A New Fuzzy Knowledge-Based Optimisation System for Management of Container Yard Operations

Al Bazi, A., Palade, V. & Abbas, A. H. H. J.

Original citation & hyperlink:
Al Bazi, A, Palade, V & Abbas, AHHJ 2021, 'A New Fuzzy Knowledge-Based Optimisation System for Management of Container Yard Operations', International Journal on Artificial Intelligence Tools, vol. 30, no. 2, 2150003.
https://dx.doi.org/10.1142/S0218213021500032

DOI 10.1142/S0218213021500032
ISSN 0218-2130
ESSN 1793-6349

Publisher: World Scientific Publishing

International Journal on Artificial Intelligence Tools
Vol. XX, No. X (2020) 1–41

A New Fuzzy Knowledge-Based Optimisation System for Management of Container Yard Operations

Ammar Al-Bazi*
Faculty of Engineering, Environment and Computing, Coventry University
Coventry, West Midlands, CV1 5FB, UK
aa8535@coventry.ac.uk

Vasile Palade
Faculty of Engineering, Environment and Computing, Coventry University
Coventry, West Midlands, CV1 5FB, UK
ab5839@coventry.ac.uk

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

Managing the container yard operations can be challenging as a result of various uncertainties associated with storing and retrieving containers from the yard. These associated uncertainties occur because the arrival of a truck to pick up the container is random, so the departure time of the container is unknown. The problem investigated in this paper emerges when newly arrived containers of different sizes, types and weights require storage operation in the same yard where other containers have already been stored. This situation becomes more challenging when the time of departure of existing container is not known. This study develops a new Fuzzy Knowledge-Based optimisation system named 'FKB_GA' for optimal storage and retrieval of containers in a yard that contains long stay pre-existing containers. The containers' duration of stay factor is considered along with two other factors such as the similarity (containers with same customer) and the quantity of containers per stack. A new Multi-Layered Genetic Algorithm module is proposed 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.

Keywords: Fuzzy Knowledge-Based Model; Multi-Layer Genetic Algorithm; Fuzzy Rules; 'ON/OFF' Strategy; Container Yard Operations.

1. Introduction

As a result of globalisation and economic growth, the need for container transportation has become very significant, which consequently leads to competition between container terminals. Thus, efficient handling operations in container terminals are becoming increasingly important [1]. As most of the terminal operations are concerned with the containers storage and movement in or out of the yard, the efficiency of these operations is a very important issue [2].

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 a container is random, so the departure time of the container is unknown. Storing an import container on the top of another that 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 [3].

Several techniques have been developed for import container storage operation in yards with an unknown departure time for short durations of stay such as the segregation and non-segregation strategies discussed by [4], and a fuzzy logic-based rule model by [5]. However, there is still a lack of advanced optimisation systems for the storage of import containers, given their unpredictable departure behaviour. These systems will assist the planners of terminal operations to achieve the most efficient allocation of containers which will eventually contribute to minimise the total amount of re-handlings, reducing the re-handling times and consequently improving the management productivity of the overall yard operations.

In this study, a new fuzzy knowledge-based optimisation system is proposed to manage the operations of import container yard especially when containers stay for long duration, which may 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 comprise the amount of containers in each stack, as well as the containers similarity i.e. that has the same customer and the time length of the container at the stack peak. The proposed system is associated with number of constraints such as the type and size of containers as well as weather they are empty or full. In addition, the Fuzzy Knowledge-Based optimisation system has been developed to optimise the fuzzy rules that contribute to the retrieval and storage processes of import containers.

The novelty of this work is the development of an innovative multi-layer Genetic Algorithms module embedded in the fuzzy knowledge-based system proposed by [6]. The new system based multi-layered GA is used to optimise large number of dynamic sets of rules allocated to container stacks to minimise container retrieval. A comparison study is made with two relevant storage techniques to evaluate the performance of the proposed stack allocation system using a genetic algorithm (GA).

Although the container yard operations under constraints has been studied in [6] using a fuzzy knowledge-based system to manage the problem, the fuzzy rules used in the developed system were subjectively chosen. Some of these rules, especially those allocated for each container stack, might be unnecessary or redundant, and hence it is imperative optimise the fuzzy rules for higher accuracy decisions. This work proposes an optimisation module based multi-layer GA to refine and improve these fuzzy rules for better decision-making for storing containers.

The remainder of this paper is structured thus: in section 2 previous studies are presented, and section 3 defines the challenging of storing and retrieving container within the yard. The approach adopted in this paper is discussed in section 4. The description of the experiments and analysis of results is presented in Section 5. 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 containers storage/stacking problems with an unknown time of departure are discussed. A fuzzy logic-based model was developed in [5] to solve the problem of allocation of storage space for seaport terminal containers. The containers that depart randomly had a short duration of stay in the yard. The study in [7] compared two techniques of stacking for the static assignment of the correct slot for 150 containers with a short duration of stay in the yard. In [4], a mathematical function based on a uniform distribution was used to examine non-segregation and segregation strategies for the static container storage problem when the departure time of containers was random, and with a duration of stay based on the number of ship arrivals. In [8], difference equations were applied to extend the segregation strategy proposed by [4]. These equations took into account constant, cyclic, and dynamic arrival patterns for containers with a duration of stay of only 3 to 6 days, to achieve efficient storage operation. Also, a mathematical equation was formulated by [9] to tackle the problem of stacking containers. The main variable of the formulation was the number of containers that were removed with a short duration of stay. The study of [10] introduced two stacking methods to deal with the problem of container storage operation. The methods were used to decide how incoming containers are stacked up with the existing ones. The introduction of these methods was for impact assessment of short container duration of stay on storage procedures. A fuzzy knowledge-based model was developed in [6] for containers storage and retrieval in a yard. The allocation decisions of containers were made based on subjectively chosen fuzzy rules. However, the factor of long duration of container stay in the yard was considered in this study. A rolling horizon approach was discussed in

[11] for improving the allocation process of containers storage space with unknown pick up times. The total number of planning periods in a planning horizon was only 18 hours. In the study of [12], an intelligent neural network was modelled which focused on fuzzy logic for container yard operations scheduling. The study provided improvement of the containers' storage as well as the scheduling operation for container durations of stay of a limited number of days. An algorithm was introduced in [13] with the aim of ensuring better performance to solve the problem relating to the Online Container Relocation (OCR) with periodic and sequential containers retrieval time. The algorithm aimed to reduce the number of relocations of containers with a short duration of stay in the yard.

Optimisation techniques are also utilised to solve the same storage/stacking problems. In this work of [14], a fuzzy-based optimisation model was developed to allow better allocation of container storage space. One of the objectives of the study was 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. Also, the work of [15] aimed to achieve optimal storage for containers with 3 to 4 days duration of stay by proposing a mathematical model. The model which was based on probability distribution functions was used to reduce redundant movements of containers within the yard. A Multi-Objective Integer Programming (MOIP) model was introduced in [16]. The model helped to solve the problem relating to Stacking Position Determination (SPD) of container in a situation where there are fewer days and time of stay. The study of [17] aimed to minimise the reservation numbers, clusters, of each export container group by introducing a mathematical model. This has to do with when storage space is been allocated for containers for time of stay was less than a week. In [18], a Genetic Algorithm model was proposed to optimise positions of containers of different types with random delivery dates. The model was used to determine an optimal storage for containers operation with a short duration of stay to minimise re-handlings and improve the possibility of achieving customer delivery target. Another Genetic Algorithm model was developed by [19] aimed for storage-space allocation optimisation for export and import containers. The model helped to reduce the re-handling operations and efficiently organise containers with a short duration of stay in the available storage space. The storage space assignment problem for containers was the focus in [20], where a simulation-based genetic algorithm was proposed to optimise storage rules for containers with a short duration of stay in the yard. The algorithm was purposely designed to minimise unnecessary container movement. In [21], a stochastic dynamic programming model was proposed 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. Also, in [22] a reward-based algorithm was proposed for solving the outbound container stacking problem for containers with duration of stay of a few days. The algorithm aimed at reducing the amount of containers re-handlings.

Even though the reviewed literature revealed several allocation techniques and optimisation approaches for solving the containers storage/stacking problems with an unknown time of departure, the focus of all these techniques was made on containers with a short duration of stay. None of them has considered key factors related to long stay durations such the container duration of stay factor and the use of fuzzy techniques based optimisation to predict the likely departure and to assess the effect of other factors on the storage and retrieval plans. Although [6] developed stack allocation decisions based on subjectively chosen fuzzy rules that took into consideration the factor of long duration of container stay. However, this paper presents an improved approach from previous work [6] for optimal allocation techniques for unknown departure times of containers based on optimisation of fuzzy rule-based systems. The optimisation of each set of rules allocated per stack of containers has not been investigated yet and hence, the focus of this work is established. In general, two major contributions added to the above literature: the consideration of long duration of stay of containers and the optimisation of the fuzzy rules used in the fuzzy rule-based systems.

Hence, this study presents a new Fuzzy Knowledge-Based optimisation system named 'FKB_GA', 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 factor.

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 may stay for unpredictable times before departing the yard. If this happens the topmost containers may require moving to access the container to be despatched. In a situation where the time of departure of containers are not known and these containers are allowed to stay for long stay time (maximum one month period) before a notification is sent to customers. This kind of event can occur in a situation where customers arrange their container collection by 3PL companies without notifying the yard operators before-hand. Hereby the delivery company proceeds to the terminals to collect containers with no further, hereby creating problems for storage and retrieval operations of containers in the yard. However, most yard operators are happy for containers to stay for a long time given that customers pay a pre-defined daily storage fee.

For the number of containers that spend longer time in the yard, the period of their stay is considered vital and relatively associated to their time of departure. This implies that the more time a container spend in the yard is directly proportional to their chance of being retrieved and collected.

However, an unknown departure time makes it difficult to determine how to stack and store containers in the yard by just taking into account their duration of stay. It becomes even more problematic when the containers at the peak of the stack have been stored for roughly the same time. For this reason, additional further factors such as the amount of containers in a stack and containers similarities (i.e. container with the similar customer) need to be considered for storage operation optimisation and then minimise the amount of re-handlings, as depicted in Fig. 1 below.

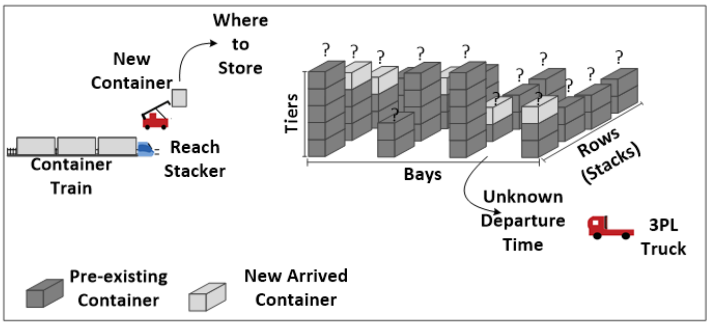

Fig. 1 Schematic representation for the layout of a pre-existing container yard.

In order to model the fuzzy aspect associated with each of the three storage factors both individually and combined, a novel Fuzzy Knowledge-Based optimisation system is proposed. Then, the optimisation process identifies which fuzzy rules allocated per stack should be selected and then refined by keeping the most influential ones to achieve the best allocation plan for containers. This improves the retrieval operation and minimising the total number of re-handlings for containers.

The connected approaches adopted for the container yard management system modelling in this paper are explained in the next section.

4. Development of 'FKB_GA' – The Fuzzy Knowledge-Based Genetic Algorithm System

Major headings should be typeset in boldface with the first letter of important words capitalized. The fuzzy knowledge-based model is introduced in this section for problem relating to allocation of containers stack with unknown departure time. This model is used also for re-handling operations of containers during retrieval. This technique is selected because the arrival time of trucks to take containers to their destination is unknown. In addition, most of values of the factors being considered in this allocation problem cannot always be precisely predicted. Table 1 shows the source and reason of fuzzy in the values of the considered storage factors.

Table 1. Reason of selecting fuzzy logic

Factors	Source of Fuzziness	Reason of being fuzzy
Number of Containers in the Stack (N)	The arrival (date/time) of truck is unknown due to the fact that customers arrange for the collection of their containers by 3PL companies without any prior notice being given to the yard operators.	The number of containers to be picked up from each stack in a given period of time is uncertain and hence, is considered to be a fuzzy variable.
Duration of Stay of Containers (T)		This factor relates directly to containers' locations. As these locations are continuously changing in response to the rapid retrieval operations for containers that need to depart at unknown times, there is no deterministic pattern for the duration of stay of the topmost containers.
Similarity of Containers (S)	The unknown arrival (date/time) of truck to pick up a container leads to vague and imprecisely defined inputs.	This factor relates to both the number of containers which are stored in each stack i and also the number of containers that depart from each stack i, hence it is considered as a fuzzy variable.

The number of containers in each stack i (N_i) is considered for use in this model as a first input (N). The incoming container similarity to the existing in the stack i (S_i) is considered to be the second input (S). Finally, the time of stay of the containers at the peak of each stack i (T_i) is implemented in the fuzzy knowledge-based model as a third input (T). This input T_i affects the output in that the longer the duration of stay of containers in the stack, the lower the acceptability for a new incoming container for the stack i (α_i).

A Multi-Layered 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 a fuzzy system is applied on an input [23].

In order to imitate the events for arrival, storage, retrieval and departure of containers, the Discrete Event Simulation approach is used for this purpose. Below is the explanation of proposed 'FKB_GA' system framework.

4.1. *Framework of the 'FKB_GA' System*

The Fuzzy Knowledge-Based Genetic Algorithm system framework that is has been proposed is presented in this section. This framework includes input, process and output components. The input component is comprised of the specification information with storage factors and constraints information. The Discrete Event Simulation (DES) approach is utilised to mimic the events of arrival, resource status, storage, retrieval and departure of containers, and these are fed as global inputs in terms of storage and handling times to the other fuzzy knowledge-based and genetic algorithm modules. A number of techniques including Fuzzy Knowledge- Based Rules and Genetic Algorithms are combined for inputs processing are included with the component of the process. A number of key performance indicators comprise the output process and are classified under the criteria such as operational and yard. The utilisation of yard can be considered as one of the yard criteria for terminal throughput evaluation. For the system framework, see Fig. 2.

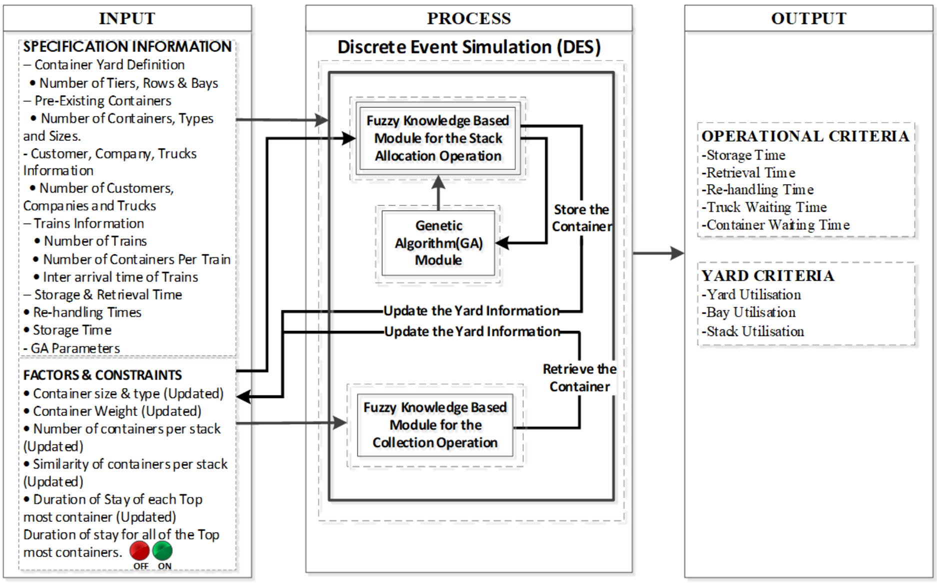

Fig. 2. The 'FKB_GA' System Framework

The container yard settings are included with the specification information of input component, details for pre-existing containers, trains, transportation times (contents shown in the framework above), and finally the GA model information are fed into the system. The GA model information consists of population size, probabilities of crossover and mutation and stopping criterion.

The storage factors and constraint information are also fed into both the storage and collection modules. These factors comprise of container similarities, the containers number in each stack, and the time of stay (i.e. the duration the container at the peak has stayed in each stack). The constraints are container size, weight (full or empty) and type. Information regarding the duration of stay for all the topmost containers collectively is also fed into the storage and retrieval modules. As the container time of stay dynamically changes with time, a strategy called ON/OFF' is required to decide that factors like the time of stay will be considered in succeeding processing.

There are two modules within the process component, they are the Genetic Algorithm (GA) and the Fuzzy Knowledge-Based module. The process commences with the specific information been inputted into the Fuzzy Knowledge-Based module and the factors and constraints information is inputted into both the retrieval and storage modules for processing. Based on the above input, the Fuzzy Knowledge-Based module decides which stack the container would be stored by firing a number of fuzzy rules per stack then determining a level of acceptance (αi) for individual stack. The GA module 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 level of acceptance values of the stacks (αi). The stack that has the highest level of acceptance 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.

4.2. *The 'FKB_GA' System Components*

4.2.1. *The FKB Module*

The Fuzzy Knowledge-Based module is proposed to assess the store location for the incoming container by applying fuzzy reasoning, considering certain constraints and factors and then assign level of acceptance for storage value (α_i) to individual stack. The level of acceptance of storage (α) is the model output, which is an uninformed value that implies the current stack value in the process of decision. This uninformed value is known as the level of acceptance of an incoming container to the stack i (α_i). For every stack i available in the container yard, a value (α) is created in relation to the input constraints and factors. This level of acceptance enables the evaluation of the most appropriate stack location for the incoming container. The stack that has the highest level of acceptance value will be assigned new container to store.

From the container yard are the input known as the crisp inputs that require fuzzy sets to be fuzzified, characterised by their individual membership functions. The membership functions from the fuzzy sets are allocated to individual variable together with linguistic definitions [24] and a triangular "shape" will be applied for all the membership functions. The idea of using fuzzy sets (linguistic variables) rather than crisp representation was introduced by [25] in order to mimic human thinking in systems. The linguistic variables were subjectively determined using expert opinions and experience characterised by literature in this case. According to [26], in the work by [27], triangular membership functions were the most common and suitably represented the behaviour of data.

The factors and constraints explained above, together with their fuzzy sets are shown in Table 2 and explained below.

Table 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 [28].

A triangular membership function was assigned by the output variable (α_i) of this model with six linguistic variables. These linguistic variables are defined by 6 fuzzy sets with their corresponding membership functions depicted in Fig. 3a. The fuzzy sets comprise Very Low 'VL', Low 'L', Medium Low 'ML', Medium 'M', Medium High 'MH', and High 'H'.

The triangular membership functions were assigned three linguistic variables for the first input variable (N_i). The triangular membership function is defined with three fuzzy sets (linguistic variables for n_i which are 'Low', 'Medium', and 'High' as shown in Fig. 3b. Similar to N_i, the second input variable (S_i) has triangular shaped membership functions. The linguistic variables determined for s_i are 'Low', 'Medium', and 'High' as Fig. 3c represents. The third input variable considered in this work is (T_i). Fuzzy sets have triangular membership functions, there are three selected linguistic variables for T_i, which are 'Low', 'Medium' and 'High' as Fig. 3d shows.

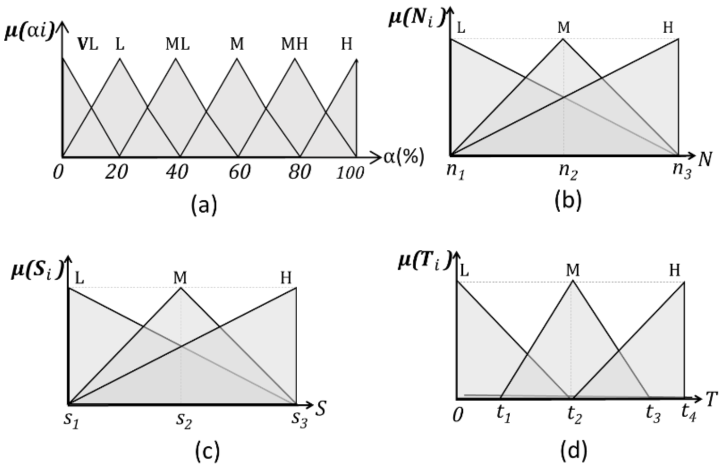

Fig. 3. The fuzzy membership functions: the output (a) the number of containers factor (b) the similarity of containers factors (c) the duration of stay of containers factor (d) the fuzzy membership function of the input factor (T)

In Fig. 3 membership functions depend on the interval value of variables considered by [29]. $\mu(\alpha i), \mu(Si), \mu(Ni), and\ \mu(Ti)$ represent fuzzy membership of the output (α_i), Similarity of containers, (S_i), number of containers (N_i), and Duration of Stay of containers (T_i) respectively. For the number of containers per stack, the interval value is set from 0 to 5 containers in each stack i. Because 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. Because the period of time for the duration of stay of containers is long, a 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 Fig. 4 (a) for W_i, Fig. 4 (b) for F_i and Fig. 4 (c) for Y_i.

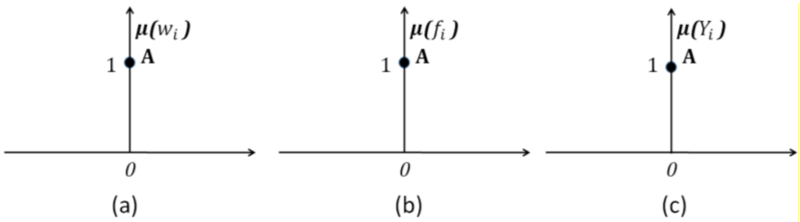

Fig. 4. The defined crisp membership functions of the constraints, (a) The membership function of the weight (W_i,), (b) The membership function of the size (F_i) and (c) The membership function of the type (Y_i)

The fuzzy inference component that comprises of aggregation, will regenerate the information obtained in fuzzy format based on fuzzy rules. Fuzzy rules are determined in order to define the connection between the inputs and the output. The result of the integration of each input variable on the output is defined by these rules [30]. For each input factor 27 different rules with their respective levels are determined and their interactions analysed for the selected variables N_i, T_i, and S_i. Divers rules with the total of 27 are acknowledged with individual levels for their input factor. The rules, which follow an 'IF-Then' structure consider the effect each input variable has on the output. Literature, observation and logic are the main determinants of the rules, based subjectively on expert opinion. Minimising the re-handlings number during the containers retrieval operation is the objective and to achieve this the availability of a location for the incoming container is reflected in the rules. See Table 3 which lists all the fuzzy rules used.

Table 3. The defined fuzzy rules.

Rule No.	Ni	Si	Ti	αi	Rule No.	Ni	Si	Ti	αi
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					

To manipulate the format of the above rules in fuzzy format an aggregation process is used. The minimum operator is applied for the aggregation on completing the rules [31]. The approach used for container stack allocation, Eq. (1), is shown below; for each rule a truncated value (T_j) being calculated.

$$T_j = min\left\{\mu_{(\widetilde{N})}\, n_i, \mu_{(\widetilde{S})} s_i, \mu_{(\widetilde{T})} t_i, \mu_{(\widetilde{W})} w_i,\ \mu_{(\widetilde{F})} f_i,\ \mu_{(\widetilde{Y})} y_i\right\} \qquad (1)$$

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 (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_i 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.

In the previous case of W_i, F_i and $Y_{i,}$ the operator was minimum, so, when the degree of membership for W_i, F_i and Y_i in any rule is calculated to be 0, the final output for all T_j will also be 0. The fuzzy output will then be de-fuzzified to transform the fuzzy output set into a crisp output using the centroid method derived from the centre strategy of gravity method. According to [32], [33] and [34], it is the most common as well as the most physically appealing and prevalent method used. In the de-fuzzification stage, for each rule, the centre value (y_j) is discovered by applying the truncated value reflected on the output fuzzy sets. This is then used to compute the overall centre of gravity. When the truncated value T_j and the output $\tilde{\alpha}$ where the rule defines the outcome to be the level-p is considered, the centre value as shown in Fig. 5 is given by the Eqs. 2 to 5. Eq (6) shows how the centre of gravity method is used to compute the crisp output value y^* by discovery the matching centre values for respective defined rules j (y_j).

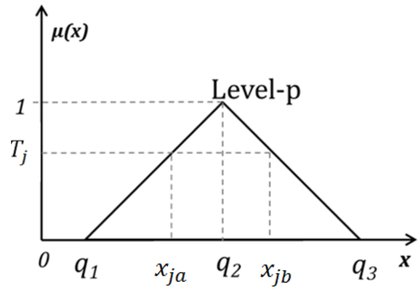

Fig. 5. Truncated value on the output fuzzy set

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

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

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

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

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

Equation (2) is used to find the centre value of the output fuzzy set (y_j) from the boundary values (x_{ia}, x_{ib}). Equations (3) and (4) are used to find boundary values (x_{ia}, x_{ib}) 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).

The level acceptance value (i.e. crisp value) of individual stack (α_i) to be applied for the incoming containers assignment is then calculated. The stack with the highest level of acceptance value will then be selected to store container, while sustaining all conditions relating to input at the same time. As soon as the container is stored, the yard information is updated by the system for the subsequent container coming. The optimisation of the storage operation is achieved through the application of the proposed Genetic Algorithm module. This module holds all the fired fuzzy rules for each incoming container, for all the possible stacks on which they could be stored then releases them for the optimisation process.

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 Figs. 6 (a), 6 (b), and 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 [35]. The truncated value T_1 is calculated by using equation (1) and equal to 0.2, see Fig. 6 (d).

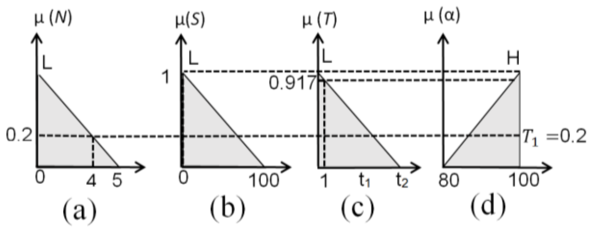

Fig. 6. The matched degrees and truncated value in rule 1

(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 Fig. 5, the boundary (x_{1a}, x_{1b}) is constructed first from the truncated value T_1 in rule 1 as shown in Fig. 6. The values of q_1 and q_2 in Fig. 6 are 80 and 100 respectively. For the high fuzzy set of the output membership function, see Fig. 7.

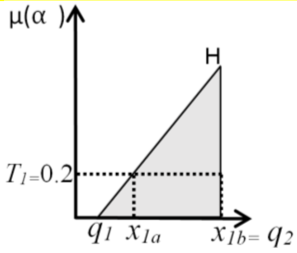

Fig. 7. The boundary of centre value for rule 1

Now the centroid method is applied to calculate the values of x_{1a} and x_{1b} using equation (4);

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

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

By using equation (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_i and centre values y_i 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 (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.

4.2.2. *The Proposed 'ON/OFF' Strategy*

The level of acceptance value for individual stack is calculated by the FKB model using the three input factors and other related constraints. To store the container, the stack with the highest level of acceptance value is assigned. The duration of stay factor is used to determine the level of acceptance values for the stacks. This factor is dynamic because, as time passes, the duration of stay for a container increases so each could have a varying time of stay. As containers are retrieved off the top of the stacks the container underneath has a varying time of stay as the updated containers at the peak in the selected stack.

Because the time of stay of a container is dynamic, an 'ON/OFF' strategy is proposed to activate/deactivate the time of stay factor in the system if the times of stay for the containers at the peak of stack do not significantly differ. The 'ON/OFF' strategy for the time of stay factor is shown in Fig. 8 below.

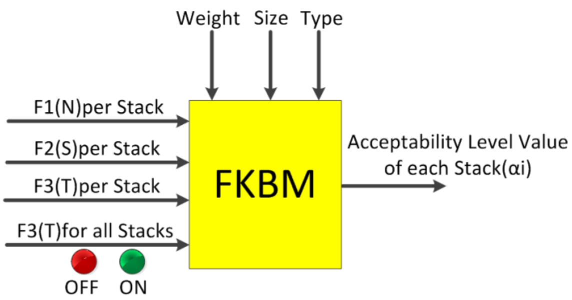

Fig. 8. 'ON/OFF' strategy of Duration of Stay factor [6]

As soon as the time of stay factor is activated (i.e. ON) to the system as an input, all factors (N, S, and T) are applied to determine the level of acceptance values for the operation of container storage. But when the time of stay factor is temporarily deactivated (i.e. 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 defined fuzzy rules determine how the acceptability level values (i.e. the output) are affected by the mixture of various linguistic variables for each input factor. Table 3 identifies 27 fuzzy rules which demonstrate the effect of each input factor on the output. Together with the other two factors, 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 operation of container storage (i.e. the output). When the time of stay factor is deactivated (i.e. OFF), only the other two factors (N and S) are used to determine the level of acceptance values for the stacks. This reduces the number of the defined fuzzy rules to 9 and the acceptability level values are updated. This is shown in Table 4 below.

Table 4. The reduced fuzzy rules

Rule No.	N_i	S_i	T_i/OFF	α_i	Rule No.	N_i	S_i	T_i/OFF	α_i
1	L	L	L	~~H~~ M	15	M	H	M	M
2	L	M	L	~~H~~ MH	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~~ VL
6	L	H	M	H	20	H	M	L	L
7	L	L	H	MH	21`	H	H	L	~~ML~~ M
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~~ ML	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					

The rules emphasised in green will be the only rules used by the system when the duration of stay factor is deactivated (OFF), as shown in Table 4. To determine the level of acceptance values for the stacks in the operation of container storage in this case, only the number of containers and the container similarity factors will be used. The linguistic variables for the time of stay factors are displayed in the highlighted column in red. The linguistic variables for the similarity and container number factors are shown in the rows highlighted in green in the second and third columns. The linguistic variables for the acceptability levels (i.e. output) are highlighted in green in the last column. The Table 4 above represents how the linguistic variables for the levels of acceptance (i.e. output) are updated based on the linguistic variables for the two input factors.

The proposed GA will then select some of the rules out of all the fired fuzzy rules for each incoming container, for each stack, utilising the selected rules for de-fuzzification to re-calculate the acceptability level values of the stacks (α_i). The stack that has the highest level of acceptance value is the optimal stack and will be allocated to store the incoming container fired fuzzy rules per container and possible stack. The proposed GA module selects the best fired fuzzy rules from all the rules fired per stack to achieve the minimum number of container re-handlings.

The next section presents in detail the development of the optimisation module (GA).

4.2.3. *Optimisation of Fuzzy Rules of the FKB module Using GA*

Although fuzzy-knowledge-based models were previously used to select the fuzzy rules from the rule base [36], [37], [38], and [39]. 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.

Using the binary coding, the GA model starts by selecting only the fired rules per stack (set to '1') 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 (set to '0'). 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 Fig. 9 for an explanation of the GA module rules selection per stack.

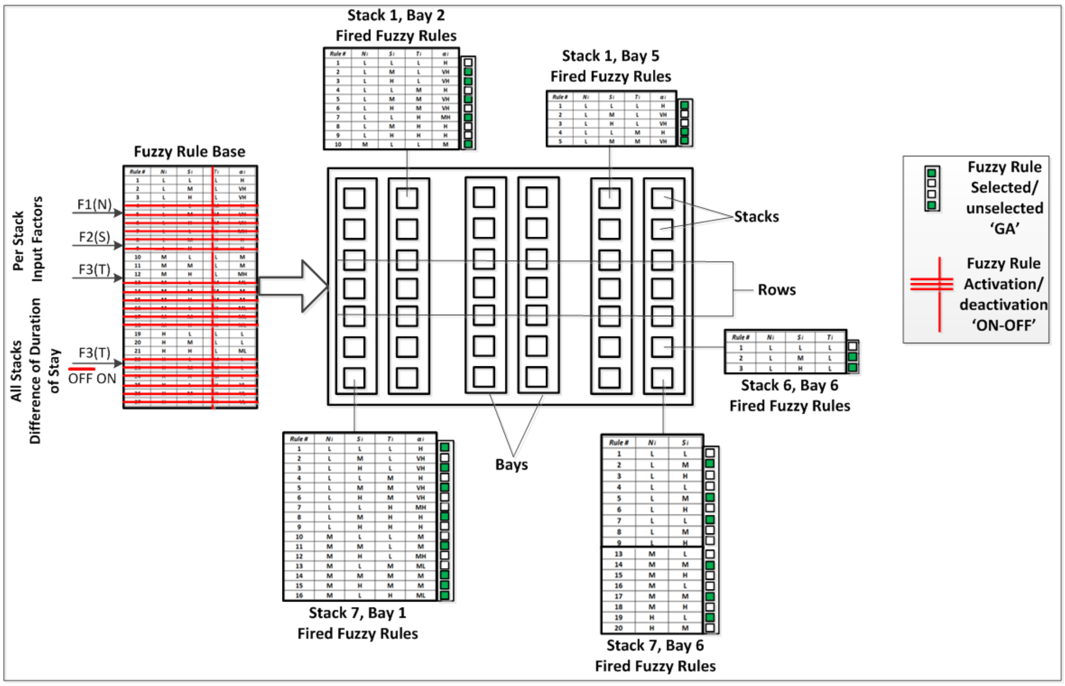

Fig. 9. The proposed GA for rules selection per stack

In Fig. 9, 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.

4.2.4. *The Multi-Layer GA Optimisation Module*

In this section, a multi-layer 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 multi-layer Genetic Algorithm (GA) module was developed to optimise the process of selecting stacks for container storage.

In previous studies ([40]; [41] and [42]), multi-layer Genetic Algorithms models were considered, where each layer/level represented a separate traditional GA model. In this study, each GA layer is assigned per container which includes all possible stacks and their rules. This provides the GA with more flexibility for dealing with large sets of information and the capability of solving the problem for selecting the optimal/near optimal rule(s) out of a set of fired fuzzy rules per container. The Genetic Algorithm steps used in this study include: the design of the chromosome structure, objective function, generation of an initial population, selection method, cross-over and mutation operators which are presented in Fig. 10.

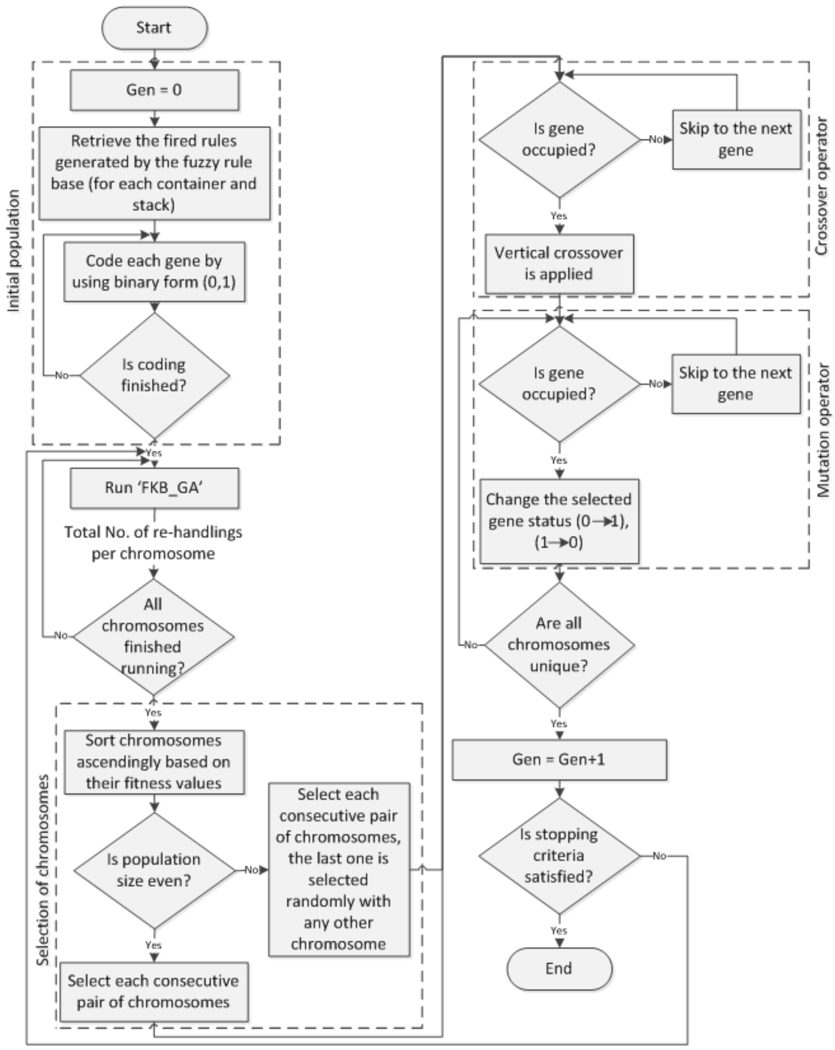

Fig. 10. The flow chart of the proposed Multi-Layer GA

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.

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 is 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 is equal to the number of fired (i.e. used) fuzzy rules for a specific container and possible stacks, and the number of layers is equal to the number of containers. The height dimension for the possible stacks for storing each container is attached with each gene. The fired fuzzy rules are placed in the length dimension (i.e. string) which was a chromosome. This chromosome includes a number of genes that represented the fired fuzzy rules for a container for all possible stack(s). The container number is 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 is 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. 11 shows the proposed Multi-Layer chromosome structure.

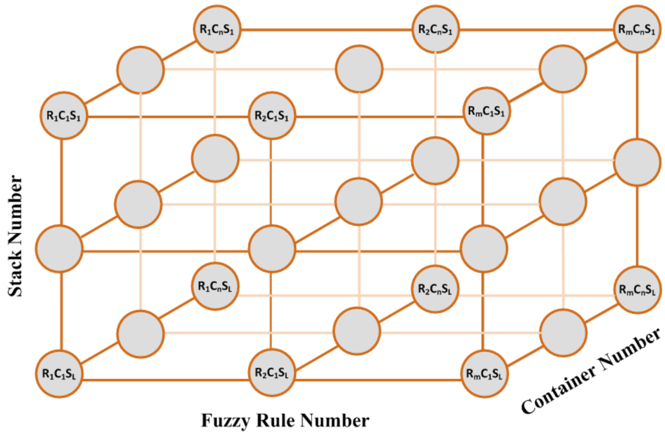

Fig. 11. The Multi-Layer chromosome structure for fuzzy rule representation of n containers

The reason behind this multi-layer chromosome structure is 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 are 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 are calculated to store the containers. For each container and possible stacks, a number of fuzzy fired rules are stored in the generations of a chromosome. The front (i.e. first) layer of the chromosome represents the first container with its fired fuzzy rules and all the possible stack(s). The second layer of the chromosome represents the second container with its fired fuzzy rules and possible stacks. The number of layers depends on the total number of containers. Each gene of each layer is 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 are then stored in multiple layers.

The Objective function

The objective function is formulated to evaluate the performance of the developed ‘FKB_GA’ system in terms of total number of re-handlings of containers. The total number of re-handlings obtained by executing each chromosome is 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.

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.

The Selection Method

After the chromosomes were sorted ascendingly based on their fitness values (total number of re-handlings of containers), the Rank Selection method is applied in which, 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 was 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:

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 Binary Probabilistic Vertical Crossover operator 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 [43]. This crossover operator is used to present the best random exchanging of genes between each pair of chromosomes. See Fig. 12 for an illustration of the crossing-over of two selected chromosomes.

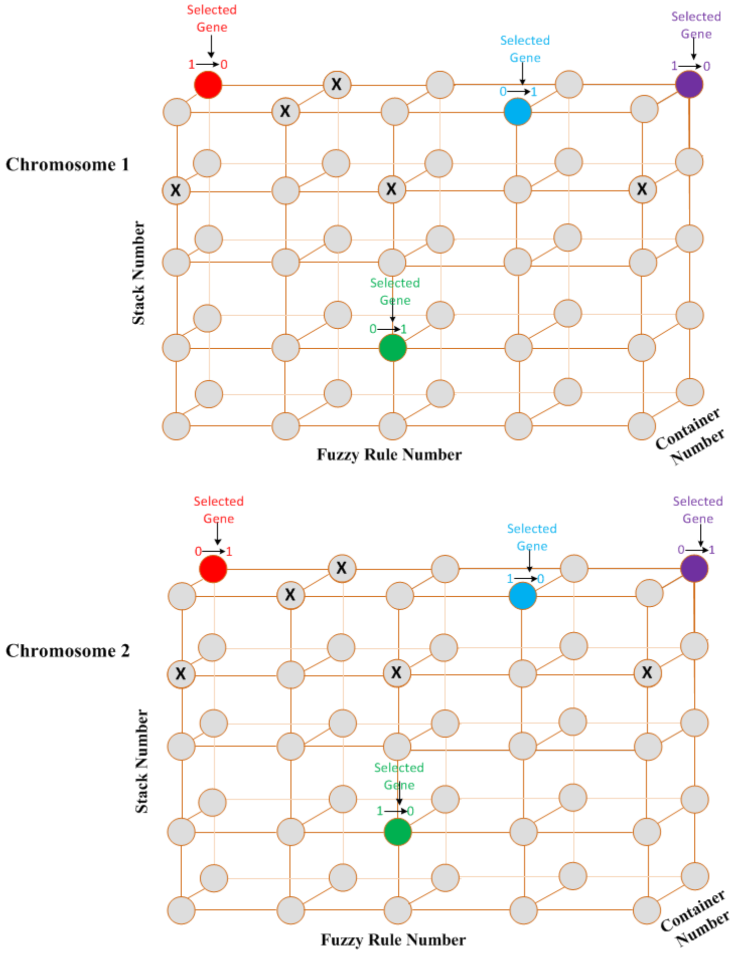

Fig. 12. The Crossing-over of genes in Multi-Layer chromosome

The genes marked with x are empty genes that do not include fired fuzzy rules. The probability of crossover value decides the number of genes to be exchanged at each chromosome. The crossover was skipped when genes contained no fired rule. This operator excludes any marked genes from the mutation operation and considers only genes with fired fuzzy rules. 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.

Mutation Operator: A Binary Probabilistic mutation operator is applied on new chromosomes that were generated from the crossover operation. This operator changes the status of fuzzy rules 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 and flipped from 0 to 1 and vice versa. The proposed GA is 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.

This operator excludes any genes marked with x 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. See Fig. 13 for the mutation operator.

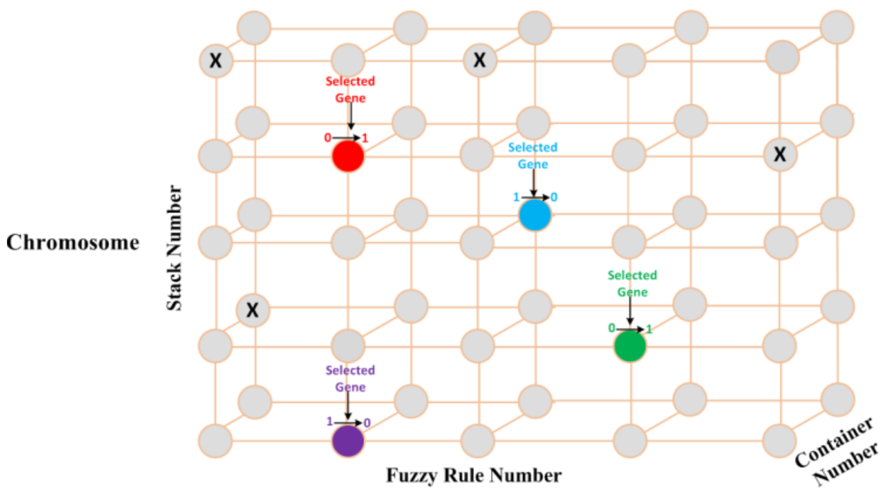

Fig. 13. The Mutation of genes in Multi-layer chromosome

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 yard dimension in terms of number of rows, bays and tiers. 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 3PL companies were also captured. The number of trucks available at each company was assumed to be between 20 and 30 trucks. These trucks are used to pick up containers directly from the yard and deliver them to their destinations (i.e. customers).

Factors of yard occupation by pre-existing containers are provided by Maritime Company based on their calculations on the yard on hand. In order to test the behaviour of the developed 'FKB_GA' system, two real life scenarios were considered including a 'Busy' yard with a significant number of pre-existing containers, and a 'Quiet' yard with a small number of pre-existing containers. In these two scenarios, the proposed Fuzzy Knowledge-Based Genetic Algorithm system was used to calculate the total number of re-handlings. This KPI was selected due to the fact that the total number of re-handlings for retrieving a container is dependent on the total number of containers in the yard and this will clearly reflect the performance of the proposed system in the container retrieval operations, considering the two different volumes of containers scenarios [9].

The impact of embedding the GA in the FKB model was also identified. The system was coded using Visual Basic for Applications (VBA) in MS Office Excel.

5.1. Validation of the developed System

After running the 'FKB_GA' system for Maritime, it was important to determine if the simulation outputs were close to reality. This validation procedure is applied to justify the accuracy and quality of the generated outputs by the system compared with the current way applied by the Maritime Company.

Before running the system, a sample of collected data types and their time were recorded as summarized in Table 5.

Table 5. Sample of data types and time recorded.

Data Type	Time Recorded (minutes)
The storage of a container in the first bay	2
The storage of a container in a neighbouring bay	0.78
The uploading of a container onto a truck	0.70
The re-handling of containers to a neighbouring row	0.76
The re-handling of a container to a neighbouring bay	0.80

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 for 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 key operators. The ‘Current‘approach of handling containers was simulated using the developed system. The company used to store 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.

5.2. Busy Yard Scenario

This scenario used a factor agreed with Maritime 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 were tested against different crossover and mutation probabilities. Each population size consisted of a predefined number of chromosomes and each chromosome covered 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 with this population size. 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. Table 6 shows the parameters tuned by the GA along with their optimal/ near optimal total number of re-handlings.

Table 6. 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

As shown by Table 6, a population size of 15 chromosomes, probability of crossover 0.90 and probability of mutation 0.10 were selected as the most appropriate tuned parameters of GA. The minimum number of re-handlings achieved was 1353. Fig. 14 demonstrates the current adopted approach by the company current, ‘FKB’ and GA results of the ‘busy yard’ scenario.

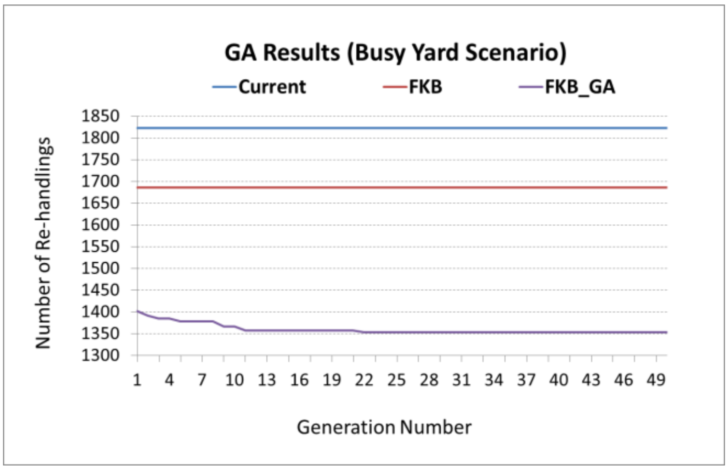

Fig. 14. The busy yard scenario experiment results

In Fig. 14, 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 ‘FKB_GA’. 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 ‘FKB_GA’ (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 that led to the minimum number of re-handlings. For the ‘FKB_GA’ 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 investigation 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 4.2.3, the total number of re-handlings has not been further improved after a greater 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. This also led to reducing the total retrieval times of containers, see Fig. 15.

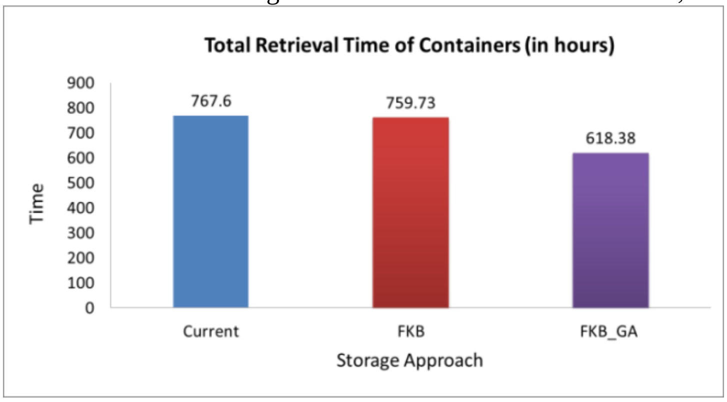

Fig. 15. The total retrieval time of containers

In Fig. 15, by comparing the 'FKB_ GA' approach with the current approach, the total retrieval time of containers was decreased by 19.4%. This is because of the large number of containers within the yard which restricted the time savings in the retrieval process. 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 'FKB_GA' was applied.

When the 'FKB_GA' 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 current and 'FKB' approaches, and hence, this led to the reduction of the total retrieval time.

5.3. Quiet Yard Scenario

To ensure the best search for results for a quiet scenario, a number of experiments were made to tune the GA parameters. The optimal GA parameters were obtained with a population size of 15 and crossover and mutation probabilities equal to 0.75 and 0.20 respectively. The stopping criterion was set to 50 generations and no further improvements were obtained after 50. This scenario considered that 20%-30% of the yard was occupied. Table 7 shows the parameters tuned by the GA along with their optimal/ near optimal total number of re-handlings.

Table 7. The tuned GA parameters for the Quiet Yard Scenario.

Time	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	901	26
	0.90	0.05	1168	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

As shown in Table 7, the minimum number of re-handlings was achieved in a population size of 10 chromosomes, with probabilities of crossover and mutation equal to 0.45 and 0.20 respectively. The maximum number of re-handlings was obtained using the current approach. When the 'FKB' approach was used, the number of re-handlings of containers was less than the current approach. A dramatic reduction in the number of re-handlings was achieved by using the 'FKB_ GA' approach. See Fig. 16 for the quiet yard scenario experiment results.

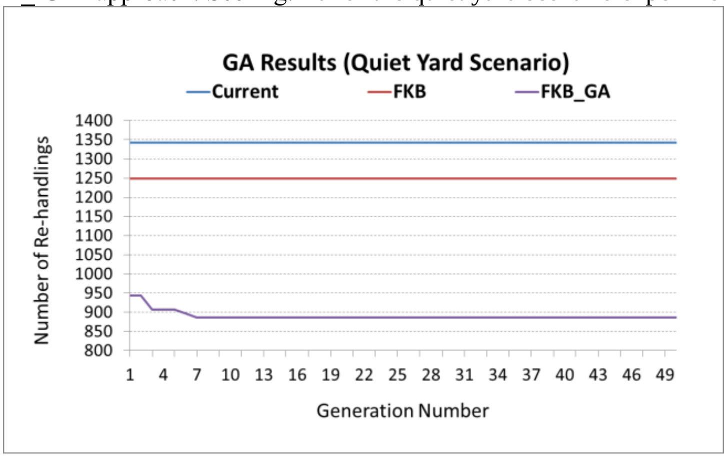

Fig. 16. The quiet yard scenario experiment results

In Fig. 16, it can be seen that the current approach resulted in 1343 re-handlings which was the highest number of re-handlings when compared with the other approaches. The current storage approach allocated containers to the same stack if they were destined for the same customer (i.e. when identified by that specific attribute). By applying the ‘FKB’ approach, the number of re-handlings was reduced to 1249 which was less than the number of re-handlings obtained by the current storage approach. By applying the ‘FKB’ approach, all rules, including weak/inappropriate ones, were selected from the fuzzy rule base which led to the allocation of improper stacks for container storage resulting in a high number of re-handlings of containers. A significant reduction in the number of re-handlings was obtained in the first generation by using the ‘FKB_GA’ approach as can be seen in Fig. 12. In this example, the number of re-handlings started from 943, which meant that in the initial population, a set of strong rules were randomly selected from the fired fuzzy rules that led to the generation of better solutions. A remarkable reduction of 36 re-handlings was obtained at generation 3 which meant that the GA operators had activated more robust fuzzy rules that further reduced the number of re-handlings. A slight reduction in the number of re-handlings was obtained at generation number 6. It can be seen that after generation 6 the minimum number of re-handlings was obtained (886 re-handlings). This minimum number of re-handlings was achieved because the system selected appropriate stacks for containers which led to a reduced number of re-handlings. By using the ‘FKB_GA’ approach, the total retrieval of all containers was minimised, see Fig. 17.

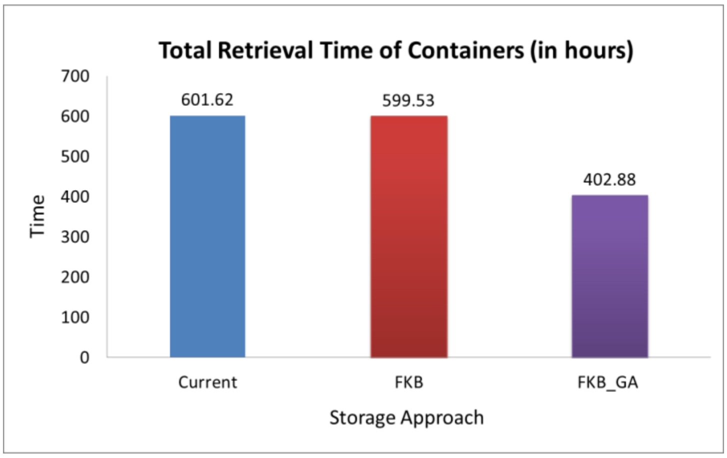

Fig. 17. The total retrieval time of containers

In Fig. 17, by comparing the 'FKB_ GA' approach with the current approach, the total retrieval time of containers was decreased by 32.8%. In addition, the total retrieval time of containers obtained by using the 'FKB' approach was less than the current approach by 196.65 hours.

6. Comparison with Other Approaches

Since other approaches mentioned in the literature would not have direct comparison factors with the same experimental setting and methods suggested in this study, hence, a comparison study with two possible and popular approaches is done to justify the superiority of the proposed approach.

In this section, a comparison study is conducted by comparing the proposed Fuzzy Knowledge-Based Genetic Algorithms approach with Constrained-Probabilistic Stack Allocation (CPSA) approach, and the Constrained-Neighbourhood Stack Allocation (CNSA) approaches [44]. The 'CNSA' approach stores containers and re-handles them to the nearest stacks taking into consideration certain constraints such as the container size, type and weight constraints [45]. 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. Both the CPSA and CNSA approaches are selected as approaches of allocating containers to stack. This is because the problem is specific in terms of considering long durations of stay of containers in the yard and cannot be solved using other approaches. However, for a fair evaluation, CPSA and CNSA are compared with the proposed one. The reason behind this comparison is to test the superiority of the proposed approach over the mentioned approaches.

The comparison was conducted under the 'busy yard' scenario. The proposed 'FKB_GA' 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. The CPSA approach searched for the closest stack possible to the original stack that complies with the constraints of the container. 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. 18 shows the comparison of the total number of re-handlings obtained under the busy yard scenario.

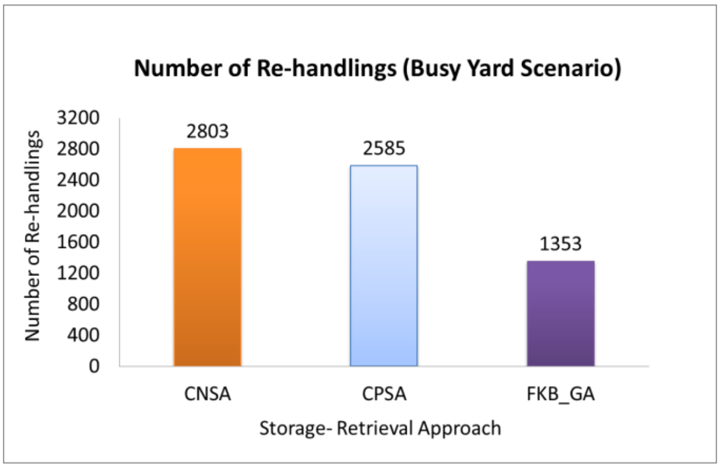

Fig. 18. Comparison between total numbers of re-handlings obtained by using different approaches

In Fig. 18, it can be seen that the 'FKB_GA' achieved a considerable reduction equal to 52% and 48% in the number of re-handlings compared with CNSA and CPSA respectively. The CNSA approach also led to a 8% higher number of re-handlings (2803) when compared with the CPSA approach. This was due to the fact that the GA led in activating the strong rules from the fired fuzzy rules for the input factors in the stacks, which led to the allocation of the optimal stacks and the minimum number of re-handlings of containers.

Additional comparison was conducted to show the performance of the proposed 'FKB_GA' approach in terms of minimising the number of re-handlings obtained in the quiet yard scenario, see Fig. 19.

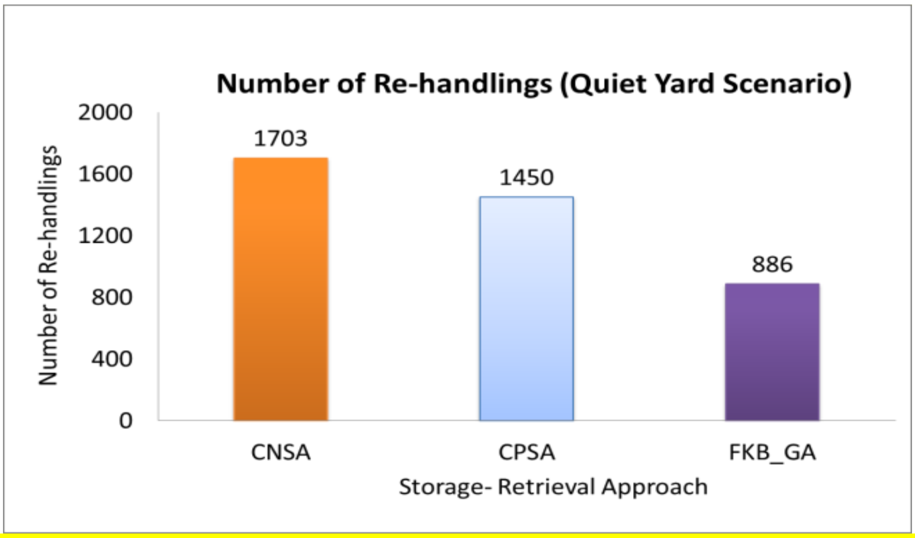

Fig. 19. Comparison between numbers of re-handlings obtained using different approaches (quiet yard scenario)

In Fig. 19, 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.

However, 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.

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. An innovative research framework was presented to model and then 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. In addition, this system was developed to assist operators and planners of container terminal yards to manage effectively and efficiently their storage and retrieval operations, which is crucial to customer satisfaction.

The combination of the FKB approach with GA achieved an optimal/near optimal stack allocation for 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 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 busy scenario, the proposed 'FKB_GA' revealed that the total number of re-handling is reduced by 25.7% when compared to the current approach used by the company, while, for the quiet scenario, the 'FKB_GA' reduced the total number of re-handlings by 34% compared to the current approach. In general, it can be concluded that the Fuzzy Knowledge-Based Genetic Algorithm 'FKB_GA' system reduced rehandling times in all conditions compared with 'Current' and other proposed contianer allocation approaches. However, the largest benefit occurred in the quite yard scenario.

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 in 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 'FKB_GA' system to handle other factors affecting the allocation of containers. The environmental impact of CO_2 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.

References

1. L. K. Tranberg, Optimizing yard operations in port container terminals, *In Proceedings of the 10th EWGT Meeting* (2005).
2. T. Chen, Yard operations in the container terminal-a study in the 'unproductive moves', *Maritime Policy & Management* 26(1) (1999) 27–38.
3. S. C. Ting, J. S. Wang, S. L. Kao, F. M. Pitty, Categorized stacking models for import containers in port container terminals, *Maritime Economics & Logistics* 12(2) (2010) 162–177.
4. B. De Castillo, C. F. Daganzo, Handling strategies for import containers at marine terminals, *Transportation Research Part B: Methodological* 27(2) (1993) 151–166.
5. J. Ries, R. G. González-Ramírez, P. Miranda, A Fuzzy Logic Model for the Container Stacking Problem at Container Terminals, Computational Logistics. *Springer International Publishing* (2014).
6. A. *Abbas A., A. Al-Bazi, V. Palade, A Constrained Fuzzy Knowledge-Based System for the Management of Container Yard Operations, International Journal of Fuzzy Systems 20(4) (2018) 1205–1223.*
7. *L. W. Lan, C. K. Kao, P. D. Student, A comparison of stacking efficiency for various strategies of slot assignment in container yards 4(1) (2001) 303–318.*
8. *K. H. Kim, H. B. Kim, Segregating space allocation models for container inventories in port container terminals, International Journal of Production Economics 59(1–3) (1999) 415–423.*
9. *K. H. Kim, Evaluation of the Number of Rehandles in Container Yards. Computers & Industrial Engineering 32(4) (1997) 701–711.*
10. *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.*
11. *C. Zhang, J. Liu, Y. Wan, K. G. Murty, R. J. Linn, Storage space allocation in container terminals, Transportation Research Part B: Methodological 37(10) (2003) 883–903.*
12. *C. Jin, X. Liu, P. Gao, An Intelligent Simulation Method Based on Artificial Neural Network for Container Yard Operation. In Yin, J, Wang, & C, Guo (Eds.), Advances in Neural Networks - ISNN 2004. Springer Berlin Heidelberg (2004).*
13. *E. Zehendner, D. Feillet, P. Jaillet P., An algorithm with performance guarantee for the Online Container Relocation Problem. European Journal of Operational Research 259(1) (2017) 48–62.*
14. *Y. Liu, H. Kang, P. Zhou, Fuzzy optimization of storage space allocation in a container terminal, Journal of Shanghai Jiaotong University (Science) 15(6) (2010) 730–735.*
15. *S. Saurí, E. Martín, Space allocating strategies for improving import yard performance at marine terminals. Transportation Research Part E: Logistics and Transportation Review 47(6) (2011) 1038–1057.*
16. *X. Yang, N. Zhao, Z. Bian, J. Chai, C. Mi, An intelligent storage determining method for inbound containers in container terminals. Journal of Coastal Research 73(sp1) (2015) 197–204.*
17. *Y. J. Woo, K. H. Kim, Estimating the space requirement for outbound container inventories in port container terminals, International Journal of Production Economics 133(1) (2011) 293–301.*

18. *I. Ayachi, R. Kammarti, M. Ksouri, P. Borne, A Genetic algorithm to solve the container storage space allocation problem. In 2010 International conference on Computational Intelligence and Vehicular System (CIVS), Cheju, Republic of Korea (2010a).*
19. *I. Ayachi, R. Kammarti, M. Ksouri, P. Borne, Harmony search to solve the container storage problem with different container types. International Journal of Computer Applications 48(22) (2013b) 26–32.*
20. *C. Junqueira, A. T. De Azevedo, T. Ohishi, Stowage planning and storage space assignment of containers in port yards. Pianc-Copedec IX (2016).*
21. *D. Ku, T. S. Arthanari, Container relocation problem with time windows for container departure. European Journal of Operational Research 252(3) (2016) 1031–1039.*
22. *S. Ozcan, D. T. Eliiyi, A reward-based algorithm for the stacking of outbound containers. Transportation Research Procedia (22) (2017) 213–221.*
23. *Kacprzyk, J., Pedrycz, W. (2015). Springer handbook of computational intelligence. Berlin, Heidelberg: Springer.*
24. *L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning—I. Information Sciences 8(3) (1975) 199–249.*
25. *L. A. Zadeh, Fuzzy Sets. Information and Control 8(3) (1965) 338–353.*
26. *M. Amendola, M. M. Neto, V. F. Cruz, Using Fuzzy sets theory to analyze environmental condition in order to improve animal productivity. Biomatemática, Campinas (15) (2005) 29–40.*
27. *T. Yanagi Junior, H. Xin, R. Gates, L. Ferreira, Fuzzy logic model to predict laying hen body temperature rise during acute heat stress. In Congresso Brasileiro de Engenharia Agrícola, João Pessoa: SBEA (2006)*
28. *R. Menaka, V. Ranganathan, B. Sowmya, Improving Performance of Fuzzy Based Routing Through Optimization for Manet. Middle-East Journal of Scientific Research (Techniques and Algorithms in Emerging Technologies, (2016), 24, 249–260.*
29. *Y. J. Wang, C. S. Kao, An application of a fuzzy knowledge system for air cargo overbooking under uncertain capacity. Computers & Mathematics with Applications 56(10) (2008) 2666–2675.*
30. *L. A. Zadeh, A theory of approximate reasoning, Machine Intelligence. New York: Halstead Press (1979).*
31. *H. J. Zimmermann, Fuzzy Set Theory and - Its Applications. New York: Springer Science & Business Media (2011).*
32. *J. L. Castro, Fuzzy Logic Controllers Are Universal Approximators. IEEE Transactions on Systems, Man, and Cybernetics 25(4) (1995) 629–635.*
33. *C. C. Lee, Fuzzy logic in control systems: fuzzy logic controller. I. IEEE Transactions on Systems, Man, and Cybernetics 20(2) (1990) 404–418.*
34. *A. Morim, E. Sa Fortes, P. Reis, C. Cosenza, F. Doria, A. Goncalves, Think Fuzzy System: Developing New Pricing Strategy Methods for Consumer Goods Using Fuzzy Logic. International Journal of Fuzzy Logic Systems 7(1) (2017) 1–17.*

35. *Y.J. Wang, C.S. Kao, An Application of a Fuzzy Knowledge System for Air Cargo Overbooking under Uncertain Capacity. Computers & Mathematics with Applications, (2008), 56 (10), 2666–2675*
36. *O. Nelles, M. Fischer, B. Muller, 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).*
37. *P. Carmona, J. Castro, J. Zurita, Strategies to Identify Fuzzy Rules Directly from Certainty Degrees: A Comparison and a Proposal'. IEEE Transactions on Fuzzy Systems, (2004), 12 (5), 631–640.*
38. *M. Cintra, C. De Arruda, Fuzzy Rules Generation Using Genetic Algorithms with Self-Adaptive Selection. In IEEE International Conference on Information Reuse and Integration, (2007), Las Vegas, Nevada, 261–266*
39. *P.C. Shill, M.A.H. Akhand, M. Asaduzzaman, K. Murase, Optimization of Fuzzy Logic Controllers with Rule Base Size Reduction Using Genetic Algorithms. International Journal of Information Technology & Decision Making, (2015), 14 (05), 1063–1092*
40. *M. F. Abdulhalim, B. A. Attea, Multi-layer Genetic Algorithm for Maximum Disjoint Reliable Set Covers Problem in Wireless Sensor Networks. Wireless Personal Communications 80(1) (2015) 203–227.*
41. *G. Kelareva, M. Negnevitsky, Multi-Layered Genetic Algorithm for Maintenance Scheduling with Multiple Parameters. Australian Journal of Intelligent Information Processing Systems 7(3/4) (2002) 122–131.*
42. *M. Negnevitsky, G. Kelareva, Development of a multi-layer genetic algorithm for maintenance scheduling in power systems. In Transmission and Distribution Conference and Exposition, 2008. T&D. IEEE/PES. Chicago, USA: IEEE (2008).*
43. *X. Tang, Y. Liu, J. Zhang, W. Kainz, W. Advances in Spatio-Temporal Analysis. Bristol, PA, USA: Taylor & Francis, Inc* (2007).
44. *J. Ries, R.G. González-Ramírez, P. Miranda, A Fuzzy Logic Model for the Container Stacking Problem at Container Terminals. Proceedings of the 5th International Conference on Computational Logistics (ICCL), Valparaiso, Chile, (2004), September 24-26, pp 93-111.*
45. *A. Lim, Z. Xu, A critical-shaking neighborhood search for the yard allocation problem. European Journal of Operational Research (2006)*