ZIBELINE INTERNATIONAL PUBLISHING

ISSN: 2222-7059 (Print)
EISSN: 2222-7067 (Online)

Advances In Industrial Engineering And Management (AIEM)

DOI: http://doi.org/10.7508/aiem.01.2021.01.09

ARTICLE

AN IMPROVED FUZZY KNOWLEDGE-BASED MODEL FOR LONG STAY CONTAINER YARDS

Ammar Al-Bazi[1*], Vasile Palade[1], Rami AL Hadeethi[2], Ali Abbas[1]

[1]*Faculty of Engineering, Environment and Computing, Coventry University, Coventry CV1 2JH, UK*
[2]*School of Engineering, Department of Industrial Engineering, The University of Jordan, Amman 11942, Jordan*
**Corresponding Author E-mail: aa8535@coventry.ac.uk*

ARTICLE DETAILS

Article History:

Received 22 March 2021
Accepted 15 May 2021
Available online 10 June 2021

ABSTRACT

This paper considers the problem of allocating newly arrived containers to stacks of existing containers in a yard when the departure date/time for containers is unknown. Many factors and constraints need to be considered when modelling this storage allocation problem. These constraints include the size, type and weight of the containers. The factors are the number of containers in a stack and the duration of stay of the topmost container in the stack. This paper aims to develop an improved Fuzzy Knowledge-Based 'FKB' model for best allocation practice of long-stay containers in a yard. In this model, the duration of stay factor does not need to be considered in the allocation decision if the duration of stay for the topmost containers in a stack is similar; hence, a new 'ON/OFF' strategy is proposed within the Fuzzy Knowledge-Based model to activate/deactivate this factor in the stacking algorithm whenever is required. Discrete Event Simulation and Fuzzy Knowledge-Based techniques are used to develop the proposed model. The model's behaviour is tested using three real-life scenarios, including allocating containers in busy, moderately busy and quiet yards. The total number of re-handlings, the number of re-handlings per stack, and the number of re-handlings for containers were considered KPIs in each scenario.

KEY WORDS

Fuzzy Knowledge-Based Model, Fuzzy Rules, Long Stay Containers, Duration of Stay Factor, 'ON/OFF' Strategy, Container Yard Operations, Unknown Departure Time.

1. INTRODUCTION

With the growth in international container shipping worldwide owing to the offshoring of manufacturing, there has been an increased interest in improving the operations in container terminals [1]. Container terminals involve many operations, including rail side, container-yard side, and gate-side operations. The most important operation is the container-yard side involving the storage and retrieval of the containers. The management of container yard operations is complex due to uncertainties inherent in container storage and/or retrieval operations. For achieving efficient utilisation of container yards, container storage operation is essential. Proper storage operation reduces the container yard operations' cycle time [2].

Different types of problems are faced in storage operations where the departure time for containers is unknown, including storage space allocation and location assignment [3]. In the location assignment (stacking) problem, the allocation of containers to stacks is the main focus. Liu *et al.* (2010) and Saurí & Martín (2011) have performed studies concerning the location assignment problem for containers with an unknown departure time where they considered different deterministic, probabilistic and fuzzy factors [4,5].

However, sometimes the factors considered in the container storage operation are affected by real-life conditions, which render them less effective in the container storage decision. For example, the duration of stay factor of containers changes dynamically over time, and it becomes less influential in the storage process when the duration of stay for all the topmost containers of stacks are similar. Hence, a decision is required on whether or not to consider this factor in subsequent processing. Therefore, temporarily removing such factors from the list of factors until the conditions affecting them are no longer present will make the container storage operation decision more effective.

This paper aims to develop an improved Fuzzy Knowledge-Based model for container storage and retrieval operations when container departure time is unknown. It also introduces a new strategy for modelling factors when some are being affected by temporary storage conditions. The factors will be re-activated when the condition is changed.

The rest of this paper is organised as follows: Previous work is reviewed in Section 2. Section 3 presents the container storage problem. The research methodology is presented in Section 4. Section 5 provides the experimental part, results analysis and discussion, followed by the conclusion and future work in the final section.

2. RELATED WORK

In this section, existing works on the container stacking problem for outbound containers with random/unknown departure time are reviewed. This review includes investigating the considered factors and constraints that affect the storage decision and others related to the container duration of stay in the yard. Ozcan and Eliiyi (2017) proposed a reward-based algorithm for solving the storage allocation of outbound containers staying only a few days in the yard [6]. The number and similarity of containers in each stack were considered. A mathematical model was proposed byWoo and Kim (2011) to allocate containers with less than a week duration of stay in a yard, based on the number of containers in each stack[7]. Ayachi *et al.* (2013a) developed a Genetic algorithm model for optimising the static storage space allocation for import and export containers staying only for a short time in the storage space. Different types of containers of the same size were considered [8].

Ku and Arthanari (2016)proposed a stochastic dynamic programming model to calculate the minimum number of expected reshuffles for containers with an assumed stay of only a few days. However, the model did not consider the containers' actual stay in the yard [9]. Tang *et al.* (2015) studied the reshuffling of containers with a short range of stay time in both static and dynamic environments. Although containers of the same size and type were considered, other criteria, including either the grouping of containers or stack height was not taken into account [10]. Park *et al.* (2011) suggested an online search algorithm for improving the container stacking operation, with an assumed duration of stay of one week [11]. Although containers were grouped according to size or weight, however, the utilisation of each stack was not considered. Casey and Kozan (2012) developed a mathematical model for the storage problem of containers with a short duration of stay in the yard. Containers of the same size and type were considered [12]. However, both the duration of stay or the customer's grouping was not considered. Borgman *et al.* (2010) developed a simulation model to evaluate the performance of various online stacking strategies [13]. The storage operation places single size containers that might leave shortly before each other on top of each other. However, different sizes and residence times or categories of these containers were not considered. Ndiaye *et al.* (2014) proposed a mathematical model to optimise storage plans of containers, considering additional storage constraints such as stack capacity and the compatibility between containers and stacks. However, the duration of the stay of containers was not considered [14]. Zhu *et al.* (2020) developed an integer programming model and two-stage search algorithm to simultaneously obtain the proper unloading sequence and optimal yard stacking distribution simultaneously, considering the relation between the expected number of re-handles the stack height [15]. However, the model did not consider the duration of the stay of containers in the yard. Bazzazi *et al.* (2009) presented an efficient Genetic Algorithm (GA) to solve an extended Storage Space Allocation Problem (SSAP) in a container terminal, in which the type of container (as well as different sizes) was considered without paying enough attention to the duration of stay of containers in the yard [16]. Ries *et al.* (2014) developed a fuzzy logic-based rule model for the storage space allocation problem of the same size and type containers. Grouping these containers was based on specific attributes, including the number of containers stored at each stack rather than their duration of stay [17]. Jin *et al.* (2004) developed an intelligent neural network based on fuzzy logic to schedule static container yard operations when the container stay time was only a few days [18]. The model did not consider the weight, size, type, or inter-arrival time of containers or the grouping of containers by customer. Saurí and Martín (2011) proposed a mathematical model based on probability distribution functions to achieve optimal container storage with a short duration stay in the yard. However, the containers would still be stored on top of those with a shorter stay in the yard [5].

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 as the container duration of stay factor and other auxiliary factors such as the number and similarity of containers.

The difference of our study from its predecessors and, in turn, its contribution lies in proposing an improved Fuzzy Knowledge-Based approach that considers containers with a longer duration of stay in the yard along with the other auxiliary factors, which will produce more efficient storage and retrieval plans as the duration of stay of containers varies over time. This challenge appears when the duration of container stay is long because containers with a long duration of stay have more chance of departing the yard, making the stack allocation process for container storage a complex task.

3. PROBLEM DESCRIPTION

The problem starts when a train arrives with a load of containers of different size, type and weight. The reach stacker will then be needed to move containers from the train platform to be stored in a yard containing pre-existing containers. The departure date/time for these containers are unknown because customers deal directly with third-party logistics companies (3PLs) to deliver their containers. The 3PL companies have a limited number of trucks for container collection and transportation. Customers request the collection of their containers, and 3PL companies send trucks to the terminals to collect them without any advance notice being given to the yard operators, making the storage operation challenging.

As a consequence of these containers having an unknown departure date/time, other factors such as the number of containers to be picked up (i.e. depart) from each stack in a given time are unknown/uncertain and are considered to be fuzzy variables. In addition, the duration of stay of the topmost containers is also considered a fuzzy variable because it relates directly to the location of the containers. These locations are continuously changing in response to the rapid retrieval operations for containers that need to depart at unknown times, and hence there is no deterministic pattern for the duration of stay for the topmost containers.

However, many of the topmost containers would often have similar durations of stay as these containers arrive by the same train with inter-arrival times equal to zero. Thus any decisions regarding the allocation of containers based on the duration of stay factor will be unrealistic, leading to a greater retrieval and re-handling time.

Therefore, a Fuzzy Knowledge-Based model has been developed to model these fuzzy factors and other container sizes, type and weight constraints for deciding the best place to store containers. An 'ON/OFF' strategy is proposed to activate the stay of the topmost containers when noticeable differences in the duration of stays of containers are identified. When there is no noticeable difference, this factor will be temporarily deactivated. In this case, decisions will be made based on other considered factors and constraints such as the number of containers per stack, and the size, type and weight of containers.

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

4. FRAMEWORK OF THE IMPROVED FUZZY KNOWLEDGE-BASED (IFKB) MODEL

This section explains the Improved Fuzzy Knowledge-Based 'IFKB' model. The model framework comprises the input, process and output components, as shown in Figure 1. The input component consists of the specification and container yard information.

The 'IFBK' core process consists of collecting techniques that work together to process the inputs. Finally, the output component includes a number of Key Performance Indicators categorised based on the operational criteria.

In the input component, the specification information consists of several input parameters such as container yard settings, number of pre-existing containers, number of customers, number of companies, number of trucks, truck travel time, number of container trains, and the inter-arrival time of container trains. In addition, it includes the transportation time of containers from train to bay, bay to bay, and row to row. The Container yard information involves a number of related factors alongside real-life constraints, including the container size, type, and weight (both empty and full) of the topmost containers in each stack. The factors considered are the number of containers in each stack and the duration of stay (i.e. the length of time the topmost container has stayed in each stack). All this information is fed into the model to generate the required outputs. The 'ON/OFF' strategy to determine whether or not the duration of stay factor is taken into account in subsequent processing will be discussed in detail in section (2).

The 'IFKB' process component is comprised of three main modules, including the Discrete Event Simulation 'DES' technique, the Fuzzy

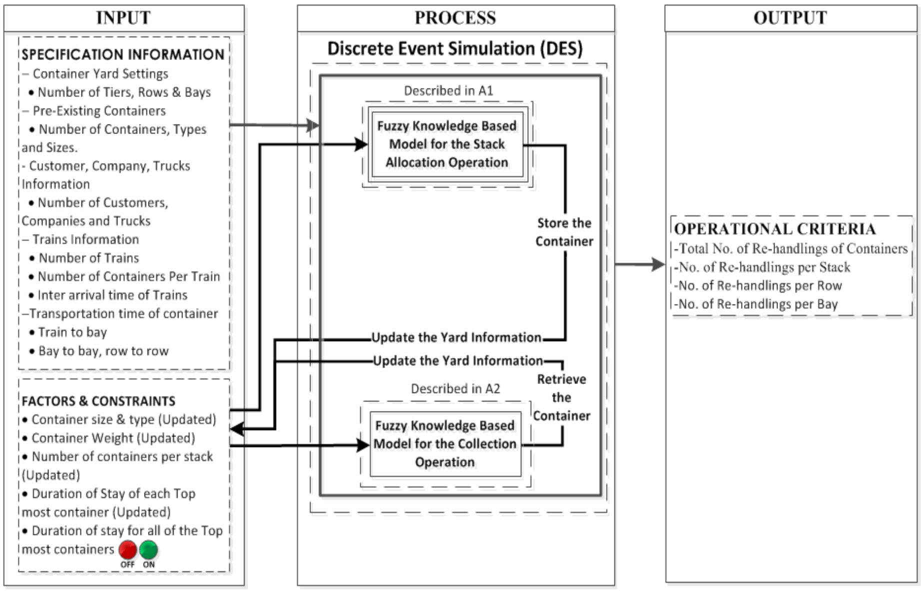


Figure 1: The ‘FKB’ model framework.

Knowledge-Based ‘FKB’ model, along with the proposed ‘ON/OFF’ strategy module. The process starts when the container yard information is fed into the Fuzzy Knowledge-Based model, and the specification information is fed into both the storage allocation operation module and the collection operation module to be processed. The specification information, which includes the container yard settings (i.e. number of bays, number of rows, and number of tiers), number of pre-existing containers and the number of new containers in each train, is fed into the storage allocation operation to initiate the storage of containers in the yard. Also included in the specification information is the number of trucks that collect the containers to deliver to customers, fed into the collection operation.

Using the input information, the Fuzzy Knowledge-Based model determines (i.e. allocates) the stack to store and re-handle the container. It achieves this by first calculating an acceptability level ($\acute{a}_i$) for each stack; then, the container is allocated to the stack with the highest acceptability level. The container is stored/re-handled in the allocated stack, and the yard information will be updated. The discrete event approach simulates both trains and vehicles’ arrival and departure processes and storage and retrieval yard operations. The events of all entities include containers as temporary entities and trains, reach stackers, and vehicles as permanent entities. As shown in Figure 1, the output module includes the total number of re-handlings of containers and the number of re-handlings per stack, row and bay.

The Fuzzy Knowledge Based model components will be explained in more detail in the following section.

4.1 Fuzzy Knowledge-Based model (FKB) for storage operation

This Fuzzy Knowledge-Based model consists of a number of stages, including the fuzzification process, 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, 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 ($\acute{a}_i$). For every stack i available in the container yard, a value α is generated based on the input factors and constraints discussed below. The acceptability level allows for assessing the most suitable stack location for the incoming container. The stack with 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 the lower the acceptability level for the new incoming container to the stack i ($\acute{a}_i$) will be obtained. If the truck arrival time for collecting a container is unknown, then the probability for the service time is longer, (i.e. owing to the number of re-handlings that would need to happen for a condensed container stack) be high. Equally, when the number of containers in a stack is high, the number of re-handlings will also be high in that stack. Therefore, the input N_i is implemented to consider the number of containers for every stack i.

Factor 2: Duration of Stay of Containers (DoS)

The second input (T) is the duration of stay of the topmost 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 the lower will be the acceptability level for a new incoming container for the stack i ($\acute{a}_i$). Based on the work discussed by Saurí and Martín (2011) [5], 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, the probability of departing it in the future is increased if a container is not collected. Since the duration of stay of the containers will be updated.

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

(1) 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 and linguistic definitions [19], and a triangular “shape” will be used for all the membership functions.

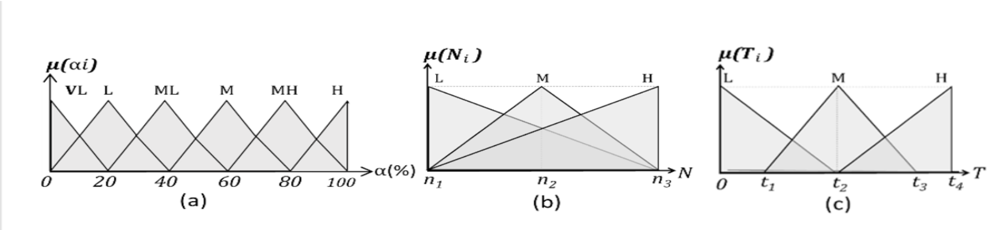


Figure 2: a) The fuzzy membership function of the output. b) The fuzzy membership function of the input factor (N). c) Fuzzy membership function of the input factor (T).

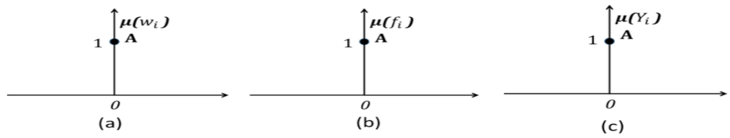


Figure 3: The defined, crisp membership functions of the constraints: a) The membership function of the weight. b) The membership function of the size. c) The membership function of the type.

Table 1: 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

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 Figure 2a. 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 Figure 2b, 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 Figure 2c.

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 is presented in Figure 3a for W_i , Figure 3b for F_i and Figure 3c for Y_i . W_i , F_i and Y_i have the same membership function.

(2) The fuzzy inference- fuzzy rules determination stage

Fuzzy rules have been determined to define the relationship between the inputs and outputs. These rules define the outcome of the interaction of each input variable on the output [20].

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 available for the incoming container to minimise the number of re-handlings of containers during the retrieval operation. Table 1 provides all the fuzzy rules defined in this study.

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

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

Previously, the exceptional condition of W_i ,F_i & Y_i is discussed. As our operator is minimum, in any rule, if the degree of membership of a given value for W_i , F_i and Y_i is computed to be 0, the final output for all T_j will also be 0.

(3) The de-fuzzification stage

The de-fuzzification step involves the operations to transform the fuzzy output into a crisp output. There are various methods for de-fuzzification, including the centre of gravity, mean of maximum and centre average, etc. [22,23]. For this study, the centroid (i.e. a specific implementation of the central strategy of gravity method) is used for the de-fuzzification process.

The strategy finds the centre value (y_j) for each rule using the truncated value reflected on the output fuzzy sets. Then, the overall centre of gravity is computed. Consider the truncated value T_j and the output $\tilde{\alpha}$ where the rule defines the outcome to be the level-p. The centre value is given by the following Eqs. (2 to 5) applied with Figure 4. Upon finding the corresponding centre values for each of the rules, j (y_j) as defined, the crisp output value defined as (y^*) is computed with the centre of gravity method as shown in Eq. (6).

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

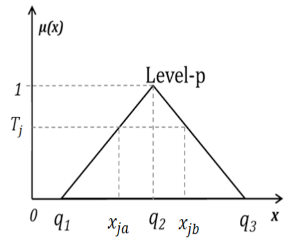


Figure 4: Truncated value on the fuzzy output se.

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

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

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

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

The proposed 'ON/OFF' strategy will be explained in more detail in the following section.

4.2 The proposed 'ON/OFF' strategy

To provide the acceptability level values for the stacks in the presence of unknown departure times of containers, the Duration of Stay of a container is considered one of the input factors and this, as explained earlier, changes dynamically over time. As the duration of stay for containers increases and varies over time, an 'ON/OFF' strategy is proposed to activate/deactivate the duration of stay factor in the model according to the difference in the lengths of stay of the topmost containers in all the stacks. See Figure 5 for the 'ON/OFF' strategy for the duration of stay factor.

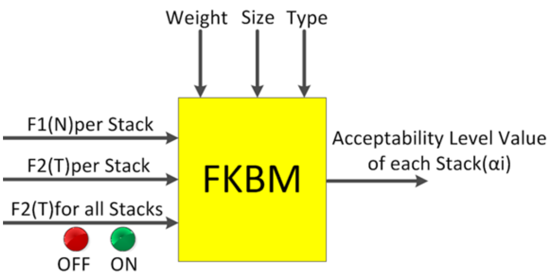


Figure 5: The 'ON/OFF' strategy of the duration of stay factor.

When the duration of stay factor is activated (i.e. ON) to the model, all factors (N and T) are used to calculate the acceptability level values for container storage operation. However, when the duration of stay factor is deactivated (i.e. OFF) to the model, only one factor (N) is used to calculate the acceptability level values for container storage operation (i.e. for stack allocation).

The defined fuzzy rules determine how different linguistic variables for each input factor affect the output (i.e. acceptability level values). For this purpose, 9 fuzzy rules are identified, as stated in Table (1), which define the outcome of the interaction of each input factor on the output. When the duration of stay factor is activated (i.e. ON) with the other factor to the model, all defined rules (9 rules) are fed to the fuzzy inference engine to calculate the output (i.e. acceptability level values for each stack) for container storage operation.

However, when the duration of stay factor is deactivated (i.e. OFF) to the model, the other factor (N) is utilised to calculate the acceptability level values for the stacks. In this case, the number of defined fuzzy rules is reduced to 3, as shown in Table 2 below.

Table 2: The reduced fuzzy rules.

Rule No.	Ni	αi
1	L	H
2	M	M
3	H	L

In Table 2, 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 container factors and the other container size, type, and weight constraints are used to calculate the acceptability level values for the stacks in the container storage operation.

The linguistic variables for the output membership function (i.e. acceptability levels) are updated based on the linguistic variables for the input factor (N), as shown in Figure 6. In Figure 6, the output membership function has three linguistic variables, including 'Low', 'Medium', and 'High'.

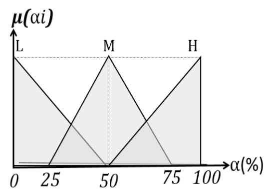


Figure 6: The updated linguistic variables of the output membership function.

(1) The incremental of container length of stay

Once stored in the yard, the length of stay for a container is incremented continually until it departs. The updating process for the container duration of stay must be executed each time a container is stored or departed from the yard. This update assists the decision of when to store newly-arrived containers with pre-existing ones. After some time, each of the containers in the yard will have different lengths of stay. See Figure 7, which illustrates the differing lengths of stay for containers over time.

In Figure 7, there can be seen a number of pre-existing containers which have been stored in the yard for some time (i.e. Containers in Red). When a container arrives to be stored with pre-existing ones, the new

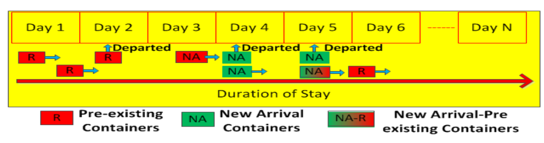


Figure 7: The time incremental for the container length of stay.

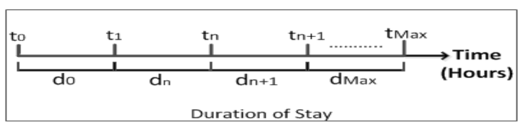


Figure 8: The length of stay progression approximation.

containers will be stored based on the acceptability level values obtained from the FKB model, as explained in section (1). While the new arrivals are being stored (i.e. Containers in Green), some pre-existing containers may depart. Over time, those new containers will become pre-existing (i.e. Containers in half Green and half Red), the duration of stay for the containers will be updated, and each could have its own quite different duration of stay.

(2) The increment of length of stay approximation algorithm

As the length of stay of the topmost containers in each stack changes, the duration of stay factor is updated and provided to the model dynamically over time. The approximation algorithm's notations are defined below then followed by its steps in details.

DoS: Duration of Stay of topmost container in each stack

t_o : Minimum DoS in hours

t_{Max} : Maximum DoS in hours

d : DoS in day

d_o : Minimum DoS in day

d_{Max} : Maximum DoS in day

t_n : DoS between t_o and t_{Max}

d_n : DoS between d_o and d_{Max}

The steps of the algorithm are explained below in detail:

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

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

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

Step 2.1.1: If t_o < DoS ≤ t_1 , then approximate the DoS to d_o

Step 2.1.2: If t_1 < DoS ≤ t_n , then approximate the DOS to d_n

Step 2.1.3: If t_n < DoS ≤ t_{n+1} , then approximate the DoS to d_{n+1}

Step 2.1.4: If t_{n+1} < DoS ≤ t_{Max} , then approximate the DoS to d_{Max}

Step 3: Check the approximated duration of stay

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

Step 3.2: Calculate the number of different durations of stay

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

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

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

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

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

Obtaining the duration of stay for the topmost container in each stack was the first step of the algorithm, then the next step was calculating the possibility percentage for container storage (i.e. the chance of the container being successfully stored in a stack). To calculate the possible percentage for container storage, approximating the duration of stay for containers was necessary. Figure 8 shows the duration of the stay approximation process.

In Figure 8, when the duration of stay is t_1 hours or less, then the DoS is approximated to d_o days, however, when the duration of stay is more than t_1 hours and less than or equal to t_n hours, then the duration of stay is approximated to d_n days. If the duration of stay is more than t_n hours and less than or equal to t_{n+1} hours, then the duration of stay is approximated to d_{n+1} days. But, when the duration of stay is more than t_{n+1} hours or equal to t_{Max} hours, then the duration of stay is approximated to d_{Max} days. The next step was to check the approximated duration of stay for the topmost container of all stacks and consider the stacks with the same approximate duration of stay values as possible (success) stacks for storage. This checking was essential to calculate the number of different durations of stay for containers in the yard. The possibility percentage for container storage was calculated as the number of different durations of stay, divided by the total number of stacks in the yard. If the possibility percentage for the container (success) is greater than or equal to a specific percentage (i.e. provided by the user), then the DoS factor is activated (i.e. ON) to the model. However, if the percentage of storage possibility is less than a specific percentage (provided by the user), then the DoS factor is deactivated (i.e. OFF) to the model.

5. EXPERIMENTAL STUDY, RESULTS AND DISCUSSION

In order to test the behaviour of the developed model, three scenarios are developed, including busy, moderately busy and quiet yard scenarios alongside either considering or not the Duration of Stay (DoS) factor in processing within the model. The proposed 'On/OFF' strategy is applied if the Duration of Stay (DoS) factor is considered. The three scenarios were tested using the Fuzzy Knowledge-Based model for the storage and retrieval operations. This section reports the results of testing the developed model for container yard operations against the mentioned real-life scenarios.

The model's performance was evaluated both with and without the DoS factor being used in the calculation. The developed model was coded using the Visual Basic for Applications (VBA) language within MS Office Excel.

5.1 INPUT PARAMETERS

The input parameters for testing the developed model are presented and applied for the three scenarios. Different resources are utilised in container yard operations, including a container yard, a reach stacker, container trains, and trucks. The container yard is divided into 8 bays, each bay consists of 6 rows, and each row (stack) holds up to 5 containers. The container yard had a number of pre-existing containers. The pre-existing containers had been stored in the yard for 2 to 10 days. The number of container trains was 3 to 5 trains a day for 1 week. Each train had 50 to 70 containers with varying weight size and type. Three sizes of containers are included, which are 20ft (Small), 30ft (Medium) and 40ft (Large), with different types for each size. For each container, the values for parameters used were: Weight: empty or full, size: small, medium or large, Type: 2 of small size, 3 of medium size and 4 of large size. The transportation time for each container in the first bay from the train side was set to 1 minute, and the transportation time per extra bay was set to 1 minute.

In order to activate/deactivate the duration of stay factor in the model, the duration of stay was set to 40%. When the difference in length of stay

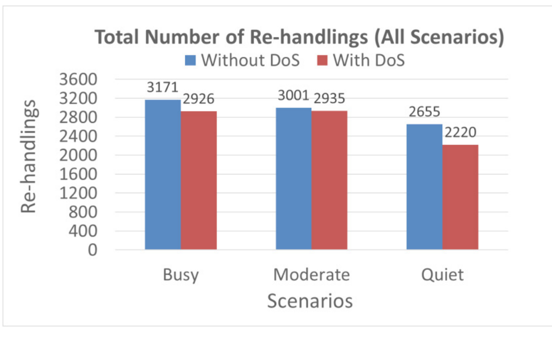
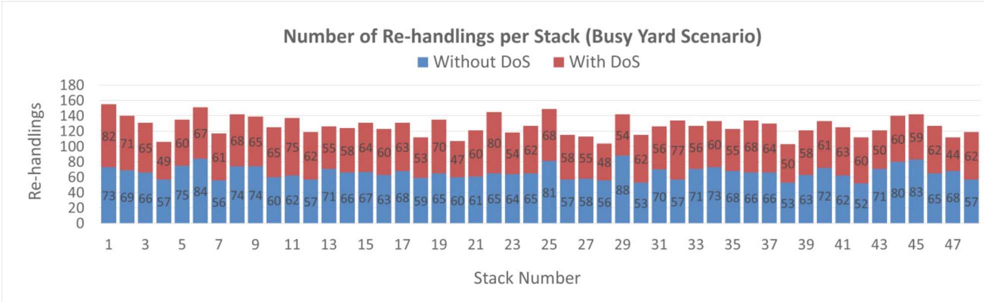
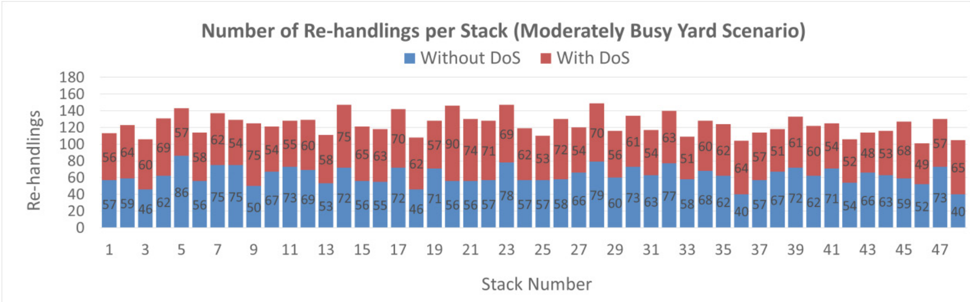


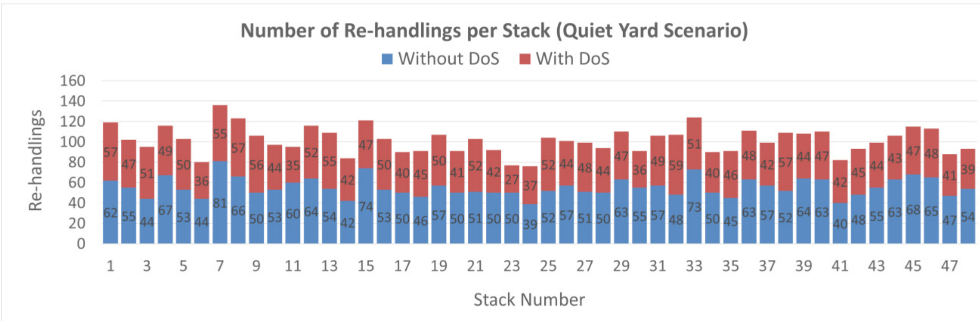
Figure 9: The total number of re-handlings (all scenarios).



(a)

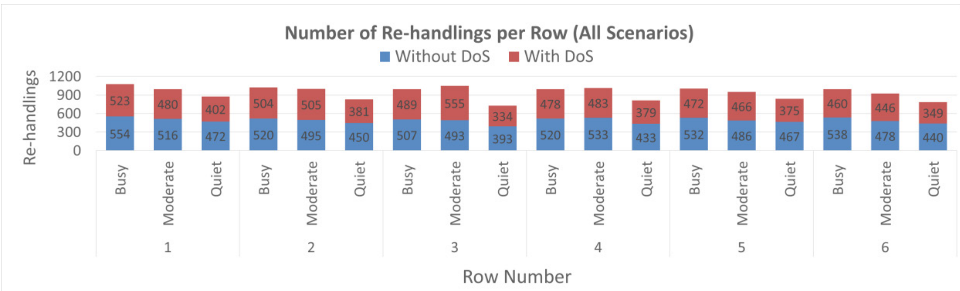


(b)

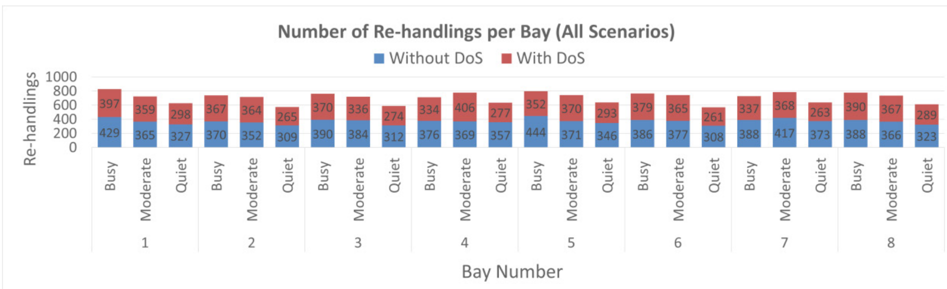


(c)

Figure 10: The number of re-handlings per stack: a) The busy yard scenario. b) The moderately busy yard scenario. c) The quiet yard scenario.



(a)



(b)

Figure 11: The number of re-handlings for all scenarios: a) per row. b) per bay.

of the containers was 40% or above, then the duration of stay factor is activated (ON); otherwise, it is OFF. When the required container was at the top of the target stack, the retrieval time was set to 1 minute, but when there was a container on top of the required one, the re-handling time of that container was set to 1 minute per row and bay. The inter-arrival time between trains was 4 hours. The containers were picked up and delivered to customers by 7 Third Party Logistic (3PL) companies. Each company had 2 to 20 trucks with 15 customers, and each customer had 3 to 10 containers in each train. The travel time for each truck to deliver containers to the customer and return was set from 60 to 200 minutes. The results are presented in the next section with figures and comments.

5.2 RESULTS

The performance of the 'ON/OFF' strategy is presented in the figures below, showing the number of re-handlings of containers. The total number of re-handlings in the busy scenario is reduced by 7.7% when the model considered the DoS factor. When the DoS factor was considered, the model allocated containers to the correct stacks. Figure 9 shows the total number of re-handlings of containers in all scenarios.

In Figure 9, for the moderately busy yard scenario, the total number of re-handlings when considering the DoS factor was reduced by 2.1%, while the total number of re-handlings was reduced by 16.3% when the DoS factor was activated in the quiet yard scenario. However, reducing the total number of re-handlings by activating the DoS factor is noteworthy in all scenarios. When the DoS factor was considered, the model successfully allocated the containers, resulting in the minimum number of re-handlings. Stacks with a lower number of containers and shorter duration of stay were selected. This stacking plan led to a reduced number of re-handlings because the storage operation was efficient. However, when the DoS factor was not considered, the containers in the allocated stacks had a longer duration of stay. This stacking plan led to a higher number of containers being re-handled than in the previous scenario because the storage operation had not considered the topmost containers' duration. However, as shown in Figure 10, the total number of re-handlings obtained in the busy yard scenario was the highest compared to moderately busy and quiet yard scenarios. The number of stored containers in the busy yard scenario was higher than the other two scenarios.

Figures 10a, 10b, and 10c show the number of re-handlings per stack in the busy, moderately busy and quiet yard scenarios taking into account whether the DoS factor was both considered and not considered.

In Figures 10a, 10b and 10c, when comparing the number of re-handlings obtained in stacks by considering the DoS factor, it can be concluded that the highest number of re-handlings was obtained in most of the stacks when the DoS was not considered, while the lowest number of re-handlings in all scenarios, was achieved in most of the stacks when the DoS factor was taken into consideration. As mentioned in Figure 10, the total number of re-handlings in the yard was achieved in all scenarios by considering that the DoS factor was low, and the lowest number of re-handlings in most stacks was thus achieved. For example, as shown in Figure 10a, the highest number of 88 re-handlings was obtained at stack number 29 when the DoS factor was not considered, while the highest number of 80 re-handlings was obtained at stack number 22 when the DoS factor was considered. As seen in Figure 10b, regarding the moderately busy yard scenario, the highest number of 86 re-handlings was obtained at stack 5 when the DoS factor was not considered, and the highest number of 90 re-handlings was achieved at stack 20 when the DoS factor was taken into consideration.

As Figure 10c shows, the highest number of 81 re-handlings when the DoS factor was not taken into consideration can be seen at stack 7. In the same figure, the highest number of 56 re-handlings was obtained when the DoS factor was considered can be seen. However, the number of re-handlings per stack was lowest when applying the DoS factor because the containers were spread equally across the yard for easier and faster retrieval in all scenarios. This model took the number of containers per stack and the duration of stay of the topmost containers per stack factors when selecting a stack for the container storage and retrieval operations. Container yards included a number of rows and bays, and the number of re-handlings achieved at these row and bays during the retrieval operation is shown in Figures 11a and 11b.

The following results can be seen when the number of re-handlings achieved at rows and bays in Figure 11a and Figure 11.b are compared. By considering the DoS factor, the lowest number of re-handlings in most of the rows and bays was obtained in all scenarios, while the number of re-handlings obtained in rows and bays in the three scenarios was higher when the DoS factor was not considered. This was because the total number of re-handlings obtained in all scenarios was the lowest when the DoS factor was considered, as explained in Figure 11, consequently achieving the lowest number of re-handlings rows and bays. However, the number of re-handlings obtained at rows and bays in the busy yard scenario was higher than the moderately busy and quiet yard scenarios, as the number of stored containers in the busy yard scenario was higher than the other two scenarios.

6. CONCLUSION AND FUTURE WORK

An Improved Fuzzy Knowledge-Based model was developed to solve the stack allocation problem for storing newly arrived and re-handled containers with several pre-existing containers in the yard. An 'ON/OFF' strategy was successfully developed to respond efficiently to variations in the duration of stay (DoS) factor, especially when the validity/effectiveness of this factor changes over time due to certain storage conditions. The proposed model was proven to be used for the efficient storage and retrieval of containers and considers a number of factors and constraints and the problem of the unknown departure date/time for containers in the yard. The results indicated that when the DoS factor was considered during containers' storage and retrieval operations, the number of re-handlings was reduced in the busy, moderately busy and quiet yard scenarios.

More factors and constraints could be included in the developed Fuzzy Knowledge-Based model for better container storage operation decisions for future work. An optimisation module-based Genetic Algorithms could be developed to optimise the fuzzy rules allocated per stack for more robust container storage allocation.

REFERENCES

[1] Galle, V., Boroujeni, S. B., Manshadi, V. H., Barnhart, C., Jaillet, P. The stochastic container relocation problem. ArXiv Preprint ArXiv, (2017):1703.04769.

[2] Zehendner, E., Feillet, D., Jaillet, P. An algorithm with a performance guarantee for the online container relocation problem. European Journal of Operational Research, (2017) 259(1): 48–62.

[3] Chen, L., Lu, Z. The storage location assignment problem for outbound containers in a maritime terminal. International Journal of Production Economics, (2012) 135(1): 73–80.

[4] Liu, Y., Kang, H., Zhou, P. Fuzzy optimisation of storage space allocation in a container terminal. Journal of Shanghai Jiaotong University (Science), (2010) 15(6): 730–735.

[5] Saurí, S., Martín, E. Space allocating strategies for improving import yard performance at marine terminals. Transportation Research Part E: Logistics and Transportation Review, (2011) 47(6): 1038–1057.

[6] Ozcan, S., Eliiyi, D.T. A reward-based algorithm for the stacking of outbound containers. Transportation Research Procedia 22, (2017): 213–221.

[7] Woo, Y.J., Kim, K.H. Estimating the space requirement for outbound container inventories in port container terminals. International Journal of Production Economics, (2011) 133 (1): 293–301.

[8] Ayachi, I., Kammarti, R., Ksouri, M., Borne, P. A heuristic for re-marshalling unbound and outbound containers. in Systems and Computer Science (ICSCS), 2013 2nd International Conference On. held 2013. IEEE, (2013a): 291–296.

[9] Ku, D., Arthanari, T.S. Container relocation problem with time windows for container departure. European Journal of Operational

Research, (2016) 252 (3): 1031–1039.

[10] Tang, L., Jiang, W., Liu, J., Dong, Y. Research into container reshuffling and stacking problems in container terminal yards. IIE Transactions, (2015) 47 (7): 751–766.

[11] Park, T., Choe, R., Hun Kim, Y., Ryel Ryu, K. Dynamic adjustment of container stacking policy in an automated container terminal'. International Journal of Production Economics, (2011) 133 (1): 385–392.

[12] Casey, B., Kozan, E. Optimising container storage processes at multimodal terminals'. Journal of the Operational Research Society, (2012) 63 (8): 1126–1142.

[13] Borgman, B., van Asperen, E., Dekker, R. Online rules for container stacking. OR Spectrum (2010) 32 (3): 687–716.

[14] Ndiaye N.F., Yassine A. DiarrassoubaI, A branch-and-cut algorithm to solve the container storage problem. ICONS 2014: The Ninth International Conference on Systems, (2014).

[15] Zhu H., Ji M., Guo W. Two-stage search algorithm for the inbound container unloading and stacking problem. Applied Mathematical Modelling, (2020) 77(2): 1000-1024.

[16] Bazzazi M., Safaei N., Javadian N. A genetic algorithm to solve the storage space allocation problem in a container terminal. Computers & Industrial Engineering, (2009) 56(1): 44-52.

[17] Ries, J., González-Ramírez, R.G., Miranda, P. A fuzzy logic model for the container stacking problem at container terminals. in Computational Logistics: 5th International Conference, ICCL (2014), Valparaiso, Chile, September 24-26. Proceedings. Lecture Notes in Computer Science, vol. 8760, Springer, pp. 93-111, 5th International Conference, ICCL (2014), Valparaiso, Chile, Chile, 24/09/14. https://doi.org/10.1007/978-3-319-11421-7_7

[18] Jin, C., Liu, X., and Gao, P. 'An intelligent simulation method based on artificial neural network for container yard operation'. International Symposium on Neural Networks, Dalian, China, August 19-21, (2004), Part II.

[19] Zadeh, L, A. The concept of a linguistic variable and its application to approximate reasoning—I. Information Sciences, (1975) 8(3): 199–249.

[20] Zadeh, L, A. A theory of approximate reasoning. Machine Intelligence 9, J. Hayes, D. Michie, and LI Mikulich (eds.), (1979): 149-194. New York: Halstead Press.

[21] Zimmermann, H.-J. Fuzzy set theory and - its applications. New York: Springer Science & Business Media, (2001).

[22] Zadeh, L, A. fuzzy sets, Information and Control. (1965) 8(3): 338–353.

[23] Zimmermann, H. J. Fuzzy set theory and its applications. Boston: Kulwer, (1991).