approach, a number of weak/inappropriate rules were selected from the fuzzy rule base which led to the allocation of inappropriate stacks for container storage resulting in a high number of re-handling of containers during the retrieval operation. The figure shows that a significant reduction was obtained in the first generation. The number of re-handlings in the ‘FKB_GA’ approach started from 943, this meant that in the initial population, a set of strong rules were selected from the fired fuzzy rules that led to the generation of more robust solutions. A remarkable reduction in the number of re-handlings was obtained at generation 3, where the number of re-handlings was reduced by 36. This meant that the GA operators had activated more robust fuzzy rules that reduced the number of re-handlings. A slight reduction in the number of re-handlings was obtained at generation number 6 and after this the lowest number of 886 re-handlings was achieved. This was because the approach had selected the most appropriate stacks for containers which led to a reduced number of re-handlings of containers.

As discussed in busy yard scenario, in section 6.2.1.1.1, figure 6.1, there is no improvement in the number of re-handlings after generation 22. While, in the moderately busy yard scenario, there is no improvement in the number of re-handlings after generation 28 as explained in section, figure 6.5. However, in the quiet yard scenario, the minimum number of re-handlings is achieved at generation 7. The reason is, the GA has selected the most appropriate fuzzy rules at generations 22, 28 and 7 for busy, moderately busy and quiet yard scenarios respectively. After these generations of each scenario, whatever rules are selected by the GA, the number of re-handlings of containers did not improve.

For container storage operation in the yard, a stack needs to be selected. Figure 6.10 and figure 6.11 show the average stack utilisation and the number of re-handlings per stack receptively for all the storage and retrieval approaches.

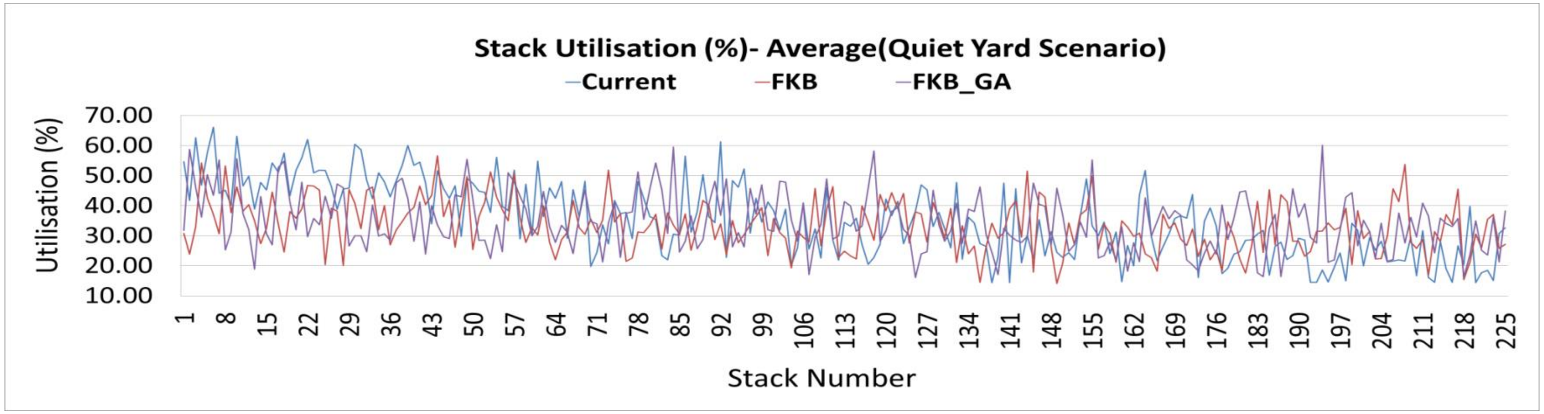


Figure 6.10: The average stack utilisation (quiet yard scenario)

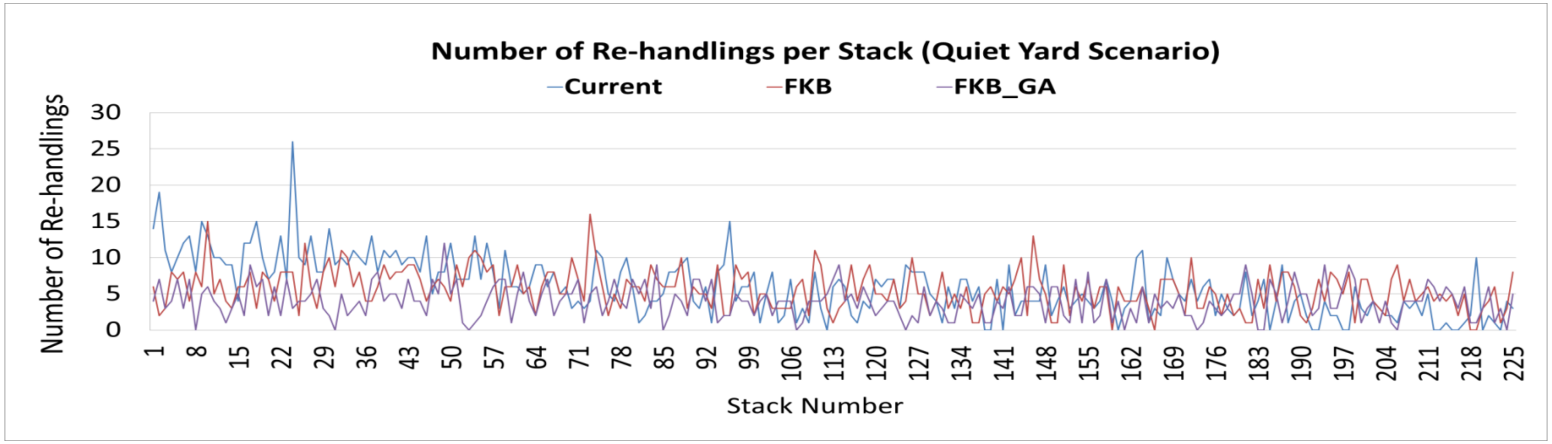


Figure 6.11: The number of re-handlings per stack (quite yard scenario)

Figure 6.10 shows that the ‘Current’ approach achieved the highest stack utilisation of 65.12% at stack number 6. This was because the number of stored containers during the storage operation was high. The ‘FKB_GA’ approach achieved the highest utilisation of 60.15% at stack 194. The ‘FKB’ approach achieved the lowest utilisation of 14.12% at stack number 149. However, the ‘Current’ approach achieved the highest average stack utilisation in most of the stacks. The lowest stack utilisation achieved by the FKB_GA approach was 15.12% at stack 125 (i.e. 15.12%). The highest utilisation achieved by the ‘FKB’ approach was 54.25% at stack 4 and the lowest utilisation was 14.12% at stack 149. The lowest stack utilisation was achieved by the ‘FKB’ approach because the number of stored containers was the lowest in this approach. The average stack utilisation is the lowest by applying the FKB_GA, the reason is discussed in section 6.2.1.1.1, figure 6.2.

The ‘FKB_GA’ achieved the lowest number of container re-handlings of all the approaches as shown in figure 6.9. This was also reflected in most of stacks as shown in figure 6.11 and contributed to the low number of re-handlings in most of stacks. The ‘Current’ approach achieved the highest number of re-handlings in most of the stacks, and could also lead to higher retrieval times in the stacks. The highest number of re-handlings achieved by the ‘Current’ approach was 26 at stack 24. The highest number of re-handlings achieved by the ‘FKB’ approach was 16 at stack 73. However, the lowest number of re-handlings was 0 which was achieved by the ‘FKB_GA’ approach and can be seen in stacks 8, 31, 53, 85, 107, 125, 161, 173, 183, 184, 206 and 224.

The number of re-handlings per stack is the lowest by applying the FKB_GA in this scenario, the reason in discussed in section 6.2.1.1.1, figure 6.3.
By adopting the ‘FKB_GA’ storage approach, the total retrieval time for all containers was minimised, which can be seen in figure 6.12.

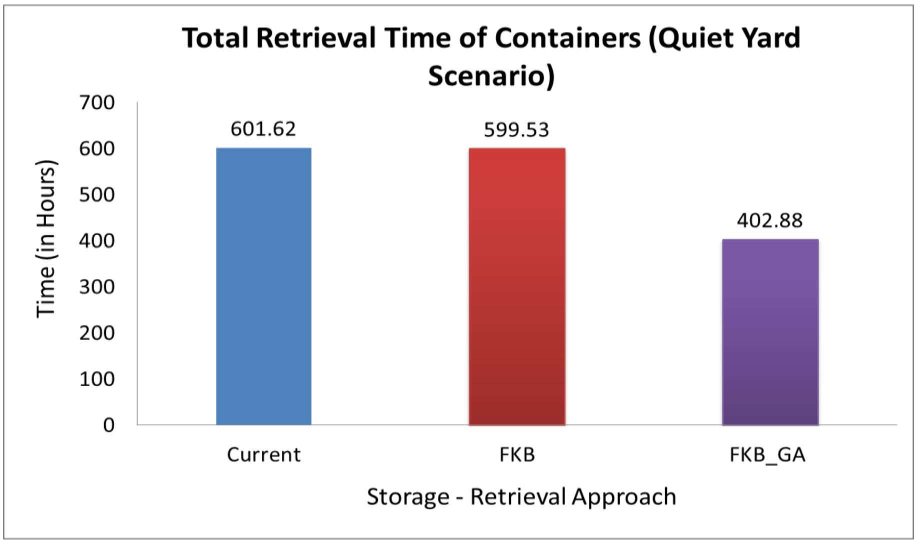


Figure 6.12: The total retrieval time of containers (quiet yard scenario)

Figure 6.12, shows that the total retrieval time achieved by the 'FKB_GA' approach was 32.8% less than the 'FKB' storage approach. This was because, as shown in figure 6.9, the total number of re-handlings achieved by the 'FKB_GA' approach was also lower than the 'FKB' approach. The highest total retrieval time was achieved by the 'Current' approach because the total number of re-handlings, as shown by figure 6.9 was also the highest of all the approaches.

6.2.1.4 Comparison Study between the Proposed Scenarios

In this section, a comparison study identifies the number of re-handlings, and total retrieval time of containers for the busy yard, moderately busy yard and quiet yard scenarios. This comparison is made between the 'Current' and 'FKB_GA' storage approaches. Figure 6.13 shows the number of re–handlings in both storage approaches for all scenarios.

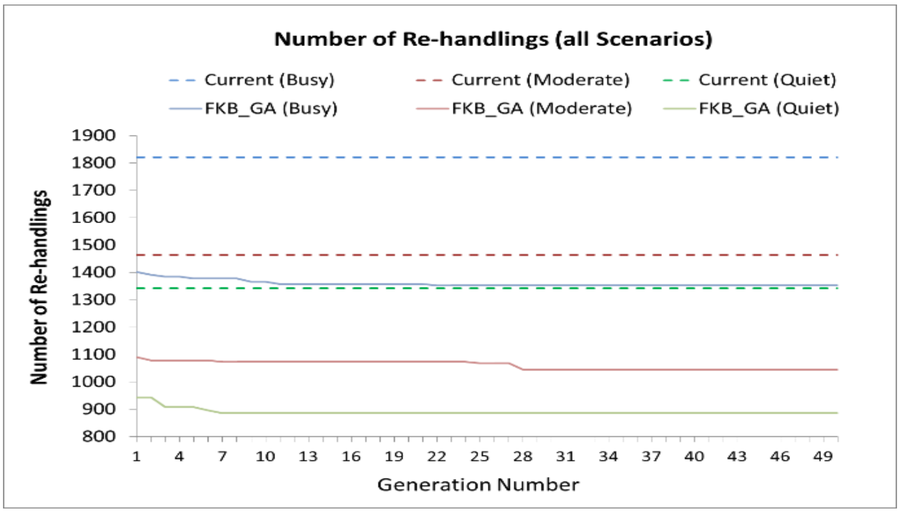


Figure 6.13: The number of re-handlings (all scenarios)

It can be seen that the 'FKB_GA' approach achieved a lower number of number of re-handlings than the current approach for both the moderately busy and quiet yard scenarios, while the 'Current' approach obtained the maximum number of re-handlings in both the busy yard and moderately busy yard scenarios. The difference in the number of re-handlings between the 'FKB_GA' approach in the busy scenario and the 'Current' approach in the quiet scenario was slight. This meant that the 'FKB_GA' approach has allocated stacks for containers storage more appropriately than the 'Current' approach although the number of containers in the 'FKB_GA' was high.

After all containers were retrieved, the total retrieval time for containers in each of the scenario was obtained as figure 6.14 shows.

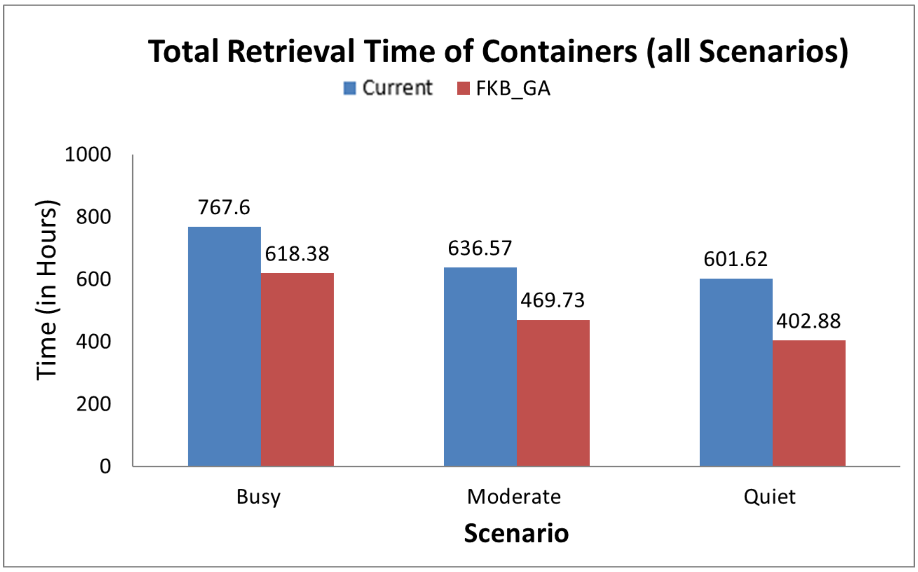


Figure 6.14: The total retrieval time of containers (all scenarios)

In figure 6.14, total retrieval time reduction was obtained by running the 'FKB_GA' storage approach in the busy yard scenario. The FKB_GA' approach drove the total retrieval time down to 618.38 hours producing a 19.4% reduction when compared with the 'Current' approach. Applying the 'FKB_GA' in the moderately busy yard scenario reduced the total retrieval time by 25.2%. While, the total retrieval time was reduced by 33% when the 'FKB_GA' storage approach was applied in the quiet yard scenario. However, the reduction of the total retrieval time by the 'FKB_GA' approach was noteworthy in all scenarios. As discussed in figure 6.13, the minimum number of re-handlings of containers was obtained by applying the 'FKB_GA' approach in all scenarios which led to a reduced total retrieval time.

In general, it can be concluded that the Fuzzy Knowledge Based Genetic Algorithm ('FKB_GA') system showed the most beneficial improvements in the quiet yard scenario, as in this scenario, the total number of re-handlings of containers was reduced by 34.03%.

In order to test the developed system with slightly different inputs, a number of experiments under three different sets of GA parameter settings for the three scenarios including busy, moderately busy and quiet yards was considered. See Appendices G, H, and I for the result diagrams for these different parameters. These parameters included the population size and the probabilities of crossover and mutation. See Appendices J, K, L for the different sets of GA parameter settings.

6.2.1.5 Effect of the Arrival Rate on the Operations Performance

In this section, further experiments to test the behaviour of the Fuzzy Knowledge Based GA (FKB_GA) System are described. These further experiments include the rush scenario when the inter-arrival time of trains that bring containers was low, and the slow scenario when the inter-arrival of trains was high. Each of these scenarios was tested under different yard conditions. These conditions were busy yard (i.e. with a significant number of existing containers), moderately busy yard (i.e. with a moderately large number of existing containers) and quiet yard (i.e. with a small number of existing containers) The results of these two scenarios are explained and analysed below in details with their experimental results.

6.2.1.5.1 Rush Arrival of Container Trains Scenario

The rush scenario assumes the inter-arrival time of container trains is small, so the number of arrival trains will be large. In this scenario, the inter-arrival time between trains is 6 hours and 4 trains arrive each day with 30 to 60 of containers on each. All the other inputs of the system are as listed in Chapter 5, table 5.1. Figure 6.15 demonstrates the GA results of the rush scenario for the busy, moderately busy and quiet yard conditions.

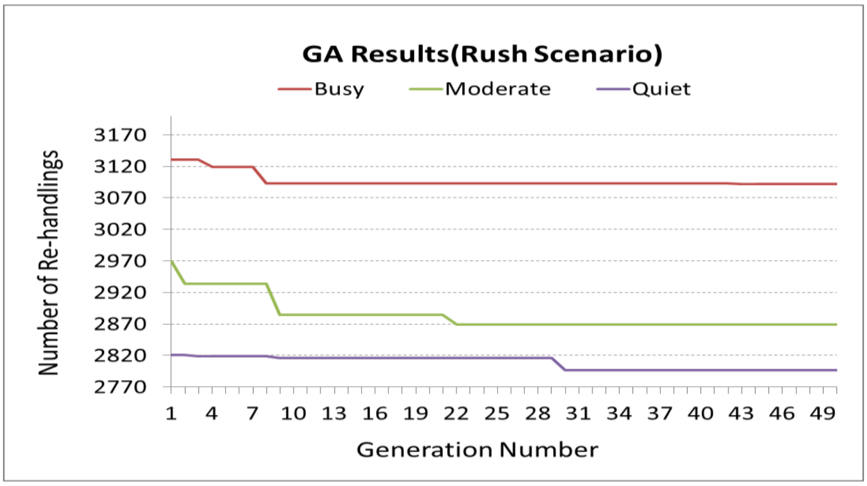


Figure 6.15: The rush scenario results

In this figure, the system obtained the highest number of re-handlings of containers (3092 re-handlings) in the busy yard scenario, while the system achieved the lowest number of re-handlings in the quiet yard scenario. In the busy yard scenario, the system reduced the number of re-handlings from 3131 to 3092, this is because the GA optimisation has selected the best rules out of the fired fuzzy rules in each stack which led to the allocation of containers to stacks. A reduction in the number of re-handlings was obtained at the 4th generation, and then another reduction was achieved at the 8th generation (26 re-handlings). The number of re-handlings was reduced to 3092 re-handlings at the 43th generation In the moderately busy yard scenario, the number of re-handlings was 2970 re-handlings in the initial population. This number was reduced to 2869 re-handlings, the reason for this reduction was because of the GA operators tuning (i.e. crossing over and mutation) that resulted in activating the optimal fuzzy rules in the stacks to allocate the optimal stacks for container storage.

Remarkable reductions were achieved at the 2nd and 9th generations, the lowest number of re-handlings was achieved in this scenario at the 22nd generation. The number of re-handlings in the quiet yard scenario started from 2821 at the 1st generation then small reductions were obtained at the 3rd and 9th generations. At the 30th generation, a worthwhile decrease in the number of re-handlings was achieved. This late improvement in the number of re-handlings was because the GA selected the most appropriate fuzzy rules at the 30th generation.

After the retrieval operation was complete, the total retrieval time of containers in hours was obtained as shown in figure 6.16 for each scenario.

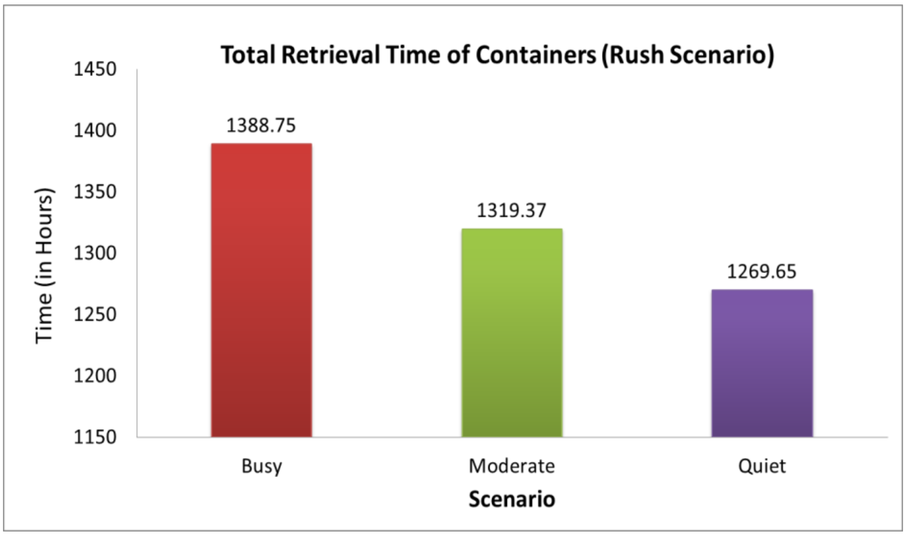


Figure 6.16: The total retrieval time of containers (rush scenario)

In figure 6.16, the highest retrieval time for containers was obtained in the busy yard scenario (1388.75 hours). The total retrieval time for containers in the moderately busy yard scenario was lower than the busy yard scenario and higher than quiet yard scenario. As discussed in figure 6.15, the total number of re-handlings was the highest in the busy yard scenario which resulted in a high total retrieval time. While the total number of re-handlings was the lowest in the quiet yard scenario as explained in figure 6.15, this resulted in a low retrieval time for containers during the retrieval operation. The total retrieval time of containers was similar for both the moderately busy and quiet yard scenarios. As discussed in figure 6.15, the total number of re-handlings in these two scenarios were similar which also led to similar total retrieval times

6.2.1.5.2 Slow Arrival of Container Trains Scenario

In this scenario, the inter-arrival time of trains was high which led to the arrival a small number of container trains. The inter-arrival time of trains was assumed to be 24 hours in this scenario, representing one train per day with 30 to 60 containers on each train. Figure 6.17 shows the GA results of the slow scenario for the busy, moderately busy and quiet yard scenarios.

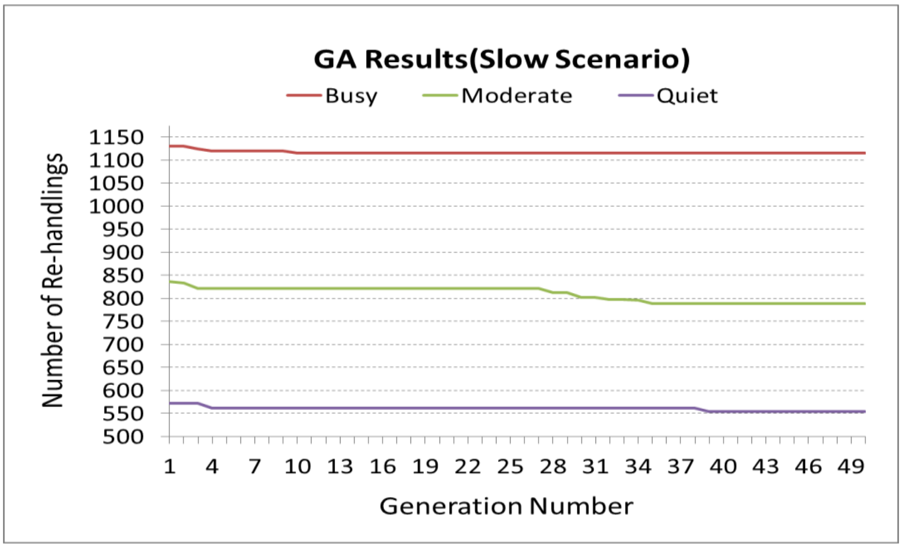


Figure 6.17: The slow scenario results

The number of re-handlings for the busy yard scenario in the initial population was 1131 re-handlings as shown in figure 6.17. After applying the GA operators, a reduction was obtained in the number of re-handlings was 1124 re-handlings at the 3rd generation. Then a slight reduction to 1120 re-handlings was achieved at the 4th generation. After the 9th generation, the minimum number of re-handlings achieved in this scenario was 1115 re-handlings. The reduction in the number of re-handlings was obtained because the GA had activated the strong fuzzy rules from the fired ones which resulted in the allocation of the best stacks for the container storage operation. In the moderately busy yard scenario, the number of re-handlings was 837 at the 1st generation. A small reduction to 833 re-handlings was achieved at the 2nd generation.

A large reduction down to 802 re-handlings was obtained from the 3rd to the 30th generation. At the 32nd and 34th generations, a further reduction down to 796 was achieved. It can be seen that the minimum number of re-handlings of 789 re-handlings was achieved after the 34th generation. This reduction in the number of re-handlings was obtained due to the fact that the most promising rules were selected in the initial population, which led to more reductions after applying the GA operators. In the quiet yard scenario, the number of re-handlings was 573 re-handlings at the beginning, then this number was reduced to 562 re-handlings at the 4th generation, being further reduced to 554 re-handlings at the 39th generation which was the minimum number of re-handlings achieved in this scenario. This reduction was achieved in the quiet yard scenario because the optimisation engine (GA) selected the active fuzzy rules for allocating the optimal stacks for container storage in the yard.

See Figure 6.18 for the total retrieval time of containers in the slow scenario.

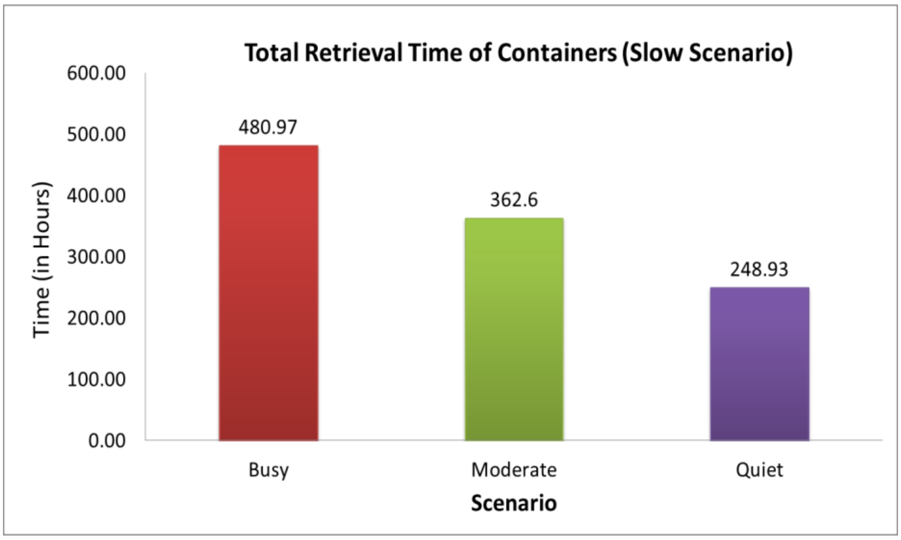


Figure 6.18: The total retrieval time of containers (slow scenario)

In this figure, the busy yard scenario resulted in a higher total retrieval time of containers than the moderately busy and the quiet yard scenarios by 24.6% and 48.2% respectively. The higher number of re-handlings in the busy yard scenario as discussed in figure 6.17 led to the highest total retrieval time of containers. The moderately busy yard scenario also achieved a high total retrieval time when compared with the quiet yard scenario, as a. higher number of re-handlings resulted in a higher retrieval time of containers during the retrieval operation (see figure 6.17).

6.2.1.5.3 The Number of Re-handlings Comparison between 'Current' and 'FKB_GA' stack allocation approaches (Rush and Slow Scenarios)

A comparison study of the number of re-handlings of containers in both rush and slow arrival of container train scenarios are explained in this section. This comparison study includes both the 'Current' and 'FKB_GA' approaches for busy, moderately busy and quiet yard conditions. The aim of the comparison is to show that the 'FKB_GA' storage approach is superior to the 'Current' approach. Figure 6.19 shows the total number of re-handlings for containers in both storage approaches for the rush arrival train scenario.

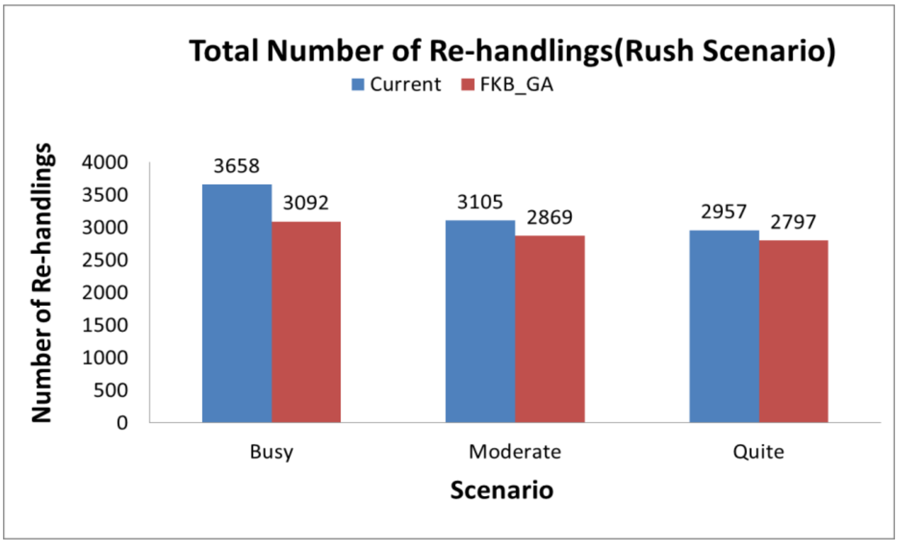


Figure 6.19: The total number of re-hamdings (rush scenario)

In figure 6.19, The ‘FKB_GA’ approach reduced the total number of re-handlings for containers by 15.4% when compared to the ‘Current’ approach in the busy yard environment. However, the ‘FKB_GA’ approach in both the moderately busy and quiet yard conditions when compared with the ‘Current’ approach reduced the number of re-handlings by 7.6% and 5.4 respectively. The total number of re-handlings for containers was also reduced under all yard conditions by the ‘FKB_GA’ approach for the rush arrival of trains scenario (see figure 6.15). The reduction was achieved because the GA optimisation process selected the strong fuzzy rules which led to the allocation of the best stacks for the storage of containers in the yard.

See Figure 6.20 for the total number of re-handlings in the slow arrival of trains scenario.

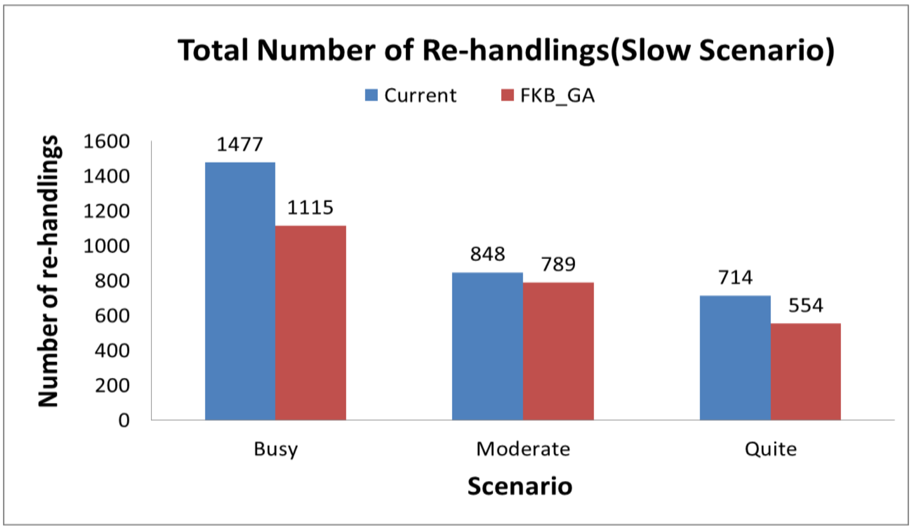


Figure 6.20: The total number of re-handings (slow scenario)

As shown in Figure 6.20, the total number of re-handlings of containers for the ‘FKB_GA’ storage approach was low when compared with the ‘Current’ approach in the busy, moderate and quiet yard environments. The total number of containers was reduced by 24.5%, 5.9% and 22.4% respectively by applying the ‘FKB_GA’ storage approach. The reason behind this reduction was because of the selection by the GA of the active fuzzy rules during the storage operation of containers, which led to a reduced total number of re-handlings in the retrieval operation.

In general, it can be concluded that the Fuzzy Knowledge Based Genetic Algorithm (‘FKB_GA’) system obtained the best results in the busy yard scenario when the arrival of container trains was slow, because the total number of re-handlings of containers was reduced by 24.5% which was the highest of all the scenarios tested.

6.2.1.6 Significant of the Yard Scenario Results

The significant of the yard scenario results is presented in this section in terms of the number of re-handlings and retrieval time of containers. The significant percentage for each technique in all the scenarios are shown in the table 6.2.

Table 6.2: The significant of the yard scenario results

Scenario	Technique	Total Number of Re-handlings of Containers	Total Retrieval Time of Containers (Hours)	Improvement Percentage (%)	
				No. of Re-handlings	Retrieval Time
Busy Yard	Current	1822	767.6	-	-
	FKB	1686	759.73	7.46	1.03
	FKB_GA	1353	618.38	25.74	19.44
Moderately Busy Yard	Current	1464	636.57	-	-
	FKB	1326	604.45	9.43	5.05
	FKB_GA	1045	469.73	28.62	26.21
Quiet Yard	Current	1343	601.62	-	-
	FKB	1249	599.53	7	3
	FKB_GA	886	402.88	34.03	33
Rushed Arrival of Container Trains with a Busy Yard	Current	3658	1520.4	-	-
	FKB_GA	3092	1388.75	15.47	8.66
Rushed Arrival of Container Trains with a Moderately Busy Yard	Current	3105	1357.67	-	-
	FKB_GA	2869	1289.52	7.6	5.02

Table 6.2: The significant of the yard scenario results

Scenario	Technique	Total Number of Re-handlings of Containers	Total Retrieval Time of Containers (Hours)	Improvement Percentage (%)	
				No. of Re-handlings	Retrieval Time
Rushed Arrival of Container Trains with a Quiet Yard	Current	2957	1269.65	-	-
	FKB_GA	2797	1259.52	5.41	2
Slow Arrival of Container Trains with a Busy Yard	Current	1477	615.83	-	-
	FKB_GA	1115	480.97	24.51	21.9
Slow Arrival of Container Trains with a Moderately Busy Yard	Current	848	393.87	-	-
	FKB_GA	789	362.6	6.96	7.94
Slow Arrival of Container Trains with a Quiet Yard	Current	714	310.58	-	-
	FKB_GA	554	284.93	22.41	8.26

The techniques included table 6.2 are: ‘Current’, ‘FKB’, and ‘FKB_GA’ approaches. The slow and rushed arrival of container trains for busy, moderately busy, and quiet yards were included in these scenarios.

The ‘FKB’ approach reduced the number of re-handlings and retrieval time of containers by 7.46% and 1.03% respectively when compared with the ‘Current’ approach in the busy yard scenario. The ‘FKB’ technique utilised all the fuzzy rules from the rule base for containers rather than selecting just the active rules, while the ‘FKB_GA’ approach reduced the number of re-handlings and retrieval time of containers by 25.74 and 19.44% respectively when compared with the ‘Current’ approach for the same scenario. The ‘FKB_GA’ technique selected the active rules from the fired fuzzy rules for stacks in the container storage operation. For the rushed arrival of container trains in a busy yard scenario, the ‘FKB_GA’ technique reduced the number of re-handlings and the total retrieval time of containers by 15.47% and 8.66% respectively when compared with the ‘Current’ approach. The ‘Current’ approach allocated stacks for containers based on a specific attribute which ensured that containers for a particular customer were stored in the same stack, while the ‘FKB_GA’ technique allocated stacks for storage of containers by selecting the optimised rules from a set of fired fuzzy rules for each stack.

As shown in the table, for the ‘FKB_GA system, significant improvements in the number of re-handlings and retrieval time for containers occurred for both the quiet yard scenario (e.g. 34.03% and 33%), and the slow arrival of container trains with the busy yard scenario (e.g. 24.51% and 21.9%).

6.2.2 Effect of the ‘ON/OFF’ Strategy on the Operations Performance

In order to test the contribution of the proposed ‘ON/OFF’ strategy together with the effect of Duration of Stay as one of the storage factors, the following experiment part was established.

This experiment was designed to consider whether or not the duration of stay (DoS) factor was to be processed within the system (i.e. activated or de-activated respectively). The experiment assumed there were already containers in the yard, used the fuzzy knowledge-based model for the storage strategy and the neighbourhood heuristic algorithm which is named Constrained-Neighbourhood Stack Allocation

(CNSA) as a re-handling strategy. Two scenarios were tested, the first scenario included the duration of stay factor, the second scenario did not.

The ‘ON/OFF’ strategy was tested with a container yard size of 8 bays, 6 rows with up to 5 containers in each stack. The duration of stay of containers in the yard was 2 to 10 days. The number of container trains was 3-5 trains a day for 1 week with an inter-arrival time of 4 hours, and 60-70 containers in each train. The containers were different weights, sizes (i.e. small, medium, large) and types (2 of small, 3 of medium and 4 of large size). The transportation time for containers: from the train side to the first bay was 3 minutes, and 2 minutes for the extra bay. The retrieval time for containers was set to 2 minutes, and the re-handling time set to 1 minute per row and 2 minutes per bay. The containers were picked up by 7 third-party logistic companies with 2–20 trucks and 15 customers each. Each customer had 3-10 containers on each train.

The contribution of the ‘ON/OFF’ strategy towards the system functionality is demonstrated by a number of KPIs including the number of re-handlings, retrieval time of containers and stack utilisation in the yard. Figure 6.21 shows the total number of re-handlings of containers in both scenarios. As can be seen, the total number of re-handlings was reduced by 5% when the DoS factor was activated.

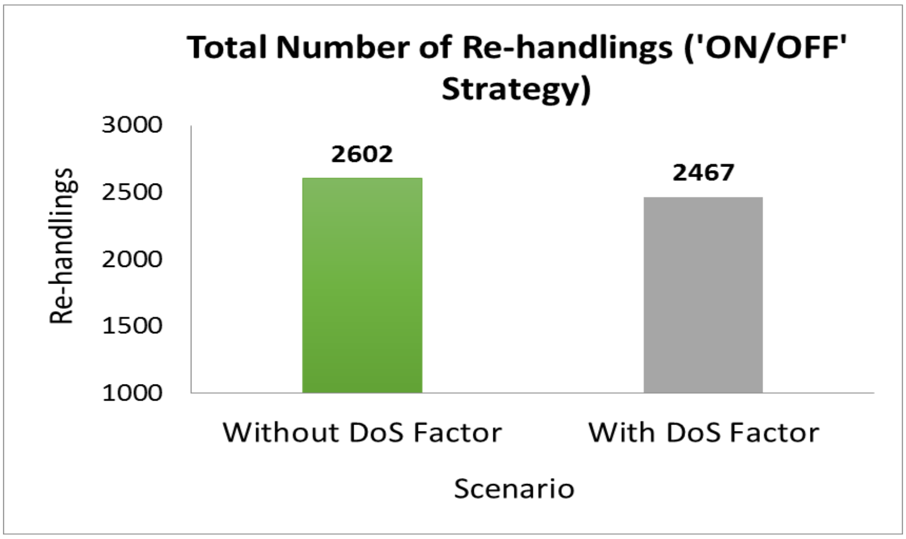


Figure 6.21: Total number of re-handlings (‘ON/OFF’ strategy)

When the DoS factor was activated, the system successfully allocated the containers, resulting in the minimum number of re-handlings. Stacks with a lower number of containers, shorter stay time, and a larger number of containers that belonged to the same customers were selected. This led to a reduced number of re-handlings because the storage operation was efficient. However, when the DoS factor was not activated, the containers in the allocated stacks were higher in number and had a longer stay time. This led to a higher number of containers being re-handled than the previous scenario because the storage operation had not taken into consideration the length of stay of the topmost containers.

Figure 6.22 shows the number of re-handlings per stack for both scenarios. Comparing the number of re-handlings achieved in stacks when the duration of stay factor was de-activated, the following results can be observed.

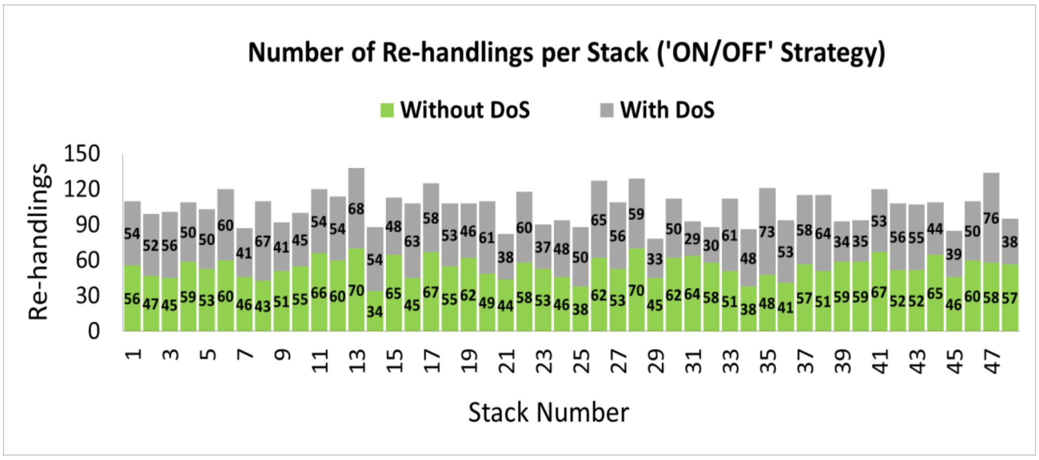


Figure 6.22: The number of re-handlings per stack ('ON/OFF' strategy)

The highest number of re-handlings can be seen at both stacks 13 and 38 (i.e. 70 re-handlings), while the lowest number of re-handlings can be seen at stack 14 (i.e. 34 re-handlings). This meant that the number of stored containers during the operation at stacks 13 and 38 was higher than the other stacks, while in stack 14 the number of stored containers was lower. When the duration of stay factor deactivated, the system selected stacks that had a long stay time for containers, which resulted in the largest number of re-handlings before the containers left the yard. However, when the duration of stay factor was activated, the highest number of re-handlings was at stack 47 (i.e. 76 re-handlings), while the lowest number of re-handlings was at stack 30 (i.e. 32 re-handlings). This meant that the number of stored containers during the operation was higher than the other stacks, but in stack 30 the number of stored containers was lower.

When the duration of stay factor was activated, the system selected stacks with containers of a shorter stay time. However, the high number of re-handlings occurred in some stacks owing to the fact that the system allocated a large number of containers (up to a maximum of five) to each of these stacks because of other constraints, such as size and weight. With regard to stack utilisation, figure 6.23 shows the average utilisation of stacks.

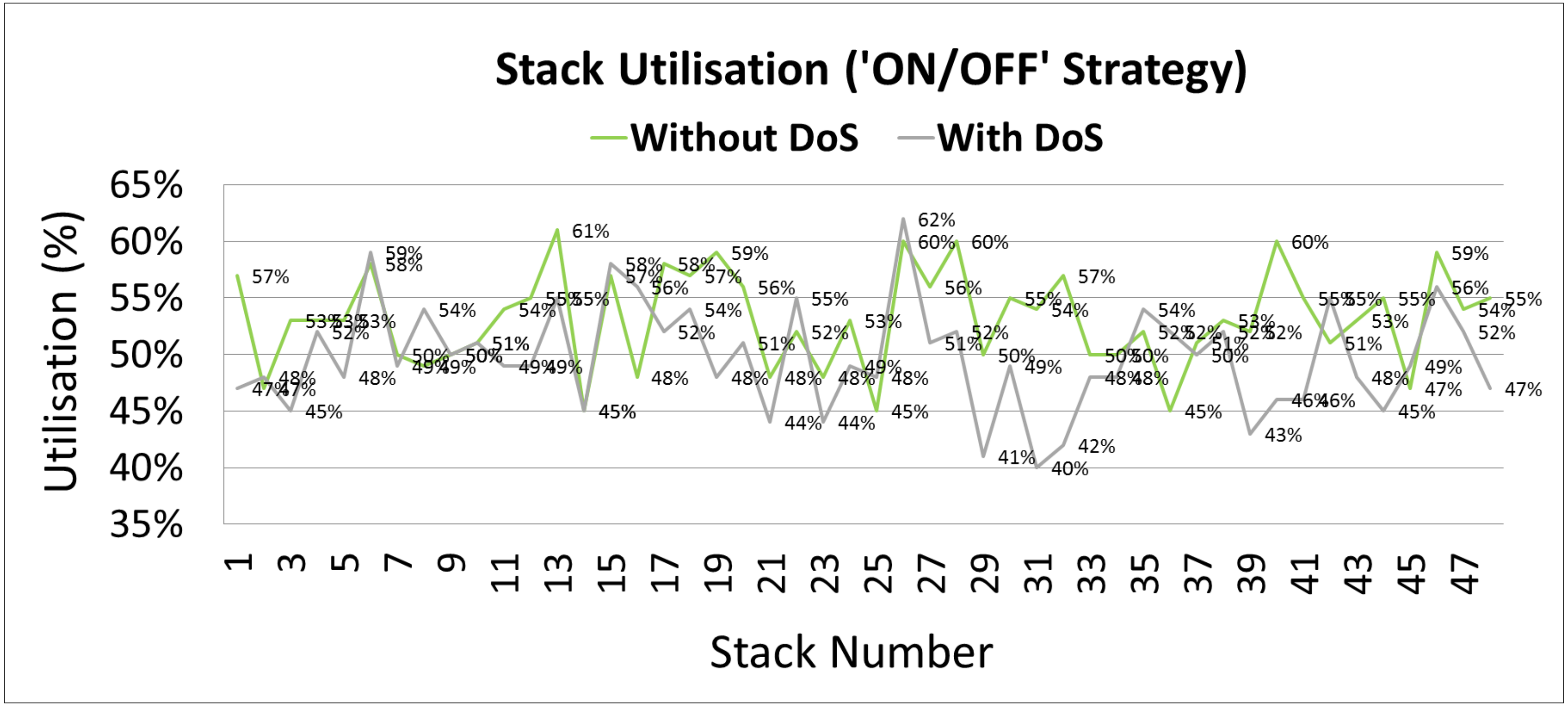


Figure 6.23: The average utilisation of stacks (‘ON/OFF’ strategy)

Figure 6.23 shows that stack 26 was the highest when the DoS factor was activated, and stack 13was the highest when the DoS factor was deactivated. Although the difference in stack utilisation was slight in both scenarios, the number of containers stored was almost the same in both scenarios in all stacks. In general, the utilisation of the stacks was lower when adopting the DoS factor because the retrieval of containers was more efficient and they required less re-handling. All the stored containers during the storage operation were retrieved after a period of time based on a random departure that followed a triangular distribution. Figure 6.24 shows the total retrieval time for all containers for both scenarios.

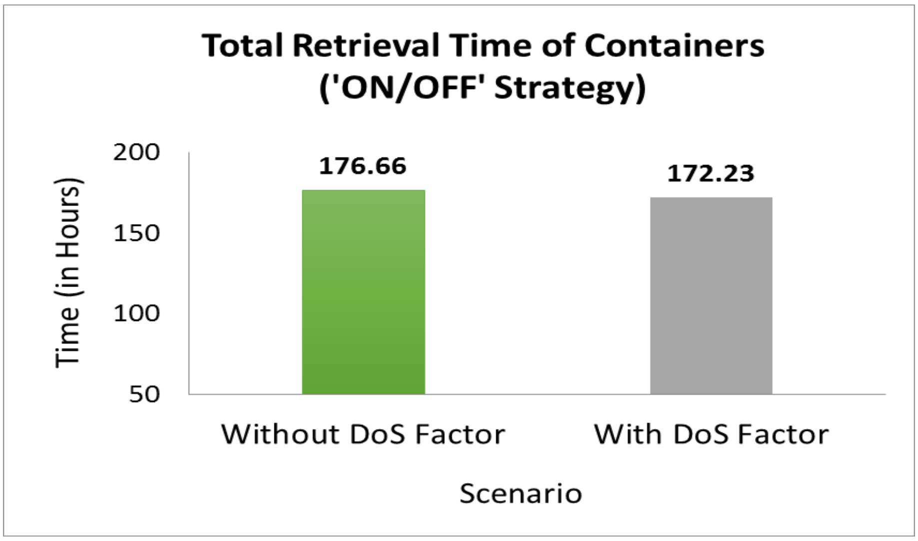


Figure 6.24: The total retrieval time of containers (‘ON/OFF’ strategy)

When the DoS factor was activated, the total retrieval time for all containers was also reduced, which can be seen in figure 6.24. This shows that a reduction in the total number of re-handlings minimises the total retrieval time for containers.

6.2.2.1 Significant of the 'ON/OFF' Strategy Results

In this section, the significant of the 'ON/OFF' strategy results are presented as shown in table

Table 6.3: The Significant of the 'ON/OFF' Strategy Results

Effect of the 'ON/OFF' Strategy on the Operation Performance				
Scenario	**Total No. of Re-handlings of Containers**	**Total Retrieval Time of Containers (Hours)**	**Improvement Percentage in the No. of Re-handlings (%)**	**Improvement Percentage in the Retrieval Time (%)**
Without DoS	2602	176.66	-	-
With DoS	2467	172.23	5.19	2.51

The effect of the proposed 'ON/OFF' strategy on the operation performance is presented table 6.3. Two scenarios were considered with this strategy including Without DoS (Duration of Stay) and with DoS. As shown in the table, the improvement percentage in the number of re-handlings and the retrieval time of containers were 5.19% and 2.51% respectively when the DoS was considered by the system developed in this research.

6.2.3 Comparison with Popular Storage-Retrieval Approaches

In this section, a comparison of the results from testing other widely used approaches for storage/ retrieval focussing on the total number of re-handlings achieved is described. The reason behind this comparison was to justify the superiority of the 'FKB_GA' (Fuzzy Knowledge Based GA) system. The comparison was conducted under different yard occupancy scenarios including busy yard, moderately busy yard and quiet yard scenarios. Each scenario was then investigated using different storage-retrieval approaches including: the proposed Fuzzy Knowledge-Based GA approach 'FKB_GA', the Constrained-Probabilistic Stack Allocation approach 'CPSA', and the Constrained-Neighbourhood Stack Allocation approach 'CNSA'. Figure 6.25 shows the comparison of total number of re-handlings obtained under the busy yard scenario.

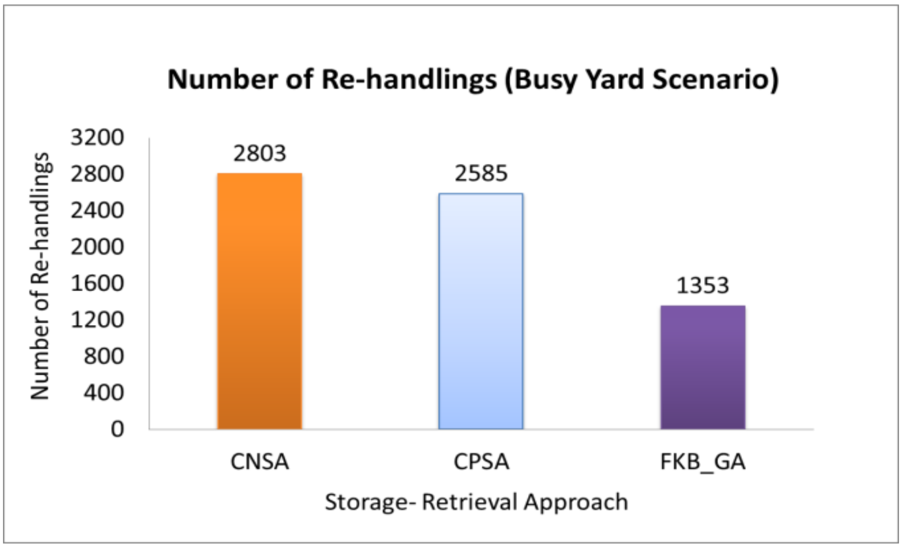


Figure 6.25: Comparison between total numbers of re-handlings obtained using different approaches (busy yard scenario)

In this figure, it is noticed that the 'FKB_GA' approach minimised the number of re-handlings. This is because the system achieved the best allocation of stacks for container storage operation that led to a considerable reduction in the number of re-handlings. This was because the GA played a vital rule in selecting promising fuzzy rules which optimised the stack allocation decision for container storage operation.

The 'CNSA' approach led to a 7% higher number of re-handlings (i.e.2803 re-handlings) than the 'CPSA' approach. The 'CNSA' approach stores containers and re-handles them to the nearest stacks taking into consideration certain constraints. The stacks in the 'CPSA' strategy were allocated taking into consideration the container size, type and weight constraints. These three storage factors were not taken into consideration during the storage and retrieval operations with this approach. The total number of re-handlings achieved by the 'FKB_GA' approach improved by 47.6% and 51.7% when compared with the 'CPSA', and 'CNSA' approaches respectively. Figure 6.26 shows the comparison of the total number of re-handlings obtained in the moderately busy yard scenario.

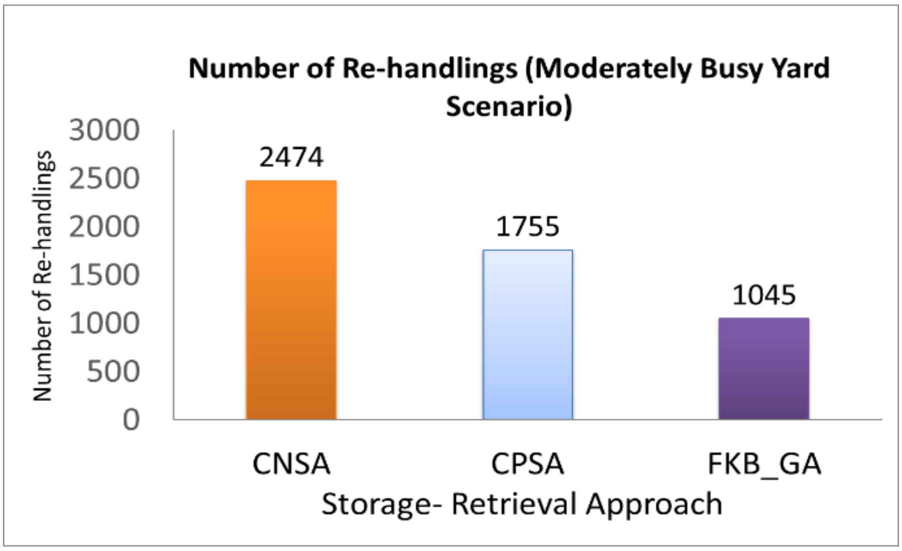


Figure 6.26: Comparison between numbers of re-handlings obtained using different approaches (moderately busy yard scenario)

In figure 6.26, it can be seen that that the ‘CNSA’ approach achieved 2474 re-handlings which was the highest of all the approaches. This approach selected the stacks based on the certain constraints for container storage and re-handled them to the closest stacks to the target stack with considering these constraints.

The number of re-handlings obtained was reduced to 1755 handlings by the ‘CPSA’ approach but this was still high when compared with the ‘CNSA’ approach. In the ‘CPSA’ approach, the stacks were allocated using the constraints for storage and retrieval of containers without taking into consideration the distance between stacks.
As can be seen in the above figure, the number of re-handlings was the lowest for the ‘FKB_GA’ approach. By comparing the total number of re-handlings with both the ‘CNSA’, and ‘CPSA’ approaches, the number of re-handlings was reduced to 57.7% and 40.4% respectively. This was because the GA allocated the best stacks for the container storage operation based on an optimised set of rules per stack that yielded a dramatic reduction in the number of re-handlings for containers during the retrieval operation (i.e. down to 1045 re-handlings). Figure 6.27 shows the comparison of the number of re-handlings obtained in the quiet yard scenario.

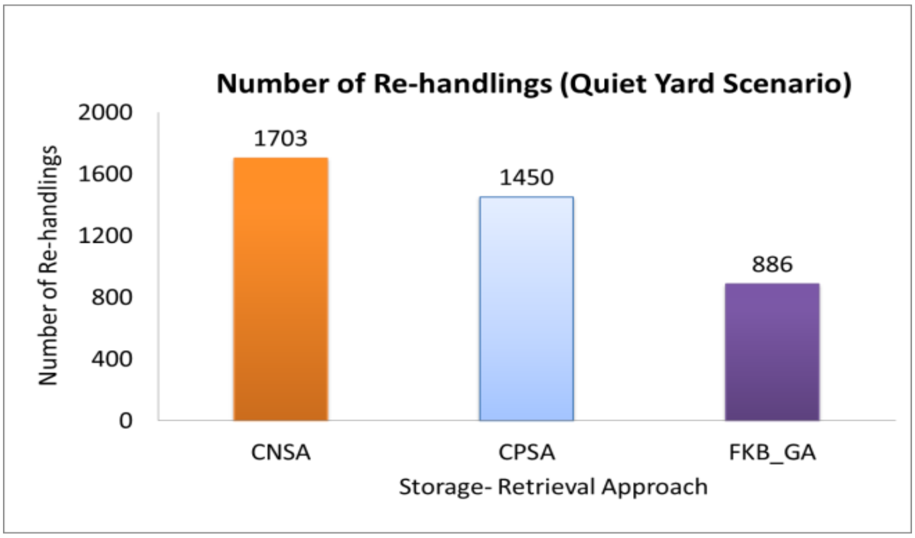


Figure 6.27: Comparison between numbers of re-handlings obtained using different approaches (quiet yard scenario)

The number of re-handlings obtained by the ‘CPSA’ approach was reduced by 14.8% to 1450 re-handlings when compared with the ‘CNSA’ approach as shown in figure 6.27. The containers were stored and re-handled according to the constraints in the yard by using the ‘CPSA’ approach.

While, the ‘CNSA’ approach achieved 1703 re-handlings which was the highest of all the approaches, the containers were stored taking into consideration weight, size and type constraints. The containers were re-handled by using the ‘CNSA’ approach to the nearest stacks using the ‘Neighbourhood’ algorithm.

The ‘FKB_GA’ approach achieved the minimum number of re-handlings. The reason is that the system selected the best stacks for the container storage operation that resulted in a dramatic reduction in the number of re-handlings. The GA selected the strong rules from the fired fuzzy rules taking into account the input factors in the stacks which led to the allocation of the optimal stacks and a minimum number of re-handlings for the containers.

The ‘FKB_GA’ approach achieved a decrease in the total number of re-handlings of containers of 47.9% and 38.8% respectively when compared with the ‘CNSA’, and ‘CPSA’ approaches.

6.3 Chapter Summary

In this chapter, the application of ‘FKB_GA’ system the resulting outputs were discussed. Three real life scenarios alongside with three storage approaches were designed to conduct the required analysis considering the total number of re-handlings, retrieval time of containers, yard utilisation, bay utilisation, and row utilisation. For proving the benefits of the approach the total number of re-handlings of containers and the utilisation of the yard compared with the ‘Current’ and ‘FKB’ approaches. By using the chromosome structure derived by the GA, and comparing the results with the ‘Current’ and ‘FKB’ approaches the efficiency of the proposed structure was determined. In addition, the system was tested a slow and rush arrival strategy for container trains. The effect of the proposed ‘ON/OFF’ strategy on the number of re-handlings and the retrieval time of containers was presented in this chapter. Significant of both the yard scenario results and the ‘ON/OFF’ strategy were discussed in details in this chapter. A comparison study with other storage and retrieval approaches was also conducted.

A detailed analysis of each scenario and storage approach was discussed in terms of the total number of re-handlings, number of re-handlings per stack, average utilisation per stack and total retrieval time of containers. The results showed that involving real-life storage factors contributed in the reduction of the number of re-handlings and retrieval time for containers in the yard. A number of experiments to test the sensitivity of the ‘FKB_GA’ system showed that decision variables such as fuzzy rules can significantly affect the number of re-handlings of containers and create more efficient storage and retrieval operations.

CHAPTER 7

CONCLUSION AND RECOMMENDATION

7.1 Introduction

In this research, the integration of FKB model with a Genetic Algorithm (GA) model was presented and discussed to develop a container stack allocation system named 'FKB_GA'. This allocation system was used to allocate suitable stacks for storage and retrieval operations for containers with an unknown departure time taking into consideration real-life factors and constraints. This allocation process led to optimised container yard operations by selecting a set of optimal/near optimal rules from the fired fuzzy rules per container and stacks to ensure the minimum operation time. The best stack allocation approach, to achieve the minimum number of re-handlings of containers being the objective of developing the stack allocation system.

This chapter summarizes a number of highlights concluded from each system component. In addition, the system behaviour was discussed in terms of the improvements achieved and their significance to the benefits of the proposed system.

7.2 Overall Conclusion of the Stack Allocation System

The system presented in this research was successfully developed to include various functions. These were:

- Optimise the storage and retrieval operations for containers.
- Simulate stack utilisation, similarity of containers in stacks and the duration of stay of containers in the yard.
- Provide for unknown departure times for containers which were transported by third party logistics trucks that arrive without any prior notice.
- Deal with differing durations of stay for containers during the storage and retrieval operations.
- Provide for the efficient management of container yards as the long duration of stay for containers makes the stack allocation decision complex in yards.

7.3 Conclusion from Literature Review

The literature review of previous works for container yard operations management was useful in order to identify the tools and techniques used in solving the problems associated with the storing and retrieving of containers. A gap in knowledge was identified through the review of a number of related container yard operations research contributions. The gap related to a number of issues, one of them being a lack of information regarding the departure time for containers as well as consideration for a collection of real life factors in the storage and retrieval operations. This issue was tackled by using the Fuzzy Knowledge Based (FKB) model. The other issue was considering the duration of stay factor for containers in the yard when performing storage and retrieval operations. The duration of stay for containers in a yard varies dynamically over time and because these containers have differing durations of stay these operations become even more challenging. To respond efficiently to these differing durations of stay for containers, an 'ON/OFF' strategy was developed to activate/deactivate this factor. In order to optimise the storage and retrieval operations, a Genetic Algorithm model was developed and embedded into the FKB model to select the most appropriate fired fuzzy rules per container and stack. This optimisation was achieved by selecting the optimal/near optimal rules from the fired fuzzy rules per stack rather than the rule base. The literature review was useful in identifying the requirement for more advanced stack allocation systems in the storage industry and to develop the allocation system incorporating all the relevant theoretical knowledge.

7.4 Fuzzy Knowledge Based (FKB) Modelling

The container yard operations management system was imitated by developing a FKB model that involved in storage and retrieval operations of containers. The input to the model included a number of factors, and the output was useful to test and compare the model by considering factors both individually and collectively in terms their effect on the total number of re-handlings. These factors were the similarity of containers in a stack, the number of containers in a stack and the duration of stay of containers in the yard.

In the busy yard scenario, when the model considered only the similarity of containers factor, the total number of re-handlings were higher compared to when both the similarity and the number of containers were considered. When the model considered all three factors the total number of re-handlings was less compared to when the similarity of containers only was considered. When the model considered all three factors the total number of re-handlings was less compared to when both the similarity and the number of containers factors were considered.

In the moderately busy yard scenario, when the model considered only the similarity of containers factor, the total number of re-handlings of containers was higher compared to when both of the other two factors were considered. When the model only considered the similarity of containers factor, the total number of re-handlings was higher than when all the factors were considered. However, when the model considered all three factors, the total number of re-handlings was less compared to when both the number, and similarity of containers factors were considered.

In the quiet yard scenario, when the model considered the combination of the similarity and number of containers factors the total number of re-handlings of containers was less compared to when the similarity factor only was considered. When the model considered all three factors the total number of re-handlings was less compared to when the similarity factor only was considered. When the model considered the similarity factor the total number of re-handlings was less compared to when the combination of both the similarity and number of containers factors was considered.

7.5 Conclusion on the 'ON/OFF Strategy'

As the duration of stay for containers increases and varies over time, an 'ON/ OFF' strategy was developed to respond efficiently to changes in the length of stay of containers in the yard. A length of stay approximation algorithm was proposed to estimate the duration of stay for each container in days. A neighbourhood algorithm was developed for an efficient retrieval operation. In general, the integration of the fuzzy knowledge-based concept together with the proposed algorithms and the suggested strategy above resulted in an advanced container operations management system. This system can be used to optimise storage and retrieval operations in yards with pre-existing containers by handling the problem of the unknown departure time taking into consideration some of the factors and constraints described in this research. The system was developed to reduce the number of re-handlings given the unknown departure time for containers. The system successfully handled this problem by taking into account the duration of stay for containers in the yard. The results indicated that stacks with more containers had a higher number of re-handlings. The duration of stay (DoS) factor played a vital role in achieving lower retrieval times (6.6%), less re-handlings (5.5%), and lower average utilisations in bays (6.3%), rows (5.9%), and tiers (9.9%). This meant that when the DoS factor was activated, containers were allocated to the correct stacks, which led to reduced operational times and less re-handling during the retrieval operation. However, the suggested approach required a large amount of data including; the length of stay; destination customer and the type, size, and weight of the container. In addition, if the fuzzy membership parameters were set subjectively, some of the generated set of rules per stack, or for some stacks, might be redundant leading to the inappropriate selection of stacks and hence such rules need to be eliminated.

7.6 Development of the Optimisation Model

In order to increase the capability of the model to search for the most appropriate stack for containers, an optimisation model was developed. The optimisation model was designed to be embedded within the FKB model for improved searching capability. This model was based on an evolutionary concept provided by a Genetic Algorithm (GA) in order to explore more promising solutions in a large solution space.

The modelling ability in the GA to handle the complexity of selecting rules out of a set of fired fuzzy rules per container and possible stacks was the reason for choosing the GA as an optimisation model, as based on the fired fuzzy rules per stack, acceptability level values were calculated for each stack. The acceptability level values have been optimised by selecting the optimal/near optimal rules. The flexibility available in the GA operators provided a convenient tool for modelling such complex container stack allocation problems.

Based on the process of selecting rules per stack for the container storage operation explained, the fired fuzzy rules for each container and each possible stack were represented as a three-dimensional vector. The fired fuzzy rules for one container per possible stack were represented by a uni-layer chromosome. For more than one container and possible stacks, the fired fuzzy rules were stored in a multi-layer chromosome structure. The height dimension of the chromosome, for the possible stacks for storing each container, was attached with each gene. The fired fuzzy rules were placed in the length dimension, which was a chromosome. This chromosome included a number of genes that represented the fired fuzzy rules per container per possible stack(s). The container number was placed in the width dimension; each container being represented in one layer with its fired fuzzy rules and possible stacks.

The GA enabled a comprehensive evaluation of the simulated container yard operations performance under different storage approaches. The operation time of containers, and the number of re-handlings for containers were successfully identified. The optimal container stack allocation for container storage and retrieval operations was identified using the proposed optimisation engine.

For testing the ‘FKB_GA’ system approach the number of re-handlings of containers was compared with the current approach used by the company under busy, moderately busy and quiet yard conditions.

For the busy scenario, the application of the proposed ‘FKB_GA’ approach resulted in a 25.7% reduction in the total number of re-handlings. In the moderately busy yard scenario, a reduction of 28.6% was obtained, while, for the quiet scenario, a reduction of 34% was achieved. These reductions were achieved because the GA engine, after a number of generations, identified the best stacks for the storage and retrieval operations of containers. The reason behind the best stack selection was that the GA engine selected the optimal rules from the fired fuzzy rules per container and stacks and unselected the weak/unnecessary ones.

In addition, the reductions achieved in the number of re-handlings for containers in both slow and rush arrival strategies for container trains were identified. For the rush arrival of container trains, in the busy yard environment, the total number of re-handlings for containers was reduced by 15.4% . By comparing the total number of re-handlings obtained with the rush arrival strategy for moderately busy and quiet yard environments, the total number of re-handlings was reduced by 7.6% and 5.4 respectively. When the slow arrival of trains strategy was considered, the total number of containers was reduced by 24.5%, 5.9% and 22.4% in busy yard, moderately busy and quiet yard environments respectively.

7.7 Calibration of GA Parameters and Computational Time of Experiments

Different population sizes were tested against different crossover and mutation probabilities to tune the GA parameters. The best set of parameters including population size, probability of crossover and probability of mutation was selected based on the minimum achieved total number of re-handlings for containers.

The population sizes were tested through 50 generations. Each population size included 5, 10 and 15 chromosomes, each chromosome contained the stored fired fuzzy rules for all stacks in the yard. The container yard considered in this case study included: 45 bays, 5 rows and 225 stacks led to the selection of these low population sizes. With this large yard size each iteration of the algorithm consumed a large amount computational time which led to the selection of these population sizes.

The computational time depended on the number of arriving containers along with yard size (number of stacks) taking into consideration the storage constraints. Each arriving container had a possible of 225 fuzzy rule sets allocated across all stacks by taking into account the storage constrains. Hence, the computational time was increasing based on the number of arriving containers and 225 or less fuzzy rule sets allocated per arriving container across all stacks.

Crossover probabilities of 0.45, 0.75 and 0.90 and mutation probabilities of 0.05, 0.10 and 0.20 were tested. These high probabilities were selected to bounce the searching process away so that a more promising part of the large solution space was explored.

7.8 Other Storage-Retrieval Techniques

The resulting outputs of the proposed ‘FKB_GA’ stack allocation system were compared with the widely used Constrained- Probabilistic Stack Allocation (CPSA) and Constrained-Neighbourhood Stack Allocation (CNSA) approaches.

In the busy yard scenario, the total number of re-handlings achieved by the ‘FKB_GA’ approach improved by 47.6% and 51.7% when compared with the ‘CPSA’, and ‘CNSA’ approaches respectively.

In the moderately busy yard scenario, by comparing the total number of re-handlings obtained by the ‘FKB_GA’ approach with both the ‘CNSA’, and ‘CPSA’ approaches, the number of re-handlings was reduced to 57.7% and 40.4% respectively.

In the quiet yard scenario, the ‘FKB_GA’ approach achieved a decrease in the total number of re-handlings of containers of 47.9% and 38.8% respectively when compared with the ‘CNSA’, and ‘CPSA’ approaches. It can be canculuded that the ‘FKB_GA’ system for stack allocation for containers outperformed all the other approaches.

7.9 Knowledge Obtained from Experimentations

A wide range of experimentations have been conducted to veirfy the behaviour of the developed system. A number of comparisons have been done to justify the superioty of the developed system.

The knowledge obtained from the experimentations and comparisons are set in a number of points including:

- The developed system has outperformed other approaches such as the current used approach and the fuzzy knowledge based model in terms of total number of re-handlings obtained considerable redcutions in the current used approach.
- The developed system outperformed more approaches including Constrained-Probabilistic Stack Allocation and Constrained-Neighbourhood Stack

Allocation approaches in which remarkable reductions in total number of re-handlings was obtained.

- The developed system is ideal to be used for quiet yard scenarios in which an improvement was obtianed using the proposed system.
- The developed system is ideal to be used for busy yard scenarios alongside slow arrival of container trains in which a noticable improvement was obtained.
- The proposed 'ON/OFF' strategy has increased the performacne of the developed FKB model by increasing its functionalty for a reduction in the total number of re-handlings was obtained.
- The proposed GA model has advanced the developed FKB model by selecting the most promising rules and deselecting the other weak rule for the best number of re-hanldings. This resulted in a noteworthy reduction of the total number of re-handlings.
- In the developed GA model, small size populations along with long number of generations were found to be ideal for running the developed system due to the considered size and dimention of the yard. This is because each chromosome represents a full yard information including fired fuzzy rules per container and stack.
- The developed system runs shorter for small size yards while this eventually increases when the yard size and dimension expands.

7.10 Limitations of the Developed Stack Allocation System

In the proposed stack allocation system, a number of limitations arose when developing the system and applying it in the allocation process. The first limitation was that the environmental impact was not considered in container yard operations. The inability to deal with unexpected sitiuitions such as reach stacker breakdown can also be considered as a second limitation. Another limitation was the computational time of the system, in case a large container yard scales were considered, the computational time of the system was long.

7.11 Recommendations for Future Study

A number of recommendations for future study have been suggested:

1. Additional factors and real-life constraints could be defined in the stack allocation system for the container storage operation, especially if they have a significant effect on the overall system performance. The duration of stay for all containers stored in a stack can be considered as one of the most influential factors that affect the allocation process of containers to stacks, and hence more features could be added to the current system to handle such container stack allocation problems. The duration of stay of the oldest container in stacks could be considered as one of the influential factors for storage and retrieval operations.

2. Consideration of the environmental impact of CO_2 produced by allocating incoming containers to each tiers of stacks in conjunction with the Vehicle Routing Problem (VRP).

3. The development of a GA model that optimises the sequence of container handling from trains to the yard can be considered. In this case, the storage time of containers might be effected.

4. Heuristic rules could be used to model storage and retrieval operations for containers, flexible retrieval operation being considered in such heuristics. Flexible retrieval could be applied to a situation when a container needs to be re-handled during the retrieval process and the truck to transport it is ready to depart, but is waiting somewhere in the queue. An alternative to re-handling that container would be to retrieve it out-of-order.

5. Resource breakdown could be considered as a random factor when a quick response is required for the sustainable storage and retrieval operations of containers.

6. Multi-objective optimisation is still worthy of consideration in solving this type of stack allocation problem. Different key performance indicators could be considered including retrieval time, number of re-handlings of containers and utilisation of transporters in order to minimise each of them in a satisfactory way.

REFERENCES

ABDOUN, O., ABOUCHABAKA, J., and TAJANI, C. (2012) 'Analyzing the Performance of Mutation Operators to Solve the Travelling Salesman Problem'. *ArXiv Preprint ArXiv:1203.3099* 18

Abdulhalim, M.F. and Attea, B.A. (2015) 'Multi-Layer Genetic Algorithm for Maximum Disjoint Reliable Set Covers Problem in Wireless Sensor Networks'. *Wireless Personal Communications* 80 (1), 203–227

Al-Bazi, A., Dawood, N., and Dean, J.T. (2010) 'Improving Performance and the Reliability of Off-Site Pre-Cast Concrete Production Operations Using Simulation Optimisation'. *Journal of Information Technology in Construction (ITcon)* 15, 335–356

Amendola, M., Neto, M.M., and Cruz, V.F. (2005) 'Using Fuzzy Sets Theory to Analyze Environmental Condition in Order to Improve Animal Productivity'. *Biomatemática, Campinas* 15, 29–40

van Asperen, E., Borgman, B., and Dekker, R. (2013) 'Evaluating Impact of Truck Announcements on Container Stacking Efficiency'. *Flexible Services and Manufacturing Journal* 25 (4), 543–556

Ayachi, I, Kammarti, R., Ksouri, M., and Borne, P. (2013a) 'A Heuristic for Re-Marshalling Unbound and Outbound Containers'. in *Systems and Computer Science (ICSCS), 2013 2nd International Conference On*. held 2013. IEEE, 291–296

Ayachi, I., Kammarti, R., Ksouri, M., and Borne, P. (2010b) 'A Genetic Algorithm to Solve the Container Storage Space Allocation Problem'. in *2010 International Conference on Computational Intelligence and Vehicular System (CIVS),* '2010 International conference on Computational Intelligence and Vehicular System (CIVS)'. held 2010 at Cheju, Republic of Korea. 1–4

Ayachi, I., Kammarti, R., Ksouri, M., and Borne, P. (2013c) ‘Harmony Search to Solve the Container Storage Problem with Different Container Types’. *International Journal of Computer Applications* 48 (22), 26–32

Bazzazi, M., Safaei, N., and Javadian, N. (2009) ‘A Genetic Algorithm to Solve the Storage Space Allocation Problem in a Container Terminal’. *Computers & Industrial Engineering* 56 (1), 44–52

Bian, Z. and Jin, Z. (2013) ‘Optimization on Retrieving Containers Based on Multi-Phase Hybrid Dynamic Programming’. *Procedia - Social and Behavioral Sciences* 96, 844–855

Bielli, M., Boulmakoul, A., and Rida, M. (2006) ‘Object Oriented Model for Container Terminal Distributed Simulation’. *European Journal of Operational Research* 175 (3), 1731–1751

Booch, G., Rumbaugh, J., and Jacobson, I. (1999) *The Unified Modeling Language User Guide*. MA: Addison-Wesley

Borgman, B., van Asperen, E., and Dekker, R. (2010) ‘Online Rules for Container Stacking’. *OR Spectrum* 32 (3), 687–716

Borjian, S., Manshadi, V.H., Barnhart, C., and Jaillet, P. (2015) ‘Managing Relocation and Delay in Container Terminals with Flexible Service Policies’. *ArXiv Preprint ArXiv:1503.01535*

Budipriyanto, A., Wirjodirdjo, B., Pujawan, I.N., and Gurning, S. (2017) ‘A Simulation Study of Collaborative Approach to Berth Allocation Problem under Uncertainty’. *The Asian Journal of Shipping and Logistics* 33 (3), 127–139

Carmona, P., Castro, J.., and Zurita, J.. (2004) ‘Strategies to Identify Fuzzy Rules Directly from Certainty Degrees: A Comparison and a Proposal’. *IEEE Transactions on Fuzzy Systems* 12 (5), 631–640

Caserta, M., Voß, S., and Sniedovich, M. (2011) ‘Applying the Corridor Method to a Blocks Relocation Problem’. *OR Spectrum* 33 (4), 915–929

Casey, B. and Kozan, E. (2012) ‘Optimising Container Storage Processes at Multimodal Terminals’. *Journal of the Operational Research Society* 63 (8), 1126–1142

de Castillo, B. and Daganzo, C.F. (1993) ‘Handling Strategies for Import Containers at Marine Terminals’. *Transportation Research Part B: Methodological* 27 (2), 151–166

Castro, J.L. (1995) ‘Fuzzy Logic Controllers Are Universal Approximators’. *IEEE Transactions on Systems, Man, and Cybernetics* 25 (4), 629–635

Chafik, R., Benadada, Y., and Boukachour, J. (2016) ‘Stacking Policy for Solving the Container Stacking Problem at a Containers Terminal.’ in *In 6th International Conference Information System Logistics and Supply Chain (ILS2016),* ‘In 6th International Conference Information System Logistics and supply chain (ILS2016)’. held 2016 at KEDGE Business School-France. 1–9

Chen, L. and Lu, Z. (2012) ‘The Storage Location Assignment Problem for Outbound Containers in a Maritime Terminal’. *International Journal of Production Economics* 135 (1), 73–80

Cintra, M.. and de Arruda, C.. (2007) ‘Fuzzy Rules Generation Using Genetic Algorithms with Self-Adaptive Selection’. in *IEEE International Conference on Information Reuse and Integration, 2007. IRI 2007*, ‘IEEE International Conference on Information Reuse and Integration, 2007. IRI 2007’. held 2007 at Las Vegas, Nevada. IEEE, 261–266

Dekker, R., Voogd, P., and van Asperen, E. (2006) ‘Advanced Methods for Container Stacking’. *OR Spectrum* 28 (4), 563–586

Fancello, G., Errico, G.D., and Fadda, P. (2008) ' PROCESSING AND ANALYSIS OF SHIP-TO-SHORE GANTRY CRANE OPERATOR PERFORMANCE CURVES IN CONTAINER TERMINALS'. *Journal of Maritime Research* V (2), 39-58

Forster, F. and Bortfeldt, A. (2012) ‘A Tree Search Procedure for the Container Relocation Problem’. *Computers & Operations Research* 39 (2), 299–309

Galle, V., Barnhart, C., and Jaillet, P. (2018) 'A New Binary Formulation of the Restricted Container Relocation Problem Based on a Binary Encoding of Configurations'. *European Journal of Operational Research* 267 (2), 467–477

Galle, V., Boroujeni, S.B., Manshadi, V.H., Barnhart, C., and Jaillet, P. (2017) 'The Stochastic Container Relocation Problem'. *ArXiv Preprint ArXiv:1703.04769* 50

Ghanbari, M.R., Azimi, P., and Abdollahi, F. (2012) 'Simulation-Based Optimization in Performance Evaluation of Marshaling Yard Storage Policy in a Container Port'. *World Academy of Science, Engineering and Technology, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering* 6 (7), 1912–1919

Gharehgozli, A., Mileski, J.P., and Duru, O. (2017) 'Heuristic Estimation of Container Stacking and Reshuffling Operations under the Containership Delay Factor and Mega-Ship Challenge'. *Maritime Policy & Management* 44 (3), 373–391

Gheith, M., Eltawil, A., Harraz, N., and Mizuno, S. (2014a) 'An Integer Programming Formulation and Solution for the Container Pre-Marshalling Problem'. in *44th Int. Conf. on Computers and Industrial Engineering, Istanbul, Turkey*. held 2014

Gheith, M., Eltawil, A.B., and Harraz, N.A. (2016b) 'Solving the Container Pre-Marshalling Problem Using Variable Length Genetic Algorithms'. *Engineering Optimization* 48 (4), 687–705

Gheith, M.S., ElTawil, A.B., and Harraz, N.A. (2013c) 'A Proposed Heuristic for Solving the Container Pre-Marshalling Problem'. in *The 19th International Conference on Industrial Engineering and Engineering Management* [online] ed. by Qi, E., Shen, J., and Dou, R. Berlin, Heidelberg: Springer Berlin Heidelberg, 955–964. available from <http://link.springer.com/10.1007/978-3-642-37270-4_91> [7 April 2018]

Guerra-Olivares, R., González-Ramírez, R.G., and Smith, N.R. (2015) ‘A Heuristic Procedure for the Outbound Container Relocation Problem during Export Loading Operations’. *Mathematical Problems in Engineering* 2015

Güven, C. and Eliiyi, D.T. (2014) ‘Trip Allocation and Stacking Policies at a Container Terminal’. *Transportation Research Procedia* 3, 565–573

Hakan Akyüz, M. and Lee, C.-Y. (2014) ‘A Mathematical Formulation and Efficient Heuristics for the Dynamic Container Relocation Problem’. *Naval Research Logistics (NRL)* 61 (2), 101–118

Han, Y., Lee, L.H., Chew, E.P., and Tan, K.C. (2008) ‘A Yard Storage Strategy for Minimizing Traffic Congestion in a Marine Container Transshipment Hub’. *OR Spectrum* 30 (4), 697–720

Hassanein, O.I., Aly, A.A., and Abo-Ismail, A.. (2004) ‘Genetic - PID Control for a Fire Tube Boiler’. in *In Proceeding of Second IEEE International Conference on Computational Cybernetics, 2004. ICCC 2004*, ‘in Proceeding of Second IEEE International Conference on Computational Cybernetics, 2004. ICCC 2004’ [online] held 2004 at Vienna. IEEE, 19–24. available from <http://ieeexplore.ieee.org/document/1437656/> [7 April 2018]

Holguín-Veras, J. and Jara-Díaz, S. (1999) ‘Holguín-Veras, J. and Jara-Díaz, S. (1999) ’Optimal Pricing for Priority Service and Space Allocation in Container Ports.Pdf’. *Transportation Research Part B: Methodological* 33 (2), 81–106

Hottung, A. and Tierney, K. (2016) ‘A Biased Random-Key Genetic Algorithm for the Container Pre-Marshalling Problem’. *Computers & Operations Research* 75, 83–102

Huang, S.H. and Lin, T.H. (2012) ‘Heuristic Algorithms for Container Pre-Marshalling Problems’. *Computers & Industrial Engineering* 62 (1), 13–20

Huang, Y., Liang, C., and Yang, Y. (2009) ‘The Optimum Route Problem by Genetic Algorithm for Loading/Unloading of Yard Crane’. *Computers & Industrial Engineering* 56 (3), 993–1001

Huynh, N. (2008) ‘Analysis of Container Dwell Time on Marine Terminal Throughput and Rehandling Productivity’. *Journal of International Logistics and Trade* 6 (2), 69–89

Ji, M., Guo, W., Zhu, H., and Yang, Y. (2015) ‘Optimization of Loading Sequence and Rehandling Strategy for Multi-Quay Crane Operations in Container Terminals’. *Transportation Research Part E: Logistics and Transportation Review* 80, 1–19

Jiang, X., Lee, L.H., Chew, E.P., Han, Y., and Tan, K.C. (2012) ‘A Container Yard Storage Strategy for Improving Land Utilization and Operation Efficiency in a Transshipment Hub Port’. *European Journal of Operational Research* 221 (1), 64–73

Jiang, X.J. and Jin, J.G. (2017) ‘A Branch-and-Price Method for Integrated Yard Crane Deployment and Container Allocation in Transshipment Yards’. *Transportation Research Part B: Methodological* 98, 62–75

Jin, C., Liu, X., and Gao, P. (2004) ‘An Intelligent Simulation Method Based on Artificial Neural Network for Container Yard Operation’. in Yin, F.-L., Wang, J., and Guo, C. (eds.) *Advances in Neural Networks - ISNN 2004*. held 2004. Springer Berlin Heidelberg, 904–911

Jones, E. and Michael Walton, C. (2002) ‘Managing Containers in a Marine Terminal: Assessing Information Needs’. *Transportation Research Record: Journal of the Transportation Research Board* (1782), 92–99

Jovanovic, R., Tuba, M., and Voß, S. (2017) ‘A Multi-Heuristic Approach for Solving the Pre-Marshalling Problem’. *Central European Journal of Operations Research* 25 (1), 1–28

Jovanovic, R. and Voß, S. (2014) ‘A Chain Heuristic for the Blocks Relocation Problem’. *Computers & Industrial Engineering* 75, 79–86

Junqueira, C., de Azevedo, A.T., and Ohishi, T. (2016) ‘Stowage Planning and Storage Space Assignment of Containers in Port Yards’. *Pianc-Copedec IX*

Kacprzyk, J. and Pedrycz, W. (2015) *Springer Handbook of Computational Intelligence*. Berlin, Heidelberg: Springer

KAYA, Y., UYAR, M., and TEKĐN, R. (2011) ‘A Novel Crossover Operator for Genetic Algorithms: Ring Crossover’. *ArXiv Preprint ArXiv:1105.0355* 4

Kelareva, G.. and Negnevitsky, M. (2002) ‘Multi-Layered Genetic Algorithm for Maintenance Scheduling with Multiple Parameters’. *Australian Journal of Intelligent Information Processing Systems* 7 (3/4), 122–131

Khatri, V., Ram, S., Snodgrass, R.., and Terenziani, P. (2014) ‘Capturing Telic/Atelic Temporal Data Semantics: Generalizing Conventional Conceptual Models’. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING* 26 (3), 528–248

Kim, K.H. (1997) ‘Evaluation of the Number of Rehandles in Container Yards’. *Computers & Industrial Engineering* 32 (4), 701–711

Kim, K.H. and Bae, J.W. (1998) ‘RE-MARSHALING EXPORT CONTAINERS IN PORT CONTAINER TERMINALS’. *Computers & Industrial Engineering* 35 (3–4), 655–658

Kim, K.H. and Kim, H.B. (1999a) ‘Segregating Space Allocation Models for Container Inventories in Port Container Terminals’. *International Journal of Production Economics* 59 (1–3), 415–423

Kim, K.H. and Kim, H.B. (2002b) ‘The Optimal Sizing of the Storage Space and Handling Facilities for Import Containers’. *Transportation Research Part B: Methodological* 36 (9), 821–835

Kim, K.H., Park, Y.M., and Ryu, K.-R. (2000) ‘Deriving Decision Rules to Locate Export Containers in Container Yards’. *European Journal of Operational Research* 124 (1), 89–101

Kourounioti, I., Polydoropoulou, A., and Tsiklidis, C. (2016) ‘Development of Models Predicting Dwell Time of Import Containers in Port Container

Terminals – An Artificial Neural Networks Application'. *Transportation Research Procedia* 14, 243–252

Ku, D. and Arthanari, T.S. (2016) 'Container Relocation Problem with Time Windows for Container Departure'. *European Journal of Operational Research* 252 (3), 1031–1039

LAN, L.W., KAO, C.K., and student, P.D. (2001) *A COMPARISON OF STACKING EFFICIENCY FOR VARIOUS STRATEGIES OF SLOT ASSICNMENT IN CONTAINER YARDS*. 4 (1), 303–318

Lee, C.C. (1990) 'Fuzzy Logic in Control Systems: Fuzzy Logic Controller. I'. *IEEE Transactions on Systems, Man, and Cybernetics* 20 (2), 404–418

Lee, Y. and Chao, S.L. (2009) 'A Neighborhood Search Heuristic for Pre-Marshalling Export Containers'. *European Journal of Operational Research* 196 (2), 468–475

Lee, Y. and Hsu, N.-Y. (2007) 'An Optimization Model for the Container Pre-Marshalling Problem'. *Computers & Operations Research* 34 (11), 3295–3313

Lee, Y. and Lee, Y.J. (2010) 'A Heuristic for Retrieving Containers from a Yard'. *Computers & Operations Research* 37 (6), 1139–1147

Li, W., Xiaoning, Z., Zhengyu, X., and Wei, Y. (2013) 'Container Slot Allocation Model of Railway Container Terminal Yard Based on Mixed Storage Mode'. *Advances in Information Sciences and Service Sciences* 5 (10), 370

Lim, A. and Xu, Z. (2006) 'A Critical-Shaking Neighborhood Search for the Yard Allocation Problem'. *European Journal of Operational Research* 174 (2), 1247–1259

Lin, D.Y., Lee, Y.J., and Lee, Y. (2015) 'The Container Retrieval Problem with Respect to Relocation'. *Transportation Research Part C: Emerging Technologies* 52, 132–143

Liu, Y., Kang, H., and Zhou, P. (2010) 'Fuzzy Optimization of Storage Space Allocation in a Container Terminal'. *Journal of Shanghai Jiaotong University (Science)* 15 (6), 730–735

Lodderstedt, T., Basin, D., and Doser, J. (2002) 'SecureUML: A UML-Based Modeling Language for Model-Driven Security'. in *In International Conference on the Unified Modeling Language*, 'In International Conference on the Unified Modeling Language'. held 2002 at Berlin, Heidelberg. Springer, 426–441

Menaka, R., Ranganathan, V., and Sowmya, B. (2016) 'Improving Performance of Fuzzy Based Routing Through Optimization for Manet'. *Middle-East Journal of Scientific Research (Techniques and Algorithms in Emerging Technologies)* 24, 249–260

Mi, W., Yan, W., He, J., and Chang, D. (2009) 'An Investigation into Yard Allocation for Outbound Containers'. *COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* 28 (6), 1442–1457

Misliah, Samang, L., Adisasmita, R., and Sitepu, G. (2012) 'CONTAINER STACKING YARD OPTIMUM UTILIZATION ANALYSIS OF OPERATOR AND USER ORIENTATION'. *International Journal of Civil & Environmental Engineering IJCEE-IJENS* 12 (3), 76–83

Morim, A., Sa Fortes, E., Reis, P., Cosenza, C., Doria, F., and Goncalves, A. (2017) 'Think Fuzzy System : Developing New Pricing Strategy Methods for Consumer Goods Using Fuzzy Logic'. *International Journal of Fuzzy Logic Systems* 7 (1), 1–17

Negnevitsky, M. and Kelareva, G. (2008) 'Development of a Multi-Layer Genetic Algorithm for Maintenance Scheduling in Power Systems'. in *In Transmission and Distribution Conference and Exposition, 2008. T&D. IEEE/PES*, 'In Transmission and Distribution Conference and Exposition, 2008. T&D. IEEE/PES'. held 2008 at Chicago, USA. IEEE, 1–5

Nelles, O., Fischer, M., and Muller, B. (1996) ‘Fuzzy Rule Extraction by a Genetic Algorithm and Constrained Nonlinear Optimization of Membership Functions’. in *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, 1996*, ‘Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, 1996’ [online] held 1996. IEEE, 213–219. available from <http://ieeexplore.ieee.org/document/551744/> [7 April 2018]

Ozcan, S. and Eliiyi, D.T. (2017) ‘A Reward-Based Algorithm for the Stacking of Outbound Containers’. *Transportation Research Procedia* 22, 213–221

Pace, D.K. (2000) ‘Ideas About Simulation Conceptual Model Development’. *JOHNS HOPKINS APL TECHNICAL DIGEST* 21 (3), 327–336

Park, T., Choe, R., Hun Kim, Y., and Ryel Ryu, K. (2011) ‘Dynamic Adjustment of Container Stacking Policy in an Automated Container Terminal’. *International Journal of Production Economics* 133 (1), 385–392

Pawlukowicz, P. (2012) ‘Automatic Learning of Fuzzy Logic With the Use of Genetic Algorithms’. *Advances in Manufacturing Science and Technology* 34 (4), 98–108

Rekik, I., Elkosantini, S., and Chabchoub, H. (2017) ‘Integration of Case Based Reasoning in Multi-Agent System for the Real-Time Container Stacking in Seaport Terminals’. in Martínez de Pisón, F.J., Urraca, R., Quintián, H., and Corchado, E. (eds.) *Hybrid Artificial Intelligent Systems*. held 2017. Springer International Publishing, 435–446

Ries, J., González-Ramírez, R.G., and Miranda, P. (2014) ‘A Fuzzy Logic Model for the Container Stacking Problem at Container Terminals’. in González-Ramírez, R.G., Schulte, F., Voß, S., and Ceroni Díaz, J.A. (eds.) *Computational Logistics*. held 2014. Springer International Publishing, 93–111

Saurí, S. and Martín, E. (2011) ‘Space Allocating Strategies for Improving Import Yard Performance at Marine Terminals’. *Transportation Research Part E: Logistics and Transportation Review* 47 (6), 1038–1057

Serban, C. and Carp, D. (2017) ‘A Genetic Algorithm for Solving a Container Storage Problem Using a Residence Time Strategy’. *Studies in Informatics and Control* 26 (1), 59–66

Sharif, O. and Huynh, N. (2013) ‘Storage Space Allocation at Marine Container Terminals Using Ant-Based Control’. *Expert Systems with Applications* 40 (6), 2323–2330

Shill, P.C., Akhand, M.A.H., Asaduzzaman, M., and Murase, K. (2015) ‘Optimization of Fuzzy Logic Controllers with Rule Base Size Reduction Using Genetic Algorithms’. *International Journal of Information Technology & Decision Making* 14 (05), 1063–1092

Sriphrabu, P., Sethanan, K., and Arnonkijpanich, B. (2013) ‘A Solution of the Container Stacking Problem by Genetic Algorithm’. *International Journal of Engineering and Technology* 5 (1), 45–49

Stahlbock, R. and Voβ, S. (eds.) (2008) *Vehicle Routing Problems and Container Terminal Operations–an Update of Research. In The Vehicle Routing Problem: Latest Advances and New Challenges* [online] Operations Research/Computer Science Interfaces. vol. 43. Boston, MA: Springer US. available from <http://link.springer.com/10.1007/978-0-387-77778-8> [8 April 2018]

Tanaka, S. and Tierney, K. (2018) ‘Solving Real-World Sized Container Pre-Marshalling Problems with an Iterative Deepening Branch-and-Bound Algorithm’. *European Journal of Operational Research* 264 (1), 165–180

Tang, L., Jiang, W., Liu, J., and Dong, Y. (2015) ‘Research into Container Reshuffling and Stacking Problems in Container Terminal Yards’. *IIE Transactions* 47 (7), 751–766

Tang, X., Liu, Y., Zhang, J., and Kainz, W. (2007) *Advances in Spatio-Temporal Analysis*. in ISPRS Book Series. 5 vols. Bristol, PA, USA: Taylor & Francis, Inc

Ting, C.J. and Wu, K.C. (2017) ‘Optimizing Container Relocation Operations at Container Yards with Beam Search’. *Transportation Research Part E: Logistics and Transportation Review* 103, 17–31

Ting, S.C., Wang, J.S., Kao, S.L., and Pitty, F.M. (2010) ‘Categorized Stacking Models for Import Containers in Port Container Terminals’. *Maritime Economics & Logistics* 12 (2), 162–177

Türsel Eliiyi, D., Mat, G., and Ozmen, B. (2013) ‘Storage Optimization for Export Containers in the Port of Izmir’. *PROMET - Traffic&Transportation* 25 (4), 359–367

Ünlüyurt, T. and Aydın, C. (2012) ‘Improved Rehandling Strategies for the Container Retrieval Process’. *Journal of Advanced Transportation* 46 (4), 378–393

Valdes-Gonzalez, H., Reyes-Bozo, L., Vyhmeister, E., Salazar, J.L., Sepúlveda, J.P., and Mosca-Arestizábal, M. (2015) ‘Container Stacking Revenue Management System: A Fuzzy-Based Strategy for Valparaiso Port’. *DYNA* 82 (190), 38–45

Vis, I.F.A. and de Koster, R. (2003) ‘Transshipment of Containers at a Container Terminal: An Overview’. *European Journal of Operational Research* 147 (1), 1–16

Wang, L., Zhu, X., and Xie, Z. (2014) ‘Storage Space Allocation of Inbound Container in Railway Container Terminal’. *Mathematical Problems in Engineering* 2014, 1–10

Wang, N., Jin, B., and Lim, A. (2015) ‘Target-Guided Algorithms for the Container Pre-Marshalling Problem’. *Omega* 53, 67–77

Wang, N., Jin, B., Zhang, Z., and Lim, A. (2017) ‘A Feasibility-Based Heuristic for the Container Pre-Marshalling Problem’. *European Journal of Operational Research* 256 (1), 90–101

Wang, Y.J. and Kao, C.S. (2008) ‘An Application of a Fuzzy Knowledge System for Air Cargo Overbooking under Uncertain Capacity’. *Computers & Mathematics with Applications* 56 (10), 2666–2675

Woo, Y.J. and Kim, K.H. (2011) ‘Estimating the Space Requirement for Outbound Container Inventories in Port Container Terminals’. *International Journal of Production Economics* 133 (1), 293–301

Wu, K. and Ting, C. (2012) ‘Heuristic Approaches for Minimizing Reshuffle Operations at Container Yard’. in *Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference*. held 2012. 1407–51

Yanagi Junior, T., Xin, H., Gates, R., and Ferreira, L. (2006) ‘Fuzzy Logic Model to Predict Laying Hen Body Temperature Rise during Acute Heat Stress’. in *Congresso Brasileiro de Engenharia Agrícola, João Pessoa: SBEA,* ‘Congresso Brasileiro de Engenharia Agrícola, João Pessoa: SBEA’. held 2006. 35–43

Yang, X., Zhao, N., Bian, Z., Chai, J., and Mi, C. (2015) ‘An Intelligent Storage Determining Method for Inbound Containers in Container Terminals’. *Journal of Coastal Research* 73 (sp1), 197–204

Yousefi, H., Jafari, H., Rash, K., Khosheghbal, B., and Dadkhah, A. (2012) ‘Evaluation of Causes of Delay in Container Handling Operation at Lebanese Container Ports (Case Study Beirut Container Terminal)’. *International Journal of Accounting and Financial Management* 5, 249–262

Yu, M. and Qi, X. (2013) ‘Storage Space Allocation Models for Inbound Containers in an Automatic Container Terminal’. *European Journal of Operational Research* 226 (1), 32–45

Zadeh, L.A. (1979) *A Theory of Approximate Reasoning, Machine Intelligence 9, J. Hayes, D. Michie, and LI Mikulich (Eds.), 149-194*. New York: Halstead Press

Zadeh, L.A. (1965) ‘Fuzzy Sets’. *Information and Control* 8 (3), 338–353

Zehendner, E., Feillet, D., and Jaillet, P. (2017) ‘An Algorithm with Performance Guarantee for the Online Container Relocation Problem’. *European Journal of Operational Research* 259 (1), 48–62

Zhang, C., Liu, J., Wan, Y., Murty, K.G., and Linn, R.J. (2003) ‘Storage Space Allocation in Container Terminals’. *Transportation Research Part B: Methodological* 37 (10), 883–903

Zhang, C., Zhang, Z., Zheng, L., and Miao, L. (2011) ‘A Decision Support System for the Allocation of Yard Cranes and Blocks in Container Terminals’. *Asia-Pacific Journal of Operational Research* 28 (06), 803–829

Zhang, H., Guo, S., Zhu, W., Lim, A., and Cheang, B. (2010) ‘An Investigation of IDA* Algorithms for the Container Relocation Problem’. in García-Pedrajas, N., Herrera, F., Fyfe, C., Benítez, J.M., and Ali, M. (eds.) *Trends in Applied Intelligent Systems*. held 2010. Springer Berlin Heidelberg, 31–40

Zhao, W. and Goodchild, A.V. (2010) ‘The Impact of Truck Arrival Information on Container Terminal Rehandling’. *Transportation Research Part E: Logistics and Transportation Review* 46 (3), 327–343

Zhen, L., Jiang, X., Lee, L.H., and Chew, E.P. (2013) ‘A Review on Yard Management in Container Terminals’. *Industrial Engineering and Management Systems* 12 (4), 289–304

Zimmermann, H. J. (1991) *Fuzzy Set Theory and Its Applications*. Boston: Kulwer

Zimmermann, H.J. (2001) *Fuzzy Set Theory and - Its Applications*. New York: Springer Science & Business Media

APPENDICES

Appendix A: The Summary of the Previous Work (Deterministic Departure Time of Containers)

Author(s)	Contribution in Container Yards
(Kim et al. 2000)	Discuss locating containers in storage yards.
(Dekker et al. 2006)	Address container stacking policies in a yard.
(Borgman et al. 2010)	Develop container stacking rules in container yards.
(Chen and Lu 2012)	Consider storage allocation problem of containers.
(Li et al. 2013)	Propose container slot allocation problem.
(van Asperen et al. 2013)	Identify a number of container stacking rules.
(Güven and Eliiyi 2014)	Discuss stacking strategies containers at a container terminal.
(Valdes-Gonzalez et al. 2015)	Consider container stacking problem.
(Bazzazi et al. 2009)	Address storage space allocation problem.
(Sharif and Huynh 2013)	Discuss storage space allocation problem at a container terminal.
(Jiang et al. 2012)	Apply storage space management problem in a container terminal for export containers.
(Zhang et al. 2011)	Deal with blocks allocation problem at container terminals.
(Han et al. 2008)	Address storage yard operations at a terminal.
(Mi et al. 2009)	Tackle yard block allocation problem for export containers.
(Lim and Xu 2006)	Discuss container yard allocation problem.
(Türsel Eliiyi et al. 2013)	Present determination of export container locations in the yard.
(Ghanbari et al. 2012)	Identify a yard storage policy in a container terminal.
(Wang et al. 2014)	Consider storage space allocation problem of inbound containers.
(Gheith et al. 2016a)	Tackle container pre-marshalling problem in yards.
(Gheith et al. 2013b)	Address the container pre-marshalling problem.
(Gheith et al. 2014c)	Discuss the pre-marshalling problem of containers.

Appendix A: The Summary of the Previous Work (Deterministic Departure Time of Containers)

Author(s)	Contribution in Container Yards
(Hottung and Tierney 2016)	Propose the container pre-marshalling problem.
(Wang et al. 2015)	Tackle the container pre-marshalling problem.
(Jiang and Jin 2017)	Investigate the container allocation problem in transhipment yards.
(Gharehgozli et al. 2017)	Discuss container stacking and reshuffles problem in ports.
(Serban and Carp 2017)	Deal with the container storage problem.
(Chafik et al. 2016)	Propose the container stacking problem.
(Rekik et al. 2017)	Tackle container storage process problem.
(Kourounioti et al. 2016)	Discuss the storage operation problem of containers
(Wu and Ting 2012)	Discuss relocation problem of containers based on difference of retrieval priorities between a reshuffled container and other containers
(Jones and Michael Walton 2002)	Discuss the information needs of managing of import containers in a storage yard
(Zhao and Goodchild 2010)	Evaluate the impact of truck arrival information on containers re-handling problem
(Lin et al. 2015)	Consider container retrieval problem with respect to relocation of containers in a stack yard.
(Bian and Jin 2013)	Optimising working plan to retrieve containers from a yard according to a given order
(Lee and Lee 2010)	Study the problem of retrieving containers from a yard in a given sequence.
(Galle et al. 2018)	Investigate a restricted container relocation problem.
(Kim and Bae 1998)	Study the re-marshalling export containers problem among bays at container terminals.
(Yu and Qi 2013)	Address containers retrieval operation.
(Lee and Hsu 2007)	Discuss the container re-marshalling problem.
(Huang and Lin 2012)	Consider container re-marshalling problem during the retrieval operation in yards.
(Huang et al. 2009)	Investigate container retrieval problem.
(Caserta et al. 2011)	Study retrieving problem of a subset of containers from the yard in a given order.
(Ting and Wu 2017)	Apply the container relocation problem that involves in retrieval all containers from the yard.
(Sriphrabu et al. 2013)	Introduce container stacking problem.
(Ting et al. 2010)	Discuss import container yard management problem.

Appendix A: The Summary of the Previous Work (Deterministic Departure Time of Containers)

Author(s)	Contribution in Container Yards
(Ji et al. 2015)	Container re-handling problem during retrieval operation in terminals.
(Tanaka and Tierney 2018)	Consider the container re-marshalling problem.
(Lee and Chao 2009)	Tackle re-marshalling export containers in a container yard.
(Forster and Bortfeldt 2012)	Study the container relocation problem during the retrieval processes.
(Ünlüyurt and Aydın 2012)	Consider the retrieval problem of containers from stacks.
(Hakan Akyüz and Lee 2014)	Study the container relocation problem during retrieval processes.
(Borjian et al. 2015)	Discuss the container relocation problem.
(Zhang et al. 2010)	Investigate the container relocation problem.
(Jovanovic and Voß 2014)	Address the BRP (Block Relocation Problem) in a terminal.
(Guerra-Olivares et al. 2015)	Deal with the container relocation problem in the yards.
(Jovanovic et al. 2017)	Study the pre-marshalling problem of containers in a yard.
(Wang et al. 2017)	Consider the container pre-marshalling problem (CPMP).

Appendix B: Fuzzy Sets Definitions and Membership Function

$Definition\ 1.\ Assume\ U\ is\ a\ universal\ set\ and\ \tilde{X}\ is\ a\ fuzzy\ set\ in\ U.$

$\tilde{X}\ is\ defined\ by\ the\ membership\ function\ \mu_{\tilde{X}}(x) \to [0, 1].where\ \mu_{\tilde{X}}(x), \forall x \in U.$

$Definition\ 2.\ Fuzzy\ set\ \tilde{X}\ is\ normal\ if\ and\ only\ if\ sup_x \mu_{\tilde{X}}(x) = 1.$

$Definition\ 3.\ Fuzzy\ set\ \tilde{X}\ is\ a\ convex\ set, if$

$\mu_{\tilde{X}}(\lambda x_1 + (1-\lambda)x_2) \geq \min\left(\mu_{\tilde{X}}(x_1), \mu_{\tilde{X}}(x_2)\right), \forall x_1, x_2 \in U, \forall \lambda \in [0,1].$

$Definition\ 4.\ Let\ \tilde{X}\ be\ a\ normal\ and\ convex\ fuzzy\ set\ in\ U, the\ support\ of\ \tilde{X}\ is\ the\ crisp\ set\ defined\ by\ S(\tilde{X}) = \{x | \mu_{\tilde{X}}(x) > 0\}.$

$Definition\ 5.\ Let\ \tilde{X}\ be\ a\ normal\ and\ convex\ fuzzy\ set\ in\ U, the\ \alpha\ level\ set\ of\ \tilde{X} is\ the\ crisp\ set\ defined\ by\ X_\alpha = \{x \in X | \mu_{\tilde{X}}(x) \geq \alpha\}.$

$Definition\ 6.\ A\ triangular\ fuzzy\ set\ \tilde{X}\ is\ defined\ as\ (a, b, c).$

$The\ membership\ function\ \mu_{\tilde{X}}(x)\ then\ is;$

$$\mu_{\tilde{X}}(x) = \begin{cases} \frac{x-a}{b-a}, & a \leq x < b, \\ 1, & x = b, \\ \frac{c-x}{c-b}, & b < x \leq c, \\ 0, & otherwise. \end{cases}$$

Appendix C: The Steps of Constrained-Probabilistic Stack Allocation Approach 'CPSA' for Container Storage and Retrieval Operations

Step 1: Select a stack in the yard.

Step 2: If the selected stack is empty, then go to Step 3, else then go to step 4.

Step 3: Store the container in the stack and then go to step 7.

Step 4: If the selected stack is not empty, then go to step 6 else then go to step 5.

Step 5: If all stacks are full, then go to step 1.

Step 6: Check the size, type, and weight of the arriving container with the topmost container in that stack.

- *Step 6.1: If the arriving container has the same size of the topmost container then go to step 6.2, else go to step 1.*
- *Step 6.2: If the arriving container has the same type of the topmost container, then go to step 6.3, else go to step 1.*
- *Step 6.3: If the arriving container has the same weight or less of the topmost container, then go to step 3, else go to step 1.*

Step 7: Terminate in case the storage operation is completed, else repeat Steps 1-6.

Appendix D: The Steps of Constrained-Neighbourhood Stack Allocation Approach 'CNSA' for Container Retrieval Operation

Step 1: Search for an available slot in the closest stack to the original stack.

Step 2: If the found stack is empty, then go to Step 3, else then go to step 4.

Step 3: Re-handle the container to the stack and then go to step 7.

Step 4: If the found stack is not empty, then go to step 6 else then go to step 5.

Step 5: If all stacks are full, then go to step 1.

Step 6: Check the size, type, and weight of the container being re-handled with the topmost container in that stack.

> *Step 6.1: If the container being re-handled has the same size of the topmost container then go to step 6.2, else go to step 1.*
>
> *Step 6.2: If the container being re-handled has the same type of the topmost container, then go to step 6.3, else go to step 1.*
>
> *Step 6.3: If the container being re-handled has the same weight or less of the topmost container, then go to step 3, else go to step 1.*

Step 7: Terminate in case the retrieval operation is completed, else repeat steps 1-6.

Appendix E: A Sample of the Collected Data

Customer	Date Out	Container No	Size Type	Entry Time	Time 1	Time 2	Exit Time	Carrier	Rego	Release Number	Release Method
MTLACL	02/10/13 06:50	GCNU1215433	20 DC				06:50	MTL	AV11XBL	NR	R
MTLACL	02/10/13 05:17	GCNU1229118	20 DC				05:17	MTL	GN05TFU	NR	R
MTLACL	03/10/13 16:03	GCNU2006572	20 OT				16:03	MTL	AY60YR	NR	R
MTLACL	04/10/13 17:21	GCNU1097623	20 DC				17:21	MTL	DE11XVL	NR	R
MTLACL	04/10/13 16:35	GCNU1229118	20 DC				16:35	MTL	AY62AUP	NR	R
MTLACL	04/10/13 02:19	GCNU1246136	20 DC				02:19	MTL	VX55XGA	NR	R
MTLACL	04/10/13 05:37	GCNU8000946	20 DC				05:37	MTL	EU54EUM	NR	R
MTLACL	07/10/13 17:10	GCNU1098996	20 DC				17:10	MTL	EU57HFW	NR	R
MTLACL	08/10/13 07:17	GCNU1111594	20 DC				07:17	MTL	S77MSC	NR	R
MTLACL	08/10/13 07:17	GCNU1124863	20 DC				07:17	MTL	S77MSC	NR	R
MTLACL	08/10/13 16:46	GCNU1243219	20 DC				16:46	MTL	AV11XBZ	NR	R
MTLACL	08/10/13 06:56	GCNU1243219	20 DC				06:56	MTL	DE11UCZ	NR	R
MTLACL	09/10/13 17:44	GCNU8000782	20 REEFER				17:44	MTL	AY62AAU	NR	R
MTLACL	10/10/13 07:13	GCNU1232662	20 DC				07:13	MTL	AV11XBX	NR	R
MTLACL	14/10/13 08:39	GCNU1223110	20 DC				08:39	MTL	AY62AAU	NR	R
MTLACL	14/10/13 20:39	GCNU1223110	20 DC				20:39	MTL	AV11XBX	NR	R
MTLACL	18/10/13 06:47	GCNU1104790	20 DC				06:47	MTL	EU05AGO	NR	R
MTLACL	19/10/13 06:38	AMFU3054010	20 DC				06:38	TFR		AMFU3054010	T
MTLACL	22/10/13 14:16	TRIU0312336	20 OT				14:16	MTL	EU05AGO	NR	R
MTLACL	23/10/13 15:14	GCNU1131158	20 DC				15:14	MTL	KN13UCJ	NR	R
MTLACL	24/10/13 06:52	ACLU2781616	20 DC				06:52	MTL	DK61UEY	NR	R
MTLACL	24/10/13 05:00	GCNU8000680	20 DC				05:00	MTL	DE11UZC	NR	R
MTLACL	26/10/13 01:06	ACLU2782588	20 DC				01:06	MTL	AY60YSL	NR	R
MTLACL	26/10/13 10:50	GCNU1161656	20 DC				10:50	MTL	EU11FPF	NR	R
MTLACL	30/10/13 09:07	GCNU1090526	20 DC				09:07	MTL	DE60OJY	NR	R
MTLACL	30/10/13 08:46	GCNU1264438	20 DC				08:46	MTL	AY60YSL	NR	R
MTLACL	31/10/13 09:02	GCNU1138532	20 DC				09:02	MTL	KU61EEG	NR	R
MTLACL	31/10/13 09:00	GCNU1217307	20 DC				09:00	MTL	EU61CYL	NR	R

Appendix F: Structured Interview Questions

Interview Question	Interviewee	Response	Action Taken
1. What is the size of the current container yard?	Operations Manager	The yard includes 225 stacks for container storage. 3 to 5 containers (tiers) can be stored in each stack. The yard has 45 bays, each bay consists 5 rows.	The number of tiers, bays and rows were used in the as inputs to identify the container yard size for the modelling process.
2. How long has each pre-existing container been in the yard before new containers arrive?	Operations Manager	• Each pre-existing container was in the yard for between 2 to 4 days before new containers arrived. • The pre-existing containers are stored together with the new ones. • All containers (i.e. pre-existing and new ones) departed without prior notice from customers.	• The duration of stay of pre-existing containers was used to specify the length of stay spent by containers in the yard before new containers arrived. • The new containers were stored together with the pre-existing ones in the yard.

Appendix F: Structured Interview Questions

Interview Question	Interviewee	Response	Action Taken
3. How many trains arrive per day in the yard?	Business Development Manager	1 to 2 container trains could arrive at the yard per day.	• The number of arrivals of trains was used in the system. • The Discrete Event Simulation model was used in the system to identify the arrival of container trains at the yard.
4. How many containers can be transported by the trains?	Business Development Manager	30 to 60 containers could be transported by each train.	The DES technique was used to identify the number of container arrivals for each train at the yard.
5. What is the inter-arrival time of container trains?	Operations Manager	The inter-arrival time between two container trains is fixed at 12 hours.	The inter-arrival time of trains was used in the system using DES to identify the duration between train arrivals,

Appendix F: Structured Interview Questions

Interview Question	Interviewee	Response	Action Taken
6. How many container sizes can be stored in the yard, and what are they?	Operations Manager	There were 3 sizes of containers could be stored in the yard (i.e. 20ft, 30ft, and 45ft.	• The number of different sizes for a container was used in the system. • The containers in the yards were stored on top of others with the same size.
7. How many types of container are there for each size?	Operations Manager	There were: 5 types of size 20ft, 5 types of size 30ft, and 5 types of size 45ft. The types included DC, HC, PW, OT and Reefer.	• The number of different types of container was used in the system. • Containers in the yard were stored on top of containers of the same type.
8. What is the transportation time of a container from the train to the first bay in the yard?	Operations Manager	The transportation time of a container from the train to any row in the first bay was estimated at 2 Minutes.	This transportation time was used in the system to identify time for unloading and moving a container from the train to the first bay.

Appendix F: Structured Interview Questions

Interview Question	Interviewee	Response	Action Taken
9. What is the transportation time of a container between two neighbouring bays in the yard?	Operations Manager	The transportation time of a container between two neighbour bays was 0.78 Minute.	This transportation time was used in the system to specify the time to transport a container to a neighbouring bay.
10. What is the uploading time of a container onto a truck?	Operations Manager	The uploading time of a container onto a track was estimated as 0.70 Minutes.	The uploading time of a container was used in the system to define the departure time of the container using the DES model.
11. What is the transportation time of a container between two neighbouring rows during the re-handling process in the yard?	Operations Manager	The transportation time of a container between two neighbour rows was approximately 0.76 Minutes.	The transportation time of a container was used in the system to identify the time for re-handling a container between neighbouring rows.

Appendix F: Structured Interview Questions

Interview Question	Interviewee	Response	Action Taken
12. What is the transportation time of a container between two neighbouring bays during the re-handling process in the yard?	Operations Manager	The transportation time of a container between two neighbour bays was 0.80 Minutes.	This transportation time of a container was used in the system to specify the time for re-handling a container between neighbouring bays.
13. How many 3PL companies deal with the yard operators?	Operations Manager	On average, approximately 7 companies deal with the yard operators.	The number of 3PL transportation companies was used as an input of the system.
14. How many trucks does each 3PL company own?	Operations Manager	Each 3PL company owns 20 to 30 trucks.	The number of trucks for each 3PL company was used as an input for the system.
15. How many customers does are dealt with by a 3PL company?	Operations Manager	Each 3PL company deals with 5 Customers. Each customer has a number of containers stored in the yard.	The number of customers that had containers was used in the system as an input parameter.

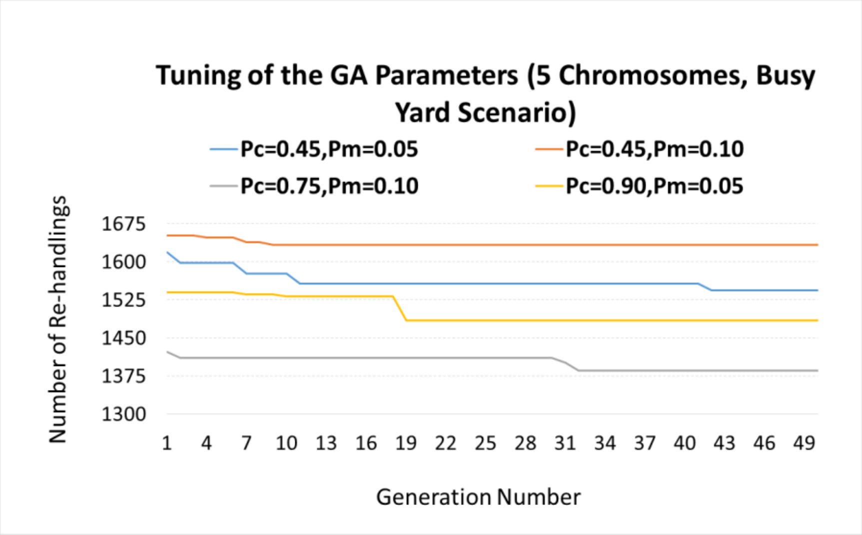
Appendix F: Structured Interview Questions

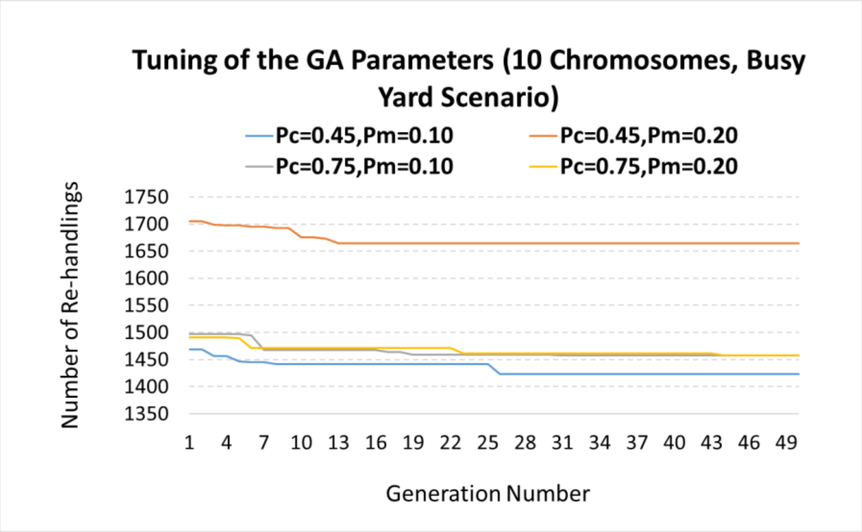
Interview Question	Interviewee	Response	Action Taken
16. What is the travel time of a truck to deliver containers to customers?	Operations Manager	The travel time of a truck to transport a container to a customer is between 1 hour to 3 hours and 20 Minutes.	The travel time of trucks was used in the system to identify the time taken to deliver containers to customers.
17. How many containers are there on the train for a customer?	Operations Manager	There are 3 to 10 containers for each customer in a train.	The number of containers per customer per train was utilised in the system.
18. What is the departure times for containers?	Operations Manager	The containers depart between 10 and 30 days, but most of them depart on day 15.	Probability density function for a triangular distribution was selected in order to estimate the departure time for containers.
19. How are the containers uploaded onto the trucks?	Operations Manager	• Trucks go to stacks where the required containers are stored. • The containers will be retrieved and uploaded onto the trucks.	This method for uploading was considered by the system as the departure event for containers.

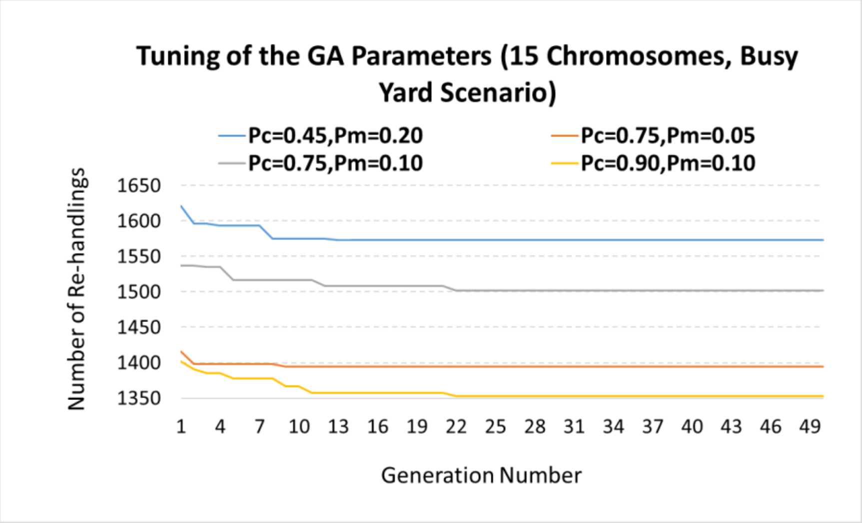
Appendix F: Structured Interview Questions

Interview Question	Interviewee	Response	Action Taken
20. How many resources are used to handle containers in the yard, and what are they?	Business Development Manager	1 reach stacker was used to handle the containers in the yard.	The number and type of resources used was considered by the system for the storage, re-handling and uploading of containers in the yard.
21. How are the containers stored in the yard?	Operations Manager	• The containers were first stored on the ground, side by side in the container yard. • Then they were stacked one on top of the other. • The containers were stored in the yard based on their customers (i.e. containers that belonged to the same customer were stored in the same stack).	• A flowchart was developed to model the storage operation of containers. • The method used for the storage operation of containers was considered by the system.

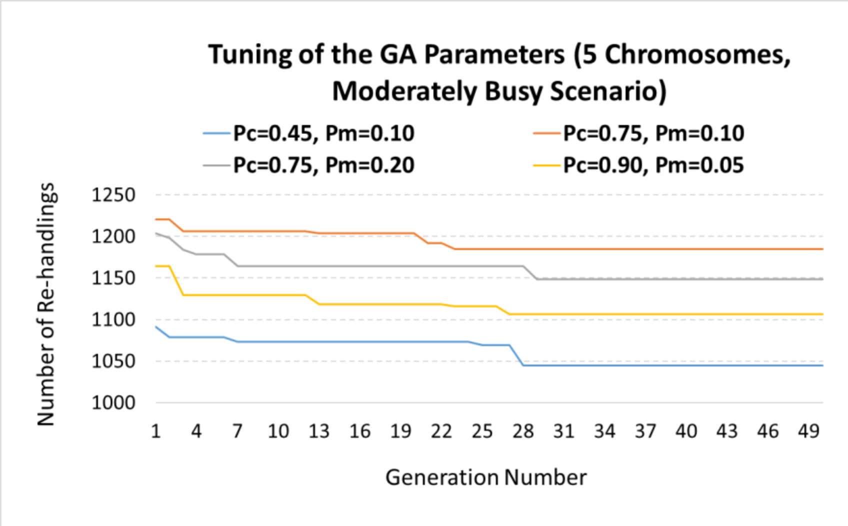
Appendix G: Tuning Diagrams of the GA Parameters-Busy Yard Scenario

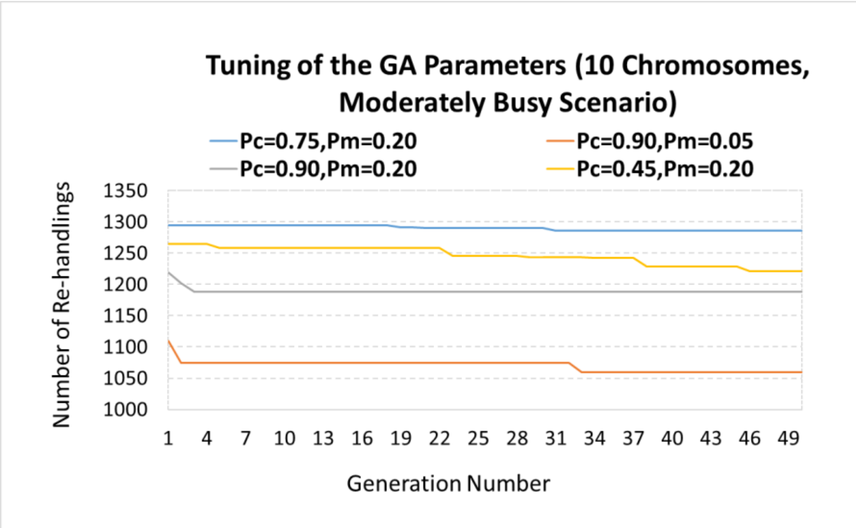


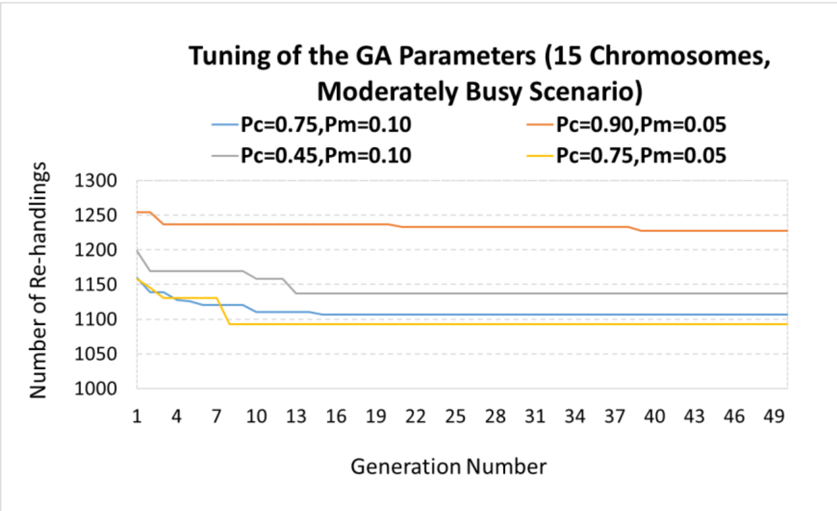




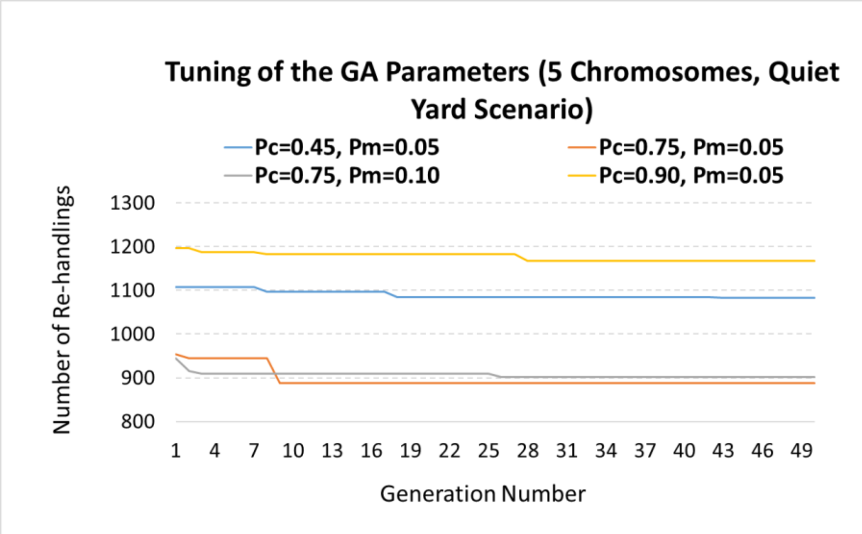
Appendix H: Tuning Diagrams of the GA Parameters-Moderately Busy Yard Scenario

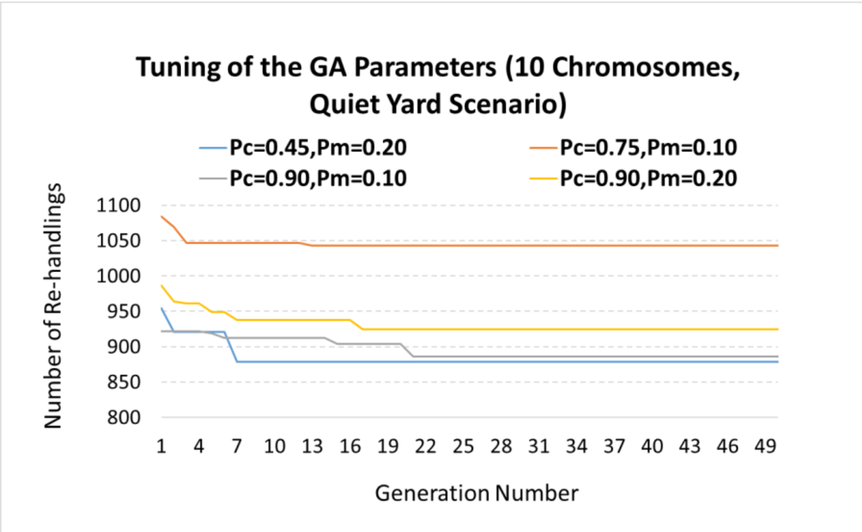


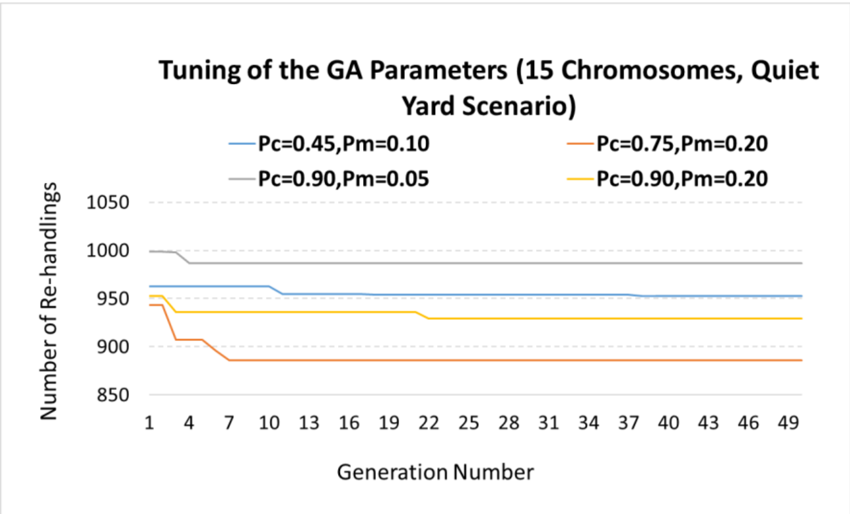




Appendix I: Tuning Diagrams of the GA Parameters-Quiet Yard Scenario







Appendix J: The Tuned GA Parameters for the Busy Yard Scenario

Population Size	Probability of Crossover	Probability of Mutation	Minimum total number of re-handlings	At which generation
5	0.45	0.05	1544	42
	0.45	0.10	1633	9
	0.75	0.10	1385	32
	0.90	0.05	1484	19
10	0.45	0.10	1423	26
	0.45	0.20	1664	13
	0.75	0.10	1457	31
	0.75	0.20	1458	44
15	0.45	0.20	1573	13
	0.75	0.05	1394	9
	0.75	0.10	1502	22
	0.90	0.10	1353	22

Appendix K: The Tuned GA Parameters for the Moderately Busy Yard Scenario

Population Size	Probability of Crossover	Probability of Mutation	Minimum total number of re-handlings	At which generation
5	0.45	0.10	1045	28
	0.75	0.10	1185	23
	0.75	0.20	1148	29
	0.90	0.05	1106	27
10	0.75	0.20	1286	31
	0.90	0.05	1060	33
	0.90	0.20	1188	3
	0.45	0.20	1221	46
15	0.75	0.10	1107	15
	0.90	0.05	1227	39
	0.45	0.10	1137	13
	0.75	0.05	1093	8

Appendix L: The Tuned GA Parameters for the Quiet Yard Scenario

Population Size	Probability of Crossover	Probability of Mutation	Minimum total number of re-handlings	At which generation
5	0.45	0.05	1083	43
	0.75	0.05	888	9
	0.75	0.10	1148	26
	0.90	0.05	901	28
10	0.45	0.20	879	7
	0.75	0.10	1043	13
	0.90	0.10	886	21
	0.90	0.20	925	17
15	0.45	0.10	953	38
	0.75	0.20	886	7
	0.90	0.05	987	4
	0.90	0.20	929	22