FUZZY OPTIMIZATION AND MODELLING 1(2) (2020) 16-42

E-ISNN: 2676-7007

Contents lists available at FOMJ

Fuzzy Optimization and Modelling

Journal homepage: http://fomj.qaemiau.ac.ir/

Fuzzy Optimization and Modeling

Paper Type: Original Article

Developing a Fuzzy Knowledge Based Optimisation System for Storage and Retrieval Operations of Long Stay Containers

Ali Hadi Hussain Joma Abbas

Faculty of Engineering, Environment and Computing, Coventry University, Coventry, West Midlands, CV1 5FB, UK

ARTICLE INFO

Article history:
Received 25 December 2019
Revised 7 April 2020
Accepted 15April 2020
Available online 25 April 2020

Keywords:
Fuzzy Knowledge Based System
Multi-Layer Genetic Algorithm
Fuzzy Rules
'ON/OFF' Strategy
Storage and Retrieval Operations of Long Stay Containers

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 based on genetic algorithm named '*F*KBGA' for the management of container yard operations that take into consideration factors and constraints of long stay containers 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 module 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. 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 Constrained-Probabilistic Stack Allocation "CPSA" and Constrained-Neighbourhood Stack Allocation "CNSA" storage and retrieval techniques.

1. Introduction

As a result of globalisation and economic growth, the need for container transportation has become very significant, and consequently leads to competition between container terminals. Thus, efficient handling operations in container terminals are becoming increasingly important as in [35]. As most of the terminal operations are concerned with the storage and movement of containers in or out of the yard, the efficiency of these operations is a very important issue as mentioned in [8].

[*] Corresponding author
E-mail Address: bbuk9999@gmail.com (Ali Hadi Hussain Joma Abbas)

One of the most complex tasks in the management of container yards is the storage operation of import containers. This is 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 which is due to go out of the yard first can lead to unnecessary handling by the yard cranes which is a costly and time-consuming operation as defined in [34].

Several techniques have been developed for import container storage operation in yards with an unknown departure time for short duration of stay such as the segregation and non-segregation strategies discussed in [10], and a fuzzy logic based rule model [29]. However, there is still a lack of advanced optimisation systems for the storage of import containers that would stay for long duration given their unpredictable departure behaviour. These containers could stay for long duration of stay in yards due to that their customers do not have enough storage facilities. These systems will assist the planners of terminal operations to achieve the most efficient allocation of containers which will eventually contribute to a much reduced total number of re-handlings, reducing the re-handling times and consequently improving the management productivity of the overall yard operations.

This study presents an innovative fuzzy knowledge-based optimisation system for the management of import container yard operations especially when containers stay for long duration, which will cause a disruption in the storage plan of the import containers. It considers real-life factors and constraints that have an effect on the storage operation of import containers. These factors include the number of containers per stack, and the duration of stay of the topmost container of the stack. The constraints of the proposed system include weight (full or empty), size, and type of containers. In addition, the fuzzy knowledge based optimisation system is developed to optimise the fuzzy rules that are contributing to the storage and retrieval operations of import containers.

The remainder of this paper is structured as follows: previous work is presented in Section 2 and Section 3 defines the container yard operations problem. The research methodology is discussed in Section 4. Section 5 describes the experiments and results analysis. A comparison study with other approaches is illustrated in Section 6 and Section 7 concludes the research.

2. Literature review

In this section, most of the existing approaches for solving the storage/stacking problems for containers with an unknown departure time are discussed. To solve this problem, Ries et al. [29] used a fuzzy logic based rule model which reduced the relocation ratio. 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 containers arrived every 5 minutes and departed randomly within a simulated time of 0 to 500 minutes. Ayachi et al. [3] proposed a Genetic Algorithm model to optimise positions of containers of different types with random delivery dates. The objective of the proposed model was to identify an optimal storage operation for containers with a short duration of stay to reduce the number of re-handlings and increase the likelihood of meeting the customer delivery deadline. Another genetic algorithm model was developed by Ayachi et al. [4] for optimising the storage-space allocation for import and export containers. The objective of the model was to minimise the re-handling operations and to organise efficiently containers with a short duration of stay in the available storage space. Junqueira et al. [15] studied the problem of storage space assignment for containers by developing a simulation-based genetic algorithm to optimise storage rules for containers with a short duration of stay. The purpose of the algorithm was to reduce unnecessary movement of containers. Huynh [12] introduced two stacking methods, mixed and non-mixed, to solve the storage operation problem, which controlled whether or not newly arrived containers were stacked on top of existing containers. These methods were introduced to evaluate the effect of short container duration of stay on storage policies based on a number of criteria, such as imported throughput, storage density, and re-handling productivity. Lawrence and Chwan-Kai [22] compared ordered and random stacking strategies for the assignment of the correct slot for 150 containers with a short duration of stay in the yard. The method used techniques which simulated the stacking of containers having both known and unknown departure times in both single and twin storage areas. The aim of the comparison was to establish the number of unproductive movements of containers for each strategy. Ozcan and Eliiyi [26] proposed a reward-based algorithm for solving the outbound container stacking problem for containers with a duration of stay of a few days. The aim of the algorithm was to minimise the number of re-handlings of containers and the travelling time for cranes 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. Different sizes, weights, types, expected departure time (EDT) and ports of destination (PoD) for containers were also taken into account. A heuristic algorithm for the dynamic remarshalling of inbound and outbound containers was presented in [5]. 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. [32] 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. [27] suggested an online search algorithm for improving the container stacking operation when their duration of stay was a week. This algorithm was used to reduce the re-handling and retrieval time for containers, which were grouped according to size or weight. Zehendner [41] 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 was not considered. Jin et al. [14] developed an intelligent neural network based on fuzzy-logic for scheduling container yard operations to improve storage of containers. The aim of the model was to reduce the total operation time. This model included system status evaluation, operation rule and stack height regulation, and operation scheduling for container duration of stay of a few days. Liu et al. [23] introduced a fuzzy based optimisation model for optimising the storage space allocation process. 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 in the yard. The model considered containers with an unknown departure time, as well as the storage of different groups of containers in the same block and stack. A mathematical model was proposed in [36] to reduce the number of reservations (i.e. clusters) for each export container group when allocating storage space for containers where the duration of stay 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. Researchers [10] used mathematical functions to analyse both segregation and non-segregation strategies for the container storage problem. The main aim was to reduce the handling effort for containers when their duration of stay was based on the number of ship arrivals. In the segregation strategy, cargoes from different ships were separated, while in the non-segregation strategy, containers from different ships were stacked together. Kim and Kim [19] further improved on the segregation strategy proposed in [10] by using mathematical difference equations.

This method considered constant, cyclic, and dynamic arrival patterns for containers with a duration of stay of 3 to 6 days, as well as the number of containers stored in each bay to achieve efficient storage operation. Saurí and Martín [30] proposed a mathematical model based on probability distribution functions to achieve optimal storage for containers with 3 to 4 day duration of stay, while minimising unnecessary movements. 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. Kim [18] suggested mathematical equations to solve the container stacking problem which were used to estimate the total number of re-handlings required to pick up all the containers in a bay. This work took into account unknown departure times for containers with a short duration of stay until all the containers were removed without considering additional containers being added. The main variables of the formulation were: the number of containers, the number of rows and the distribution of stacking heights in the bay. Zhang et al. [42] discussed a rolling horizon approach for improving the storage space allocation process. 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. Ku and Arthanari [21] 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. Yang et al. [37] developed a Multi-Objective Integer Programming Model (MOIPM) for solving the container Stacking Position Determination Problem (SPDP) when the container duration of stay was a few days.

The objective of the model was to increase the container circulation, reduce unbalanced workloads and reduce the movements of the yard crane. Casey and Kozan [7] 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. See table 1 for the summary of previous work that have been reviewed.

Although the literature above has presented various optimal allocation techniques for containers with unknown departure times, the focus of these techniques was on containers with a short duration of stay. None of them have considered containers with long durations of stay and the use of fuzzy techniques to predict the likely departure and to assess the effect of other factors on the storage and retrieval plans. Hence, this study presents an innovative fuzzy knowledge based optimisation system based on Genetic Algorithm named ‘FKBGA’, which is specially developed for efficient container storage and retrieval operations taking into account a number of realistic factors including the container duration of stay.

Table 1: The summary of previous related works

Technique	Author(s)	Container Staying Time Length Short/Long Run of Container Stay in Yard
Fuzzy Logic, FKB	[29]	Short
GA (Genetic Algorithm)	[3]	Short
	[4]	Short
	[5]	Short
Heuristics	[12]	Short
	[22]	Short
	[26]	Short
	[5]	Short
	[32]	Short
	[27]	Short
	[41]	Short
FKB Based on GA	[14]	Short
	[23]	Short
Mathematical Model	[36]	Short
	[10]	Short
	[19]	Short
	[20]	Short
	[30]	Short
	[18]	Short
	[42]	Short
	[21]	Short
Mathematical Model Based on GA	[37]	Short
Mathematical Model Based on Heuristic Algorithm	[7]	Short

3. Problem definition

The problem starts when containers of many different types, sizes, and weights brought by a train need to be stacked in a yard on other containers that can stay for long time before depart the yard. If this happens then containers on top need to be moved to access the container to be dispatched. This problem is made more complex when the departure time for the containers is unknown and these containers are allowed to stay for long stay time (maximum one-month period) before a notification is sent to customers. This can happen when customers arrange for the collection of their containers by 3PL companies without any prior notice being given to the yard operators. The 3PL companies send their trucks to the terminals for collection without any advance notice which makes the storage and retrieval operations more challenging. However, most of the yard operators are happy for containers to stay for long time given that customers should pay a pre-defined daily storage fee as customers do not have enough storage facilities in their depots.

As far as the containers are allowed to stay in the yard for long time, the duration that a container spends in the yard is crucial to be considered as it relates to its departure time, (i.e. the longer 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 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 Fig. 1 for a schematic representation of the problem.

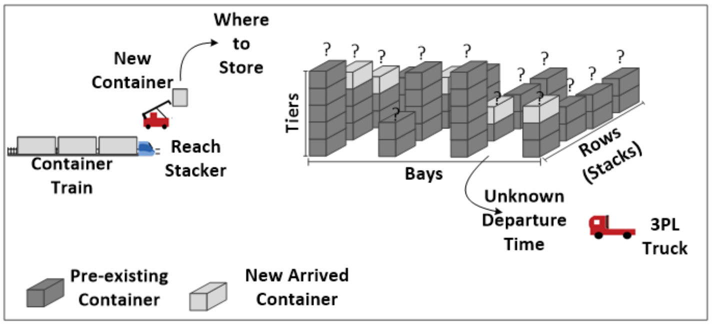


Figure 1: Schematic representation for the layout of a pre-existing container yard

A fuzzy knowledge based optimisation system is proposed to model the fuzzy aspect associated with each of the two 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.

In the next section, the tools and techniques used for modelling the container yard management system are discussed.

4. Development of ‘FKBGA’ - The Fuzzy Knowledge Based on Genetic Algorithm System

The fuzzy knowledge based model is introduced in this section for stack allocation problem of containers where the departure time is unknown. This model is used also for re-handling operation of containers during the retrieval operation. A Genetic Algorithm model is proposed and integrated with the fuzzy knowledge based model for the optimal/near optimal rules selection from a set of fired fuzzy rules for each possible stack in the yard. The term “fired rules” means the rules which are likely to fire (i.e. to a degree greater than 0) when an input is applied to a fuzzy system as in [16].

In order to imitate the events for arrival, storage, retrieval and departure of containers, the Discrete Event Simulation approach is used for this purpose. The conceptual model of the proposed ‘FKBGA’ system is explained below.

4.1. Conceptual Model of the ‘FKBGA’ System

The proposed Fuzzy Knowledge Based on Genetic Algorithm system conceptual model is presented in this section. A number of techniques including Fuzzy Knowledge based Rules and Genetic Algorithms that work together for storage and retrieval operations of containers. For the system conceptual model, see Fig. 2.

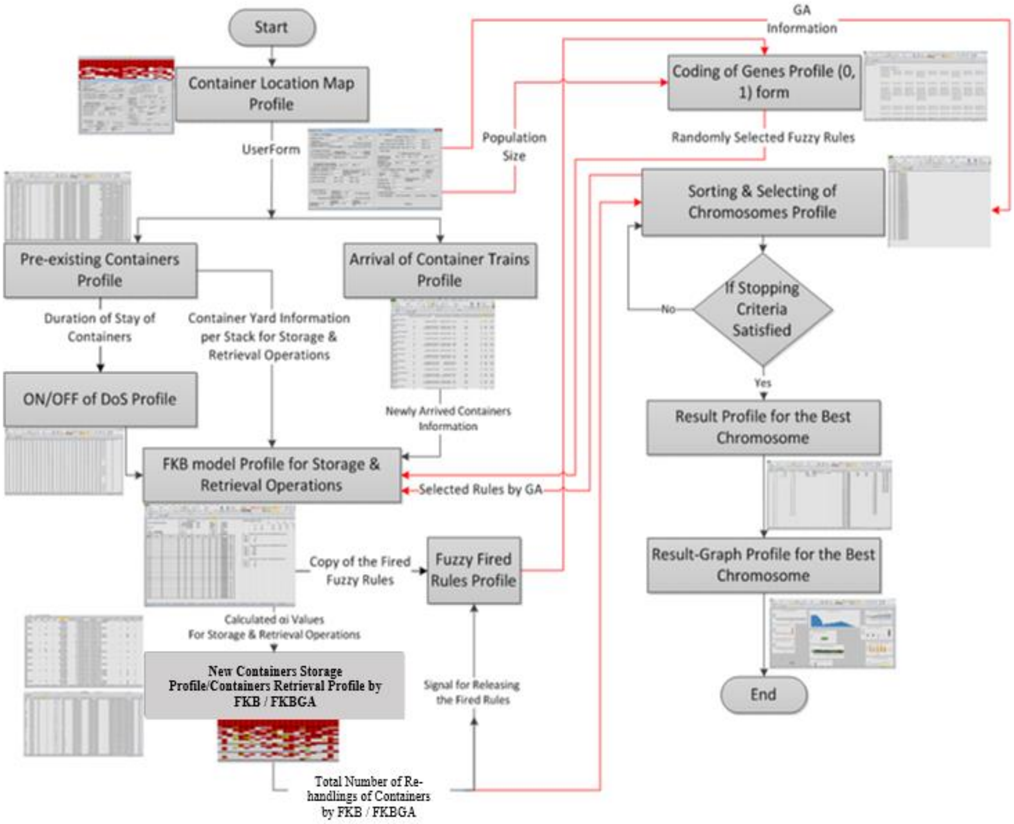

Start
Container Location Map Profile
UserForm
Population Size
GA Information
Coding of Genes Profile (0, 1) form
Randomly Selected Fuzzy Rules
Sorting & Selecting of Chromosomes Profile
Pre-existing Containers Profile
Arrival of Container Trains Profile
Duration of Stay of Containers
Container Yard Information per Stack for Storage & Retrieval Operations
If Stopping Criteria Satisfied
No
Yes
ON/OFF of DoS Profile
Newly Arrived Containers Information
Result Profile for the Best Chromosome
FKB model Profile for Storage & Retrieval Operations
Selected Rules by GA
Copy of the Fired Fuzzy Rules
Fuzzy Fired Rules Profile
Result-Graph Profile for the Best Chromosome
Calculated αi Values For Storage & Retrieval Operations
New Containers Storage Profile/Containers Retrieval Profile by FKB / FKBGA
Signal for Releasing the Fired Rules
End
Total Number of Re-handlings of Containers by FKB / FKBGA

Figure 2: Detailed Conceptual Model of the FKBGA System

The conceptual model 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 concept of the ‘ON/OFF.’ Of DoS will be explained in section 4.2.1.1. The information from ‘Arrival of Container Trains Profile’, ‘ON/OFF of DoS Profile’, and ‘Pre-existing Containers profile’ is fed to the ‘FKB model Profile’. Based on the fired fuzzy rules per stack and container, the ‘FKB model Profile’ calculates the αi values for the storage and retrieval operation for containers. These rules are fired based on factors and constraints that are considered in the system. These factors are: the number of containers per stack and duration of stay of topmost containers per stack. The constraints are container size, type, and weight (empty or full).

The α_i values calculated either by the ‘FKB’ Model or by the ‘FKBGA’ 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 αi values for the stack allocation process. After the storage and retrieval operations are completed by the ‘FKBGA’ 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 ‘FKBM Profile’ to recalculate the optimised αi 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’. In the next section, the ‘FKBGA’ system components are explained in more detail.

4.2. The ‘FKBGA’ System Components

This section discusses all the components used to develop the proposed ‘FKBGA’ system. The components are explained here to provide 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 and retrieval strategy for containers with an unknown departure time.

In the ‘FKBGA’ system, the FKB model 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 model, 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 model. 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 [38, 44] 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 model. This model 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.

An ‘ON/OFF’ strategy is used to activate/de-activate the duration of stay factor (i.e. length of stay) for containers, to prevent it being used in the calculation if the value varies significantly over time, this strategy will be explained in section 4.2.1.1. However, the imposed constraints play an important role in the storage process as they are providing the system with crisp sets. If the constraints for either the containers or stacks do not match, the acceptability levels for those stacks will be zero. See Fig. 3 for the ‘FKBGA’ system core components.

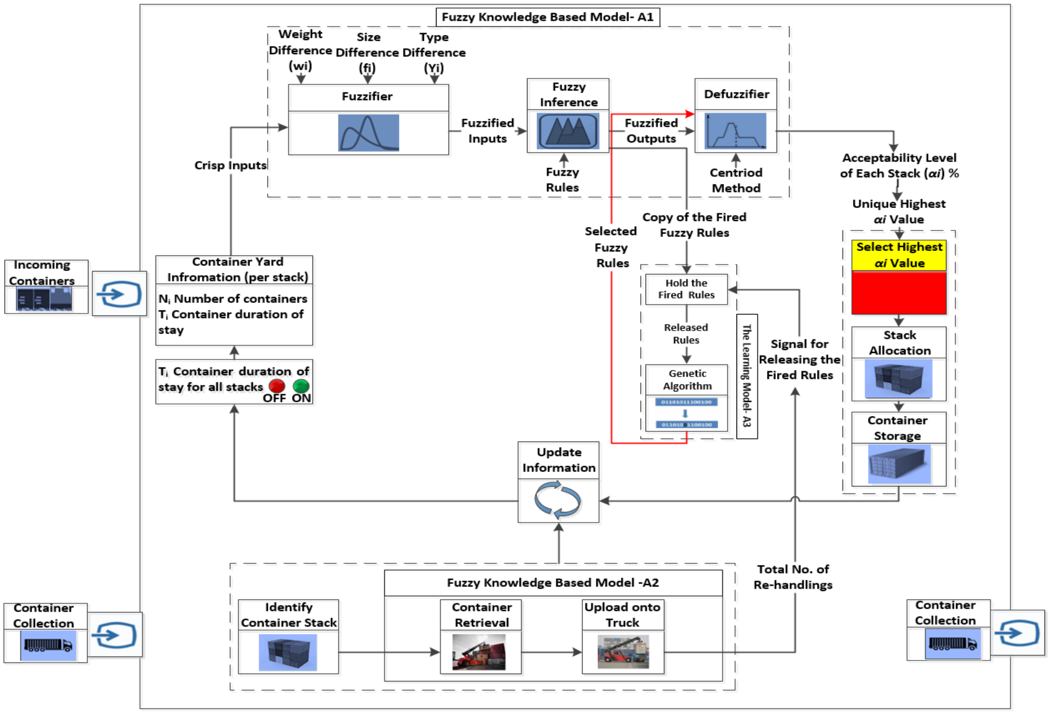


Figure 3: Components of the ‘FKBGA’ system

In Fig.3, 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 model 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 model 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 what follows, the components of the KKB_GA system will be explained in more detail.

4.2.1 Fuzzy Knowledge Based (FKB) Model for Storage and Retrieval Operations

This Fuzzy Knowledge Based Model consists of a number of stages, including the fuzzification process, fuzzy inference-fuzzy rule implementation and de-fuzzification stage. These stages will be discussed in detail.

The acceptability level of storage (α) is the output from the model, 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 which are discussed below. 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. Two types of factors are considered in this model including:

Factor 1: 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: Duration of Stay (DoS) of Containers

The second 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 [30], 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 4.2.1.1 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.

Storage Constraints: Weight, Size and Type of Containers

In addition to the above, three constraints (W, F & Y) are considered by the FKB 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 is determined by subtracting the size and type of the incoming container from 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.

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

The Fuzzification Stage

Fuzzification is the stage where fuzziness is introduced to the inputs (control variables) and the output (solution variable). Fuzzy sets and related membership functions are assigned to each variable along with linguistic definitions [40] and a triangular "shape" will be used for all the membership functions.

Firstly, the output variable (α_i) is assigned a triangular membership function with six linguistic variables. 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 Fig.4a. 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. The triangular membership function is defined, three fuzzy sets (linguistic variables) decided for the n_i are '*Low*', '*Medium*', and '*High*'. In Fig. 4b, the membership function of input (N_i) is presented.

The second input variable considered in this paper is (T_i). Fuzzy sets have triangular membership functions, there are three linguistic variables (levels) that are selected for T_i; '*Low*', '*Medium*' and '*High*' as shown in Fig. 4c.

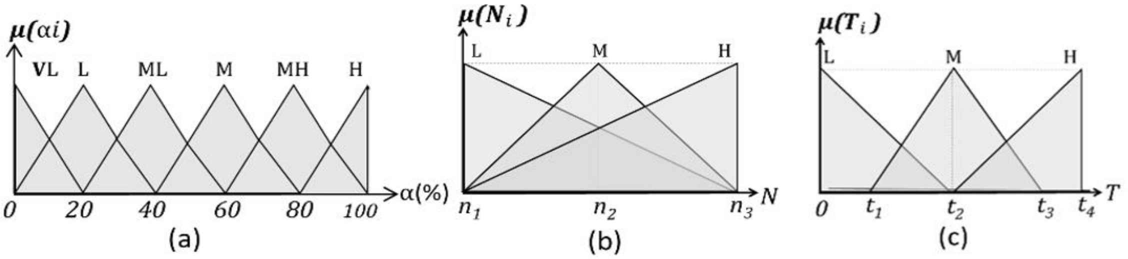


Figure 4: a) The fuzzy membership function of the output. b) The fuzzy membership function of the input factor (N). c) Fuzzy membership function of the input factor (T).

The three constraints w_i and F_i & Y_i have only one set called '*Accept*' or crisp membership functions.

The graphical representation of their membership functions are presented in: Fig. 5a for W_i, Fig. 5b for F_i and Fig. 5c for Y_i. W_i , F_i *and* Y_i have the same membership function.

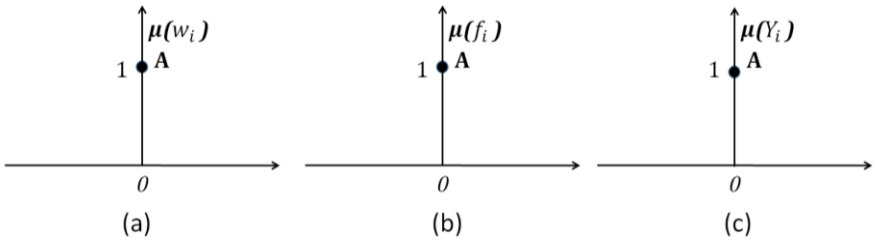


Figure 5: The defined crisp membership functions of the constraints: a) The membership function of the weight. b) The membership function of the size. c) The membership function of the type.

The Fuzzy Inference- Fuzzy Rules Determination Stage

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 [39]. For this purpose, the selected input variables (N_i, and T_i) and their interactions are analysed and the rules are determined. A total of 9 different rules are identified with respective levels for each input factor. The rules follow the 'If-Then' structure. The rules are decided based on expert opinions, which in this case, are based on the literature, observation and logic regarding the effect each input variable has on the output. In addition, the rules are proposed to reflect the location availability for the incoming container to minimise the number of re-handlings of containers during the retrieval operation. Table 2 provides all the fuzzy rules defined in this study.

Table 2: The defined fuzzy rules

Rule No.	*Ni*	*Ti*	*αi*
1	L	L	H
2	L	M	MH
3	L	H	M
4	M	L	MH
5	M	M	M
6	M	H	ML
7	H	L	M
8	H	M	L
9	H	H	VL

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 [44]. Eq. (1) is introduced for the proposed approach for container stack allocation. For each rule j, a truncated value (T_j) is calculated as follows:

$$T_j = min\left\{\mu_{(\widetilde{N})}\, n_i, \mu_{(\widetilde{T})} t_i, \mu_{(\widetilde{W})} w_i,\ \mu_{(\tilde{F}\,)} f_i,\ \mu_{(\widetilde{Y})} \mathrm{y}_i\right\} \tag{1}$$

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

The De-fuzzification Stage

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 [38, 43]. 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.

This strategy finds the centre value ($\mathrm{y_j}$) 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 $\mathrm{T_j}$ and the output $\tilde{\alpha}$ where the rule defines the outcome to be the level-p. The centre value is given by the equations (2 to 5), as shown in Fig.6 below. Upon finding the corresponding centre values for each of the rules j ($\mathrm{y_j}$) as defined, the crisp output value defined as (y^*) is computed with the centre of gravity method as shown in equation (6).

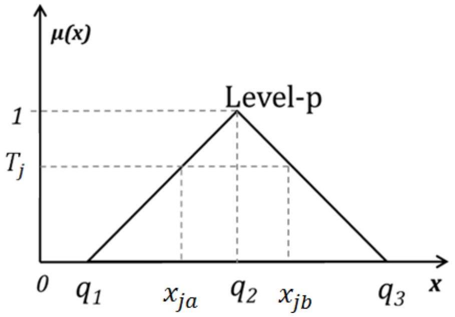


Figure 6: Truncated value on the output fuzzy set

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

$$T_j = \frac{x_{ja} - q_1}{q_2 - q_1} = \frac{q_3 - x_{jb}}{q_3 - q_2}, \qquad where; \tag{3}$$

$$x_{ja} = q_1 + T_j(q_2 - q_1) \qquad and \qquad x_{jb} = q_3 - T_j(q_3 - q_2) \tag{4}$$

$$\therefore \quad y_j = \frac{x_{ja} + x_{jb}}{2} = \frac{q_1 + q_3 + T_j(2q_2 - q_1 - q_3)}{2} \tag{5}$$

$$y^* = \frac{\sum_{j=1}^{l} y_j\, T_j}{\sum_{j=1}^{l} T_j} \tag{6}$$

Equation (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) and (4) are used to find boundary values (x_{ja}, x_{jb}) of the centre value in any rules j. Equation (5) is used to find the centre value (y_j) of any rules j, and equation (6) is used to calculate the acceptability level values of stacks (i.e. crisp outputs).

4.2.1.1 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 Fig.7 for the 'ON/OFF' strategy for the duration of stay factor.

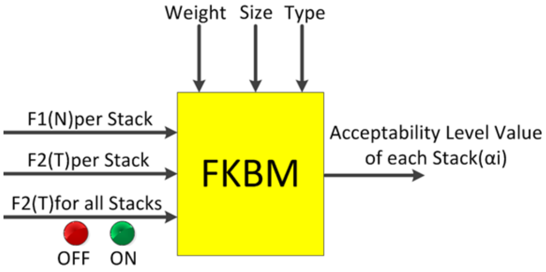


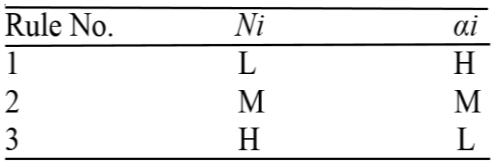
Figure 7: The 'ON/OFF' strategy of duration of stay factor

When the duration of stay factor is activated (i.e. ON) to the system as an input, all factors (N and T) are used to calculate the acceptability level values for the container storage operation. But when the duration of stay factor is temporarily deactivated (i.e. OFF), only one factor (N) is used to calculate the acceptability level values for the container storage operation (i.e. for stack allocation). A more detailed explanation including verification and validation of this strategy is provided by [1]. The contribution of this paper is to advance the capabilities of the fuzzy knowledge based model developed by [1] for the container allocation problem. This is by proposing a multi-layer genetic algorithm model that will be embedded within the developed fuzzy knowledge based model for further optimisation purposes for storage and retrieval of long stay containers. This optimisation identifies which fuzzy rules allocated per stack should be selected to achieve the best allocation plan for long duration of stay of containers rather than short stay of these containers. This contributes in turn to further improving the retrieval operation and minimising the total number of re-handlings for containers.

The defined fuzzy rules determine how the acceptability level values (i.e. the output) are affected by the combination of different linguistic variables for each input factor. Table 2 identifies 9 fuzzy rules which demonstrate the effect of each input factor on the output. Together with the other factor, when the duration of stay factor is activated (i.e. ON), all the rules are used by the fuzzy inference engine to calculate the acceptability level values for each stack, for the container storage operation (i.e. the output). De-activating the duration of stay factor is deactivated (i.e. OFF), however only utilises one factor (N) to calculate the acceptability level values for the stacks.

This reduces the number of the defined fuzzy rules to 3 and the acceptability level values are updated. This is shown in Table 3 below.

Table 3: The reduced fuzzy rules



Rule No.	*Ni*	*αi*
1	L	H
2	M	M
3	H	L

In Table 3, when the duration of stay factor is deactivated (OFF), only 3 rules will be used by the model. In this case only the number of containers factor is used to calculate the acceptability level values for the stacks in the container storage operation.

The linguistic variables for output membership function (i.e. acceptability levels) are updated based on the linguistic variables for the input factor (N) as shown in Fig. 8. In Fig. 8, the output membership function has three linguistic variables including ‘Low’, ‘Medium’, and ‘High’.

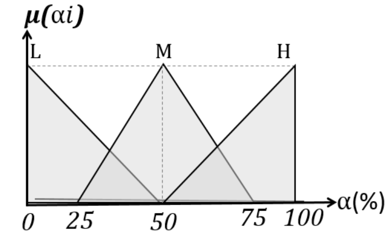


Figure 8: The Updated Linguistic Variables of the Output Membership Function

4.3. Development of the GA Optimisation Model

In this section, a GA optimisation module is proposed to be integrated with the Fuzzy Knowledge Based (FKB) model, for optimising the stack allocation of the container storage operation. A Genetic Algorithm (GA) model was developed to optimise the process of selecting stacks for container storage. The GA was selected as an optimisation search engine because of its capability to deal with a population of solutions. Genetic Algorithms (GAs) are probabilistic search methods that employ a search technique based on ideas from natural genetics and evolutionary principles (Hassanein, Aly, & Abo-Ismail, 2004). By using a FKB model together with the GA for optimisation the best set of fuzzy rules (Pawlukowicz, 2012) can be selected. The purpose for using this is now described.

4.3.1 The Purpose of Using GA

Although Fuzzy Knowledge Based models were previously used to select the fuzzy rules from the rule base [6, 9, 25, 31], a Genetic Algorithm model is proposed to tune and finally select the optimal/near optimal rules from the fired fuzzy rules, by removing the redundant rules that reduce system performance. This is because the definition of fuzzy rules and membership functions is affected by subjective decisions, and some of the fired rules would be redundant/ unnecessary, reducing the performance of the fuzzy knowledge-based system [25].

In the storage problem being investigated, a set of rules are fired for each possible stack based on its status in terms of input factors and constraints. The GA will then tune the set of fired fuzzy rules per stack, optimising them by selecting only the most effective ones in each set (i.e. those that lead to the minimum number of re-handlings for the containers). Fig.9. shows how the GA selects rules per stack. The GA 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 to keep selecting continuously the best rules per stack as the total number of re-handlings of containers is reduced. See Fig.9 for the proposed GA for rules selection per stack.

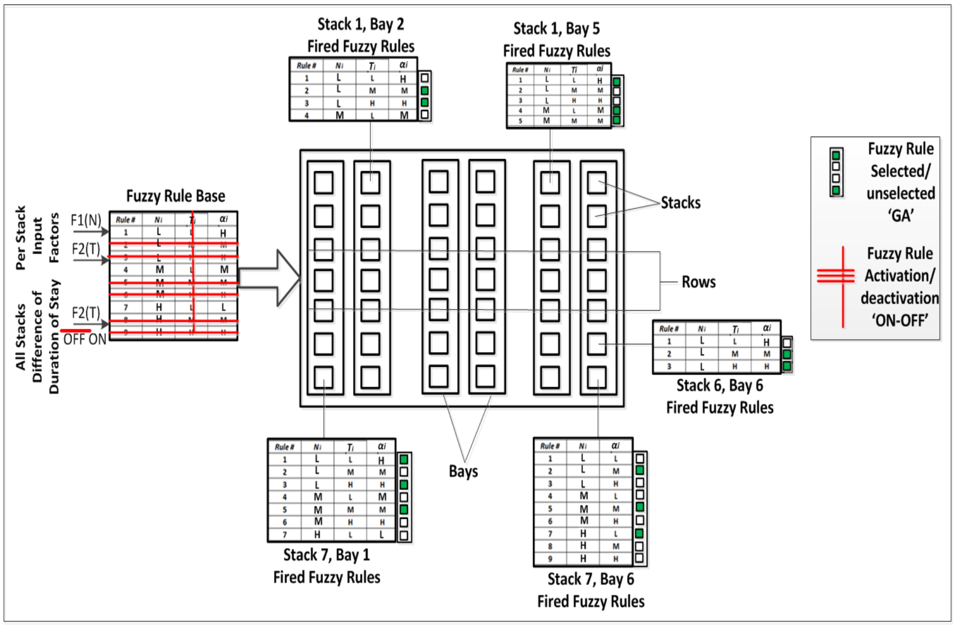


Figure 9: The proposed GA for rules selection per stack

In Fig. 9, a number of fuzzy rules from the possible stacks in the yard Fuzzy Rule Base are fired. The selection of some of the rules for each stack is then made by using the GA module. To further explain the mechanism of the 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 by the white boxes, while rules 1,4 and 5 are selected as represented by the 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 using all the 1-5 fired rules. From the process of selecting rules per stack for container storage explained above, the fired fuzzy rules for each container and each possible stack can be represented as a three-dimensional vector as shown in Fig. 10a. The fired fuzzy rules for one container for each possible stack allocation can be represented by a uni-layer chromosome as shown in Fig. 10b.

4.3.2 Multi-Layer Genetic Algorithm

In Fig. 10a, there are *m* numbers of rules fired for each possible L stack to store n containers. This can be arranged in a chromosome structure in the form of a uni-layer chromosome if only one container (C_1) is considered as shown in Fig. 10. In Fig. 10b, the fired rules for container C_1 with all possible stacks (S_1 to S_L) are stored in a one-layer chromosome. In the case of more than one container, the fired fuzzy rules of containers, together with all possible stacks (and their fuzzy rules) will be stored in a multiple-layer chromosome as explained in the next section.

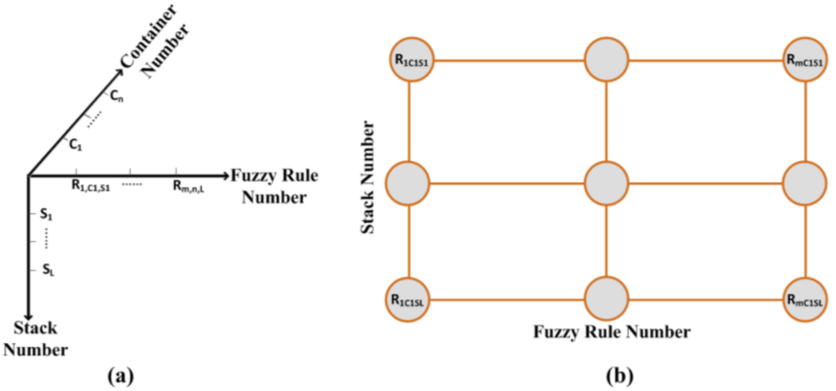


Figure 10. The fired fuzzy rules representation: The three dimension vector representation of n containers, L possible stacks and m fired rules (a) The uni-layer chromosome structure of 1 container, L stacks and m fired rules (b).

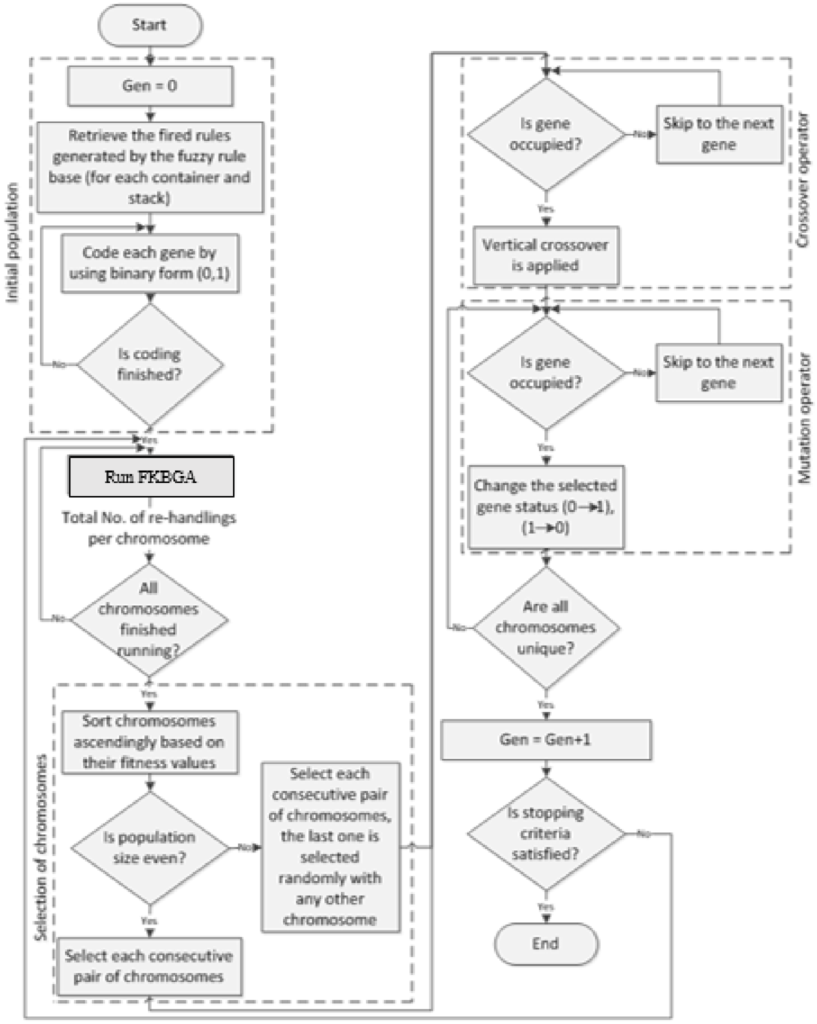

Start
Gen = 0
Retrieve the fired rules generated by the fuzzy rule base (for each container and stack)
Code each gene by using binary form (0,1)
Is coding finished?
No
Yes
Initial population
Run FKBGA
Total No. of re-handlings per chromosome
All chromosomes finished running?
No
Yes
Sort chromosomes ascendingly based on their fitness values
Is population size even?
No
Yes
Select each consecutive pair of chromosomes, the last one is selected randomly with any other chromosome
Select each consecutive pair of chromosomes
Selection of chromosomes
Is gene occupied?
No
Skip to the next gene
Yes
Vertical crossover is applied
Crossover operator
Is gene occupied?
No
Skip to the next gene
Yes
Change the selected gene status (0→1), (1→0)
Mutation operator
Are all chromosomes unique?
No
Yes
Gen = Gen+1
Is stopping criteria satisfied?
No
Yes
End

Figure 11. The flow chart of the proposed Multi-Layer GA

In the proposed Multi-Layer GA, an initial population of the selected rules out of each set of rules per stack, is randomly identified. Binary coding was applied on each chromosome layer by coding the selected rules to 1, and 0 for any other temporarily unselected rules. 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 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 are sorted into ascending order, those with the minimum number of re-handlings of containers being kept on the top of the selection list for further selection. Crossover and mutation genetic operators are then applied to create the next generation. The steps repeat until the stopping (i.e. termination) condition was satisfied.

4.3.2.1 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. In this 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 for the possible stacks for storing each container was attached with each gene. The fired fuzzy rules were placed in the length dimension (i.e. string) which was a chromosome. This chromosome included a number of genes that represented the fired fuzzy rules for a container for all 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 multi-layer chromosome structure was proposed to provide more flexibility to deal with such sets of information in order to select fuzzy rule(s) from the fired rules for each container and possible stacks. Fig. 12 shows the proposed Multi-Layer chromosome structure.

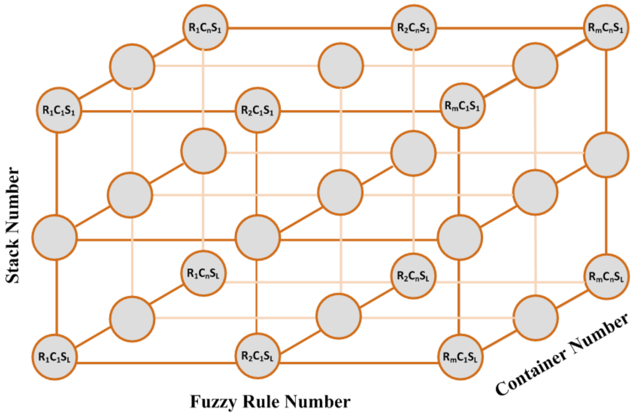


Figure 12. The Multi-Layer chromosome structure for fuzzy rule representation of n containers

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 stacks, a number of fuzzy rules were fired to store containers. Based on the fired fuzzy rules per container and possible stacks and the related degrees of membership of the input factors, the acceptability levels for the possible stacks were calculated to store the containers. For each container and possible stacks, a number of fuzzy fired rules were stored in the generations of a chromosome. The front (i.e. first) layer of the chromosome represented the first container with its fired fuzzy rules and all the possible stack(s). The second layer of the chromosome represented the second container with its fired fuzzy rules and possible stacks. The number of layers depended on the total number of containers. Each gene of each layer was used to select or not to select rules from the fired fuzzy rules using binary coding. All fired fuzzy rules per container and possible stacks were then stored in multiple layers.

4.3.2.2 The Objective function

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

$$\min \sum_{i=1}^{n} \gamma_i \tag{7}$$

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

4.3.2.3 Initial Population of Selected Fuzzy Rules

As a starting point, an initial set of selected rules are required to provide a feasible starting basic solution. After the set of fired fuzzy rules for each container together with the possible stacks were stored, binary coding was applied randomly to select some 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 allocated to store each container. The binary coding process avoided generating 0s for all the genes at each layer of the chromosomes.

4.3.2.4 The Selection Method

After the chromosomes were sorted ascendingly based on their fitness values (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 (offspring) using GA operators. This is in case the population size was even. In the case where it was odd, each pair of chromosomes with minimum fitness values were selected for further generations. The last chromosome was coupled with any randomly selected chromosome from the population for further offspring generation. The GA operators are explained in detail below.

4.3.2.5 Multi-Layer Genetic Algorithm Operators

Crossover Operator: The crossover operator for the Genetic Algorithms was based on the exchange of genes between two chromosomes when they were selected. To crossover genes in the chromosome, the genes of each chromosome were coded in binary representation (0, 1) in the Multi-Layer chromosome. This type of representation meant that the genes that were set to 1 were selected genes. The genes that were set to 0 were unselected genes. Each gene in a chromosome represented the fired fuzzy rule number per specific container and possible stacks. Based on the selected genes (i.e. rules), the acceptability level values for stacks to store a container was calculated.

With the crossover operator, the selection of genes to be exchanged depended on the probability of crossover (i.e. a specific percentage). The probability of crossing over genes determined 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 was not fired from the fuzzy rule base), then the crossing over process skips to the next gene. A vertical crossover type was used to swap 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 [33]. This crossover operator was used to present the best random exchanging of genes between each pair of chromosomes. See Fig. 13 for an illustration of the crossing-over of two selected chromosomes.

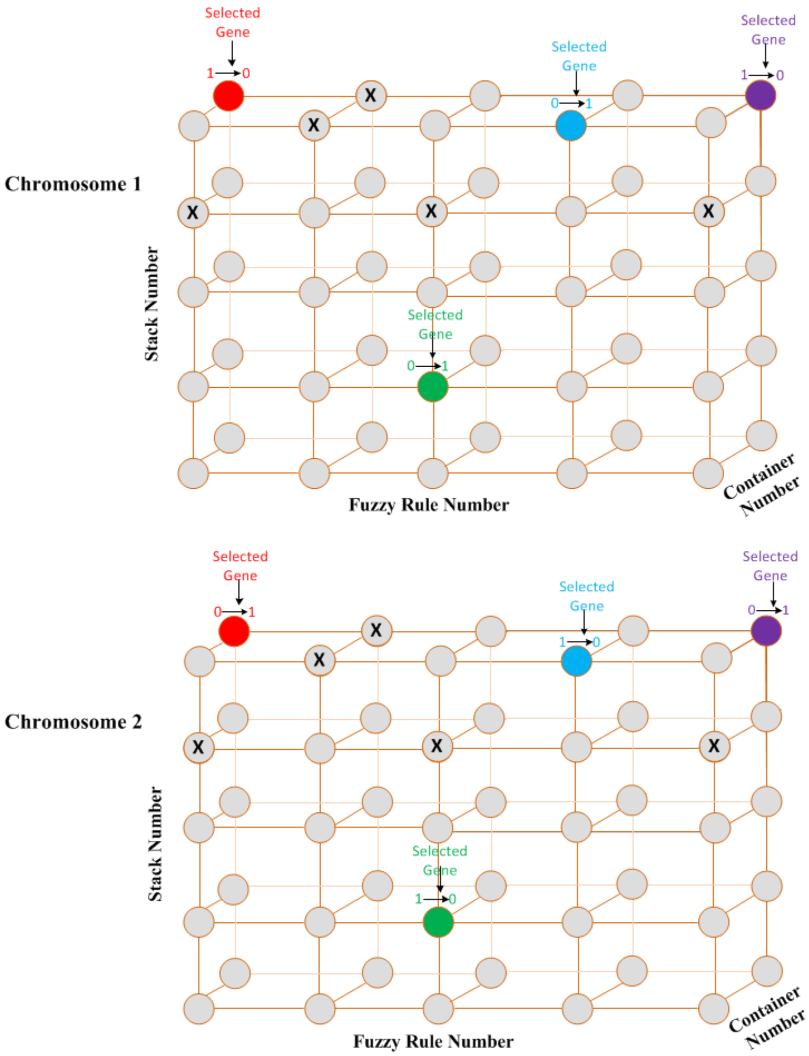


Figure 13: The Crossing-over of genes in Multi-Layer chromosome

The probability of crossover value decided the number of genes to be exchanged at each chromosome. The crossover was skipped when genes contained no fired rule. This type of crossover operator provided 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.

Mutation Operator: A mutation operation was applied on new chromosomes that were 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 was selected randomly. The proposed 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 wastes time and leads to long computations. See Fig. 14 for the mutation operator.

The crossed genes in Fig. 14 are empty genes that do not include fired fuzzy rules. This operator excluded any crossed genes from the mutation operation and considered 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

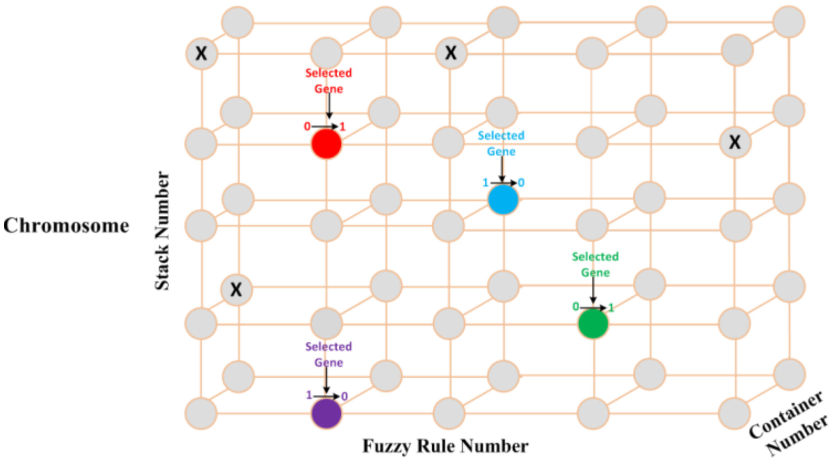


Figure 14: The Mutation of genes in Multi-layer chromosome

5. Case Study, Experiments and Results Analysis

This case study was used to justify the proposed yard management system. It was conducted in collaboration with Maritime Transport- the UK's leading Container Transport Company. Maritime is one of the UK's leading multimodal transport and container service specialists, combining road, rail & storage to become an integral element of the supply chain for its customers. This company provides highly effective UK container transport and services. Most of the system inputs were collected from the Maritime Company. This included the container yard dimension in terms of number of rows, bays and tiers. The container yard included a number of pre-existing containers. For each train, the inter-arrival time, number of containers, container attributes (e.g. for each container: the size, type, weight, destination customer, owner company and truck id), and the number of companies were also captured. The number of trucks available at each company was assumed to be between 20 and 30 trucks. In order to test the behaviour of the developed 'FKBGA' system, three real life scenarios were considered including a Busy yard with a significant number of pre-existing containers, rush and slow arrival rate of container trains. In these scenarios, the proposed Fuzzy Knowledge Based on Genetic Algorithm (FKBGA) system was used to calculate the total number of re-handlings along with total retrieval time of containers. The impact of embedding the GA in the FKB model was also identified. The 'ON/OFF' approach was tested and the promised contribution supported thoroughly by the tests performed by Abbas et al. [1]. The system was coded using Visual Basic for Applications (VBA) in MS Office Excel.

5.1. Busy Yard Scenario

This scenario uses the factor that assumes 80%-90% of the yard was occupied with pre-existing containers. In order to guarantee the best search for solutions in such a busy scenario, the GA parameters were tuned after a number of experiments. Three population sizes are tested against different crossover and mutation probabilities. Each population size consists of a predefined number of chromosomes and each chromosome covers all the yard stacks in terms of their fuzzy rules. As far as a large yard size of 225 stacks is considered in this case study, a maximum of 15 chromosomes is decided as a population size. However, 50 generations are run to explore more promising solutions under these restricted population sizes. The optimal settings of these parameters were population size (i.e. number of chromosomes) equal to 15, the probability of crossing-over genes was 0.90, and the mutating rate of genes was 0.10. The stopping condition was satisfied when the number of generations reached 50. The minimum number of re-handlings achieved was 1353. Fig. 15 demonstrates the current adopted approach by the company 'Current', 'FKB' and FKBGA results of the 'busy yard' scenario.

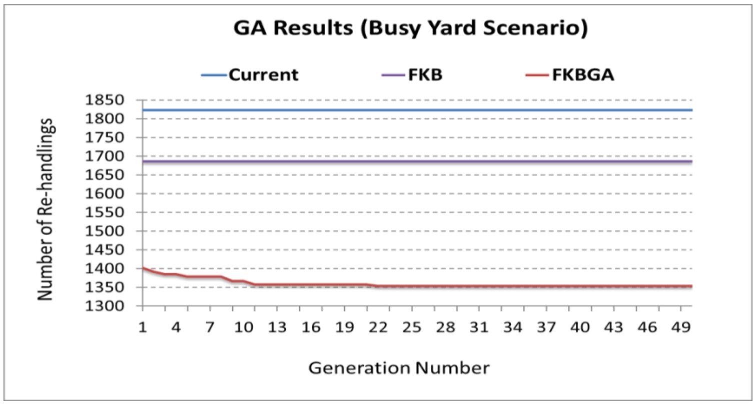


Figure 15: The busy yard scenario experiment results

In Fig. 15, the ‘Current’ approach used by the company resulted in 1822 re-handlings to deliver all its containers, which was higher than the number required by both the ‘FKB’ and ‘FKBGA’. The ‘Current’ approach stored the containers in groups (i.e. containers were grouped by customer) taking into consideration the three storage constraints. The ‘FKB’ approach achieved a significant reduction in re-handlings (i.e. from 1822 to 1686 when compared to the ‘Current’ approach. The reduction in re-handlings between the ‘FKB’ and ‘FKBGA’ (i.e 1686 to 1353) can be explained by the embedding of the GA because with the ‘FKB’ approach all the fired fuzzy rules including the unnecessary ones were utilised, rather than using only the most influential ones with the ‘FKBGA’ approach that led to the minimum number of re-handlings. For the ‘FKBGA’ an early reduction of the number of re-handlings (i.e. 1402) was obtained from the initial population. This is because the initial population randomly selected promising rules from the fired fuzzy rules. Further reductions of the re-handlings were obtained later at generation numbers 2 and 11 because the best set of GA parameters led to the selection of more effective rules from the previous rules obtained. This led to the investigating of more promising solutions to achieve the required randomness in the search process. A slight reduction in the number of re-handlings was obtained at the 3rd and 5th generations. It can be seen that after the 21st generation, the minimum number of re-handlings was obtained (i.e. 1353 re-handlings). Although repeated chromosomes are not allowed as discussed in section 4.2.2, the total number of re-handlings has not been further improved after a number of generations (22 in this scenario). This is due to the reason that a binary coding mechanism of genes was applied where each gene represents a rule and hence, selection of good genes might be affected by some other activated weak ones, and hence, the resultant outcome of number of re-handlings of all containers might be similar.

The stacks allocated by the system were the best stacks for the container storage operation which yielded the minimum number of re-handlings of containers after the retrieval operation was complete. By running the ‘FKBGA’, the stacks were allocated optimally to store containers which resulted in a minimum number of re-handlings as shown in Fig 16. This also led to reducing the total retrieval times of containers, see Fig. 16.

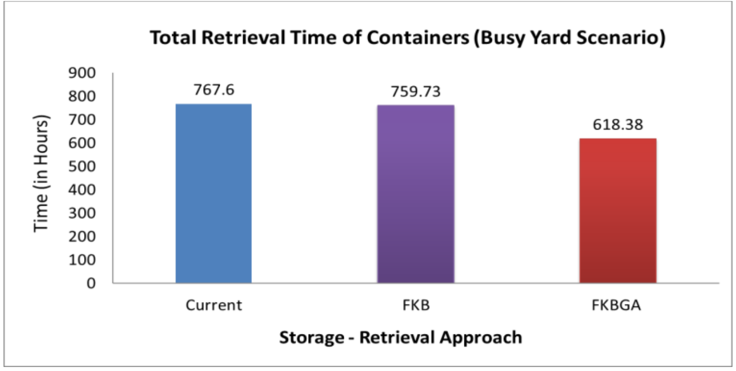


Figure 16: The total retrieval time of containers

In Fig. 16, when the ‘FKBGA’ approach is compared with the ‘FKB’ approach, the total retrieval time of containers was reduced by 18.6%. The number of re-handlings obtained by using the ‘FKB_ GA’ was the lowest when compared with the other 2 approaches, and hence, this led to the reduction of the total retrieval time. By comparing the ‘FKB_ GA’ approach with the ‘Current’ approach, the total retrieval time of containers was decreased by 19.4%. The total number of re-handlings using the ‘Current’ approach was the highest when compared with the other approaches, that is why the reduction of the total retrieval time was high when the ‘FKBGA’ was applied.

5.2. Rush arrival rate 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. This scenario was tested under busy yard condition. Fig.17 demonstrates the GA results of the rush arrival rate of container trains scenario.

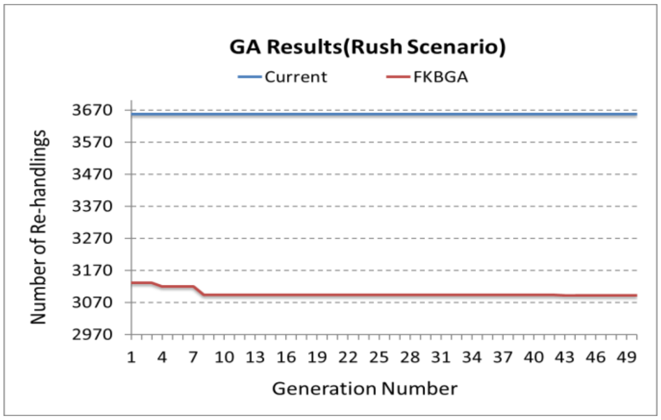


Figure 17: The rush scenario results

In Fig. 17, the ‘FKBGA’ system obtained the lowest number of re-handlings of containers (3092 re-handlings) under busy yard scenario when compared with the ‘Current’ approach. In the rush arrival rate of trains scenario, the ‘FKBGA’ 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).

After the retrieval operation was complete, the total retrieval time of containers in hours was obtained as shown in Fig.18.

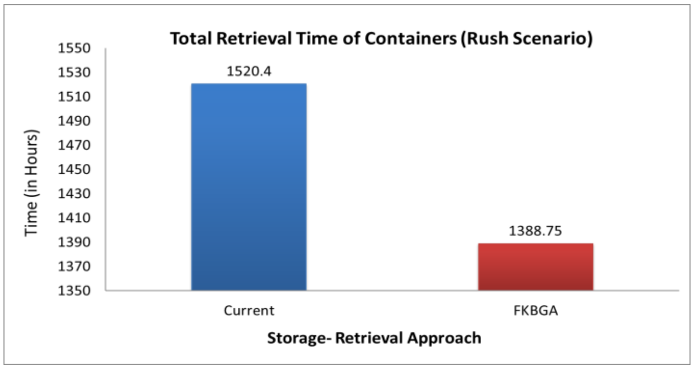


Figure 18: The total retrieval time of containers (rush scenario)

In Fig.18, the lowest retrieval time for containers were obtained by running the ‘FKBGA’ system (i.e. 1388.75 hours) when compared with the ‘Current’ approach. As discussed in Fig.17, the total number of re-handlings was the lowest by running the ‘FKBGA’ approach which led to the lowest total retrieval time of containers as well.

5.3. Slow Arrival rate 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. The busy yard condition was used to test the slow arrival rate of container trains scenario. Fig.19 shows the GA results of this scenario.

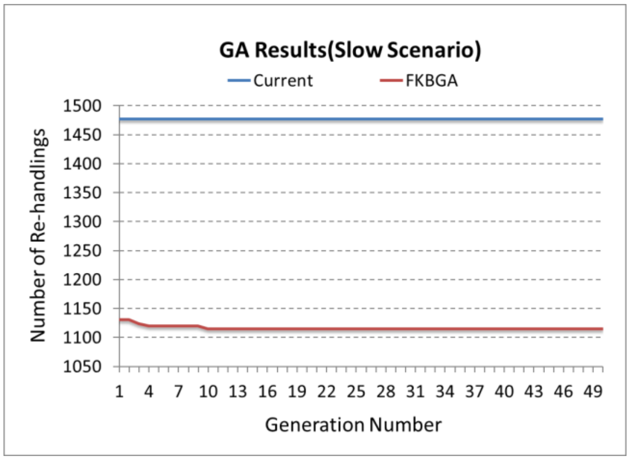


Figure 19: The slow scenario results

The number of re-handlings for the slow arrival rate of trains scenario in the initial population was 1131 re-handlings by running the ‘FKBGA’ system as shown in Fig.19. After applying the GA operators, a reduction were obtained in the number of re-handlings were 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 by the ‘FKBGA’ system in this scenario was 1115 re-handlings when compared with the ‘Current’ approach. 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. See Fig. 20 for the total retrieval time of containers in the slow arrival rate of container trains scenario.

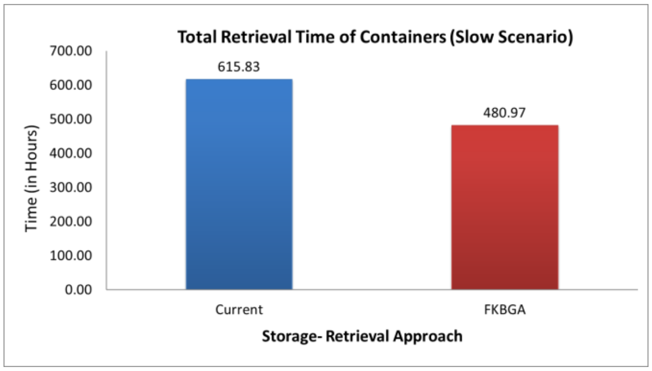


Figure 20: The total retrieval time of containers (slow scenario)

In Fig. 20, the ‘Current’ approach resulted in a higher total retrieval time of containers than the ‘FKBGA’ system. The higher number of re-handlings was achieved by applying the ‘Current’ approach as discussed in Fig.19 which led to the highest total retrieval time of containers as well.

6. Comparison with Other Approaches

In addition to the previous comparison made in section 5, other popular stack allocation approaches were selected to support the superiority of the proposed 'FKBGA' approach. The comparison was conducted under 'busy yard' scenario. This scenario was then investigated using different storage-retrieval approaches including: the Fuzzy Knowledge-Based GA approach (FKBGA), the Constrained-Probabilistic Stack Allocation (CPSA) approach, and the Constrained-Neighbourhood Stack Allocation (CNSA) approach. The proposed 'FKBGA' can also be applied in the retrieval operation by taking into consideration the aforementioned storage factors and constraints while searching for a proper stack for container storage. By using the 'CPSA' approach a container can be allocated to any possible stack given the satisfaction of the aforementioned storage constraints. The 'CNSA' approach was used here only in the retrieval operation. This approach searched for the closest stack possible to the original stack that complies with the constraints of the container (Ji et al. 2015). In the 'CNSA' approach; places of storage (i.e. stacks) have the same chances/probabilities of selection for a container providing the storage constraints are satisfied. In the retrieval operation, a container can be moved and stored at any possible storage place/stack providing the storage constraints are satisfied. Fig. 21 shows the comparison of the total number of re-handlings obtained under the busy yard scenario.

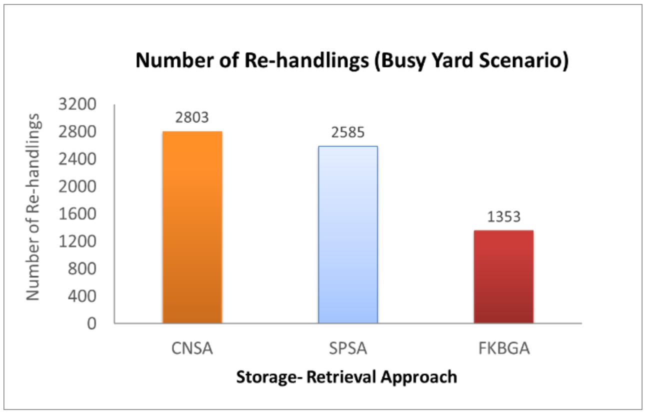


Figure 21: Comparison between total numbers of re-handlings obtained by using different approaches

In Fig. 21, it can be seen that the 'FKBGA' achieved a considerable reduction in the number of re-handlings by allocating the best stacks for the container storage operation. This was due to the fact that the GA played a vital role in selecting the optimal/near optimal fuzzy rules that contributed to the best stack allocation decision. The 'CNSA' approach also led to a 7% higher number of re-handlings (2803) when compared with the 'CPSA' approach.

7. Conclusion and future work

A new approach for solving the problem for the management of the container yard was presented. A fuzzy knowledge based optimisation system for solving stack allocation problems for containers that are allowed to stay in the yard for long time with an unknown departure time was developed. A detailed conceptual model was presented to handle the complexity of the problem. The developed system dealt successfully with other influential factors such as the duration of stay, together with other real-life constraints in order to optimise storage-retrieval operations of long stay containers in the yard.

The combination of the FKB approach with GA achieved an optimal/near optimal stack allocation for long stay container storage operation in the yard. The concept of using GA for solving this type of storage problems provided flexibility when dealing with large sets of information as well as the capability to select the most promising fuzzy rules out of a set of fired fuzzy rules per container and possible stack. The best stack was allocated for an incoming container that stayed long in the yard based on the selected fuzzy rules. In addition, a multi-layer chromosome design that can be used to deal with a large set of information was proposed along with other modified GA operators. The proposed GA chromosome structure enabled the use of a more organised set of information than the traditional multi-layer GA chromosomes structure which could be seen as a contribution to the industry. For the selected busy yard scenario, the proposed 'FKBGA' resulted in a 25.7% reduction in the total number of re-handlings when compared to the 'Current' approach used by the company. In the rush arrival rate of container trains, the 'FKBGA' reduced the total number of re-handlings for containers was reduced by 15.4% when compared with the 'Current' approach. When the slow arrival rate of trains scenario was considered, the total number of containers was reduced by 24.5%. In general, it can be concluded that the Fuzzy Knowledge Based on Genetic Algorithm ('FKBGA') system shows the most beneficial improvements in the slow arrival rate of trains scenario. The proposed 'FKBGA' outperformed the Constrained Probabilistic Stack Allocation 'CPSA' and Constrained Probabilistic Stack Allocation 'CNSA' achieving the minimum number of re-handlings.

As a further development of this research work, 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 is a stack can be considered as one of the influential factors that affect the process for the allocation of containers to stacks, and hence more features could be added to the current FKBGA system to handle other factors affecting the allocation of containers. The environmental impact of CO2 produced when using reach stackers to allocate/retrieve containers in/from the storage yard when allocating incoming containers to each tier of stacks, could also be taken into consideration.

Conflict of interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Abbas, A., Al-Bazi, A., & Palade, V. (2018). A Constrained Fuzzy Knowledge-Based System for the Management of Container Yard Operations. *International Journal of Fuzzy Systems*, 20(4), 1205-1223.
2. Abdulhalim, M. F., & 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.
3. Ayachi, I., Kammarti, R., Ksouri, M., & Borne, P. (2010). A Genetic algorithm to solve the container storage space allocation problem. *International conference on Computational Intelligence and Vehicular System* (CIVS), 1–4.
4. Ayachi, I., Kammarti, R., Ksouri, M., & Borne, P. (2013). A heuristic for re-marshalling unbound and outbound containers: 2nd IEEE International Conference. *Systems and Computer Science* (ICSCS), 291–296.
5. Ayachi, I., Kammarti, R., Ksouri, M., & Borne, P. (2013). Harmony search to solve the container storage problem with different container types. *International Journal of Computer Applications*, 48(22), 26–32.
6. Carmona, P., Castro, J., & 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.
7. Casey, B., & Kozan, E. (2012). Optimising container storage processes at multimodal terminals. *Journal of the Operational Research Society*, 63(8), 1126–1142.
8. Chen, T. (1999). Yard operations in the container terminal-a study in the 'unproductive moves'. Maritime Policy & Management, 26(1), 27–38.
9. Cintra, M., & de Arruda, C. (2007). Fuzzy rules generation using genetic algorithms with self-adaptive selection: In *IEEE International Conference on Information Reuse and Integration*, 261–266.
10. de Castillo, B., & Daganzo, C. F. (1993). Handling strategies for import containers at marine terminals. *Transportation Research Part B: Methodological*, 27(2), 151–166.
11. Hassanein, O. I., Aly, A. A., & Abo-Ismail, A. (2004). Genetic - PID control for a fire tube boiler: In *Proceeding of Second IEEE International Conference on Computational Cybernetics*, 19–24

12. 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.
13. 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.
14. Jin, C., Liu, X., & Gao, P. (2004). An Intelligent Simulation Method Based on Artificial Neural Network for Container Yard Operation. *Advances in Neural Networks*, 904–911.
15. Junqueira, C., de Azevedo, A. T., & Ohishi, T. (2016). Stowage planning and storage space assignment of containers in port yards: Ninth International Conference on Coastal and Port Engineering in Developing Countries, 1-16.
16. Kacprzyk, J., & Pedrycz, W. (2015). *Springer handbook of computational intelligence*, Springer, Berlin.
17. Kelareva, G., & Negnevitsky, M. (2001). Multi-Layered Genetic Algorithm for Maintenance Scheduling with Multiple Parameters. *Australian Journal of Intelligent Information Processing Systems*, 7(3/4), 122–131.
18. Kim, K. H. (1997). Evaluation of the Number of Rehandles in Container Yards. *Computers & Industrial Engineering*, 32(4), 701–711.
19. Kim, K. H., & 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.
20. Kim, K. H., & 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.
21. Ku, D., & Arthanari, T. S. (2016). Container relocation problem with time windows for container departure. *European Journal of Operational Research*, 252(3), 1031–1039.
22. Lawrence, W., & Chwan-Kai, K. (2001). A comparison of stacking efficiency for various strategies of slot assignment in container yards. *Journal of the Eastern Asia Society for Transportation Studies*, 4(1), 303-318.
23. Liu, Y., Kang, H., & Zhou, P. (2010). Fuzzy optimization of storage space allocation in a container terminal. *Journal of Shanghai Jiaotong University (Science)*, 15(6), 730–735.
24. Negnevitsky, M., & Kelareva, G. (2008). Development of a multi-layer genetic algorithm for maintenance scheduling in power systems: In *IEEE/PES Transmission and Distribution Conference and Exposition*, 1-5.
25. Nelles, O., Fischer, M., & 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*, 1, 213–219.
26. Ozcan, S., & Eliiyi, D. T. (2017). A reward-based algorithm for the stacking of outbound containers. *Transportation Research Procedia*, 22, 213–221.
27. Park, T., Choe, R., Hun Kim, Y., & Ryel Ryu, K. (2011). Dynamic adjustment of container stacking policy in an automated container terminal. *International Journal of Production Economics*, 133(1), 385–392.
28. Pawlukowicz, P. (2012). Automatic learning of fuzzy logic with the use of genetic algorithms. *Advances in Manufacturing Science and Technology*, 34(4), 98–108.
29. Ries, J., González-Ramírez, R. G., & Miranda, P. (2014). A fuzzy logic model for the container stacking problem at container terminals. *International Conference on Computational Logistics*, 93–111.
30. Saurí, S., & 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.
31. Shill, P. C., Akhand, M. A. H., Asaduzzaman, M., & 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.
32. Tang, L., Jiang, W., Liu, J., & Dong, Y. (2015). Research into container reshuffling and stacking problems in container terminal yards. *IIE Transactions*, 47(7), 751–766.
33. Tang, X., Liu, Y., Zhang, J., & Kainz, W. (2007). *Advances in Spatio-Temporal Analysis*. Taylor & Francis, Inc, Bristol.
34. Ting, S.C., Wang, J.S., Kao, S.L., & Pitty, F.M. (2010). Categorized stacking models for import containers in port container terminals. *Maritime Economics & Logistics*, 12(2), 162–177.
35. Tranberg, L. K. (2005). Optimizing yard operations in port container terminals: In *Proceedings of the 10th EWGT Meeting*, 386–391.
36. Woo, Y. J., & 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.

37. Yang, X., Zhao, N., Bian, Z., Chai, J., & Mi, C. (2015). An intelligent storage determining method for inbound containers in container terminals. *Journal of Coastal Research*, 73(sp1), 197–204.
38. Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, 8(3), 338–353.
39. Zadeh, L, A. (1979). *A theory of approximate reasoning, Machine Intelligence*. Halstead Press, New York.
40. Zadeh, L, A. (1975). The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences*, 8(3), 199–249.
41. Zehendner, E., Feillet, D., & Jaillet, P. (2017). An algorithm with performance guarantee for the Online Container Relocation Problem. *European Journal of Operational Research*, 259(1), 48–62.
42. Zhang, C., Liu, J., Wan, Y., Murty, K. G., & Linn, R. J. (2003). Storage space allocation in container terminals. *Transportation Research Part B: Methodological*, 37(10), 883–903.
43. Zimmermann, H, J. (1991). *Fuzzy set theory and its applications*. Kluwer-Nijhoff, Boston.
44. Zimmermann, H, J. (2001). *Fuzzy set theory and its applications*. Springer Science & Business Media, New York.