TRANSPORT
ISSN 1648-4142 / eISSN 1648-3480
2020 Volume 35 Issue 1: 37–47
https://doi.org/10.3846/transport.2019.11628

OPTIMIZATION OF CONTAINER RELOCATION OPERATIONS IN PORT CONTAINER TERMINALS

Livia MAGLIĆ*, Marko GULIĆ, Lovro MAGLIĆ

Faculty of Maritime Studies, University of Rijeka, Rijeka, Croatia

Received 2 November 2018; revised 3 June 2019, 19 September 2019; accepted 25 October 2019;
first published online 9 December 2019

Abstract. The relocation of containers is a crucial operation in container ports all around the world. The Container Relocation Problem (CRP) is focused upon to find a sequence of container retrievals in a defined order from a single yard container bay with a minimum number of relocations. The goal of this paper is to find out if Genetic Algorithm (GA) can give new insights in the problem of solving the CRP. In this paper we focus on the two-dimensional, static, offline and restricted CRP of real-world yard container bays. Four rules are proposed for determining the position of relocated containers. We applied GA to find the best sequence of container retrievals according to these four rules in order to minimize the number of relocations within the bay. The experimental testing was run on a total of 800 different instances with varying bay sizes and number of containers. The given results are compared with the results of different authors using other heuristic methods. The results show that the proposed model solves CRP and achieves near optimal solutions.

Keywords: logistics, port, container terminal, stacking area, container relocation problem, discrete optimization, genetic algorithm, performance analysis.

Notations

Abbreviations:

2D	two-dimensional;
3D	three-dimensional;
A1…A13	assumptions of the model;
AS/RS	automated storage/retrieval system;
AVS/RS	autonomous vehicle storage/retrieval system;
B&B	branch and bound;
BS	beam search;
BWP	blocks-world planning;
C1…C4	constraints of the model;
CRP	container relocation problem;
CRP-W	CRP with container weight;
DC	dual command;
DSS	decision support system;
GA	genetic algorithm;
JGAP	*Java* GA and programming;
LPF	lowest priority first;
NP	non-deterministic polynomial time problem;
PMP	pre-marshalling problem;
R1…R4	heuristic rules;
RI	reshuffle index;
RTG	rubber tired gantry;
S/R	storage/retrieval;
SC	single command;
SP	split platform;
TEU	twenty-foot equivalent unit;
TLP	tier lowest position.

Variables and functions:

C_p	container position within a bay;
F	total fitness value of all relocated container within the bay;
f_c	value of fitness function of the chromosome;
H	total number of tiers in a bay;
$H - 1$	greatest number of relocations to retrieve one container;
i	number of tiers in stack (height of stack);
i_0	maximum stacking height within the bay;
j	number of stacks in a bay (width of bay);
j_0	first stack within the bay;
l_c	length of the chromosome in the GA;
N	total number of containers stacked within a bay;
N_p	size of a population in the GA;
n_e	number of evolutions in the GA;

*Corresponding author. E-mail: *livia@pfri.hr*

P – retrieval priority of a container;
p_c – probability of crossover in the GA;
p_m – mutation probability in the GA;
R – number of container relocation within the bay;
W – total number of stacks in a bay.

Introduction

A large part of the world's total transport refers to the transport of goods by containers. A container terminal plays an important role in the container transport process as it connects the container transfer between the land and the sea. Due to the increase in world container transport in recent years, there is a need to optimize the operations of container transport that occur within a container terminal in order to improve the entire container transport process from the starting location to the defined destination.

The stacking area is a subsystem of a container terminal in which the containers are temporarily stacked while awaiting shipment. Considering the limited capacity dimensions of this stacking area within the majority of the world's container terminals (Kim, Hong 2006), containers are stacked side by side and on the top of each other forming blocks, each of them consists of a number of rows in width, a number of bays in length and number of tiers in height. In order to make better use of the limited ground space, containers are stacked on the ground and piled up. In order to get access to an individual container that is stored below others, the upper ones have to be relocated. This relocation represents the change of position of a blocking container within the bay. This activity is unproductive crane work but unavoidable if the arrival of the truck is a stochastics process (Zhao, Goodchild 2010). The relocation is practically impossible to avoid completely, because at the moment of stacking containers in the stacking area, most of the information about the future shipping of the containers is unavailable or incomplete. Although all the relevant data (e.g. the defined ship, the container's unloading port, etc.) for shipping a container are completely accurate and sent on time before the stacking of a certain container, it can happen that this very important data can be changed during the transportation process and consequently the order of retrieving the container from the stack is changed, causing the additional relocations within the stack. It is estimated that 30…40% of the outbound containers at European terminals have no correct information about the ship or the destination port (Steenken *et al.* 2004). For inbound containers the situation is even worse. During the unloading of containers from a ship, the land transport means for the further shipment is known in only 10…15% of container shipment, whilst for the remaining cases the container shipment method is chosen at the time when the container is stacked at the stacking area (Steenken *et al.* 2004). Consequently, it can be concluded that the operation of container relocation at the stacking area is performed relatively frequently. Therefore, in order to use the existing stacking surfaces within the terminal more rationally and effectively, exceptional efforts are made for solving the relocation problem with an optimization method.

The optimization method is used for minimizing the number of container relocations within the bay in order to resolve two similar, but still different problems: CRP and PMP. The basic difference between these problems is that CRP deals simultaneously with the relocation and the retrieval of containers within the bay, whilst PMP deals only with container relocation within the bay.

In this paper we propose a model based on a GA that resolves the CRP minimizing the total number of container relocations within the bay and thus speeding up container retrieval. We introduced four heuristic rules into this model. Each rule determines the position within the bay in which the current blocking container has to be relocated in order to retrieve a certain container that has to be shipped from the bay at that point. Our model, based on a GA, searches for the optimal sequence of these four heuristic rules (each rule can be used more than once within this sequence) that will minimize the total number of container relocations within the bay.

The paper is organized as follows. In Section 1 we discuss the literature review. In Section 2 the basic terminology of CRP is introduced. The proposed model for resolving the CRP is presented in Sections 3 and 4. In Section 5 the evaluation of our method is presented. Finally, the conclusion is given in the last section.

1. Literature review

The literature review is divided into two main parts. First part refers to literature review of definition, approaches, heuristic rules and methods for resolving CRP in conventional container terminal while second part refers to literature review of stacking policy and models, which deals with retrieval travel time in automated storage and retrieval system at container terminals.

The first scientific paper that studies the CRP at a stacking area was written by Sculli and Hui (1988). They presented a simulation model in which they observe the effect of the number of stacks, the number of tiers and the different types of containers on the total number of operations with the containers at a stacking area. The obtained results showed that the different types of containers themselves have the largest impact on the total number of operations. Gupta and Nau (1992) shown that the BWP problem in the theory of complexity belongs to a NP-hard. Consequently, there are many proposed models that use heuristic methods in order to resolve these optimization problems within the stacking area of container terminal. De Castillo and Daganzo (1993) and Kim (1997) have a similar approach in which they proposed heuristic rules to calculate the total expected number of relocations for inbound containers. Avriel *et al.* (1998) proposed a model based on heuristic procedures and binary programming for reducing the overall costs of container relocations on container ships. This problem is very similar to CRP because the containers loaded onto a ship have to be unloaded in several ports, therefore some containers have to be unloaded before others. Kim, K. H. and Kim, H. B. (1999) considered the dynamic CRP and proposed a formula that explains the relation between the stack height and the expected number of container relocations inside a bay. In the paper the appropriate maximum height of a stack was determined in order to minimize the number of container relocations. Murty *et al.* (2005) provided a DSS for the container terminals in Hong Kong. They divided the problem into five related sub problems, one of which is the CRP. In order to minimize the number of relocations the authors' defined rule called RI was applied in order to relocate containers. Relocated container was stacked in the stack where the reshuffling index was the smallest. For a relocated container, the RI of a stack represents the number of containers that will be picked up earlier than the relocated container. Yang and Kim (2006) defined the general concept of the CRP and presented a model in which the container that has already been relocated cannot be relocated again. The aforementioned restriction limits the usage of the model in real-world situations. Hussein and Petering (2012) proposed a model for resolving CRP-W in which the container weight is also considered while searching for the optimal relocation solution in order to minimize the fuel costs of the stacking crane. The model is based on a GA. Caserta *et al.* (2012) shown that the CRP is a NP-hard problem. Consequently, with the exact solution approaches, only small size instances can be solved in a reasonable time. Therefore, only a few papers deal with exact solution approaches. There are several exact and heuristic solution approaches for this problem. The exact solution approaches applied integer linear programming (Caserta *et al.* 2012; Lee, Hsu 2007; Petering, Hussein 2013; Tang *et al.* 2012; Wan *et al.* 2009; Zehendner *et al.* 2015). Most of the heuristic solution approaches are based on the B&B method with varying search strategies and several branching strategies (Da Silva *et al.* 2018; Caserta *et al.* 2009; Caserta, Voß 2009; Expósito-Izquierdo *et al.* 2014; Forster, Bortfeldt 2012; Jin *et al.* 2015; Jovanovic, Voß 2014; Kim, Hong 2006; Ku, Arthanari 2016; Rei, Pedroso 2013; Tanaka, Takii 2016; Tanaka, Mizuno 2018; Tricoire *et al.* 2018; Ünlüyurt, Aydın 2012; Wu, Ting 2010; Zhang *et al.* 2010; Zhu *et al.* 2012). Along with the mentioned solution approaches there are several authors using the tabu search method (Wu *et al.* 2009), the Beam search (Wu, Ting 2010) and the Corridor method (Caserta *et al.* 2011). These authors research the offline variant of the CRP where the retrieval times are known in advance. Due to the complexity of the problem, different heuristics and metaheuristics methods, that efficiently solve the CRP, were used in the majority of proposed approaches.

In this paper, we also consider the offline CRP and we use GA for solving this problem. Based on the existing literature it has been determined that GA is applied only for solving the variant of the CRP called CRP-W (Hussein, Petering 2012), whilst for solving the standard CRP a GA has not been applied. Therefore, this is the main reason why this method was chosen.

The literature dealing with the travel time modelling for unit-load AS/RS shows many different approaches. The first study on the performance of AVS/RS was proposed by Malmborg (2002) who modelled the travel time of an AVS/RS according to the number of aisles, tiers, vehicles and lifts, considering random storage policy and tier to tier configuration. Van den Berg (2002) identified the optimal dwell point position that can minimize the expected travel time from the dwell point position to the position of the first operation after the idle period. Sari *et al.* (2005) presented mathematical models for the expected travel time for a flow-rack AS/RS, which use two (S/R) machines. Hu *et al.* (2005) have presented a new kind of S/R mechanism that enables AS/RS to efficiently handle heavy loads together with a continuous travel time model for the new AS/RS, under the stay dwell point policy. Hachemi and Alla (2008) presented an optimization method of retrieval sequencing where the AS/RS was depicted by a coloured Petri net model. De Koster *et al.* (2008) used a lifting mechanism on the opposite face of the S/R machine for a flow racks AS/RS and they presented a closed-for expression of expected retrieval-time for SC cycles and derived an approximate travel time expression for dual-command cycles of the system. Ghomri *et al.* (2009) proposed an analytical expression for the average time of a single cycle of a multi-aisle AS/RS. Sari (2010) performed comparative analysis between unit-load AS/RS and flow rack AS/RS in which is considered space use and travel time parameters. Lerher *et al.* (2010) presented analytical travel time models of multi-aisle AS/RS for the computation of travel time for both SC and DC cycles. Hamzaoui and Sari (2015) have determined optimal dimensions of the same AS/RS design minimizing expected travel times of the S/R machine using enumeration technique. Liu *et al.* (2016) have presented continuous travel model for the DC in the SP-AS/RS under input and output dwell point policy and shown that the square-in-time rack incurs the smallest expected travel time and that the DC is better than SC in terms of the expected travel time. Xu *et al.* (2018) have developed a continuous travel time model for DC in 3D compact storage system and used analytical models for optimize system dimensions.

2. Problem description, assumptions and constraints

In this paper, we resolve the CRP for one bay with a certain number of columns, tiers and containers stacked within this bay. Containers within the bay are marked with a different retrieval priority P based on the departure order defined in the transport documentation. The retrieval times are known in advance (offline variant). The priorities of retrieval are marked with ordinal numbers. A container marked with the number 1 has a greatest priority and is the first one to be retrieved. The CRP focuses to find the best sequence of rules to retrieve all the containers within the given bay (respecting their priority) with the minimum number of relocations. The relocation represents the movement of a blocking container within the bay. A blocking container is a container located above the target container. The target container is a container that must be retrieved first before all the remaining containers within the bay. The retrieval represents the lifting of the target container and loading it onto a truck or other internal transport unit. Container retrievals and relocations are performed by a yard stacking crane called a RTG (Figure 1).

The problem's definition relies on the assumptions A1 to A13 and constraints C1 to C4.

The general assumptions of the model are:

- A1 – all containers are the multipurpose dry type;
- A2 – all containers are 20-foot (1 TEU);
- A3 – all containers are full;
- A4 – containers are stacked according to their weight;
- A5 – all containers are outbound and are loaded onto internal trucks and then on a ship;
- A6 – the RTG crane performs the relocation and retrieval with only one container at the same time;
- A7 – the problem is dealing with the one bay;
- A8 – the bay is defined by two dimensions: the number of stacks and number of tiers;
- A9 – the problem is defined as static – no new container arrival is allowed during the retrieval process;
- A10 – the initial configuration of the container bay is known in advance;
- A11 – under the RTG portal is a one-way direction truck lane;
- A12 – the container priorities are unique, there are no groups of containers with the same priority;
- A13 – the number of relocations is calculated only for the blocking containers; containers that are going to be retrieved directly are not taken into calculation.

The constraints of the model are:

- C1 – the maximum stacking height is defined according to the maximum retrieval height of the latest generation of RTG cranes minus the reserve height, which is necessary for container relocation and retrieval;
- C2 – the bay width is equivalent to the span of the latest generation of RTG cranes minus the width of one container, which serves as a one-direction truck lane;
- C3 – the given bay is never completely full because free slots are necessary for relocation;
- C4 – the relocation within the bay is limited (restricted variant) (Caserta *et al.* 2011); only the relocation of blocking containers is allowed, whilst the relocation of the other containers within the bay is restricted.

3. Rules for relocating containers

In the model the bay is given as a matrix where i represents the number of tiers (height of stacks), whilst j indicates the number of stacks in a bay (width) (Figure 2).

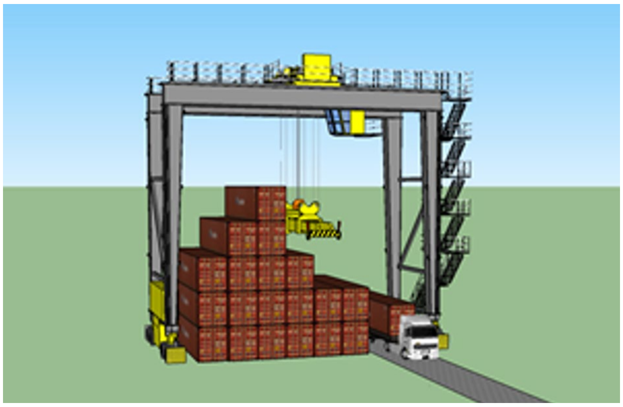


Figure 1. Yard container bay and a RTG crane

	j_0	j_1	j_2	j_3	j_4	j_5	j_6
i_0				9		8	
i_1	6		5	7		11	
i_2	10	2	1	3		4	

Figure 2. Yard bay with 3 tiers and 7 stacks

The stacks within the bay are numbered from left to right. The stack with the minimum index is marked with j_0. The maximum stacking height within the bay is marked with i_0. The container position C_p is determined with an ordered pair (i, j). The containers stacked within the bay have defined retrieval priorities P according to their retrieval times. The container marked with the number 1 has the highest priority, which means that it must be retrieved first while the container marked with the number 11 has to be the last for the retrieval. The goal is to retrieve all containers from the bay with the minimum number of relocations. In order to achieve this aforementioned optimization, we propose four rules for container relocation. These rules are applied only to the blocking containers. The fundamental purpose of the rules is the selection of the new position for each blocking container within the bay in order to reduce future relocations.

The first rule R1 is modified LPF rule proposed by Expósito-Izquierdo *et al.* (2014) and states – relocate the container to the nearest stack in which all containers will be retrieved later than the blocking container. This rule selects a stack that is nearest to the stack of the blocking container and has containers with lower priorities than the blocking container. In order to implement this rule, the following terms must be met:

- a container is located in the stack;
- there is a free position in the stack considering the maximum stacking height;
- the priority of the blocking container is higher than the priorities of all the containers that are currently located in the stack.

Thereafter, for all the stacks that represent a possible solution, the relocation distance by width (the distance between the stack in which the blocking container is located and a potential stack for container relocation) is calculated. The final solution is a destination stack with the smallest relocation distance by width. If there are several stacks with the same distance by width, then the stack with the smallest index number is chosen as the final solution. The application of rule R1 is shown in Figure 3.

In the first scenario a container marked with the priority $P = 1$ stacked on C_p (2, 2) (dark grey colour) must be retrieved. However, in order to retrieve this container, the container marked with the priority $P = 5$ stacked on C_p (1, 2) (light grey colour), which is the blocking container of the container with $P = 1$, must be relocated first. According to the rule R1, considering the terms that one stack has to satisfy to become a potential solution for the current relocation, only the first stack j_0 satisfies all the terms therefore the destination position of the blocking container with the priority $P = 5$ is the position C_p (0, 0).

In the second scenario, before the retrieval of the container marked with the priority $P = 1$ stacked on C_p (2, 2) (dark grey colour) the relocation of the blocking container marked with the priority $P = 3$ stacked on C_p (1, 2) (light grey) has to be performed. The possible stacks for stacking the blocking container are j_1, j_3 and j_4. Since stacks j_1 and j_3 have the identical distance by width as the destination position of the blocking container, a stack marked with the smaller index number i.e. stack j_1 is selected. Thus, the new position of the blocking container with the priority $P = 3$ is position C_p (0, 1).

The second rule R2 states – relocate the container to the nearest empty stack. For relocation this rule selects an empty stack, which is the nearest stack considering the distance between each empty stack and the stack of the blocking container. In order to implement this rule, the following term must be met:

- the potential stack for container relocation must be empty.

If there are several empty stacks, a stack that has a minimum relocation distance by width is chosen as the destination stack for the blocking container (Figure 4). In addition, if there are several stacks with equal minimum relocation distance by width, a stack with the lowest index number is chosen as the destination stack.

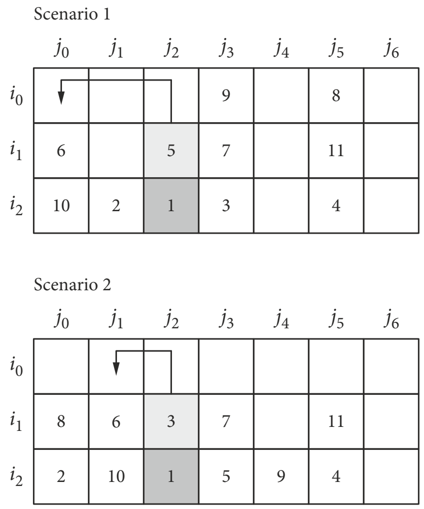
As can be seen in Figure 4, the blocking container marked with the priority $P = 5$ stacked on C_p (1, 2) (light grey colour) must be relocated before the retrieval of the container marked with the priority $P = 1$ stacked on C_p (2, 2) (dark grey colour). The position C_p (2, 1) within stack j_1 is chosen as the destination position for stacking the blocking container.

The third rule R3 is a modification of the rule called TLP rule, proposed by Zhang (2000) and states – relocate the container to the nearest and the lowest stack. This rule selects a stack with the smallest number of containers (without taking into consideration an empty stack), which is the nearest to the stack of the blocking container. In order to take into consideration one stack as the potential stack for container relocation according to rule R3, the following terms must be met:

- there is at least one container located in the stack;
- there is a free position in the stack considering the maximum stacking height.

In Figure 5 the example of applying rule R3 to the bay with size 3 × 7 can be seen. The container marked with the priority $P = 3$ stacked on C_p (1, 2) is the blocking container and must be relocated. After checking all the terms of R3, it is determined that stacks j_0 and j_4 represent possible solutions. Since both stacks have the same minimum relocation distance by the width, the position C_p (1, 0) within the stack j_0 is chosen as the destination position of the blocking container.

The fourth rule R4 states – relocate the container to the nearest stack that is not full to the maximum stacking height. This rule selects a stack that has at least one empty slot and is nearest to the stack of the blocking container (Figure 6).



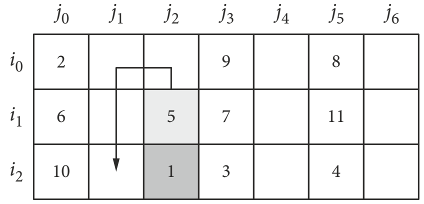
Scenario 1

	j_0	j_1	j_2	j_3	j_4	j_5	j_6
i_0				9		8	
i_1	6		5	7		11	
i_2	10	2	1	3		4	

Scenario 2

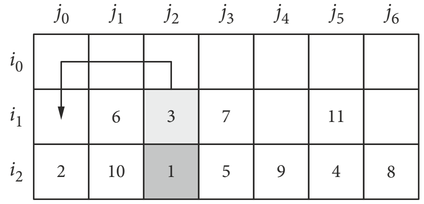
	j_0	j_1	j_2	j_3	j_4	j_5	j_6
i_0							
i_1	8	6	3	7		11	
i_2	2	10	1	5	9	4	

Figure 3. Two possible scenarios for the application of the rule R1 within the bay size 3 × 7



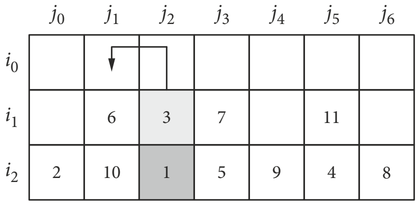
	j_0	j_1	j_2	j_3	j_4	j_5	j_6
i_0	2			9		8	
i_1	6		5	7		11	
i_2	10		1	3		4	

Figure 4. The application of the R2 within the bay size 3 × 7



	j_0	j_1	j_2	j_3	j_4	j_5	j_6
i_0							
i_1		6	3	7		11	
i_2	2	10	1	5	9	4	8

Figure 5. The application of the R3 within the bay size 3 × 7



	j_0	j_1	j_2	j_3	j_4	j_5	j_6
i_0							
i_1		6	3	7		11	
i_2	2	10	1	5	9	4	8

Figure 6. The application of the R4 within the bay size 3×7

Therefore, with this rule the stacks that represent possible solutions are not allowed to be empty nor full to the maximum stacking height. If the bay configuration shown in Figure 6 is considered then with the application of this rule the position C_p (0, 1) within the stack j_1 is chosen as the destination stack. Since the terms for the application of R4 enable the greatest number of allowable solutions, in some situations these solutions can correspond to the final solutions that are congruent to the application of R1 and R3.

It is necessary to emphasize that the aforementioned rules have equal importance and they include all possible positions in which a blocking container can be relocated, therefore there is no case that all the rules are inapplicable, at least one rule is applicable for the relocation of a blocking container within the given bay.

4. The model in the GA

In this paper, we apply a GA to solve the CRP. GA is a metaheuristic that simulates the process of natural selection. This method was first proposed by Holland (1992) and is often used to solve complex optimization problems. GA starts by creating the initial population of potential solutions (chromosomes). Subsequently, GA applies the genetic operators (selection, crossover and mutation) over the initial population in order to obtain a new and better population of solutions than the previous one. This process is called evolution. The evolution process is repeated as long as the term of stopping the process (maximum number of evolutions is reached; a satisfactory solution is obtained etc.) is not satisfied. In this section, the main components of GA as well as the procedure of applying the GA to solve the CRP are described in detail.

4.1. Chromosome

A chromosome consists of a group of genes and represents a potential solution of the CRP for the given bay. In this model the gene is an integer type and represents one of the four rules – R1…R4, which determine the stack within the bay in which the blocking container will be relocated. Each gene can have a value from 1 to 4. The code values of a gene are assigned in the following way: R1 = 1, R2 = 2, R3 = 3 and R4 = 4. For example, if the gene has the value 1, it means that rule 1 has to be executed. An example of a chromosome used for solving the CRP is shown in Figure 7. The first gene (value 2) determines that rule 2 has to be executed during the first container relocation within the bay.

2	1	3	3	1	4	4	2	3	Chromosome 1
3	2	3	1	1	2	4	2	3	Chromosome 2
4	1	4	3	1	4	4	1	3	Chromosome 3
1	1	3	4	1	3	4	2	3	Chromosome 4

Figure 8. Population of 4 chromosomes

The number of genes in a chromosome is determined by multiplying the number of containers stacked within the bay and the number 10. According to the executed experiments of our model, which are presented in the Section 5 this number of genes will ensure adequate number of rules for solving the CRP for the given bay. Not all the rules within a chromosome are used for container relocation. It may happen that the next heuristic rule that has to be executed according to the gene values within the chromosome is not applicable for the current relocation of certain container (e.g. rule R2 relocates the container to the nearest empty stack and there is no single empty stack currently available). Then, this inapplicable rule is skipped and the next applicable rule is executed. Therefore, the chromosome has to contain additional number of genes (rules), which will resolve this potential problem. The number of applied rules represents the number of container relocations within the bay i.e. the value of the fitness function. Fitness function is calculated as follows:

$$f_c = \sum R;$$
$$F = 100000 - f_c, \qquad (1)$$

where: f_c represents the value of fitness function of the chromosome; $\sum R$ represents a sum of container relocation within the bay before retrieval of target container; F represents a total fitness value of all relocated container within the bay.

4.2. Population

A population is a set of chromosomes (Figure 8). Each chromosome is evaluated with the fitness function defined within the GA. In our GA approach, the fitness function of each chromosome is determined according to the total number of relocations within the bay while resolving the current CRP with the sequence of rules defined in this chromosome.

$GENE_1$	$GENE_2$	$GENE_3$	$GENE_4$	$GENE_5$	$GENE_6$	$GENE_7$	$GENE_8$	$GENE_9$	$GENE_{10}$	...
2	1	2	3	4	2	4	3	3	2	...

Figure 7. Integer type chromosome

The initial population is always randomly generated. Thus, this generated population enables the diversity of genes (rules) in the chromosomes, i.e. the solutions of CRP. The determination of the population size is based on a trial and error method and the size is set to 8000 chromosomes.

4.3. Genetic operators

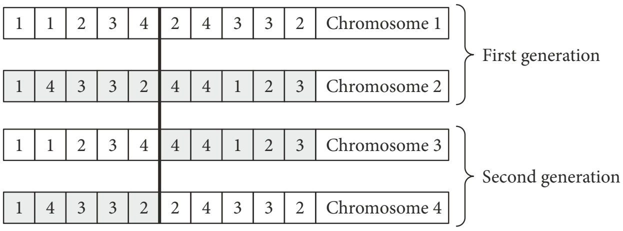
GA consists of three operators: selection, crossover and mutation.

Selection. This procedure selects a certain number of chromosomes in order to create a new generation (population of chromosomes). In our approach we implement the n selection of the best chromosomes, which is the simplest form of selection. The number of the fittest chromosomes that propagate to the next generation is set at 50% (4000 chromosomes) of the population size. The duplication of chromosomes, which appear in a generation, is also allowed.

Crossover. This procedure replaces the genes between two (parent) chromosomes according to a certain rule in order to produce new (children) chromosomes that have better value for the fitness function. A single-point crossover (Holland 1992) is used in our model. Thereafter, for each pair of chromosomes, one crossover point (crossover position) is randomly determined and the recombination of the genes within the pair of chromosomes is performed (Figure 9). After the recombination of genes is finished, the newly created (children) chromosomes are included in the following generation.

Mutation. This procedure randomly changes the value of one or several genes within the chromosome. In contrast to the crossover, which gravitates towards the preservation of the good characteristics of the chromosome, mutation gravitates towards the modification of the genetic material. The mutation used in this model is the simple mutation (Figure 10).

As shown in Figure 10 the mutation operator is applied on the fourth gene of the chromosome 1. Before the mutation, the code value of the fourth gene equals 3. After gene mutation, the code value equals 4. Considering Figure 10, it is obvious that the mutation is applied to only one gene in a way that the code value of the gene is altered. The described procedure is repeated for all the chromosomes in the population that are selected for the mutation.

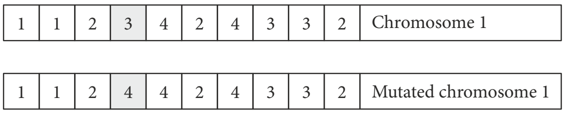

1 1 2 3 4 2 4 3 3 2 Chromosome 1

1 4 3 3 2 4 4 1 2 3 Chromosome 2

1 1 2 3 4 4 4 1 2 3 Chromosome 3

1 4 3 3 2 2 4 3 3 2 Chromosome 4

Figure 9. Single-point crossover


1 1 2 3 4 2 4 3 3 2 Chromosome 1

1 1 2 4 4 2 4 3 3 2 Mutated chromosome 1

Figure 10. Simple mutation of a chromosome

4.4. Parameters of GA

GA parameters are used for the control of the model and they influence the resolution time of the CRP. The fundamental parameters of the work of GA are the number of evolutions, the size of the population and the probability of the mutation (Mitchell 1998) (Table 1).

Table 1. Parameters of GA

Parameter	Value
Number of evolutions n_e	20
Size of population N_p	8000
Mutation probability p_m	0.01
Probability of crossover p_c	0.95

The number of evolutions n_e is a parameter that influences the quality of the solutions and the processing time of CRP (the time of running the model). A greater number of evolutions generally means a better solution, as well as a longer processing time. As the goal is to achieve the best solution in the shortest processing time, the maximum number of evolutions is set to 20. The size of the population N_p also affects the quality of the solutions; therefore, it is necessary to adjust the size of the population as well. In this model the size of the population is constant, which means that it does not change through new generations, and totals 8000 chromosomes. The mutation probability p_m determines the rate of the chromosomes mutation. This parameter extends the solution search space and prevents jamming in the local optimum and also leads to good solutions to the given problem. The mutation probability in the algorithm is set to 0.01.

The other parameters used in the GA are the probability of the crossovers and the length of the chromosome. The probability of crossovers p_c used in this algorithm is 0.95, which means that on 95% of the selected pairs of chromosomes the crossover will be executed. The length of the chromosome l_c is a parameter that is defined in advance and in this model it depends on the number of containers stacked within the given bay and total from 70 to 380 genes. In order to better understanding the principle of the model for solving CRP a flow chart of the model is shown (Figure 11).

As shown in Figure 11, at the beginning of the model the position and the priority of a target container is determined. If the target container is on the top of a stack, then the retrieval from the bay is performed, however, if there is some blocking container above the target container then a relocation is performed. The relocation is performed according to the sequence of genes (rules) in a randomly

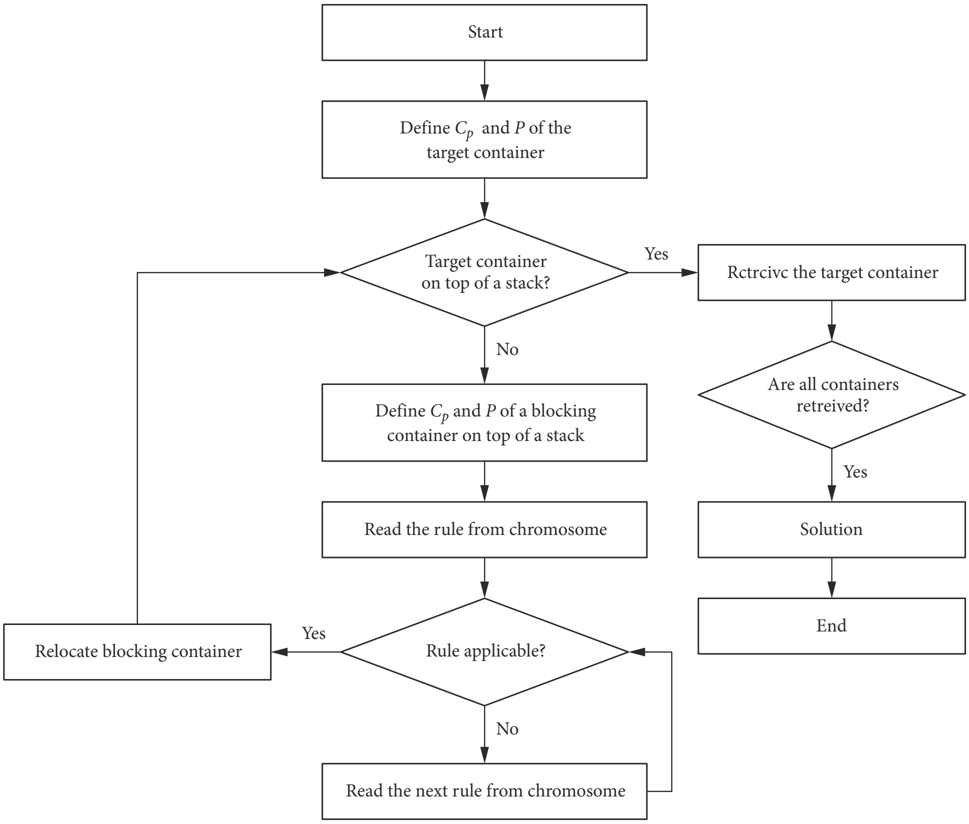

Start
Define C_p and P of the target container
Target container on top of a stack?
Yes
Rctrcivc the target container
No
Are all containers retreived?
Define C_p and P of a blocking container on top of a stack
Yes
Read the rule from chromosome
Solution
End
Relocate blocking container
Yes
Rule applicable?
No
Read the next rule from chromosome

Figure 11. Flow chart of the model

Table 2. Result presentation of the GA for a bay size 6×7

Number of model run	Number of containers	Sequence of applied rules	Fitness function
1	37	3 1 3 3 3 3 1 3 3 4 3 2 4 3 1 3 4 4 4 3 3 4 4 2 3 4 2 1	28.251
…	…	…	…
40	37	3 3 3 4 4 3 3 4 4 1 4 2 4 3 3 3 4 1 3 4 4 3 4 2 3 1	26.118
		Total:	33.192

generated chromosome. If one of the rules is not applicable, it proceeds to the next gene in the chromosome, i.e. to the following rule and so on until one of the rules is applied. When a rule is applied, the relocation of the blocking container is performed. When all the containers are retrieved from the bay the proposed model print the solution, which consist of the sequence of applied rules together with fitness function value as shown in Table 2.

5. Experiments, results and analysis

The model was implemented in *Java* using NetBeans IDE 7.4. The *Java* library JGAP is used for GA implementation within the model. The experimental testing of the model was performed with the constant size of a population, which totalling 8000 chromosomes and the constant number of evolutions set to 20 evolutions with a probability of crossover of 0.95 and a probability of mutation of 0.01.

The experiment was run on a 32-bit computer, *Windows 7 Professional* operating system with an *Intel® Core™* 2 Duo CPU T6500 2.10 GHz processor and 4 GB of RAM. The maximum tested size of the bay corresponds with the specifications of the latest technological RTG cranes in regard to the actual sizes of bays, and totals a maximum of 6 tiers and 7 stacks. The tiers parameter assumes a value of 3 to 6 whilst the stacks parameter assumes a value from 3 to 7.

The total number of containers stacked within the bay is obtained from the expression (Wu, Ting 2010):

$$N = W \cdot H - (H - 1), \tag{2}$$

where: N represents the total number of containers stacked within a bay; W represents the total number of stacks in a bay; H the total number of tiers in a bay; H – 1 represents the greatest number of relocations to retrieve one container.

This expression determines the maximum number of containers within a bay, i.e. it ensures a sufficient number of free container positions (slots) for container relocations within the bay and it was also used by the authors with whom the results of the proposed model are compared.

In order to compare the obtained results with the results of other authors, it was necessary to determine and adjust the other terms of the experiment. The experiment was run under the same terms as other authors performed it, and it refers to: the bay sizes (3×3 … 6×3, 3×4 … 6×4, 3×5 … 6×5, 3×6 … 6×6, 3×7 … 6×7), the number of containers obtained on the basis of the expression mentioned under (2) and 40 different problem instances. The initial configuration of containers within the bay is identical to all 40 instances as well as with the authors mentioned in Table 3. The test data used in comparison is taken from Jovanovic and Voβ (2014). According to Jovanovic and Voβ (2014) the same test data were used by Wu and Ting (2010). In Table 3 we present the average results for each bay size achieved by proposed model and other authors who achieved the most significant results in the resolution of CRP. The results for the approaches Zhang *et al.* (2010), Murty *et al.* (2005), Wu and Ting (2010) are taken from Wu and Ting (2010) and the results for the approaches Caserta *et al.* (2011), Jovanovic and Voβ (2014) are taken from Jovanovic and Voβ (2014).

The results of the authors Wu and Ting (2010) are known as being the best achieved results. The highlighted results (emboldened) of Wu and Ting (2010) represent the optimum results achieved by BS and the B&B algorithms.

Although the proposed model, along with the number of container relocations within the bay, calculates the relocation distance by the width and height this data is not compared with the authors in Table 3 since their models did not take it into consideration. In addition, the running time of algorithms is not taken into consideration for the same reason. Since the stacking areas of container port terminals that are equipped with RTG cranes are studied in this paper, the tested bay sizes are determined by the technical specifications of the latest generation of RTG cranes. Due to this fact, the proposed model was not tested on the so-called problem of large instances i.e. the unrealistic bay sizes.

On the basis of the results shown in Table 3, it can be concluded that the proposed model achieves the optimal solution only for a bay size of 3×3 and equals 3.38, whilst for the remaining bays it achieves a solution close to the optimal. In addition, it was noted that with the increase of the stacking height a lesser quality solution was achieved with all the tested bay widths considering the optimal solution. Based on the sum of the average number of container relocations, for all tested bay sizes proposed model achieves a better solution from the first six mentioned authors in the range from 0.8 to 23.9% i.e. it achieves from 2 to as many as 88 relocations less. The solution achieved by the proposed model represents the second-best results with a total of 280.66 average container relocations, which means that in comparison with the results of Wu and Ting (2010), with a total of 265.38 average relocations, it is worse by 5.4% i.e. approximately 15 additional container relocations were performed.

Table 3. The comparison of results of different authors and the heuristic methods for solving the CRP on real sizes of bays

Average relocations									
Tiers × stacks	No of containers	Zhang (2000) – TLP	Murty *et al.* (2005) – RI	Wu, Ting (2010) – RI	Caserta *et al.* (2009)– min–max	Jovanovic, Voß (2014) –chain	Jovanovic, Voß (2014) – chain F	Wu, Ting (2010) – BS, B&B	Authors – GA
3×3	7	3.58	3.42	3.40	3.42	3.42	3.38	3.38	**3.38**
4×3	9	6.67	6.10	5.95	5.82	5.82	5.95	5.67	5.85
5×3	11	10.60	9.80	9.45	9.10	9.10	8.70	8.40	8.93
6×3	13	15.40	13.60	13.20	12.97	12.77	12.30	11.50	12.30
3×4	10	5.67	5.03	4.92	4.95	4.95	4.95	4.85	4.98
4×4	13	10.50	9.05	8.80	8.75	8.72	8.57	8.43	8.55
5×4	16	16.30	14.50	13.70	13.12	13.02	13.17	12.20	12.63
6×4	19	23.20	19.10	17.90	17.15	17.05	16.92	15.60	16.93
3×5	13	6.95	5.90	5.75	5.75	5.75	5.80	**5.75**	5.80
4×5	17	14.40	12.20	11.80	11.40	11.35	11.40	**11.00**	11.35
5×5	21	21.00	18.10	17.40	17.07	16.95	16.65	**15.60**	16.75
6×5	25	31.80	25.60	24.10	23.92	23.52	23.57	21.10	22.60
3×6	16	8.95	7.92	7.80	7.72	7.72	7.85	**7.65**	7.88
4×6	21	16.00	13.20	12.80	12.67	12.55	12.60	**12.00**	12.48
5×6	26	26.90	22.60	21.40	20.60	20.42	20.52	19.30	20.32
6×6	31	41.20	32.60	30.10	29.02	28.77	28.57	26.10	28.50
3×7	19	11.50	10.10	9.88	9.02	9.05	9.17	**8.95**	9.15
4×7	25	19.40	20.10	18.80	16.35	16.45	16.07	15.50	16.13
5×7	31	33.00	30.90	28.30	23.15	22.92	22.70	21.40	22.60
6×7	37	45.80	45.00	41.30	34.40	34.07	34.07	31.00	33.55
Σ average relocations:		368.82	324.82	306.75	286.35	284.37	282.91	265.38	280.66

Conclusions

In this paper a GA was proposed for resolving the CRP at the stacking area of a port container terminal. The experimental testing of the performance of the proposed model was run on 20 various bay sizes, which were determined by the span and lifting height of the latest generation of RTG cranes. For each bay size forty different instances were generated according to the data set of authors mentioned in this paper, which means that the testing was run on a total of 800 different instances.

The obtained results show that the proposed model successfully minimizes the number of container relocations and achieves a near optimal solution. Based on the comparative analysis of the average number of relocations achieved by different authors, it can be concluded that the model achieves the second-best solution with regard to the known solutions of other significant authors in this field. Although the proposed model achieves near optimal solutions, its basic disadvantage is the running time of the algorithm, which depends on the bay size totalling from 15 s to 12 min. Therefore, the disadvantage is that this model cannot be implemented on real storage systems of a container terminal, since the model is applicable only for a 2D system (bay) in which only containers of a same size are stacked. Further research could proceed in several directions. The actual throughput performance of the system depends on many factors, such as technical specification of observed RTG crane, container position i.e. distance of movement, container weight, crane operator's skills, etc., where the time spent for each crane operation has a high influence. Hence, the first direction of the further research should be integrating a proposed CRP model along with the travel time analysis of a particular RTG crane. The second direction of the further research could be the testing of the proposed model on the large size instance problem. The third direction could be the extension of the model to the 3D problem i.e. to a container block, which represents a considerably more complicated problem. The fourth direction could encompass the various types and dimensions of containers at the stacking area of a terminal. The sixth direction could be to reduce the execution time of the algorithm adjusting the GA parameters differently.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Disclosure statement

The author declares that they have no relevant or material financial interests that relate to the research described in this paper.

References

Avriel, M.; Penn, M.; Shpirer, N.; Witteboon, S. 1998. Stowage planning for container ships to reduce the number of shifts, *Annals of Operations Research* 76: 55–71. https://doi.org/10.1023/A:1018956823693

Caserta, M.; Schwarze, S.; Voß, S. 2012. A mathematical formulation and complexity considerations for the blocks relocation problem, *European Journal of Operational Research* 219(1): 96–104. https://doi.org/10.1016/j.ejor.2011.12.039

Caserta, M.; Schwarze, S.; Voß, S. 2009. A new binary description of the blocks relocation problem and benefits in a look ahead heuristic, *Lecture Notes in Computer Science* 5482: 37–48. https://doi.org/10.1007/978-3-642-01009-5_4

Caserta, M.; Voß, S. 2009. A corridor method-based algorithm for the pre-marshalling problem, *Lecture Notes in Computer Science* 5484: 788–797. https://doi.org/10.1007/978-3-642-01129-0_89

Caserta, M.; Voß, S.; Sniedovich, M. 2011. Applying the corridor method to a blocks relocation problem, *OR Spectrum* 33(4): 915–929. https://doi.org/10.1007/s00291-009-0176-5

Da Silva, M. M.; Erdoğan, G.; Battarra, M.; Strusevich, V. 2018. The block retrieval problem, *European Journal of Operational Research* 265(3): 931–950. https://doi.org/10.1016/j.ejor.2017.08.048

De Castillo, B.; Daganzo, C. F. 1993. Handling strategies for import containers at marine terminals, *Transportation Research Part B: Methodological* 27(2): 151–166. https://doi.org/10.1016/0191-2615(93)90005-U

De Koster, R. B. M.; Le-Duc, T.; Yugang, Y. 2008. Optimal storage rack design for a 3-dimensional compact AS/RS, *International Journal of Production Research* 46(6): 1495–1514. https://doi.org/10.1080/00207540600957795

Expósito-Izquierdo, C.; Melián-Batista, B.; Moreno-Vega, J. M. 2014. A domain-specific knowledge-based heuristic for the blocks relocation problem, *Advanced Engineering Informatics* 28(4): 327–343. https://doi.org/10.1016/j.aei.2014.03.003

Forster, F.; Bortfeldt, A. 2012. A tree search procedure for the container relocation problem, *Computers & Operations Research* 39(2): 299–309. https://doi.org/10.1016/j.cor.2011.04.004

Ghomri, L.; Sari, Z.; Guezzen, A.; Sari, T. 2009. Continuous models for single and dual cycle times of a multi aisle automated storage and retrieval system, *IFAC Proceedings Volumes* 42(4): 1061–1066. https://doi.org/10.3182/20090603-3-ru-2001.0144

Gupta, N.; Nau, D. S. 1992. On the complexity of blocks-world planning, *Artificial Intelligence* 56(2–3): 223–254. https://doi.org/10.1016/0004-3702(92)90028-V

Hachemi, K.; Alla, H. 2008. Pilotage dynamique d'un système automatisé de stockage/déstockage à convoyeur gravitationnel, *Journal Européen des Systèmes Automatisés* 42(5): 487–508. (in French).

Hamzaoui, M. A.; Sari, Z. 2015. Optimal dimensions minimizing expected travel time of a single machine flow rack AS/RS, *Mechatronics* 31: 158–168. https://doi.org/10.1016/j.mechatronics.2014.10.006

Holland, J. H. 1992. *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press. 211 p. https://doi.org/10.7551/mitpress/1090.001.0001

Hu, Y.-H.; Huang, S. Y.; Chen, C.; Hsu, W.-J.; Toh, A. C.; Loh, C. K.; Song, T. 2005. Travel time analysis of a new automated storage and retrieval system, *Computers & Operations Research* 32(6): 1515–1544. https://doi.org/10.1016/j.cor.2003.11.020

Hussein, M.; Petering, M. E. H. 2012. Genetic algorithm-based simulation optimization of stacking algorithms for yard cranes to reduce fuel consumption at seaport container transshipment terminals, in *2012 IEEE Congress on Evolutionary Computation*, 10–15 June 2012, Brisbane, Australia, 1–8. https://doi.org/10.1109/CEC.2012.6256471

Jin, B.; Zhu, W.; Lim, A. 2015. Solving the container relocation problem by an improved greedy look-ahead heuristic, *European Journal of Operational Research* 240(3): 837–847. https://doi.org/10.1016/j.ejor.2014.07.038

Jovanovic, R.; Voß, S. 2014. A chain heuristic for the blocks relocation problem, *Computers & Industrial Engineering* 75: 79–86. https://doi.org/10.1016/j.cie.2014.06.010

Kim, K. H. 1997. Evaluation of the number of rehandles in container yards, *Computers & Industrial Engineering* 32(4): 701–711. https://doi.org/10.1016/S0360-8352(97)00024-7

Kim, K. H.; Hong, G.-P. 2006. A heuristic rule for relocating blocks, *Computers & Operations Research* 33(4): 940–954. https://doi.org/10.1016/j.cor.2004.08.005

Kim, K. H.; Kim, H. B. 1999. Segregating space allocation models for container inventories in port container terminals, *International Journal of Production Economics* 59(1–3): 415–423. https://doi.org/10.1016/S0925-5273(98)00028-0

Ku, D.; Arthanari, T. S. 2016. On the abstraction method for the container relocation problem, *Computers & Operations Research* 68: 110–122. https://doi.org/10.1016/j.cor.2015.11.006

Lee, Y.; Hsu, N.-Y. 2007. An optimization model for the container pre-marshalling problem, *Computers & Operations Research* 34(11): 3295–3313. https://doi.org/10.1016/j.cor.2005.12.006

Lerher, T.; Potrč, I.; Šraml, M.; Tollazzi, T. 2010. Travel time models for automated warehouses with aisle transferring storage and retrieval machine, *European Journal of Operational Research* 205(3): 571–583. https://doi.org/10.1016/j.ejor.2010.01.025

Liu, T.; Xu, X.; Qin, H.; Lim, A. 2016. Travel time analysis of the dual command cycle in the split-platform AS/RS with I/O dwell point policy, *Flexible Services and Manufacturing Journal* 28(3): 442–460. https://doi.org/10.1007/s10696-015-9221-7

Malmborg, C. J. 2002. Conceptualizing tools for autonomous vehicle storage and retrieval systems, *International Journal of Production Research* 40(8): 1807–1822. https://doi.org/10.1080/00207540110118668

Mitchell, M. 1998. *An Introduction to Genetic Algorithms*. MIT Press. 221 p.

Murty, K. G.; Wan, Y.-W.; Liu, J.; Tseng, M. M.; Leung, E.; Lai, K.-K.; Chiu, H. W. C. 2005. Hongkong international terminals gains elastic capacity using a data-intensive decision-support system, *Interfaces* 35(1): 61–75. https://doi.org/10.1287/inte.1040.0120

Petering, M. E. H.; Hussein, M. I. 2013. A new mixed integer program and extended look-ahead heuristic algorithm for the block relocation problem, *European Journal of Operational Research* 231(1): 120–130. https://doi.org/10.1016/j.ejor.2013.05.037

Rei, R.; Pedroso, J. P. 2013. Tree search for the stacking problem, *Annals of Operations Research* 203(1): 371–388. https://doi.org/10.1007/s10479-012-1186-2

Sari, Z.; Saygin, C.; Ghouali, N. 2005. Travel-time models for flow-rack automated storage and retrieval systems, *The International Journal of Advanced Manufacturing Technology* 25(9–10): 979–987. https://doi.org/10.1007/s00170-003-1932-3

Sari, Z. 2010. Performance evaluation of flow-rack and unit load automated storage and retrieval systems, in *International Science and Technology Conference 2010* (*ISTEC 2010*)*: Proceedings Book*, 27–29 October 2010, Cyprus, 605–616.

Sculli, D.; Hui, C. F. 1988. Three dimensional stacking of containers, *Omega* 16(6): 585–594. https://doi.org/10.1016/0305-0483(88)90032-1

Steenken, D.; Voß, S.; Stahlbock, R. 2004. Container terminal operation and operations research – a classification and literature review, *OR Spectrum* 26(1): 3–49. https://doi.org/10.1007/s00291-003-0157-z

Tanaka, S.; Mizuno, F. 2018. An exact algorithm for the unrestricted block relocation problem, *Computers & Operations Research* 95: 12–31. https://doi.org/10.1016/j.cor.2018.02.019

Tanaka, S.; Takii, K. 2016. A faster branch-and-bound algorithm for the block relocation problem, *IEEE Transactions on Automation Science and Engineering* 13(1): 181–190. https://doi.org/10.1109/TASE.2015.2434417

Tang, L.; Zhao, R.; Liu, J. 2012. Models and algorithms for shuffling problems in steel plants, *Naval Research Logistics* 59(7): 502–524. https://doi.org/10.1002/nav.21503

Tricoire, F.; Scagnetti, J.; Beham, A. 2018. New insights on the block relocation problem, *Computers & Operations Research* 89: 127–139. https://doi.org/10.1016/j.cor.2017.08.010

Ünlüyurt, T.; Aydın, C. 2012. Improved rehandling strategies for the container retrieval process, *Journal of Advanced Transportation* 46(4): 378–393. https://doi.org/10.1002/atr.1193

Van den Berg, J. P. 2002. Analytic expressions for the optimal dwell point in an automated storage/retrieval system, *International Journal of Production Economics* 76(1): 13–25. https://doi.org/10.1016/S0925-5273(01)00149-9

Wan, Y.-W.; Liu, J.; Tsai, P.-C. 2009. The assignment of storage locations to containers for a container stack, *Naval Research Logistics* 56(8): 699–713. https://doi.org/10.1002/nav.20373

Wu, K.-C.; Ting, C.-J. 2010. A beam search algorithm for minimizing reshuffle operations at container yards, in *LOGMS 2010: the 1st International Conference on Logistics and Maritime Systems*, 15–17 September 2010, Busan, Korea, 703–710.

Wu, K.-C.; Ting, C.-J.; Hernández, R. 2009. Appling tabu search for minimizing reshuffle operations at container yards, *Proceedings of the Eastern Asia Society for Transportation Studies* 7: 435–449. https://doi.org/10.11175/eastpro.2009.0.435.0

Xu, X.; Gong, Y.; Fan, X.; Shen, G.; Zou, B. 2018. Travel-time model of dual-command cycles in a 3D compact AS/RS with lower mid-point I/O dwell point policy, *International Journal of Production Research* 56(4): 1620–1641. https://doi.org/10.1080/00207543.2017.1361049

Yang, J. H.; Kim, K. H. 2006. A grouped storage method for minimizing relocations in block stacking systems, *Journal of Intelligent Manufacturing* 17(4): 453–463. https://doi.org/10.1007/s10845-005-0018-5

Zehendner, E.; Caserta, M.; Feillet, D.; Schwarze, S.; Voß, S. 2015. An improved mathematical formulation for the blocks relocation problem, *European Journal of Operational Research* 245(2): 415–422. https://doi.org/10.1016/j.ejor.2015.03.032

Zhang, C. 2000. *Resource Planning in Container Storage Yard*. PhD Thesis. The Hong Kong University of Science and Technology, Hong Kong. 342 p.

Zhang, H.; Guo, S.; Zhu, W.; Lim, A.; Cheang, B. 2010. An investigation of IDA* algorithms for the container relocation problem, *Lecture Notes in Computer Science* 6096: 31–40. https://doi.org/10.1007/978-3-642-13022-9_4

Zhao, W.; Goodchild, A. V. 2010. The impact of truck arrival information on container terminal rehandling, *Transportation Research Part E: Logistics and Transportation Review* 46(3): 327–343. https://doi.org/10.1016/j.tre.2009.11.007

Zhu, W.; Qin, H.; Lim, A.; Zhang, H. 2012. Iterative deepening A* algorithms for the container relocation problem, *IEEE Transactions on Automation Science and Engineering* 9(4): 710–722. https://doi.org/10.1109/TASE.2012.2198642