

Research Article

A Heuristic Procedure for the Outbound Container Relocation Problem during Export Loading Operations

Roberto Guerra-Olivares,¹ Rosa G. González-Ramírez,² and Neale R. Smith¹

¹Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey, Avenida Eugenio Garza Sada 2501, 64849 Monterrey, NL, Mexico

²Escuela de Ingeniería Industrial, Pontificia Universidad Católica de Valparaíso, Avenida Brasil 2950, Casilla 4059, 2374631 Valparaíso, Chile

Correspondence should be addressed to Neale R. Smith; nsmith@itesm.mx

Received 3 December 2014; Revised 22 February 2015; Accepted 22 February 2015

Academic Editor: Seungik Baek

Copyright © 2015 Roberto Guerra-Olivares et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

During export ship loading operations, it is often necessary to perform relocation movements with containers that interfere with access to the desired container in the ship loading sequence. This paper presents a real-time heuristic procedure for the container relocation problem employing reachstacker vehicles as container handling equipment. The proposed heuristic searches for good relocation coordinates within a set of nearby bays. The heuristic has a parameter that determines how far from the original bay a container may be relocated. The tradeoff between reducing relocation movements and limiting vehicle travel distances is examined and the performance of the heuristic is compared with a common practice in the smaller container terminals in Chile and Mexico. Finally, a mathematical model for the container relocation problem is presented.

1. Introduction and Literature Review

Maritime terminals are facilities with a constant need to plan space to receive ships in the quay and to store their corresponding inbound and outbound containers in the port yard. Outbound containers arrive to the port and are temporarily stored in the yard until their ship arrives and berths at the quay. Efficient space planning yields shorter ship turnaround times in the quay, avoiding delays in the ship route.

A port can be divided into three main areas: gate, yard, and quay. The gate is the interface of the port with external trucks, which claim inbound containers and deliver outbound containers. The yard is a container storage area and acts as a buffer to absorb the difference in arrival times of the external trucks and the ship. The quay is the interface of the port with ships and can be organized in several berthing sites. A container can be categorized as inbound if it arrives to the port on a ship and is requested by an external truck or as outbound if it arrives to the port in an external truck and departs on a ship. Containers can be classified into families of containers or segregations having similar attributes such

as length, being inbound or outbound, being full or empty, associated ship, port of destination, and weight.

The port yard is organized into a three-dimensional system known as Baroti. The Baroti system defines the yard as being divided in blocks, and each block is subdivided in bays. Then, each bay is organized in rows, with each row having a defined maximum number of tiers or stack height. Figure 1 shows an illustration of this coordinate system.

A common approach found in the literature to address the storage space allocation problem is by splitting it into two stages. The first stage typically employs a mathematical model to determine in what bays to store each container, usually based on containers having the same ship and port of destination. The second stage is a heuristic procedure to assign a storage location to each individual container in the bay determined by the model in the previous stage. The first stage of this approach is useful for the tactical level decisions while the second stage makes decisions at an operational level. Tapia et al. [1] propose a mathematical model to solve the first stage of the container space allocation problem and determine a set of bays to store each container.

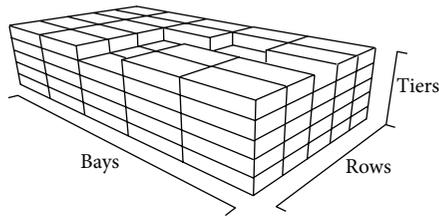


FIGURE 1: Illustration of the Baroti system.

The shipping company sends to the port a list with all outbound containers that have booked a space in the ship. The list includes container attributes such as length, port of destination, and weight. This information is input for the first stage of the approach to address the problem. By the time the outbound containers start to arrive at the gate, the port knows the total number of containers of each segregation that will arrive and in what bays they should be stored. In practice, hub ports assign time windows to receive outbound containers, so that the arrival sequence can be estimated; but that is not the case in other smaller ports. The space assignment policy deals with an unknown container arrival sequence and seeks to assign good available Baroti locations in real time as the containers arrive at the port.

The storage space allocation problem (SSAP) focuses on making decisions about the first Baroti coordinate assigned to outbound containers when they arrive at the terminal. During the loading operation, it is often the case that access to the next required container is obstructed by another container or by several containers. These obstructing or interfering containers must be removed and relocated to gain access to the next required container. Thus, a relevant decision is the relocation of interfering containers during the loading operation onto the ship. When a relocation movement is performed, a new Baroti coordinate is determined for the interfering container. This relocation problem is known in the literature as the block relocation problem (BRP). A related but different problem is the premarshalling problem, which involves moving containers to locations that allow them to be loaded on a ship directly in the stowage sequence. The premarshalling operation is usually performed before the loading operations are initiated, not in real time, as addressed herein. The premarshalling literature will not be reviewed here but the interested reader may wish to see the work by Bortfeldt and Forster [2] as an example. In the following paragraphs, a literature review is presented, considering first the SSAP contributions and then those to the BRP.

For an overview of port storage yard operations, see the paper by Carlo et al. [3] who provide a literature overview, trends, and research directions. They distinguish between the following main decision problems that arise in the storage yard operations: (1) yard design, (2) storage space assignment for containers, (3) material handling equipment dispatching and routing to serve the storage and retrieval requests, and (4) optimizing the relocation (or reshuffling) of containers.

The storage space allocation problem is addressed in the literature by several researchers. For example, K. H. Kim and H. B. Kim [4] propose a method for minimizing

the number of relocation movements under the constraint of satisfying estimated space requirements. An algorithm based on Lagrangian relaxation is provided which achieves near-to-optimal solutions. Considering the productivity of yard equipment, Young Kim and Hwan Kim [5] study the problem of routing a single straddle carrier to pick up outbound containers. They formulate an integer programming model which minimizes the total distance traveled by the straddle carrier within the yard and determines both the bay visiting sequence and the number of containers that should be picked up from each bay.

Since, for some ports, the container weight information available before container arrival is only an estimate, Kang et al. [6] present a method for deriving a strategy for stacking containers with uncertain weight information. They implemented a simulated annealing algorithm which can find a good stacking strategy in a reasonable time. Simulation experiments show that their proposed strategy reduces the number of relocation movements compared to the traditional same-weight-group stacking strategy.

Lee et al. [7] propose a simulated annealing algorithm to schedule two Rubber-Tired-Gantry cranes for loading outbound containers. A strategy for reserving space for outbound containers is analyzed by Woo and Kim [8] who suggest a series of heuristic rules to reserve a collection of adjacent stacks for each group of containers with the same attributes.

Park et al. [9] develop an online search algorithm to optimize the stacking policy in an automated terminal and introduce a set of criteria that must be considered for determining a good stacking position for each incoming container. Their algorithm continuously tries and evaluates variants of the best-so-far policy during the real operation of the terminal. Simulation experiments show that the proposed algorithm is effective in reducing the quay crane delay.

Chen and Lu [10] study the storage location assignment problem for outbound containers by splitting the problem into two stages. The first stage determines the yard bays and the number of locations in each yard bay which will be assigned to containers associated with different ships. The second stage determines the exact storage location for each container. The first stage is formulated as a lineal integer programming model that minimizes the workload imbalance in each container block of the yard and the total distance traveled by internal trucks. The second stage is a heuristic procedure which decides the exact location for each individual container in the preassigned yard bays determined by the first stage upon its arrival at the terminal. The objective is to minimize the total number of relocation movements during the loading operation. The authors propose three stacking strategies, namely, diagonal, vertical, and horizontal, and recommend the diagonal strategy because it generated the least relocation movements under their testing conditions.

Petering [11] presents several real-time container storage systems and evaluates them using a discrete-event simulation model. The author analyzes realistic scenarios and simulation experiments show that the systems can be applied in maritime terminals.

More recently, Wang et al. [12] propose a two-stage optimization model to solve the storage space allocation problem for inbound containers in railway container terminal. The objectives of the models are balancing the workload of inbound containers blocks and reducing the overlapping amounts. They solve the models employing a rolling horizon approach. The authors assume that the arrival and departure times of containers are known in advance and that RTG cranes are available to handle containers.

The BRP is addressed by several researchers. For example, Yang and Kim [13] propose a mathematical model which minimizes relocation movements in block stacking systems. It is important to point out that in [13] the generic term block is used by the authors to refer indistinctly to a container, a box, or a pallet. They analyze a static case that considers known arrival and departure times of the storage demand units and a dynamic case, in which the arrival and departure times are not known. The authors develop a genetic algorithm to find good solutions for the static case and suggest simple heuristic rules to solve the dynamic case.

The first linear integer programming model for the static container relocation problem was successfully formulated by Wan et al. [14] and is referred to as MRIP (minimize relocation integer program). By design, the MRIP retrieves all containers in a stack from a given initial configuration with the minimum number of relocation movements. The authors also extended some heuristic algorithms used in the literature and in practice in order to consider further stack configurations resulting from storing or relocating the interfering containers to various feasible slots. Later, Lixin et al. [15] improve the static version of the MRIP model formulated by Wan et al. [14] and propose five heuristics to solve it. They develop a discrete-event simulation model to test the performance of the proposed heuristics. The authors include an analysis of the worst case performance of the heuristics. The experimental results show that the improved model can obtain optimal or feasible solutions faster than the model formulated by Wan et al. [14]. The contribution of the heuristic proposed herein can be adjusted to find solutions with different amounts of total distance traveled by internal trucks. This is done depending on the congestion level at the yard. For a case in which there is a high level of congestion in the yard, the heuristic can be adjusted to find solutions in which the interfering container will be relocated only to nearby bays to reduce the distance traveled by internal trucks. On the other hand, in a low congestion season, the interfering container can be relocated to nearby or more distant positions which may result in the minimization of additional relocations.

A branch and bound algorithm to minimize the number of relocation movements during the pickup operation in block-stacking warehouses is suggested by Kim and Hong [16]. They also propose a decision rule based on probability theory and compare the performance of the decision rule with the performance of the branch and bound algorithm. The comparison shows that the total number of relocation movements calculated by the decision rule exceeded that found by branch and bound by 7.3% on average and the computational time of the heuristic rule is within the level

that can be used in real time. The authors consider some precedence relationships among the pickups of the blocks in warehouses. However, it may not always be possible to apply this strategy in terminal ports due to the constraint of maintaining the ship's physical balance in the water.

Caserta et al. [17] suggest a binary description encoding for stacking areas where homogeneous blocks are stored on stacks. It is important to point out that in [17, 18] the generic term "block" is used by the authors to refer indistinctly to a container, a box, or a pallet. Fast access to data on the current stacking area and an efficient transformation into neighboring states is required for it. They state a base for developing metaheuristic search strategies by making a randomly (roulette-wheel) guided look ahead mechanism.

Caserta et al. [18] present a metaheuristic approach for the block relocation problem (BRP). The authors indicate that this algorithm may be applied not only for stacking containers in port terminals but also for stacking boxes or pallets in warehouses. The objective is to find the block relocation pattern that minimizes the total number of movements required to comply with a given retrieving sequence. The authors consider that the blocks are handled by equipment with the ability to reach the position located at the top of any stack, which is consistent with RTG cranes. In this paper, we propose an alternative method to solve this problem employing reachstacker vehicles.

Borgman et al. [19] analyze two concepts using a discrete-event simulation tool. The first concept is to assume container departure times as known in order to limit the number of relocation movements. They stack containers leaving shortly on the top of the stacks. The second concept is the trade-off between stacking further away in the terminal and stacking close to the berth sites and accepting more relocation movements. They have employed data from real situations to generate scenarios of container movements for a mechanized container terminal by means of a simulation model. The difference between the approach of Borgman et al. [19] and the heuristic approach presented herein is that our procedure does not assume the container departure times as known, allowing more flexibility to unexpected changes in the stowage sequence due to ship imbalance.

More recently, Gharehgozli et al. [20] develop a decision-tree heuristic and a dynamic programming model focusing on minimizing the expected number of relocations when arriving containers should be stacked in a block. For small-scale problems with a small number of piles, they compare the decision-tree heuristic with the dynamic programming model results and they use the dynamic programming model to solve large-scale problems. They contrast the performance of a shared stacking policy with a dedicated stacking policy. The shared stacking policy allows containers of multiple ships to be stacked on top of each other. Their heuristic procedure solves simplified instances with a small number of ships, each with a single port of destination.

Ries et al. [21] address the storage space allocation problem and the block relocation problem in two-stage framework in combination with a fuzzy logic rule-based strategy. The framework proposed has the aim of providing real-time decision support to deal with the uncertain arrival

sequence of containers to the yard that is operated with RTGs. In contrast with the paper of Ries et al. [21], where the authors propose a fuzzy logic rule, we propose a heuristic approach to solve the problem. Another difference is the employment of reachstacker vehicles to handle containers in the heuristic presented herein.

In contrast to the previous literature, which considers Rubber-Tired-Gantry (RTG) cranes for container handling in the port yard, this paper provides a solution to the container relocation problem employing reachstacker vehicles, opening a new line of research. A reachstacker vehicle has a more limited slot access compared to an RTG crane. While an RTG crane can access any slot located at the top of any stack of the bay, the reachstacker vehicle can access only the top slot of the stack located at the end of the bay. Reachstacker vehicles are the principal container handling equipment in many Latin American ports and in ports with low cargo volume. The procedures available in the existing literature may not be applicable in ports that employ only reachstacker vehicles.

It is relevant for this paper to define two types of containers. The desired container is the container that is required to be loaded to the ship according to a stowage plan. The interfering container is the container that needs to be relocated in order to gain access to the desired container. A desired container may have one or more interfering containers.

It is desirable that each outbound container remain stored in one single Baroti location during its stay in the yard. A relocation movement is performed when it is necessary to move a container that is blocking access to another container that needs to be retrieved. This study considers that all the outbound containers have been assigned to a Baroti location for storage when they arrive to the yard according to a port operation policy. When outbound containers are loaded onto the ship, some relocation movements could be required in order to gain access to a container stacked below other containers. This paper proposes a heuristic which suggests Baroti locations for the interfering containers, so that further relocation movements are minimized.

The main contribution of this paper is a real-time heuristic procedure that determines the new Baroti coordinates for relocated containers when relocation movements are performed, assuming that the containers are handled with reachstacker vehicles and the bays are accessed from only one side. It is also important to mention that this procedure incorporates a parameter which defines the proximity of the new Baroti location with respect to the original location. This parameter allows the heuristic to be tuned to adapt the solutions to varying congestion levels in the port yard. The total number of relocation movements resulting using this strategy is compared with the strategy used in real practice in some Latin American ports.

2. Problem Description

Export containers arrive at the port several days before the estimated time of arrival of the ship and during that interval of time they are stored in the yard. Figure 2 shows a typical timeline for the container reception period. Observe that the period ends 24 hours prior to the estimated time of arrival

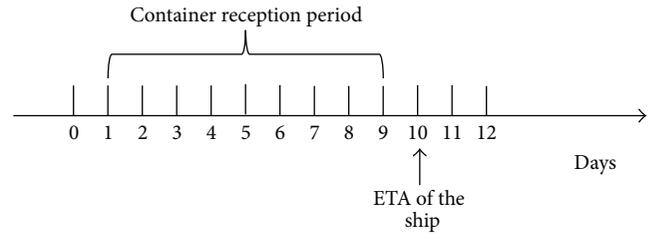


FIGURE 2: Timeline for the container reception period and the ETA of the ship.

(ETA) of the ship. A container may be received in the yard after the container reception period has ended but will incur in a penalty for the late arrival.

The stowage plan is the sequence to load containers onto a ship and is defined in a hierarchical structure. For instance, for a given ship, containers with the farthest port of destination are loaded before containers with the nearest port of destination. Furthermore, for each port of destination, heavier containers are loaded before lighter ones.

The actual container loading sequence onto the ship may be different from the stowage plan described above because it is mandatory that the distribution of the weight of the containers on the ship remain uniformly distributed. When the distribution of the weight on the ship becomes unbalanced, some containers may need to be loaded as counterweight not necessarily in the sequence determined in the stowage plan. Such deviations from the original stowage plan may require relocation movements in order to gain access to required containers in the modified stowage sequence.

During the container loading operation, outbound containers are retrieved from the yard by a reachstacker vehicle and loaded into an internal truck to be transported to the quay where the ship is berthed for the transfer of cargo. The operation of a reachstacker vehicle requires a driver and a tracker operator. The driver operates the reachstacker vehicle and the tracker operator manipulates a container tracking device to indicate to the driver the location of the container to be retrieved. When relocation movements are required, the tracker operator introduces the new Baroti coordinate of the interfering containers in the tracking device.

Lack of information of potential good Baroti locations for relocating interfering containers may lead to an increase in the number of further relocation movements when an interfering and relocated container obstructs the retrieval of another container in the new Baroti location. In this paper, we propose a heuristic procedure that determines the new Baroti coordinates of relocated containers. In order to implement this heuristic, the function of the container tracking device would be expanded to call the heuristic to suggest Baroti coordinates for interfering containers when they are relocated. The tracker operator could type the id of the interfering container into the tracking device. The required information would then be provided to the heuristic, which would then suggest a Baroti coordinate. This coordinate would then be transferred back to the tracker operator through the tracking

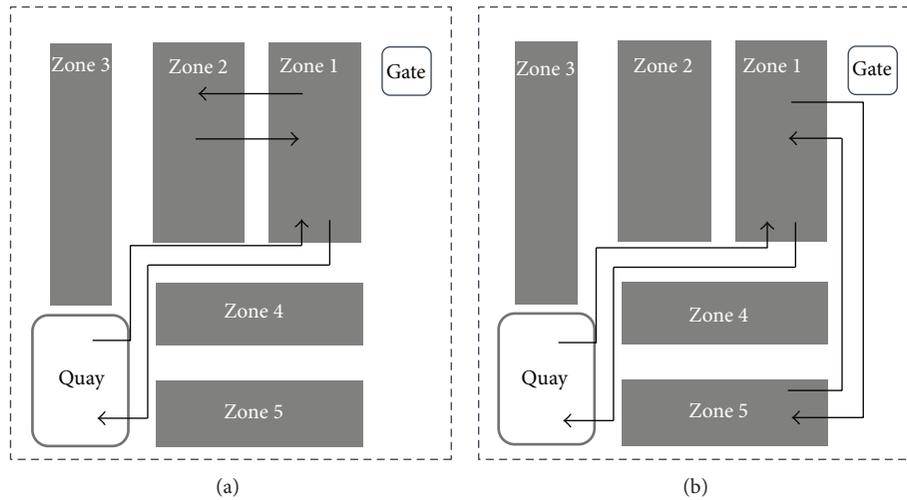


FIGURE 3: Layout of the export yard section and illustration of two possible routes for internal trucks.

device, allowing the reachstacker driver to be informed where to place the interfering container.

The strategy employed in practice in some ports is to place temporarily the interfering containers in the aisle near the retrieving bay (based on in-site interviews with yard managers at some container terminals in Chile and Mexico). Once the desired container is retrieved, the interfering containers are returned to the original bay in the opposite sequence as they were removed. The heuristic proposed in this paper suggests Baroti coordinates in other bays for interfering containers instead of placing them in the aisle while the desired container is retrieved. In this way, the number of movements per container is minimized.

The advantage of storing the interfering containers in the aisle close to the original bay is that total distance traveled by internal trucks is minimized. However, the total number of relocation movements may be greater because a total of two relocation movements are required for each interfering container, one for retrieving the interfering container from the bay to situate it in the aisle and another one to return it to the original bay. On the other hand, if interfering containers are stored in distant bays, the distance traveled by internal trucks may increase but the relocation movements may decrease significantly as the container can be placed in a better position and the movement in which the container is placed in the aisle is eliminated.

The port yard managers should define a strategy to employ according to the storage capacity and congestion levels of the port terminal. In a high congestion level season, it may be advantageous to employ a strategy which stores interfering containers in nearby or adjacent bays. Conversely, in a season with a low or moderate congestion level, interfering containers may be relocated into more distant bays, potentially reducing the number of relocation movements. In order to illustrate this tradeoff, Figure 3 shows an example of a container terminal export yard.

In the example depicted in Figure 3, let us assume that the internal truck needs to retrieve a container from a bay in

Zone 1. An interfering container may be relocated to Zone 2 (see Figure 3(a)) or to Zone 5 (see Figure 3(b)), after which the truck will return to Zone 1 for the desired container to transport it to the quay. Let us further assume that the relocation to Zone 2 results in fewer later relocations than the relocation to Zone 5. During a season with a high congestion level, the relocation to Zone 2 may be preferred since it results in shorter distances and lower travel times even though more relocations will be required. In contrast, during a season with a low congestion level, the relocation to Zone 5 may be preferred since fewer relocations will be required and the low congestion level implies that the longer travel distance to Zone 5 will not result in much longer travel times.

This paper proposes a real-time heuristic procedure to determine new Baroti coordinates to interfering containers when relocation movements are performed during loading operation, taking into consideration the congestion level of the yard. The criteria employed vary according to the congestion level of the yard that should be indicated by the yard manager.

3. Mathematical Model for the Container Relocation Problem

A mathematical model to determine a new Baroti coordinate for an interfering container is presented in this section. This model receives as input the initial inventory of containers in the yard, including their weights and positions, and the weight of the interfering container.

The assumptions of the model are the following:

- (1) The interfering container and the containers stored in the yard have the same length, either 20 or 40 ft.
- (2) This model is executed each time a relocation movement is performed to define the new Baroti coordinate to the interfering container.
- (3) This model avoids storing the interfering container in an empty bay when it is possible to store it in

another bay obtaining the same number of relocation movements.

- (4) Port yard uses a reachstacker vehicle to handle containers.
- (5) Containers may only be placed at ground level or resting on top of other containers (i.e., containers are not supported by a rack).
- (6) Heavier containers are loaded to the ship before lighter ones. Hence, the weight categories of the containers imply their loading sequence.
- (7) Containers are classified into 5 categories according to their weight. Category number 1 is the lightest and number 5 the heaviest.
- (8) In each yard bay, a container cannot be assigned to a given stack until the stack behind has been completely filled. This assumption results in good space utilization in the bay because it avoids empty slots which may be unreachable by the reachstacker vehicle.
- (9) The yard bays are accessed by the reachstacker vehicle from one end.
- (10) To determine the total number of relocation movements, it is assumed that, after positioning the interfering container in a Baroti coordinate, all containers stored in the yard are (virtually) loaded onto the ship.

The following notation is defined.

Parameters are as follows:

B : total number of bays available to allocate containers,

Q : container storage capacity of each bay,

W_{IC} : weight of the interfering container,

E : number of empty slots available in all the bays after relocating the interfering container,

I_{ij} : weight of the container stored in position (i, j) as the initial inventory; this parameter is set to 0 when no container is stored in the position (i, j) ,

L : $\max_{(i,j)} \{I_{ij}\}$,

M : a big constant,

COMB: set of ordered pairs of positions in a bay; this set is used to inspect if the arrangement will denote a relocation movement:

$$\text{COMB} = \{(a, b) \mid a \text{ in } 1 \cdots Q - 1, b \text{ in } a + 1, \dots, Q\}. \quad (1)$$

Variables are as follows:

y_{ij} : 1 if the interfering container is stored in the position j of the bay i , 0 otherwise,

$d_i^{(a,b)}$: 1 if the weight of the container stored in position a of the bay i is greater than the weight of the container stored in position b of the same bay, 0 otherwise,

D_i : total number of relocations incurred in bay i ,

P_{ij} : weight of the container stored in position j of bay i , layout after positioning the interfering container,

Z_{ij} : 1 if the position j of the bay i is occupied by a container from initial inventory, 0 otherwise,

H_i : 1 if bay i is not empty, 0 otherwise,

F_{ij}^k : 1 if k th container of weight M is stored in position j of bay i , 0 otherwise.

The mathematical formulation of the model is as follows:

$$\text{Min} \quad \sum_{i=1}^B D_i + \sum_{i=1}^B H_i, \quad (2)$$

$$\text{s.t.} \quad \sum_{i=1}^B \sum_{j=1}^Q y_{ij} = 1, \quad (3)$$

$$\sum_{i=1}^B \sum_{j=1}^Q F_{ij}^k = 1 \quad k \in \{1, 2, \dots, E\}, \quad (4)$$

$$P_{ia} - P_{ib} \leq M * d_i^{(a,b)}, \quad (5)$$

$$i \in \{1, 2, \dots, B\}, \quad (a, b) \in \text{COMB},$$

$$D_i = \sum_{(a,b) \in \text{COMB}} d_i^{(a,b)} \quad i \in \{1, 2, \dots, B\}, \quad (6)$$

$$P_{ij} = y_{ij} * W_{IC} + I_{ij} + M * \sum_{k=1}^E F_{ij}^k, \quad (7)$$

$$i \in \{1, 2, \dots, B\}, \quad j \in \{1, 2, \dots, Q\},$$

$$I_{ij} \leq L * Z_{ij} \quad i \in \{1, 2, \dots, B\}, \quad j \in \{1, 2, \dots, Q\}, \quad (8)$$

$$y_{ij} + Z_{ij} + \sum_{k=1}^E F_{ij}^k \leq 1, \quad (9)$$

$$i \in \{1, 2, \dots, B\}, \quad j \in \{1, 2, \dots, Q\},$$

$$\sum_{j=1}^Q (Z_{ij} + y_{ij}) \leq 2 * H_i \quad i \in \{1, 2, \dots, B\}, \quad (10)$$

$$y_{ij} \in \{0, 1\} \quad i \in \{1, 2, \dots, B\}, \quad j \in \{1, 2, \dots, Q\}, \quad (11)$$

$$d_i^{(a,b)} \in \{0, 1\} \quad i \in \{1, 2, \dots, B\}, \quad (a, b) \in \text{COMB}, \quad (12)$$

$$D_i \geq 0 \quad i \in \{1, 2, \dots, B\}, \quad (13)$$

$$P_{ij} \geq 0 \quad i \in \{1, 2, \dots, B\}, \quad j \in \{1, 2, \dots, Q\}, \quad (14)$$

$$Z_{ij} \in \{0, 1\} \quad i \in \{1, 2, \dots, B\}, \quad j \in \{1, 2, \dots, Q\}, \quad (15)$$

$$H_i \in \{0, 1\} \quad i \in \{1, 2, \dots, B\}, \quad (16)$$

$$F_{ij}^k \in \{0, 1\},$$

$$i \in \{1, 2, \dots, B\}, \quad j \in \{1, 2, \dots, Q\}, \quad (17)$$

$$k \in \{1, 2, \dots, E\}.$$

The first term of objective function (2) minimizes the total number of relocation movements in all bays of the yard. The second term of (2) minimizes the number of opened bays. Constraint (3) ensures that the interfering container is stored in the yard. Constraint (4) ensures that exactly E containers (with weight M) are stored in the yard. This constraint ensures that the interfering container is not relocated to a position above an empty slot. The employment of a big weight for virtual containers does not impact the count of relocation movements in the first term of the objective function. Constraint (5) sets the variables $d_i^{(a,b)}$ to the value of 1 when a relocation movement is required; and constraint (6) defines the variable D_i as the total number of relocation movements incurred in bay i . Constraint (7) defines the variables P_{ij} as the final inventory after relocating the interfering container. Constraint (8) activates the binary variables Z_{ij} when a container is stored in position (i, j) as initial inventory. Observe that L is adjusted to take the value of the maximum weight of the containers stored as the initial inventory in the yard. Constraint (9) states that in each position either the interfering container, a container from the initial inventory, or a (virtual) container of weight M can be stored. Constraint (10) defines the variables H_i . Finally, constraints (11) to (17) define the domain of the decision variables.

4. Description of the Heuristic

A real-time heuristic is proposed to assign Baroti locations to interfering containers assuming that reachstacker vehicles are the only type of available container handling equipment. The proposed heuristic assumes that the current Baroti locations of the outbound containers are known. Since this procedure is applied in real time, the heuristic is useful at the operational level of the decision making process.

The proposed heuristic considers segregations of containers with similar attributes, such as the ship in which the containers will be loaded, container weight, and port of destination. Containers of the same segregation can be handled the same way as any other container in the same segregation, allowing more flexibility in the stacking policies in the yard than if each container were considered to be unique.

The heuristic assumes that at the beginning of the procedure the current Baroti locations of the containers are known and that the sequence to load containers onto the ship is defined according to the container weight. In general, heavier containers are loaded on the ship before lighter ones. The container ship should remain balanced and in practice the container loading sequence may not be strictly followed due

Tier 4	4	8	12	16	20	24
Tier 3	3	7	11	15	19	23
Tier 2	2	6	10	14	18	22
Tier 1	1	5	9	13	17	21
	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6

FIGURE 4: Sequence to fill a yard bay using a reachstacker vehicle accessing the bay from one end.

to the need to balance the ship in the berth site. The heuristic procedure described in the remainder of this paper is referred to as the smart-relocation (S-R) heuristic since it provides Baroti coordinates for interfering containers when relocation movements are executed.

Figure 4 shows a sequence to fill a yard bay with containers using a reachstacker vehicle as handling equipment and assumes that the reachstacker vehicle can access the bay only from the right side. It is important to emphasize that this is not the only feasible sequence to fill a bay with containers (e.g., if RTG cranes are available), but it is a representative of the practice in many Latin American ports. The numbers in the figure represent the sequence in which each slot is filled. This policy to fill yard bays implies that a container is not allocated in a position with sequence k if the positions with sequences $\{1, 2, \dots, k-1\}$ have not been filled before.

Some parameters should be defined as input information to the procedure. The parameters of the S-R heuristic are as follows:

- IC: interfering container that should be relocated,
- DC: desired container that should be retrieved from the yard,
- B : set of available bays in the yard for container IC,
- OB: original bay in which the container IC is stored before relocation,
- R_b : range of bays relative to OB in which the container IC can be relocated,
- T : maximum tier to stack containers in each row of each bay,
- Q_i : container storage capacity of bay i ,
- R_1 : set of bays that are close enough to the bay OB considering the range specified in R_b ,
- R_2 : subset of R_1 including only the bays that are neither empty nor full,
- R' : set of bays that are located beyond the range specified in R_b ,
- Inventory $_i$: number of containers stored in bay i ,
- direct(a, b): rectilinear distance between bays a and b ,
- M : a large constant.

The parameter R_b indicates the range of allowed bays to inspect in search of a Baroti coordinate for the interfering

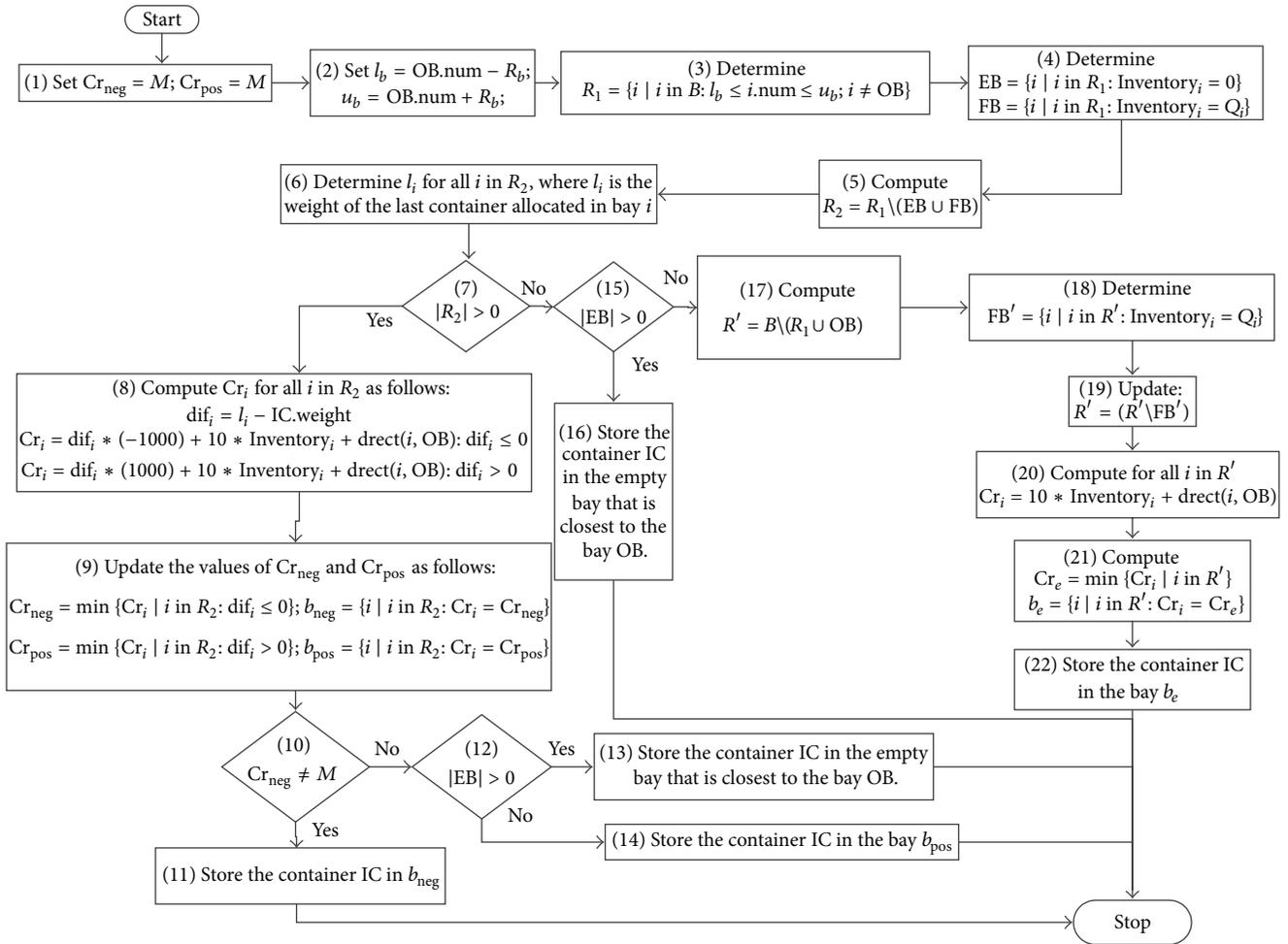


FIGURE 5: Flow diagram of the S-R heuristic.

container relative to the bay OB. For instance, if an interfering container is stacked in bay number 3 (OB = bay 3) and R_b is defined as 1, then the range of bays is 3 ± 1 ; that is, $R_1 = \{\text{bay 2, bay 4}\}$. Bay number 3 is not a candidate because it is the original bay. (Recall that this would involve two relocation movements, one to remove the container and another to return the container to the original bay.) As R_b increases, the heuristic procedure is able to inspect more distant bays but the travel time of the interfering container to the new coordinate increases. The flow diagram of the S-R heuristic shown in Figure 5 indicates that it is preferable to store the interfering container in a bay of the set R_2 which includes the nearby bays according to the parameter R_b . When it is not possible to store the interfering container in a bay of the set R_2 , the heuristic suggests storing it in the nearest empty bay. In the case when it is not possible to store the interfering container neither in R_2 nor in an empty bay, the S-R heuristic suggests storing it in a bay even though it may not be close to the bay OB. The S-R heuristic is executed each time a relocation movement is performed.

A numerical example is introduced to illustrate the operation of the S-R heuristic. Consider a port yard with

3 bays, each bay with 2 rows. Assume that the weight of the desired container (DC) is 3 and the maximum tier height in each stack is 2. The R_b is specified in this case as 1. Remember that this heuristic procedure is used to make decisions at an operational level and must be run each time that a relocation movement is performed to determine the new Baroti coordinate of the interfering container. Consider the yard layout of Figure 6. The numbers in the figure represent the weight of the containers. The desired container is located in bay 2, row 2, and tier 1. The container in bay 2, row 2, and tier 2 is interfering and it needs to be relocated.

The capacity of each bay is 4 containers. The rectilinear distance between two bays is calculated as the difference in their bay numbers. As the container IC is located in bay number 2, that bay represents the parameter OB. The values of l_b and u_b in step 2 are $l_b = 2 - 1 = 1$ and $u_b = 2 + 1 = 3$. Now the set of bays R_1 can be determined as $R_1 = \{\text{bay 1, bay 3}\}$. Since all bays in R_1 are not empty or full, $EB = FB = \emptyset$ and $R_2 = R_1$. The weight of the interfering container is 2 and the heuristic assumes that the bays can be accessed only from right side. Since bay number 1 has only one container stored and its weight is 2, $l_1 = 2$. Bay number 3 has two containers

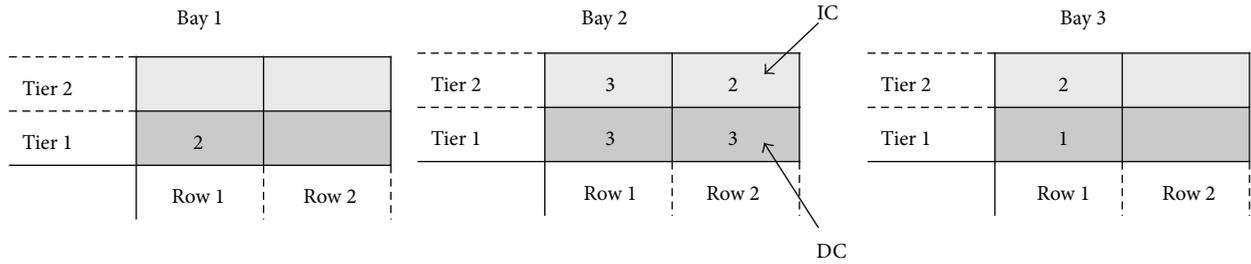


FIGURE 6: Initial configuration for numerical example 1.

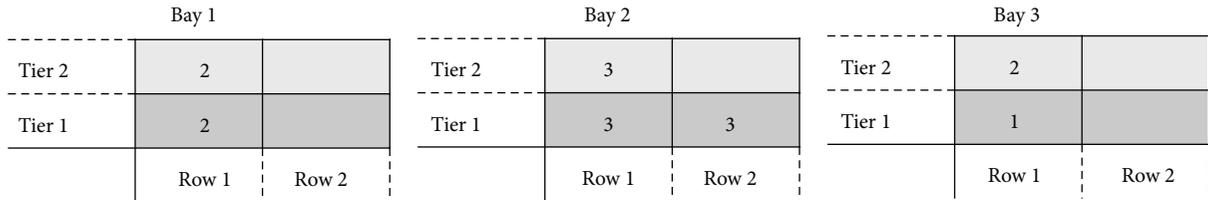


FIGURE 7: Final configuration for numerical example 1.

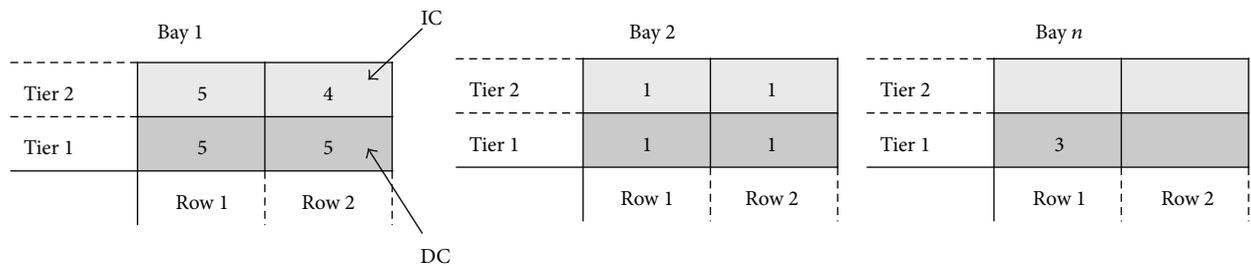


FIGURE 8: Initial configuration for numerical example 2.

stored, but the container with weight 2 is stacked above the container with weight 1, so the weight of the last container allocated in this bay is 2 defining $l_3 = 2$. The cardinality of the set R_2 is greater than zero and step number 8 is executed as follows: $dif_1 = 2 - 2 = 0$; $dif_2 = 2 - 2 = 0$. The rectilinear distance between bay number 2 and bay number 1 is 1. Similarly, the rectilinear distance between bay number 2 and bay number 3 is 1. $Cr_1 = (0)(-1000) + (10)(1) + 1 = 11$; and $Cr_2 = (0)(-1000) + (10)(2) + 1 = 21$. Step number 9 determines that $Cr_{neg} = \min\{11, 21\}$, so Cr_{neg} is 11 and b_{neg} is bay 1. The container IC can be stacked in either row 1 or row 2 of bay number 1, but row number 1 is preferred to avoid empty spaces according to the sequence shown in Figure 4. The final decision in step number 11 is to relocate the interfering container IC to bay number 1. Figure 7 shows the configuration after relocating the interfering container.

Another relevant situation is shown in Figure 8. Observe that in this case the yard also has 3 bays, but one bay is labeled “bay n ” to indicate that it is relatively far from bay 1 and bay 2. The capacities of bays are equal to 4 containers and R_b is defined as 1. The weight of desired container (DC) is 5 and the weight of the interfering container (IC) is 4.

In this case, OB is bay 1 and the values of l_b and u_b are defined as $l_b = 1 - 1 = 0$ and $u_b = 1 + 1 = 2$. R_1 contains

the set of near bays to OB according to R_b , but, in this case, there is no bay 0 and bay number 1 is the original bay (OB), so $R_1 = \{\text{bay } 2\}$. Since bay 2 is full of containers, $FB = \{\text{bay } 2\}$ and $R_2 = (R_1 \setminus FB) = \emptyset$. There are no empty bays in R_1 , so $EB = \emptyset$. In this case, the cardinality of the set R_2 is not greater than 0 and the cardinality of the set EB is 0, so step number 17 is executed and R' is determined as $R' = \{\text{bay } n\}$. As FB' in step 18 is \emptyset , R' remains with no change after the update in step 19. Bay n has only one container stored and the rectilinear distance between bay n and bay 1 is $n - 1$; so, $Cr_n = (10)(1) + n - 1 = 9 + n$. Step number 21 defines Cr_e as the minimum among the values of Cr calculated for all bays in R' ; but in this case R' consists only in one bay, so $Cr_e = 9 + n$, and $b_e = \text{bay } n$. The final decision is to store the interfering container in bay n . Figure 9 shows the configuration after relocating the interfering container.

5. Determination of the Number of Relocation Movements in the Worst Case Scenario

In order to calculate the total number of relocation movements, it is assumed that all the containers required to be moved to retrieve a given container are relocated in the same configuration as they were previously stacked at the bay. For

Bay 1			Bay 2			Bay n		
Tier 2	5		Tier 2	1	1	Tier 2	4	
Tier 1	5	5	Tier 1	1	1	Tier 1	3	
	Row 1	Row 2		Row 1	Row 2		Row 1	Row 2

FIGURE 9: Final configuration for numerical example 2.

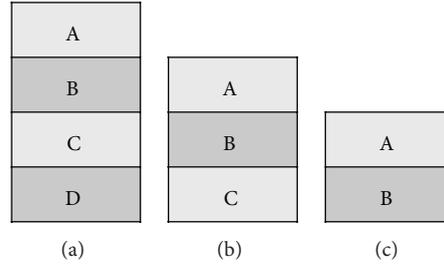


FIGURE 10: Configuration of a bay of capacity 4 during a container retrieval process.

instance, consider a bay with a capacity of 4 containers as illustrated in Figure 10(a).

Suppose that the sequence to retrieve the containers according to the stowage plan is {D, C, B, A}. To retrieve container D, three relocation movements (A, B, C) are required. Once container D is retrieved, the configuration of the bay is as shown in Figure 10(b).

Now, in order to retrieve container C, two additional relocation movements are required. The configuration at this moment is as shown in Figure 10(c).

Container B is the next one to be picked up, so one additional relocation movement should be made to reach it. The last container to be retrieved does not require any relocation movements, so the total number of relocation movements required to retrieve all the containers of this bay with capacity of four is $3 + 2 + 1 = 6$. This sequence results in the greatest number of relocation movements because the order of retrieving the containers is exactly the opposite of the order of the containers stacked in the bay.

The policy of returning to the original bay all interfering containers is referred to as the worst case scenario in the remainder of this paper. It is quite similar to the actual practice in many smaller ports and thus provides a close approximation to the current practice.

6. Numerical Results

A set of experiments employing the S-R heuristic was performed and the results were compared against the worst case scenario described in previous section. All the experiments presented in this section were performed on a personal computer with an i5 processor and 6 GB RAM. The S-R heuristic is coded in C#.

The ratio of relocation movements to total movements is used as a performance metric. Total movements are defined as relocation movements + effective movements.

In contrast with relocation movements, an effective movement is performed when a container of the requested segregation can be reached directly by the yard equipment. The ratio of relocation movements to total movements is computed as follows:

$$\frac{\text{number of relocation movements}}{\text{(number of relocation movements} + \text{number of effective movements)}^{-1}} \quad (18)$$

To illustrate this ratio, consider a bay filled with 30 containers, and assume that a total of 10 relocation movements are required to empty the bay. This yields a ratio of $10/(10+30) = 1/4 = 25\%$.

We test four values of $R_b = \{1, 2, 3, 4\}$. The R_b value used in each experiment is indicated in the header of Table 1 as S-R(R_b). A tight case is represented when $R_b = 1$ because the S-R heuristic seeks Baroti coordinates for interfering containers only in the immediate adjacent bays to OB. A medium case is represented when $R_b = \{2, 3\}$, and the relaxed case is represented when $R_b = 4$, indicating that the S-R heuristic is able to seek Baroti coordinates in all bays of the yard. We define 8 different types of instances and generate 20 different container arrival sequences for each instance type. The number of containers, maximum tier, and weight levels studied are specified in Table 1 for each instance type. All instance types consider a port yard with 5 bays and 6 rows in each bay. The average ratios of relocation movements to total movements are reported in the table. The proposed heuristic is able to suggest a new Baroti coordinate for each interfering container in less than 1 second, enabling the use of this heuristic in real port operations.

When R_b increases, the ratio of relocation to total movements decreases since the space to search Baroti coordinates

TABLE 1: Ratio of relocation movements to total movements of the procedures studied.

Instance type	Number of containers	Number of tiers	S-R(1)	S-R(2)	S-R(3)	S-R(4)	Worst case scenario
1	96	4	45.6%	39.4%	37.5%	39.0%	76.7%
2	96	4	45.6%	42.0%	39.9%	37.9%	75.2%
3	120	5	49.5%	45.3%	42.1%	46.5%	81.1%
4	120	5	55.4%	49.7%	47.8%	46.5%	82.6%
5	80	4	39.2%	33.3%	30.1%	30.8%	63.3%
6	80	4	43.0%	36.8%	34.3%	33.6%	66.8%
7	100	5	41.9%	36.6%	33.0%	32.3%	75.4%
8	100	5	48.8%	42.3%	38.4%	37.8%	77.6%

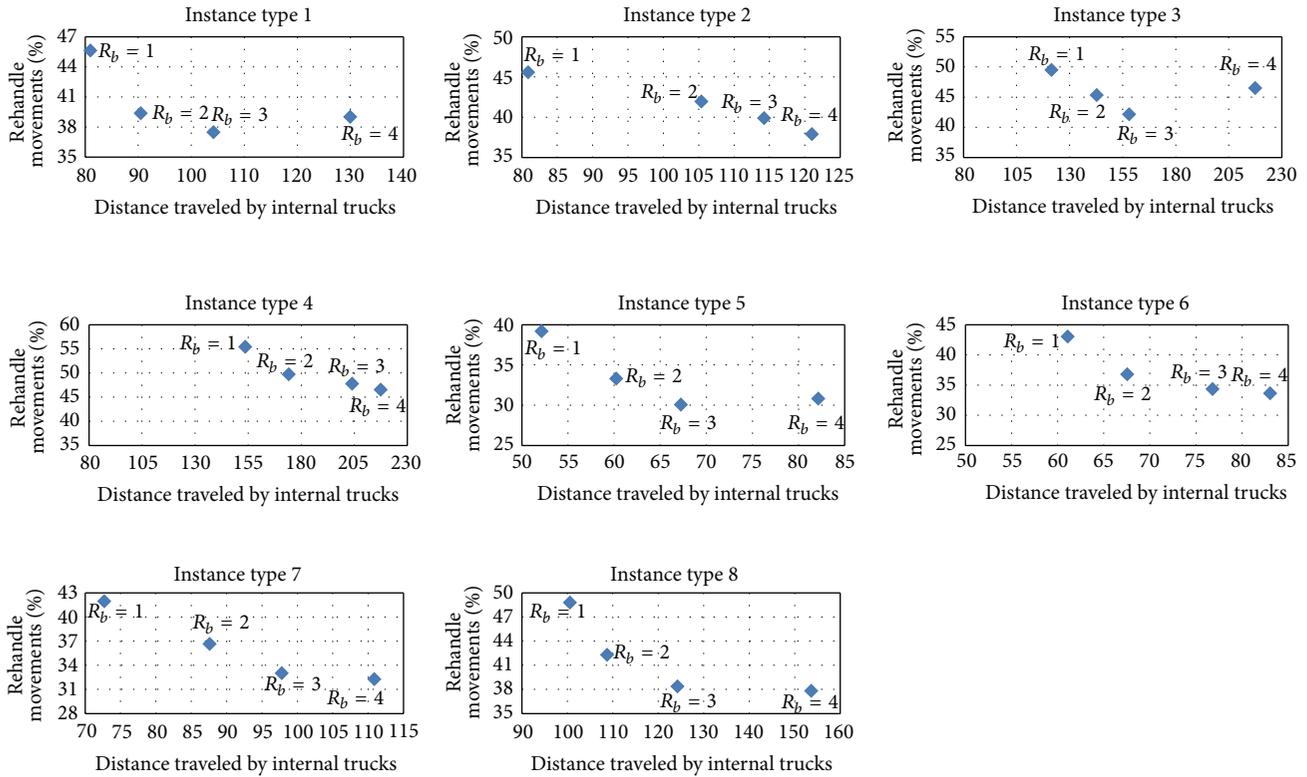


FIGURE 11: Relation between the ratio of relocation movements and the distance traveled by internal trucks.

includes more bays. Table 2 shows the gap between the S-R heuristic and the worst case scenario of each instance type.

The maximum gap reported between the S-R heuristic and the worst case scenario is 57.2% in instances with $R_b = 4$, which is expected because in this case the heuristic can suggest Baroti coordinates in any bay of the yard. The minimum gap is 32.9% and is reported in instances with $R_b = 1$. Observe that, in this case, the bigger the gap, the better the performance because the comparison is against the worst case scenario.

The determination of relocation movements performed in the worst case scenario considers as one single relocation movement the retrieval of the interfering container from the bay to situate it in the aisle and the movement to return it to the original bay. However, the movement from the bay to the aisle and the movement from the aisle back to the bay are

actually two relocation movements. Because of this, the ratio of relocation to total movements in worst case scenario in actual practice is greater than or equal to the ratio reported in Table 1. This means that the benefit of using the S-R heuristic is actually greater than implied by the values in Table 1.

The parameter R_b can be adjusted to achieve a tradeoff between a reduction in relocation movements and an increase in the distance traveled by internal trucks. The rectilinear distance traveled by internal trucks and the ratio of relocation movements for the instance types studied are shown in Figure 11. In this figure the relation between the ratio of relocation movements and the distance traveled by internal trucks is shown.

The general observed behavior is that the distance traveled by internal trucks increases with an increase of R_b . The percentage of relocation movements tends to decrease with

TABLE 2: Gap between results of S-R heuristic and the worst case scenario.

Instance type	S-R(1)	S-R(2)	S-R(3)	S-R(4)
1	40.5%	48.7%	51.2%	49.1%
2	39.4%	44.2%	47.0%	49.7%
3	38.9%	44.1%	48.0%	42.6%
4	32.9%	39.8%	42.1%	43.6%
5	38.1%	47.4%	52.5%	51.3%
6	35.6%	44.9%	48.6%	49.6%
7	44.4%	51.4%	56.2%	57.2%
8	37.1%	45.5%	50.6%	51.3%

an increase in R_b . This is observed for all tested instance types for R_b equal to 1, 2, and 3. However, for some instance types, the percentage of relocation movements is smaller with $R_b = 3$ than with $R_b = 4$, suggesting that an intermediate value of R_b may be optimal in some cases.

7. Conclusions and Recommendations for Further Research

We propose a heuristic procedure which suggests Baroti coordinates for interfering containers during the loading operation when reachstacker vehicles are the available handling equipment. To evaluate the performance of the proposed heuristic, we compute the number of relocation movements assuming the worst case scenario. The proposed heuristic has a parameter R_b , which determines the proximity of candidate bays for container relocation. When the yard wants to perform the minimum number of relocation movements, the interfering containers are relocated in any bay of the yard, regardless of its proximity to the original bay in which the container is stored. On the other hand, when the terminal needs to minimize the distance traveled by internal trucks, the interfering containers are stored in a set of bays adjacent to the original bay, but the number of relocation movements tends to be greater.

For further research, we propose developing heuristic procedures minimizing the total cost of the loading operation considering both the distance traveled by internal trucks and the relocation movements. Alternatively, the ship turnaround time could be considered as a performance metric, provided that it may account for both relocations and distance metrics expressed as the required time to serve the ship. Another approach to address this situation is to consider the problem as a biobjective optimization problem in which an efficient frontier is sought rather than a single solution.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] F. Tapia, R. Covarrubias, P. Miranda, and R. G. González-Ramírez, "On the storage space allocation problem," in *Proceedings of the 22nd International Conference on Production Research (ICPR '13)*, Iguassu Falls, Brazil, July-August 2013.
- [2] A. Bortfeldt and F. Forster, "A tree search procedure for the container pre-marshalling problem," *European Journal of Operational Research*, vol. 217, no. 3, pp. 531–540, 2012.
- [3] H. J. Carlo, I. F. A. Vis, and K. J. Roodbergen, "Storage yard operations in container terminals: literature overview, trends, and research directions," *European Journal of Operational Research*, vol. 235, no. 2, pp. 412–430, 2014.
- [4] K. H. Kim and H. B. Kim, "Segregating space allocation models for container inventories in port container terminals," *International Journal of Production Economics*, vol. 59, no. 1, pp. 415–423, 1999.
- [5] K. Young Kim and K. Hwan Kim, "Routing algorithm for a single straddle carrier to load export containers onto a containership," *International Journal of Production Economics*, vol. 59, no. 1, pp. 425–433, 1999.
- [6] J. Kang, K. R. Ryu, and K. H. Kim, "Deriving stacking strategies for export containers with uncertain weight information," *Journal of Intelligent Manufacturing*, vol. 17, no. 4, pp. 399–410, 2006.
- [7] D.-H. Lee, Z. Cao, and Q. Meng, "Scheduling of two-transterainer systems for loading outbound containers in port container terminals with simulated annealing algorithm," *International Journal of Production Economics*, vol. 107, no. 1, pp. 115–124, 2007.
- [8] Y. J. Woo and K. H. Kim, "Estimating the space requirement for outbound container inventories in port container terminals," *International Journal of Production Economics*, vol. 133, no. 1, pp. 293–301, 2011.
- [9] T. Park, R. Choe, Y. H. Kim, and K. R. Ryu, "Dynamic adjustment of container stacking policy in an automated container terminal," *International Journal of Production Economics*, vol. 133, no. 1, pp. 385–392, 2011.
- [10] L. Chen and Z. Lu, "The storage location assignment problem for outbound containers in a maritime terminal," *International Journal of Production Economics*, vol. 135, no. 1, pp. 73–80, 2012.
- [11] M. E. H. Petering, "Real-time container storage location assignment at an RTG-based seaport container transshipment terminal: problem description, control system, simulation model, and penalty scheme experimentation," *Flexible Services and Manufacturing Journal*, 31 pages, 2013.
- [12] L. Wang, X. Zhu, and Z. Xie, "Storage space allocation of inbound container in railway container terminal," *Mathematical Problems in Engineering*, vol. 2014, Article ID 956536, 10 pages, 2014.
- [13] J. H. Yang and K. H. Kim, "A grouped storage method for minimizing relocations in block stacking systems," *Journal of Intelligent Manufacturing*, vol. 17, no. 4, pp. 453–463, 2006.
- [14] Y.-W. Wan, J. Liu, and P.-C. Tsai, "The assignment of storage locations to containers for a container stack," *Naval Research Logistics*, vol. 56, no. 8, pp. 699–713, 2009.
- [15] T. Lixin, W. Jiang, J. Liu, and Y. Dong, "Research into container reshuffling and stacking problems in container terminal yards," *IIE Transactions*, 2014.
- [16] K. H. Kim and G.-P. Hong, "A heuristic rule for relocating blocks," *Computers & Operations Research*, vol. 33, no. 4, pp. 940–954, 2006.

- [17] M. Caserta, S. Schwarze, and S. Voß, “A new binary description of the blocks relocation problem and benefits in a look ahead heuristic,” in *Evolutionary Computation in Combinatorial Optimization*, vol. 5482, pp. 37–48, Springer, Berlin, Germany, 2009.
- [18] M. Caserta, S. Voß, and M. Sniedovich, “Applying the corridor method to a blocks relocation problem,” *OR Spectrum*, vol. 33, no. 4, pp. 915–929, 2011.
- [19] B. Borgman, E. van Asperen, and R. Dekker, “Online rules for container stacking,” *OR Spectrum*, vol. 32, no. 3, pp. 687–716, 2010.
- [20] A. H. Gharehgozli, Y. Yu, R. De Koster, and J. T. Udding, “A decision-tree stacking heuristic minimising the expected number of reshuffles at a container terminal,” *International Journal of Production Research*, vol. 52, no. 9, pp. 2592–2611, 2014.
- [21] J. Ries, R. G. González-Ramírez, and P. Miranda, “A fuzzy logic model for the container stacking problem at container terminals,” in *Computational Logistics*, vol. 8760 of *Lecture Notes in Computer Science*, pp. 93–111, Springer, Berlin, Germany, 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

