A Simulation Study of the Effect of a Handshake Area on the Performance of Twin Automated Stacking Cranes

Amir Hossein Gharehgozli[*,1], Floris Gerardus Vernooij[2], Nima Zaerpour[3]
[1]Maritime Administration, Texas A&M University at Galveston, Galveston, TX, USA
[2]Rotterdam School of Management, Erasmus University, Rotterdam, the Netherlands
[3]College of Business Administration, California State University San Marcos, San Marcos, CA, USA

Abstract

This paper studies the effect of a handshake area on the performance of twin automated stacking cranes (ASCs) operating on top of a stack with transfer zones at both seaside and landside. The handshake area is a temporary storage location so that one crane can start a request and leave the container there for the other crane to complete the request. By testing settings with and without such a handshake area, the goal is to find robust rules which result in the best performance, measured as (1) the makespan to finish all requests and (2) the total waiting time of the cranes due to interference or nonconsecutive delivery of containers in the handshake area (blocking time). The effect of five decision variables on the performance are tested. The decision variables are (1) the way the requests are handled by the cranes (scheduling), (2) the storage location of the containers in the handshake area, (3) the location of the handshake area in the stack, (4) the size of the handshake area and (5) the number of handshake areas in the stack. For each decision variable, multiple heuristics are developed. The results indicate that settings without a handshake area outperform settings with a handshake area for virtually all instances tested when using the same scheduling heuristic. For both types of settings, the choice for a scheduling heuristic impacts the final performance the most. In this study, we opt for simple heuristics since container terminal operators prefer to avoid any complexity in coordinating and scheduling two ASCs for safety and simplicity reasons.

Keywords: OR in Maritime Industry; Simulation Model; Seaport Container Terminal; Twin Automated Stacking Cranes; Container Stack; Collaboration; Handshake Area

1. Introduction

Due to globalization the number of containers and size of ships have increased significantly in recent years and the growth is expected in future (Fransoo & Lee, 2013). The global players in a highly competitive market are constantly seeking to improve operations and gain cost efficiency. One of the ways they do, is by selecting container terminals or ports with the best price-performance ratio. Therefore, in order for terminals to remain competitive and have the customers returning, terminals need to increase the turnover of containers and at the same time keep prices low. As a solution, terminals aim to increase storage density meanwhile increasing the level of automation in their daily operations. An example is using automated stacking cranes (ASCs), which are automated rail-mounted gantry cranes. These cranes work in the container storage yard (often referred to as stacks) where containers are temporarily stacked. The stack is one of the areas in a container terminal which is significantly affected by the throughput increase at a container terminal (Vis & Carlo, 2010). New technologies and methods are constantly being developed to efficiently handle stacking operations in order to avoid the stacks becoming the bottleneck of the terminal.

ASCs can be seen as a step in the automation of container terminals to ensure low operating cost, high utilization of yard capacity, and high availability. Among terminals using ASCs, different configurations can be found. The basic configuration is to have one ASC per stack serving requests for both landside and seaside as pick-up and drop-off points, also known as transfer points or input/output (I/O) points. In order to increase the throughput, configurations with two non-passing ASCs per stack are common in practice, in which one ASC serves the landside and the other serves the seaside. In such

[*] *Corresponding author*: Department of Maritime Administration, Texas A&M University at Galveston, PO Box 1675, Galveston, Texas 77553, USA. Tel: +1 (409) 740-4854, Fax: +1 (409) 741-4014, Email: GharehgA@TAMUG.edu.

configurations, the ASCs are non-passing, i.e. there are two identical ASCs which are unable to pass each other (also known as twin cranes, see Figure 1). In this paper we focus on a stack with twin cranes.

A container stack with twin ASCs typically has I/O points at the seaside and the landside. On the seaside, automatic guided vehicles or terminal trucks pick up and deliver containers while at the landside this function is typically performed by trucks. The difficulty is that when the landside ASC has a request close to the seaside, the seaside ASC needs to make room or (if possible) even leave the stack in order for the landside ASC to be able to handle its request. This potentially increases the ASC travel time and diminishes the terminal's performance. To address this issue, the ASCs can work together. Such a configuration would need a handshake area in which the ASCs can hand over the containers from one to another. For example, in Figure 1, the landside ASC first brings the storage container to the handshake area. Then the seaside ASC picks it up and stacks it in its final location. For the retrieval request, the seaside ASC first brings the container to the handshake area. Then the landside ASC retrieves it to the landside I/O point.

This research focuses on optimizing twin ASC operations by implementing a handshake area. Our objective is to minimize the total makespan of a given set of storage and retrieval requests. Makespan is commonly used in practice to measure the performance of twin ASC cranes. Minimizing the makespan is considered to be a good proxy for other objectives for container terminals such as minimizing the total travel time, minimizing the crane waiting time, and minimizing the delay with respect to the due date. (Gharehgozli et al, 2015; Zaerpour et al., 2014; Vis and Carlo, 2010, Speer et al, 2011).

We study the following question: *"Which handshake area setting leads to the best performance at container terminals with twin ASCs?"* With the apparent increasing number of terminals utilizing a system with twin ASCs and the lack of academic research, this question is relevant for both practitioners and academics. To answer this question, we formulate the following sub- questions:

1. *"For a given stack setting, what is the best scheduling heuristic?"*
2. *"For a given stack setting, what is the best handshake area design?"*
3. *"For a given stack setting, what is the best storage location strategy in the handshake area?"*
4. *"What is the best combination of scheduling heuristic, storage location and handshake area design?"*

To answer these questions, we have proposed a number of simple heuristics for each of the sub-questions and have performed a simulation study on each combination of the heuristics. We develop simple heuristics to answer each question. This is inspired by practice where container terminal operators prefer to avoid any complexity in coordinating and scheduling two ASC cranes for safety and simplicity reasons although a more sophisticated scheduling might bring additional time-saving benefits to the terminal.

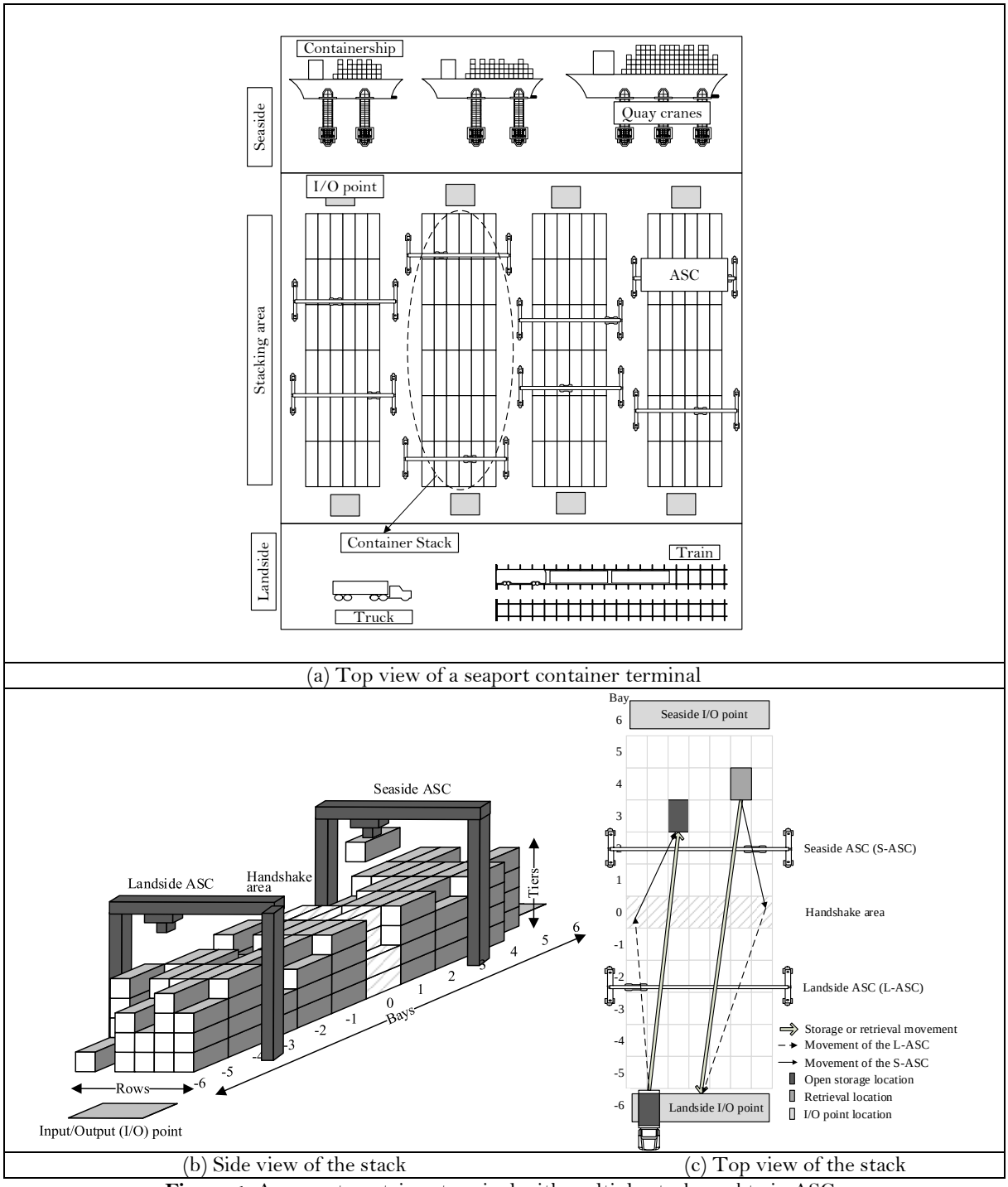


Figure 1. A seaport container terminal with multiple stacks and twin ASCs

The rest of this paper is organized as follows. In section 2, we review the related literature. In section 3, the system, concepts, and methodologies are discussed. This is followed by an explanation of the simulation model (section 4). The results are presented and discussed in section 5. This is followed by the conclusion, limitations and areas of future study in section 6.

2. Literature review

The literature on stacking operations can be in general categorized into two groups: stacking containers and scheduling yard crane(s). Container stacking focuses on properly sacking containers in the stack with the main objective of minimizing the number of reshuffles. This has been studied in three main categories (see also the recent literature reviews by Goerigk et al., 2016; Lehnfeld and Knust, 2014): (1) pre-marshalling (Cordeaua, et al., 2015; Expósito-Izquierdo et al., 2012; Bortfeldt and Forster, 2012; Huang and Lin, 2012), (2) relocating methods while retrieving containers (Ji , et al., 2015; Jin, et al., 2015; Ku & Arthanari, 2015; Caserta et al., 2011; Forster and Bortfeldt, 2012), and (3) stacking methods (Boysen and Emde, 2016; Zhang, et al., 2014a; Zhang, et al., 2014b; Gharehgozli et al., 2014b; Yu and Qi, 2013). Almost in all paper, the authors focus on reshuffling before the arrival of a ship. Meanwhile, reshuffles may occur when there is a delay in the arrival of a ship. Gharehgozli et al. (2017b) name the former as "prior reshuffles" and introduce the latter as "posterior reshuffling." They show that the delay of a ship can significantly increase the reshuffling effort required in a terminal.

Yard crane scheduling focuses on obtaining an optimal plan for one or more yard cranes stacking and retrieving containers in the stacking area of a terminal. Many authors have focused on the stacking operations by a single crane (for a complete list of studies, we refer to the works by Gharehgozli et al., 2014a; Gharehgozli et al., 2016; Gorman et al., 2014; Carlo et al., 2014; and Boysen and Stephan, 2016). On the other hand, some studies have considered settings with more than one crane sharing a stack, a similar setting as this paper (see for example, Dorndorf & Schneider, 2010; Gharehgozli, et al., 2015; Li, et al., 2012; Li, et al., 2009; Park, et al., 2010; Stahlbock & Voß, 2010; Ng, 2005; Saanen & Valkengoed, 2005; Choe, et al., 2007; Dell, et al., 2009). In most papers, the problem is formulated as a mixed integer model and minimizing makespan is the main objective (see for example, Gharehgozli et al., 2015; Vis and Carlo, 2010; Briskorn & Angeloudis, 2016; Briskorn, et al., 2016). Furthermore, in settings with more than a single crane, dealing with crane interferences is the other major research focus (Boysen, et al., 2016). However, in contrast with the literature on quay crane scheduling (Bierwirth and Meisel, 2010; Meisel and Bierwirth, 2013, Meisel, 2011; Kim and Park, 2004; Moccia et al., 2006; Choo et al., 2010, Lim et al, 2004, 2007; Lee et al., 2008; Chen et al., 2012, Liu et al., 2006), the yard crane interference has not been well studied. In order to deal with the complexity of crane scheduling, simulation is the other tool used to study the problem. For example, Kemme (2012) develops a comprehensive simulation study to compare effects of storage block layout and four automated yard crane systems including twin ASCs on the performance of seaport container terminals (see also Petering et al., 2009). However, collaboration of these cranes, e.g., through a handshake area, has not been considered or has not been the main focus. This is despite the fact that researchers have started studying collaboration in other parts of terminal operations (see for example, Gharehgozli et al., 2017a).

Carlo & Vis (2009) is the earliest study found which mentions the necessity of a handshake area for collaborating cranes. They state that for such a system two decisions need to be taken: (1) whether the requests are transported by one or two cranes and thus making use of the interchange zone and (2) in which order the requests need to be handled. Apart from this study, there is a lack of studies concerning collaboration in the literature. In fact, to our knowledge, only two studies have focused on such a system of collaborating ASCs, namely Dorndorf & Schneider (2010) and Carlo & Martínez-Acevedo (2015). Seemingly, in a broader context, there are also few studies which discuss a system of collaborating material handling equipment (Carlo & Martínez-Acevedo, 2015). One study which discusses a system of collaborating lifts is done by Carlo & Vis (2012). In their article, a system with two collaborating lifts which share a mast is studied. This is a system which can be compared with a system with twin ASCs since both have two rail mounted load-carrying devices which share a single

rail. The difference being that lifts move vertically between locations while ASC move horizontally. In the rest of this section, we focus on findings and insights from these studies.

Carlo & Vis (2012) study the request assignment problem, which lift is going to handle which request and in which order. Their goal is to minimize the total time required for the lifts to handle the requests (i.e., makespan). A methodology to identify situations in which the lifts interfere is given. By comparing various priority rules they provide evidence that such a system can achieve average improvements of 82.49% and 60.08% in terms of throughput compared to a single lift system and simple rule of thumb, respectively. Carlo & Martínez-Acevedo (2015) study a system with twin ASCs with a handshake area zone in the middle of the block. They assume that the sequence of requests to be served by each ASC is given. They compare 14 priority rules with an optimal situation found by a branch and bound approach. Their goal is to find a rule which minimizes the makespan. Results show that the rule in which priority is given to the crane that has the longest remaining time at the moment of interference outperforms all other rules. Finally, Dorndorf & Schneider (2010) study a triple cross-over collaborating ASC system in an automated stack. Here, interference occurs either between the two smaller cranes or if the larger ASC lowers its spreader. The main objective of their study is to maximize the productivity of the crane system under peak load while preventing delays in the transport of import and export containers. They use a beam-search approach for the request-to-ASC assignment. In the next step, they use a branch and bound approach to determine the optimal route. Results show that their method leads to a productivity gain of more than 20% compared to commonly used heuristics.

From the literature two conclusions can be drawn: (1) there is a lack in research which studies a system with collaborating twin ASC systems, (2) and those who have studied such a system use a fixed scheduling heuristic and assumed only one handshake area with a fixed size and location. By studying the effect of a handshake area and attempting to determine the best design of a handshake area, we aim to bridge the gap in the literature. We have performed our research using simulation, proven to be a powerful tool to deal with complex environments such as a container terminal. Angeloudis and Bell (2011) review container terminal simulation models and explain that "the sheer size of the facilities and the complexity of equipment used make it difficult to predict analytically how the terminal will operate under specific layouts and configurations." Dragović et al. (2016) reviews all simulations studies on port operations since 1961 and conclude that "over the past 50 years the use of simulation models has been increasingly favored and instrumental in the development of ports and more specifically of container terminals."

3. Problem Description & Methodology

In this section, we first describe the system under study in section 3.1, followed by a discussion of the scheduling heuristics, priority rules, handshake area designs, and storage strategies in section 3.2.

4. System description

In Figure 1, a top view of the studied system with two collaborating ASCs is shown. The L-ASC handles retrieval and storage requests to and from the landside and the S-ASC handles requests to and from the seaside. To avoid accidents, the cranes need to keep a minimum safety distance of at least 1 bay, the closest distance that the cranes can work beside each other. In the center of the stack in Figure 1, there is a 1-bay wide handshake area. The handshake area is a temporary storage location so that one crane can start a request and leave it to the other crane to complete it. In this situation, it needs to be decided which requests are split. In this study, the requests that their containers cross the handshake area are considered as split requests. These containers are handled by both cranes.

4.1.1 Interference among cranes

A crucial aspect of a setting with twin ASCs are scenarios where they interfere with each other. Thus, we need to track the location of each ASC at any given time. We follow the same schematic representation of ASC operations as Carlo & Martínez-Acevedo (2015) to find scenarios in which the ASCs interfere. In this representation, each storage or retrieval request includes four movements:

1. The crane travels (empty) to the pick-up location of the requested container (either the I/O point or to the location of retrieval).
2. The crane lowers its spreader, picks up the container and raises its spreader again.
3. The crane moves with the container to the desired location (either the I/O point or a storage location in the stack).
4. The crane lowers its spreader, deposits the container and lifts its spreader again.

In Figure 2(left), an example of a single retrieval movement can be seen, here f1-f4 represent the previously discussed steps. The y-axis represents the location of the crane in terms of bays and the x-axis represents the time. In this situation the crane moves from bay -2 to bay 2, stays there for 20 seconds to pick-up the container, moves to the desired end bay (-5) and deposits the container.

Figure 2(right) illustrates the movement of two cranes together. Crane 1 is depicted by movement f1-f4 and crane 2 by movement g1-g4. From this figure it can be observed that the two cranes interfere between times 30 and 50 seconds due to two reasons. During this period, the cranes need to work in bays 2 and 3 which is less than 1 bay apart. Furthermore, crane 2 finishes its job and cross over crane 1 to perform another job in bay 0. Clearly, in this scenario one of the cranes needs to be given priority while the other waits. The advantage of using this schematic representation is that it becomes relatively easy to find and resolve scenarios where the cranes interfere. A more detailed explanation of the usage of this schematic representation in our model is given in section 4.

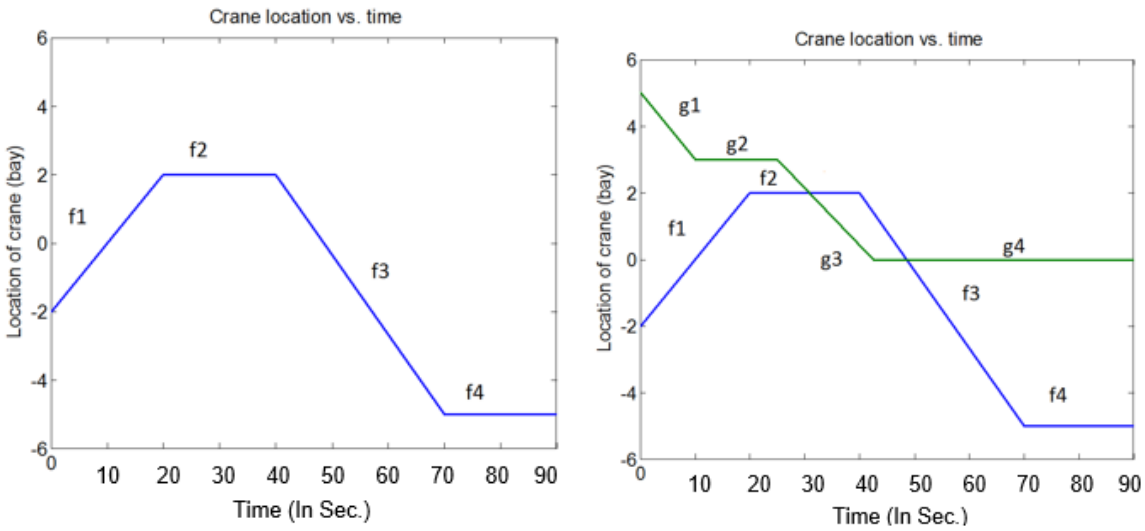


Figure 2. Example of a retrieval request with one and two cranes (adapted from Carlo & Martínez-Acevedo, 2015)

4.1.2 Settings with and without a handshake area

We test both systems with and without a handshake area. We study which combination of heuristics gives the best results for each setting. For settings without a handshake area, the main difficulty lies in deciding on how each crane schedules its jobs and how to handle interferences between the cranes. The main objective is to find a schedule which minimizes the makespan to handle all requests. A main issue in such a system is the situation in which, for instance, the seaside crane needs to handle a request

close to the landside. If the cranes interfere and the seaside crane has priority, the landside crane has to suspend its next requests to make room for the seaside crane.

For systems with a handshake area, the effect of extra movements are reduced as each crane has its own area within the stack and only in the handshake area interferences occur. So, if a crane needs to wait due to interference, it can simply wait in its own area. As a result, no additional movements occurs to, for instance, travel back and forth in order to make room for the other crane. However, for each request it needs to be decided whether it is going to be split. If so, it needs to be made sure that each split request is handled correctly and in a proper consecutive sequence. This means that it can only be picked up by a crane from the handshake area after it has been deposited by the other crane there.

5. Proposed strategies

In this section, we discuss the scheduling strategies, priority rules, and storage strategies for settings with and without a handshake area. In addition, for settings with a handshake area different handshake area designs such as location and size have been discussed.

5.1.1 Scheduling of containers for each crane

Scheduling of the cranes is the decision on when each request is to be handled and in what order. This is a complex and time-consuming exercise. For example, Gharehgozli et al. (2015) and Eilken and Fliedner (2016) show that the problem of sequencing storage and retrieval requests for the twin crane configuration is $\mathcal{NP}$-hard. Simply put, the problem can be modeled as a multiple travelling salesmen problem with interference constraints (see, for example, Vis and Roodbergen, 2009; Gharehgozli et al 2014a, 2015). In fact, $(N+1)!$, with N being the total number of requests, is the number of potential schedules for a general setting without a handshake area where both cranes can handle all requests (e.g., the landside crane leaves the stack so that the seaside crane can pick up a container at the landside I/O point). This is obtained by considering the $N!$ permutations of the N requests and then deciding where to place a marker between the requests so that the requests on the left of the marker are assigned to the seaside ASC and the containers to the right of the marker are assigned to the landside ASC. There are (N+1) ways to place the marker.

In order to deal with this scheduling complexity, we use the most commonly used heuristics to schedule ASCs. Each ASC is scheduled individually. By adding the two schedules together, a total schedule for all requests is given. Based this schedule, the total makespan to handle all requests is determined. The assignment of requests to ASCs is based on their origin or destination and is not part of the scheduling. Landside requests are carried out by L-ASC and seaside requests by S-ASC. In this study, the following scheduling heuristics are tested. The first two are used by Gharehgozli et al. (2015) as benchmark to test their adaptive large neighbor search (ALNS) heuristic. The last one is used by Vis and Carlo (2010) as an operator in their simulated annealing (SA) heuristic.

- *First come first serve (FCFS)*: This strategy handles all requests in the order in which they arrive at the stack (whether retrieval or storage). It assumes that all requests have the same importance and therefore should be handled in whatever order they arrive.
- *Nearest neighbor (NN)*: This strategy specifies that a crane should handle the request which is closest to its current location.
- *2-Opt*: This is an optimization heuristic which first sequences the requests randomly for each crane and determines the total makespan to handle all requests. This makespan is used as a reference point. Then, through a series of mutual request swaps (selecting two random requests of a crane and swapping their positions in the schedule), 2-OPT attempts to find the

schedule with the minimum makespan. In each iteration, two random requests are swapped and the total makespan is again determined. At this point, any of two situations can occur:

- If the makespan to finish all requests with the new schedule is lower than the reference point, the new schedule is accepted. The new makespan becomes the new reference point.
- If the makespan of the new schedule is higher than the reference point, the new schedule is rejected. At this point, the initial reference point stays the same.

The heuristic continues by selecting and swapping two random requests from each crane schedule. For example, based on Theorem 1, for 25 requests, up to 26! different possibilities exist for the combination of two schedules. In order to limit the completion time of the simulation within a reasonable level, 2-OPT is iterated for 1000 times for each run.

5.1.2 Prioritizing of cranes

One of the limitations of a system with twin stacking cranes is that they are not able to pass each other. Therefore, after determining the sequence in which the cranes have to carry out the requests, we need to deal with the interferences. This implies that whenever an interference occurs, one of the cranes needs to wait such that the other crane can handle its request. In every instance of interference, we need to assign priority to one of the cranes to complete its current request first. This is in essence a scheduling problem with 2^M possible priority assignments, where M being the number of interferences. One needs to note that it may be possible to simultaneously resolve all M interferences by resolving just the first interference (e.g. by shifting all of one ASC's requests later by 1 minute). Furthermore, resolving an interference prioritizes one crane and delays the other one which potentially may result in more interferences. Therefore, it is very time consuming to check for all possibilities through an exhaustive enumeration method such as branch & bound. Instead, we use the following three priority rules to assign priority. These rules are based on the ones studied by Carlo & Martínez-Acevedo (2015). In total they study 14 priority rules which also includes the ones proposed by Carlo and Vis (2012), Park et al. (2010), and Dell et al. (2009).

- *Longest total time (Prio1)*: At the beginning of the problem, determine the total time that each ASC needs to complete its requests without considering the interference with the other crane. Give priority to the crane that requires the most time to finish all the assigned requests. This rule gives priority to the bottleneck crane.
- *Longest remaining time (Prio2)*: This rule specifies that at the moment of an interference, the remaining time for each crane to handle all their requests (without considering the other crane) is calculated. The crane with the higher remaining time will be granted preference.
- *Longest time to origin (Prio3)*: This rule specifies that at the moment of interference, the crane with the longer travel time to the location of the next request after completing the current request will be granted preference. With this rule, the long unproductive (i.e., empty) and unavoidable moves need to be performed first.

Note that the schedules have been already determined based on the NN, FCFS, or 2-OPT heuristics in the previous step explained in section 3.2.1. In case of an interference, the crane without priority ensures that the crane with priority can complete its job without any interference. This means that the crane without priority waits until the priority crane has stacked or retrieved its container and left to the next container. Only then the non-priority crane can start or complete its own request.

These priorities are only for settings without a handshake area. For settings with a handshake area, the cranes need to collaborate in the handshake area to complete the split requests. Thus, one of the requirements is the availability of containers. A crane can only pick up a container after it has been brought there by the other crane. At the same time, it needs to be secured that cranes do not interfere.

Following the priority rules here, a "deadlock situation" might arise, where the non-priority crane that needs to deposit a container has to wait while the priority crane has to pick up the container which has not yet been deposited. In order to avoid this, for settings with a handshake area we will use a priority rule which gives priority to the crane which is currently handling a job. Since split requests are consecutively handled by both cranes, such a priority rule means that first a crane needs to deposit the container before that the other one can pick it up. A more detailed explanation is given in section 4.

5.1.3 Selecting the size of the handshake area

In this study, we also investigate whether the size of the handshake area has an effect on the performance. For this reason, the following two different sizes are tested. Since we consider only 20 foot containers, the size of each bay in the following cases is 20 feet.

- *One bay*: All split requests need to be temporarily stored in 1 bay. As the handshake area is the smallest, the chance of having split requests will be the least. However, since the minimum safety distance is also 1 bay, this will result in a higher chance of interferences for requests which need to be temporarily stored or retrieved from the handshake area.
- *Two bays*: All split requests need to be temporarily stored in either of 2 bays. Although a two-bay handshake area results in a higher chance of split requests, the expected number of interferences will be lower. The strategy for determining the storage location of the container in the handshake area is discussed in Section 3.2.5.

Within this study, only settings with two cranes and no reshuffling are studied. Hence, only handshake areas with a width of up to 2 bays are considered; i.e., 1 bay for each crane.

5.1.4 Selecting the number of handshake areas

We also investigate the impact of two separate handshake areas, one area per crane and each 1 bay, with some bays in the middle. The to-be tested strategies for the distance between the handshake areas are:

- *Safety distance*: In this strategy, the distance between the handshake areas is equal to the safety distance considered for the cranes. Since in our simulation the safety distance is at least 1 bay, so for this strategy we also consider that the distance between the handshake areas is 1 bay.
- *Any distance larger than safety distance*: The distance between the handshake areas is larger than the safety distance. The idea is that each crane has to cover a smaller area which shortens the travel times for stacking and retrieving containers. For requests outside this area which need longer travel times, the request are split and the cranes need to collaborate which can be again used to decrease the makespan. On the other hand, compared to the previous strategy, the chance of having a split request is higher as the handshake areas cover a larger footprint and every request crossing this area has to be split. As a result, the number of split requests can increase which may increase the makespan if the cranes are not scheduled in an efficient way.

5.1.5 Selecting a storage location in the handshake area

In settings with a two-bay handshake area, the temporary storage location of the container in the handshake area can impact the total makespan. In this study, the following two strategies are tested:

- *Nearest location to I/O point (Near IO)*: The location which is the closest to the I/O point of the request which is carried out. For example, for a split storage request at the landside I/O point, it is picked up by the L-ASC, and is stacked in the bay in the handshake area nearest to the landside I/O point. It shortens the travel time of the L-ASC but lengthens the time the S-ASC

needs to travel. For a split retrieval request to be retrieved to the seaside, it is picked up by the L-ASC, and is stacked in the bay in the handshake area nearest to the seaside I/O point. It lengthens the travel time of the L-ASC but shortens the time the S-ASC needs to travel.

– *Nearest location to retrieval or storage location (Near Req)*: The location which is closest to the retrieval or storage location. For example, the storage request, mentioned earlier, is stacked in the bay in the handshake area nearest to its storage location. It lengthens the travel time of the L-ASC but shortens the time the S-ASC needs to travel. On the other hand, the retrieval request is stacked in the bay in the handshake area nearest to the retrieval location. It shortens the travel time of the L-ASC but lengthens the time the S-ASC needs to travel.

6. Simulation

In this section the general concepts, scenarios, and parameters of the simulation model is explained. An overview of the steps taken in the simulation model is given Figure 3. All steps will be discussed in detail in the followings.

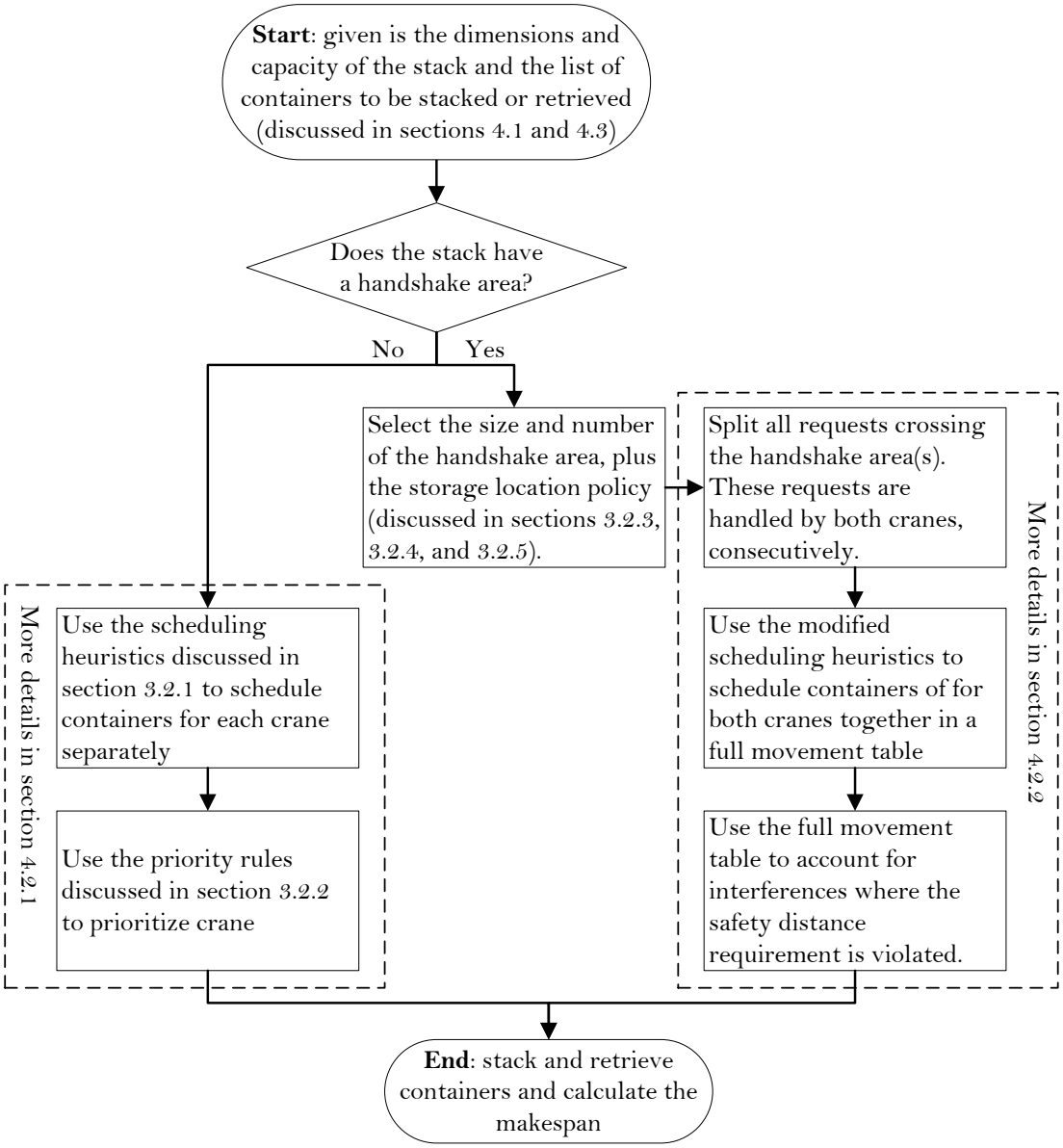


Figure 3. An overview of the simulation model

7. Conceptual design

The simulation model considers a stack with given dimensions and capacity. For simplicity, it is assumed that the inter-bay (y-movement) travel time is longer than the inter-row (x-movement) travel time. For this reason simultaneous x- and y-movements are not considered but rather only the y-movements. The stack is assumed to have a fixed total of 30 bays, not including the I/O points. For the landside, the I/O point lays at bay 1 and for the seaside crane, the I/O point lays at bay 32. Containers are assumed to be 20 feet (6.1 meters) and no separating space between containers in the stack is considered. A travel speed of 1 m/s is assumed for the ASCs which accounts for acceleration and deceleration, the space between containers, 40 foot containers, and other operational and safety measures common in a terminal (see Appendix D for more results on higher speeds). The time required to lower the spreader, release or attach the container and raise the spreader is assumed to be 30 seconds. Furthermore, the safety distance is at least 1 bay. This means there should be a minimum of 1 bay distance between the faces (closest parts) of two cranes. In our simulation study, if we notice that interference occurs, the crane without priority stops at its current location. This assures that the safety distance is always equal to or larger than 1 bay (see Figure 3). In Appendix E, we discuss instances where the safety distance is equal to or larger than 2 or 3 bays.

At the start of the simulation, all requests are assumed to be available at their locations. Normally, container terminal operators use a "block sequencing approach" to stack and retrieve containers (see, e.g., Vis and Roodbergen, 2009). In this approach, a set (i.e., block) of the most urgent storage and retrieval requests is selected from the available requests. The selected requests are then sequenced. When the ASCs are done, a new set is again selected, sequenced, and executed. For storage-requests, these are at the I/O points and for retrieval-requests these are at each container's storage location in the stack. The storage or retrieval locations for all requests are uniformly spread over the 30 bays. The requests are divided between storage and retrieval requests, with R% retrievals. The requests are divided between the landside and seaside crane with PL% of the requests for the landside crane. The model starts with determining the number of storage and retrieval requests (with R% retrievals), for each crane (with PL% requests for the landside crane). Each request is then randomly assigned a location in the stack. Based on the combination of crane (land or sea) and type of request (storage or retrieval), the start and end points of each request is determined.

As an illustrative example, consider a scenario with 3 retrieval requests at bays 6, 25 and 28 (see Table 1). Furthermore, there are 3 storage requests at bays 5, 10 and 20. Crane 1 takes care of 1 retrieval and 2 storages and crane 2 takes care of the rest. The bays lay between 2 and 31 (point 1 and 32 are the I/O points). We use this example in this section to explain the concepts of our simulation.

Table 1. Initial list for a setting without a handshake area

Start location*	End location	Crane number	Request number	Type of request
32	10	2	1	Storage
1	5	1	2	Storage
28	32	2	3	Retrieval
25	32	2	4	Retrieval
6	1	1	5	Retrieval
1	20	1	6	Storage

* Locations are specified based on bay numbers

8. Simulation of the various scenarios

For settings with and without a handshake area, the model works slightly different. Thus, first we focus on settings without a handshake area and then on settings with a handshake area.

8.1.1 Simulation of the settings without a handshake area

After the initial list is known, the final schedule needs to be determined, using the scheduling heuristics discussed in section 3.2.1, and interferences need to be taken into account, using the priority rules discussed in section 3.2.2. Based on the sequence of carrying out the requests, driven from the scheduling heuristics, a movement table is created for each crane which describes the position of the crane at each moment in time. For example, Table 2(a) and (b) show the movements of crane 1 using the FCFS heuristic. For the NN heuristic, in order to make the movement table, first, a matrix is made per crane with the inter-request travel times. Finally, the 2-OPT heuristic starts with the same movement tables as the FCFS heuristic. The difference is that for the FCFS heuristic, the simulation model stops after prioritizing the cranes. However, for the 2-opt heuristic, at the end of scheduling and prioritizing, the total makespan is used as a reference point to find the best schedules in1000 iterations.

Table 2. Movement tables of cranes in the FCFS heuristic

(a) Movement table of crane 1

Start location	End location	Crane Number	Original request number	Crane request number	Type of request
1	5	1	2	1	Storage
6	1	1	5	2	Retrieval
1	20	1	6	3	Storage

(b) Movement table of crane 2

Start location	End location	Crane number	Original request number	Crane request number	Type of request
32	10	2	1	1	Storage
28	32	2	3	2	Retrieval
25	32	2	4	3	Retrieval

Using the movement tables, the simulation model checks whether the distance between the cranes is larger than the safety distance. If not, the crane without priority waits at its location until the "priority crane" has stacked or retrieved its current container and left for the next request. We explain this further using the example presented in Table 1. In **Error! Reference source not found.**(a), the movement table of crane 1 with and without waiting is given. The movement table of crane 2 is given in **Error! Reference source not found.**(b). Comparing these tables reveals that the cranes interfere, when crane 1 moves from bay 1 to bay 20 to stack the container of request 6 and crane 2 moves from bay 32 to bay 10 to stack the container of storage 1. Thus, assuming that crane 2 is the priority crane, crane 1 needs to wait at its current location to avoid interference. In **Error! Reference source not found.**(a), the bold numbers show that crane 1 waits at bay 1 for 224 seconds. In Figures 4(a) and 4(b), the movements of both cranes with and without waiting are shown.

Table 3. The movement tables of crane 1 and crane 2

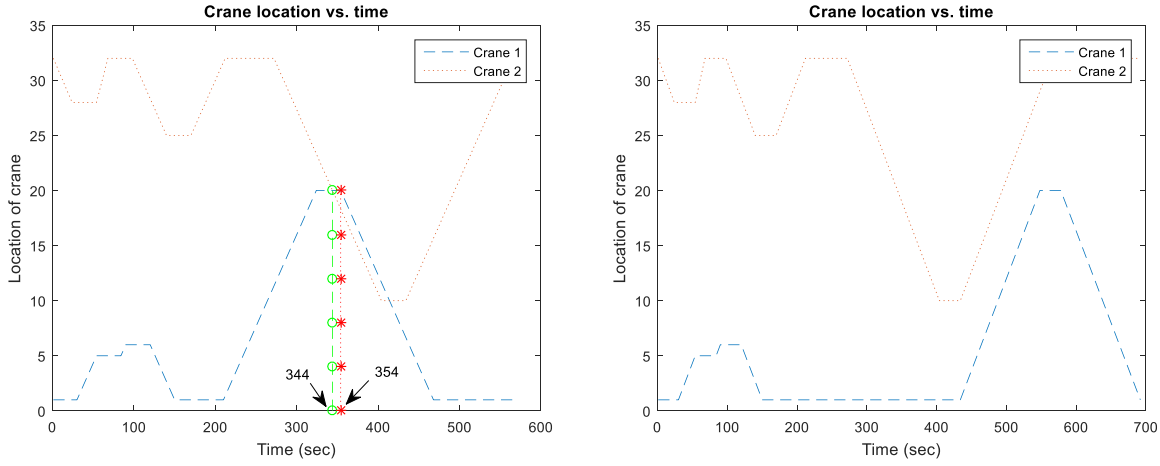
(a) Crane 1

Start location	End location	Time required	Cum. time	
1	1	30	30	
1	5	24	54	
5	5	30	84	
5	6	6	90	
6	6	30	120	
6	1	30	150	
1	1	30	180	
1	1	30	210	
1	20	114	324	Without waiting
20	20	30	354	
20	1	114	468	
1	**1**	**224**	**434**	With waiting
1	20	114	548	
20	20	30	578	
20	1	114	692	

Rest of the table with and without waiting

(b) Crane 2

Start location	End location	Time required	Cum. time
32	28	24	24
28	28	30	54
28	32	24	68
32	32	30	98
32	25	42	140
25	25	30	170
25	32	42	212
32	32	30	242
32	32	30	272
32	10	132	404
10	10	30	434
10	32	132	566



(a) Without crane 1 waiting (b) With crane 1 waiting

Figure 4. Example of waiting due to interference

8.1.2 Simulation of the settings with a handshake area

Settings with a handshake area are handled slightly different. First the actual splitting of requests is discussed. This is followed by scheduling, and finally the interferences.

Splitting of requests

The difference between settings with and without a handshake area is split requests. All requests crossing the handshake area are split and handled by both cranes, consecutively. One crane delivers the split request to the handshake area and then the other one performs the storage to the final storage location or retrieval to the I/O point. In some cases, a "deadlock situation" may arise, where the crane that needs to deposit a container has to wait while the other crane which has the priority picks up the container that has not yet been deposited. In order to ensure the consecutive delivery and retrieval of containers in the handshake area without any deadlocks, the requests of both cranes are handled in a single table called the full movement table. In this table, the end location of each split request is changed to the handshake area. Furthermore, a new request is inserted for the other crane to either store the container in its final location or deliver it to the I/O point.

The full movement table of the illustrative example is presented in Table 4. For instance, request 1 is a split request. So, the end location is changed such that crane 1 delivers the container to the handshake area. Then, a new request is inserted after request 1 in order for crane 2 to pick up and stack the container. A column shows whether a request is part of a split request and if it is the first or second move. If a request is he second move of a split request, its start time is compared with the end time of the first move. If it is smaller, the crane handling the second move needs to wait at its previous location until the other crane has deposited the container and has the handshake area. In Table 4, crane 1 waits for 156 seconds before it can pick up container 1. For request 6, crane 2 waits for 366 seconds.

Table 4. Full movement table for cranes 1 and 2 after checking for pickup times for split requests

Start location	End location	Crane	Request number	Split sequence number	Start time crane 1	End time crane 1	Start time crane 2	End time crane 2	Blocking time for split requests
32	16	2	1	1	0	0	0	156	0
16	10	1	1	2	252	348	0	0	156
1	5	1	2	0	402	486	0	0	0
28	32	2	3	0	0	0	228	312	0
25	32	2	4	0	0	0	354	456	0
6	1	1	5	0	492	582	0	0	0
1	16	1	6	1	582	732	0	0	0
16	20	2	6	2	0	0	918	1002	366

Scheduling

Scheduling the cranes and ensuring that a container can be only picked up from the handshake area once it has been deposited there is difficult. Since scheduling is not the main focus of this study, we schedule all split requests such that the initial request and the new request are always consecutive in the full movement table. The scheduling heuristics incorporate this requirement as follows.

First come first serve and 2-OPT. The first come first serve and 2-OPT heuristics work similarly for settings with a handshake area as for settings without one. The difference is that in the 2-OPT heuristic, two requests are randomly selected and their positions are swapped in the full movement table. Furthermore, if a to-be switched request turns out to be a split request, the position of the accompanying request (to or from the handshake area) is also switched.

Nearest neighbor. In order to ensure that split requests are handled in sequence, first the total makespan for each cranes is determined, assuming the other crane is idle. Priority is given to the crane with the longest time to complete all its requests. The schedule of the priority crane is then inserted into the full movement table. This table is checked for split requests. If a requests is split, the accompanying request of the other crane is inserted into the table. After handling the last split request, the NN rule is applied for the non-priority crane to determine the sequence of the remaining "non-split" requests.

Interference

As a final step, the full movement table is used to account for other interferences where the safety distance requirement is violated. Starting at the top of the table, for each request belonging to a crane, the start and end times are compared with the start and end times of all requests of the other crane appearing below this current request in the table. If interference occurs, the times in the full movement table have to be modified by prioritizing one of the cranes. In order to avoid deadlocks, the crane currently handling a request, higher in the full movement table, has priority. This is why in Table 3, the crane which brings the container of a split request to the handshake area appears first.

9. Parameters of the simulation

With the simulation tool, the various proposed heuristics are tested under different settings, depending on whether there is a handshake area. In total 864 different scenarios are tested for this study.

For settings without a handshake area there are 3 design factors: *Number of requests (5, 15 and 25)*, *percentage of requests to the landside (25%, 50%)* and *percentage of requests which are retrievals (25%, 50% and 75%)*. In terms of the strategies there are 2 factors: *Scheduling (FCFS, NN, 2-OPT)* and *priority (Prio1, Prio2, Prio3)*. Each combination of design factors represents a problem scenario (e.g. 5 requests, 25% land requests, 25% retrievals) while each heuristic-level combination represents a solution to these problems (e.g. FCFS, Prio1). This implies that a total of 162 different scenarios are tested.

For settings with one handshake area, there are 5 factors for the design: *Number of requests (5, 15 and 25), percentage of requests to the landside (25% and 50%), percentage of requests which are retrievals (25%, 50% and 75%), location of the handshake area (-5, 0 +5),* and *size of the handshake area (1 bay and 2 bays)* . In terms of the strategies there are 2 factors: *Scheduling (FCFS, NN and 2-OPT) and storage location in the handshake area (Near Req. and Near IO)*. As a result, a total of 486 different scenarios are tested.

For settings with two handshake areas, the first three design factors (*Number of requests, percentage of requests to the landside, percentage of requests which are retrievals*) are the same as for settings with one handshake area. However, the size of each handshake area is only 1 bay. Furthermore, the handshake areas are in the middle and the number of bays between them is 1 or 3 bays. Finally, the scheduling and storage strategies are the same as settings with one handshake area. In total, 216 different scenarios are tested.

10. Results

In this section we discuss our results and findings. In section 5.1, the results for settings without a handshake area are given. In Section 5.2, the results for settings with handshake area are presented. In section 5.3, the managerial insights for settings with and without a handshake area are discussed and the settings are compared with each other.

In Table 5, the variables with their notation and description can be found. Based on this table, The number of retrievals is determined using the following equation, $R(\%) \times N$. Furthermore, The number of requests to the landside is detemined using the following equation, $PL(\%) \times N$. The numbers are rounded to nearest integer. The simulations were run on a personal notebook with a 2.2GHz Intel® Core™ i7 processor and 8GB RAM. The simulation model is developed in MATLAB® R2012b. For each instance, 100 replications were run. Each run took between 0.7 and 9 seconds, averaging at 5.4 seconds per run.

Table 5. Explanation of symbols

Variable	Description	Variable	Discerption
N	Total number of requests	$LC_{L,M,S}$	Location of handshake area (landside, middle or seaside)
$R\ (\%)^{*}$	Percentage of retrievals	K	Instance number
$PL\ (\%)^{**}$	Percentage or requests to landside crane	Z_{SH}	Makespan with scheduling heuristic SH
$SL_{IO,\ Req}$	Storage location heuristic (near IO or near request)	B_{SH}	Blocking time with scheduling heuristic SH
$SZ_{1,2}$	Size of handshake area (in number of bays: 1 or 2)	$G_{SH}(\%)$	Percentage difference compared to the best found value
$DTZ_{1,3}$	Number of bays in between the two handshake areas		

11. Settings without a handshake area

In Table 6 and Table 7, the results of settings without a handshake area are given. The results of the FCFS, NN, and 2-OPT heuristics are represented by respectively $Z_{FCFS,}$ Z_{NN}, $Z_{2\text{-}OPT}$ (in seconds). For each heuristic, a sub-division is made per priority rule. The results show that with an increase in number of requests from 5 to 15 (i.e. 200%), the makespan increases from 100% to 150%. Similarly, the blocking time increases but less rapidly. This implies that the system becomes more efficient with more requests since the makespan and blocking time to number of requests ratio drops. With respect to PL(%)-R(%), the best results happen when there is a 50%-50% split. The worst case is when there is a 25%-75% split where one crane needs to do a larger number of requests. Since the blocking time for 25%-75% split is almost the same as the other instances, this setting mainly results in a long makespan for one crane. Thus, the best performance is achieved when requests are evenly distributed.

Table 6. Blocking time (in Sec.) for the settings without a handshake area

				B_{FCFS}			B_{NN}			$B_{2\text{-opt}}$		
K	***N***	***PL (%)***	***R (%)***	**Prio1**	**Prio2**	**Prio3**	**Prio1**	**Prio2**	**Prio3**	**Prio1**	**Prio2**	**Prio3**
1	5	25	25	57.75	41.31	40.79	66.58	32.31	19.50	66.58	32.31	19.50
2	5	25	50	69.92	39.17	106.90	35.08	26.13	96.98	35.08	26.13	77.94
3	5	25	75	43.92	185.88	113.22	21.06	96.78	174.36	21.06	76.98	134.16
4	5	50	25	126.63	244.44	501.52	133.27	162.35	83.25	69.46	111.40	53.31
5	5	50	50	123.63	164.02	384.17	75.75	65.08	141.58	40.10	41.60	94.04
6	5	50	75	103.26	299.94	149.52	76.68	217.62	289.68	59.16	102.90	155.82
7	15	25	25	576.75	268.85	279.40	497.25	192.58	119.77	257.42	175.96	127.15
8	15	25	50	589.27	424.79	716.02	463.50	147.46	115.62	263.65	163.90	94.10
9	15	25	75	627.72	445.44	618.90	647.70	209.76	137.70	256.44	155.22	99.60
10	15	50	25	582.46	531.17	559.73	662.83	501.29	568.56	204.52	221.25	229.21
11	15	50	50	563.13	893.48	831.46	436.90	488.71	538.33	103.85	142.27	122.19
12	15	50	75	734.40	593.40	745.02	713.58	429.54	364.26	218.64	212.40	170.40
13	25	25	25	1547.48	716.31	569.54	1200.98	445.79	318.29	658.27	394.10	294.58
14	25	25	50	1163.19	984.52	642.00	1089.98	290.83	223.67	405.81	285.58	132.98
15	25	25	75	1348.62	477.96	413.82	1232.22	390.42	249.18	451.50	285.42	173.94
16	25	50	25	1674.98	882.58	1023.46	1492.15	969.46	1120.33	447.46	412.27	337.27
17	25	50	50	1138.62	1293.87	763.38	1031.02	642.35	664.90	251.19	272.54	228.92
18	25	50	75	1482.42	883.44	818.46	1555.98	714.84	475.26	458.52	376.68	301.80
			Average	*697.45*	*520.59*	*515.41*	*635.14*	*334.63*	*316.73*	*237.15*	*193.83*	*158.16*

Table 7. Makespan (in Sec.) for the settings without a handshake area

				Z_{FCFS}			Z_{NN}			$Z_{2\text{-opt}}$			G_{FCFS}			G_{NN}			$G_{2\text{-opt}}$		
K	***N***	***PL (%)***	***R (%)***	**Prio1**	**Prio2**	**Prio3**	**Prio1**	**Prio2**	**Prio3**	**Prio1**	**Prio2**	**Prio3**	**Prio1**	**Prio2**	**Prio3**	**Prio1**	**Prio2**	**Prio3**	**Prio1**	**Prio2**	**Prio3**
1	5	25	25	1016.08	1022.88	1031.71	995.88	992.08	997.79	995.73	**991.92**	997.63	2.44%	3.12%	4.01%	0.40%	0.02%	0.59%	0.38%	**0.00%**	0.58%
2	5	25	**50**	936.46	936.87	984.98	835.79	834.87	879.29	835.23	**834.31**	869.85	12.24%	12.29%	18.06%	0.18%	0.07%	5.39%	0.11%	**0.00%**	4.26%
3	5	25	**75**	959.88	1065.48	1031.40	844.80	886.08	935.34	**844.80**	876.18	915.24	13.62%	26.12%	22.09%	0.00%	4.89%	10.72%	**0.00%**	3.71%	8.34%
4	5	**50**	**25**	819.63	873.63	1022.31	825.63	818.60	793.73	762.46	777.12	**761.88**	7.58%	14.67%	34.18%	8.37%	7.44%	4.18%	0.08%	2.00%	**0.00%**
5	5	50	**50**	837.69	853.73	986.31	770.13	754.10	804.23	733.79	**727.50**	767.48	15.15%	17.35%	35.57%	5.86%	3.66%	10.55%	0.86%	**0.00%**	5.50%
6	5	50	**75**	847.20	945.54	886.32	765.84	832.14	876.60	**740.64**	764.28	798.00	14.39%	27.67%	19.67%	3.40%	12.35%	18.36%	**0.00%**	3.19%	7.74%
7	**15**	**25**	**25**	2794.90	2758.62	2830.85	2704.85	2661.69	2684.60	**2528.65**	2529.81	2533.90	10.53%	9.09%	11.95%	6.97%	5.26%	6.17%	**0.00%**	0.05%	0.21%
8	15	25	**50**	2654.19	2695.90	2894.71	2226.06	2179.96	2214.87	2064.23	**2062.04**	2075.31	28.72%	30.74%	40.38%	7.95%	5.72%	7.41%	0.11%	**0.00%**	0.64%
9	15	25	**75**	2755.50	2810.88	2936.70	2575.92	2523.90	2547.18	2382.72	**2381.94**	2390.34	15.68%	18.01%	23.29%	8.14%	5.96%	6.94%	0.03%	**0.00%**	0.35%
10	15	**50**	**25**	2472.87	2337.06	2433.75	2428.50	2286.12	2429.65	2049.29	**2035.21**	2044.38	21.50%	14.83%	19.58%	19.32%	12.33%	19.38%	0.69%	**0.00%**	0.45%
11	15	50	**50**	2233.04	2308.90	2319.17	1997.08	1904.31	1985.31	1665.75	**1652.71**	1664.25	35.11%	39.70%	40.33%	20.84%	15.22%	20.12%	0.79%	**0.00%**	0.70%
12	15	50	**75**	2525.04	2320.98	2483.22	2327.76	2137.56	2201.76	1957.68	**1942.56**	1945.38	29.99%	19.48%	27.83%	19.83%	10.04%	13.34%	0.78%	**0.00%**	0.15%
13	**25**	**25**	**25**	4667.31	4624.15	4653.46	4403.83	4335.17	4388.48	4004.19	**4003.67**	4034.48	16.58%	15.50%	16.23%	9.99%	8.28%	9.61%	0.01%	**0.00%**	0.77%
14	25	25	**50**	4407.75	4626.81	4538.08	3663.69	3547.38	3602.77	**3400.27**	3405.46	3427.73	29.63%	36.07%	33.46%	7.75%	4.33%	5.96%	**0.00%**	0.15%	0.81%
15	25	25	**75**	4569.36	4430.7	4555.44	4168.8	4076.04	4126.14	3820.32	**3817.02**	3842.64	19.71%	16.08%	19.35%	9.22%	6.79%	8.10%	0.09%	**0.00%**	0.67%
16	25	**50**	**25**	4334.08	3848.19	4028.37	4002.00	3769.15	3947.88	3261.69	**3240.17**	3276.06	33.76%	18.77%	24.33%	23.51%	16.33%	21.84%	0.66%	**0.00%**	1.11%
17	25	50	**50**	3933.00	3718.33	3647.65	3400.50	3012.06	3095.71	2767.21	**2713.85**	2752.44	44.92%	37.01%	34.41%	25.30%	10.99%	14.07%	1.97%	**0.00%**	1.42%
18	25	50	**75**	4244.40	3722.94	3810.60	3943.44	3437.58	3486.72	3158.52	**3121.98**	3147.30	35.95%	19.25%	22.06%	26.31%	10.11%	11.68%	1.17%	**0.00%**	0.81%
		Average		2611.58	2550.09	2615.28	2382.25	2277.15	2333.22	2109.62	2104.32	2124.68	21.53%	20.88%	24.82%	11.30%	7.76%	10.80%	0.43%	0.51%	1.92%

11.1.1 Scheduling of containers for each crane

Table 8 summarizes Table 6 and Table 7, divided based on the scheduling heuristics. Based on the ranking of the makespan of the heuristics (RankZ), the 2-OPT heuristic generates the best results. This is expected since 2-OPT is an adaptation of the other two heuristics which runs for a number of iterations in an attempt to optimize the results. Furthermore, 2-OPT also generates the least blocking time. Finally, NN outperforms the FCFS heuristic both on makespan and blocking time.

Table 8. Makespan and blocking time (in Sec.) for a setting without a handshake area

SH	*P*	Z_P	$Rank_Z$	W_P	$Rank_B$
FCFS	Prio1	2611.58	8	697.45	9
	Prio2	2550.09	7	520.59	7
	Prio3	2615.28	9	515.41	6
NN	Prio1	2382.25	6	635.14	8
	Prio2	2277.15	4	334.63	5
	Prio3	2333.22	5	316.73	4
2-OPT	Prio1	2109.62	2	237.15	3
	Prio2	2104.32	1	193.83	2
	Prio3	2124.68	3	158.16	1

11.1.2 Prioritizing the cranes

In Figure 5, the makespan for each priority rule per scheduling heuristic is given. It shows that priority rule 2 slightly outperforms the other ones. It is expected that the priority rule with the lowest makespan also results in the lowest blocking time. However, as shown in Figure 6, priority rule 3 results in the lowest blocking time. This is because the makespan is determined by the maximum time of either crane to finish all its requests while the total blocking time is determined by the sum of the blocking times of both. Clearly, in some cases it is beneficial for a system to have one crane wait longer to ensure that the other crane completes more requests, instead of constantly switching priority. In the latter case, the cranes are constantly waiting for each other to complete a certain job with the result that the makespan of both cranes is leveled. Although the blocking time is an interesting measure, eventually the makespan indicates which priority rule performs best, namely priority rule 2.

Overall, no significant difference exists among the priority rules in terms of their makespan. The largest difference can be seen in NN with a difference of 4.4% for priority rule 1 compared to priority rule 2. Despite this, it is evident that priority rule 1, which gives the priority to the crane with the longest makespan assuming no interferences, is outperformed by the other two rules on the total blocking time.

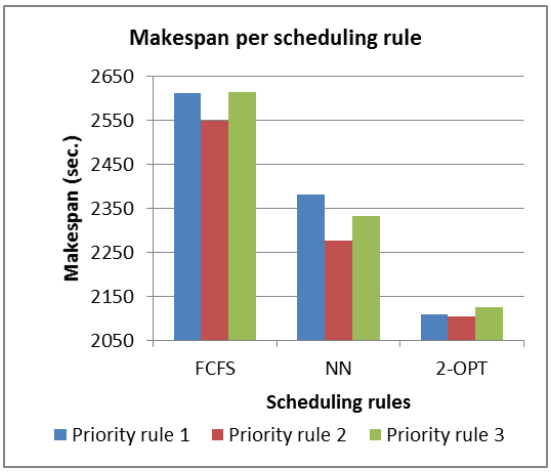


Figure 5. Makespan per priority rule per scheduling heuristic

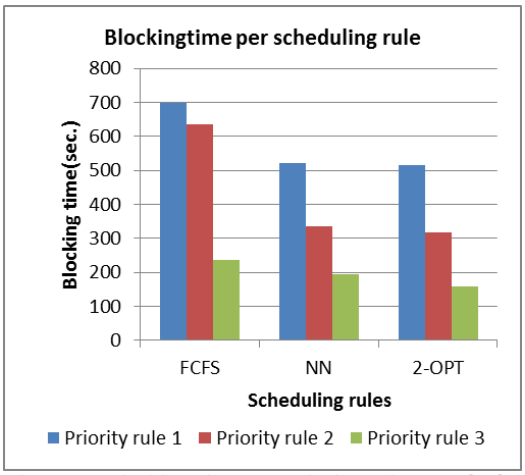


Figure 6. Blocking time per priority rule per scheduling heuristic

12. Settings with a handshake area

In appendix A to C, the complete results of the instances with one or two handshake areas are given. We here present a summary of the results in terms of both the makespan and the blocking time. In sections 5.2.1.-5.2.4, results of settings with one handshake area are presented. The results with two handshake areas are discussed in section 5.2.5. A comparative summary is given in section 5.2.6.

12.1.1 Scheduling of containers for each crane

In Table 9, a summary of the makespan and blocking time per scheduling heuristic is given. For each scheduling heuristic, the results are classified based on the size of the handshake area (*SZ*) and storage location in the handshake area (*SL*). Based on the rankings of the makespan, 2-OPT outperforms the other two heuristics based on the makespan. Furthermore, 2-OPT results in the least blocking time.

Table 9. Summary of results (in Sec.) per scheduling heuristic

	SZ	*SL*	*LC*	Z_{SH}	G_{SH}(%)	$Rank_Z$	B_{SH}	G_B(%)	$Rank_B$
FCFS	1 bay	-	2804.38	0.18	9	1516.57	0.52	6	2804.38
	2 bay	near IO	2787.05	0.17	6	1486.62	0.51	4	2787.05
		near req.	2793.68	0.17	8	1508.8	0.52	5	2793.68
NN	1 bay	-	2787.86	0.17	7	1766.83	0.59	9	2787.86
	2 bay	near IO	2755.43	0.16	4	1713.87	0.58	8	2755.43
		near req.	2760.44	0.16	5	1690.26	0.57	7	2760.44
2-OPT	1 bay	-	2330.45	0.01	3	751.06	0.04	3	2330.45
	2 bay	near IO	2306.97	0.00	1	721.66	0.00	1	2306.97
		near req.	2315.98	0.00	2	729.01	0.01	2	2315.98

12.1.2 Selecting the handshake area location

Another decision is the location of the handshake area in the stack at $1/3^{rd}$ of the stack (skewed towards the landside), in the middle or at $2/3^{rd}$ of the stack (skewed towards the seaside). In

Figure 7, the comparison for handshake area location is depicted for each combination of size of the handshake area (either 1 or 2 bays), the storage location in the handshake area (either IO or Req.) and the scheduling heuristic (FCFS, NN or 2-OPT). In this figure, only the results of instances with $PL = 50\%$ are depicted. It can be observed that the middle handshake area results in the best performance. This is not a surprising result as the requests are distributed evenly between the cranes.

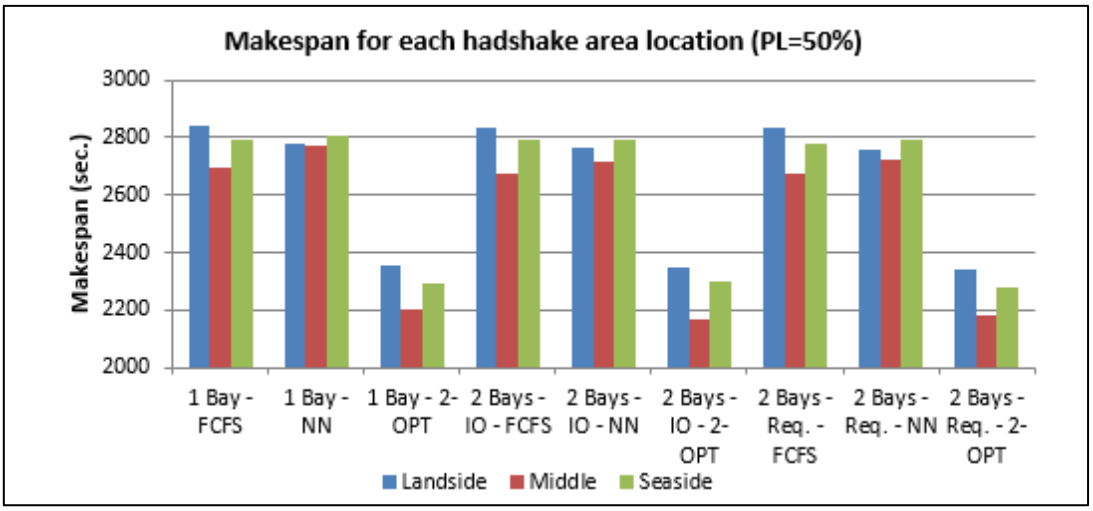
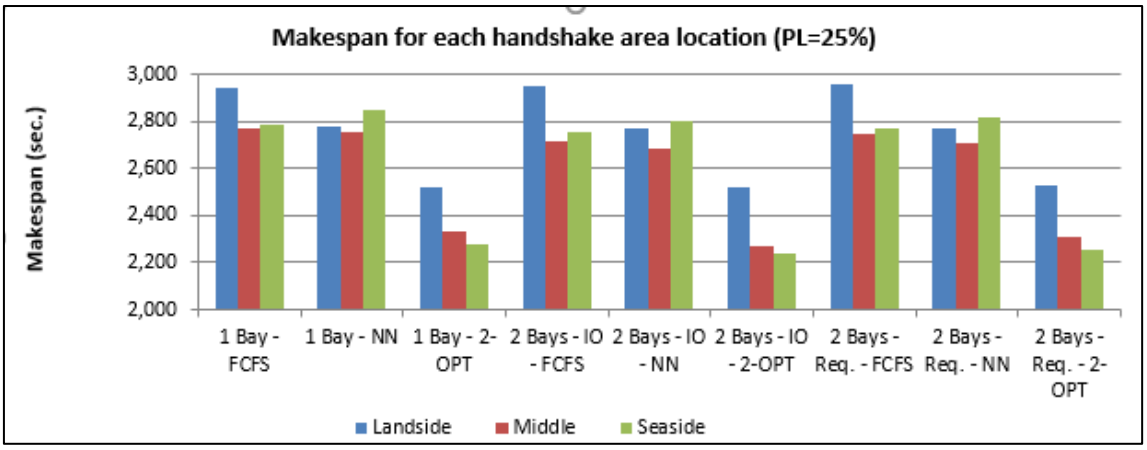


Figure 7. Makespan of the cranes split into the location of the handshake area for instances with PL=50%



In Figure 8, the three different handshake area locations are split in the same division as in Figure 7. The main difference is that, the instances with a request split of 25% are towards the landside crane (PL=25%). From this figure, we make two observations:

- For FCFS and NN, there is a preference in placing the handshake area in the middle with an uneven distribution of requests.
- For 2-OPT, there is a preference in placing the handshake area at the side which has more requests (in our case this is the seaside). Every time a split request crosses the handshake area, a seaside handshake area results in a higher chance of split requests of the seaside. This means that the landside crane will have a higher chance of additional requests compared to the seaside crane. As a result, the "busy" seaside crane will have few additional requests and can focus on its initial requests. The landside crane on the other hand will have quite a few additional requests, but it has fewer initial requests. Thus, there is a better balance of number of initial requests and additional split requests from the other crane.

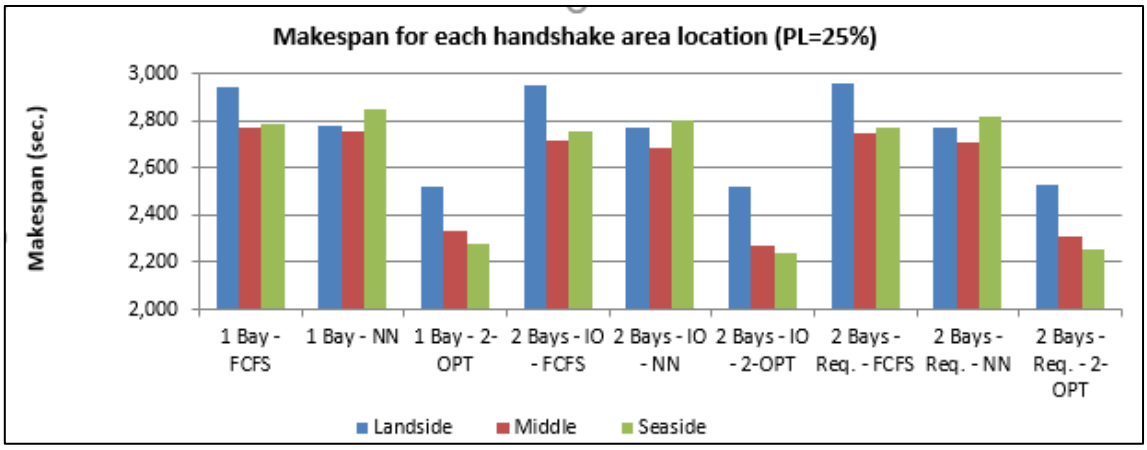


Figure 8. Makespan of the cranes split into the location of the handshake area for instances with PL=25%

12.1.3 Selecting a storage location

For a setting with a handshake area of 2 bays, it needs to be decided where to place the container in the handshake area. Based on Figure 9, the location of a container does not affect the makespan. The difference varies between 0% and 1.35%. The reason is that this decision only affects a small fraction

of containers which cross the handshake area. Furthermore, the handshake area is relatively small so selecting a storage location in the handshake area will not affect the results significantly.

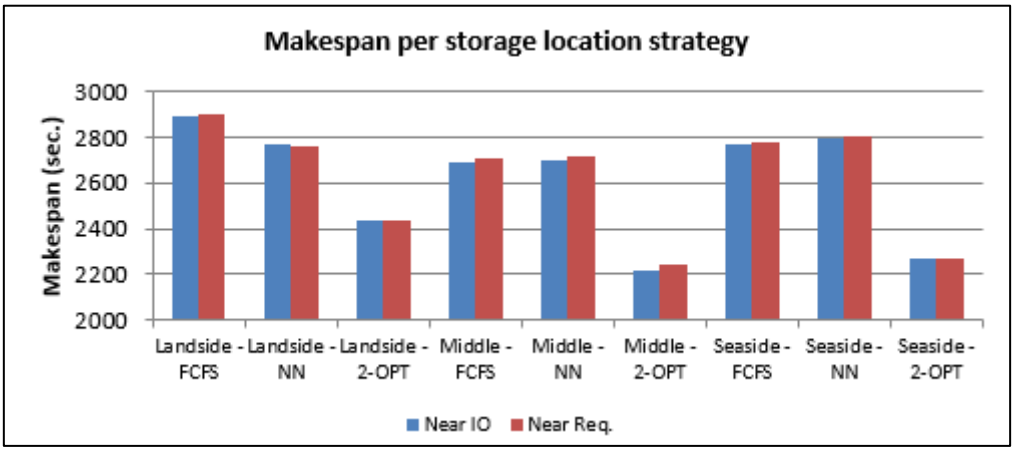


Figure 9. Makespan for each storage location strategy per stack setting

12.1.4 Selecting the size of the handshake area

Another key aspect is the size of the handshake area. The bigger the size of handshake area is, the more requests will be split. At the same time fewer interferences might occur since the cranes will only interfere when they place the containers in the handshake area (in case of I/O storage strategy) or when they pick up the containers from the handshake area (in case of close to request storage strategy). Based on Figure 10, a handshake area of 2 bays results in a better performance than a handshake area of only 1 bay. This is especially the case when the handshake area is located in the middle of the stack, then the difference can be up to 2.5%.

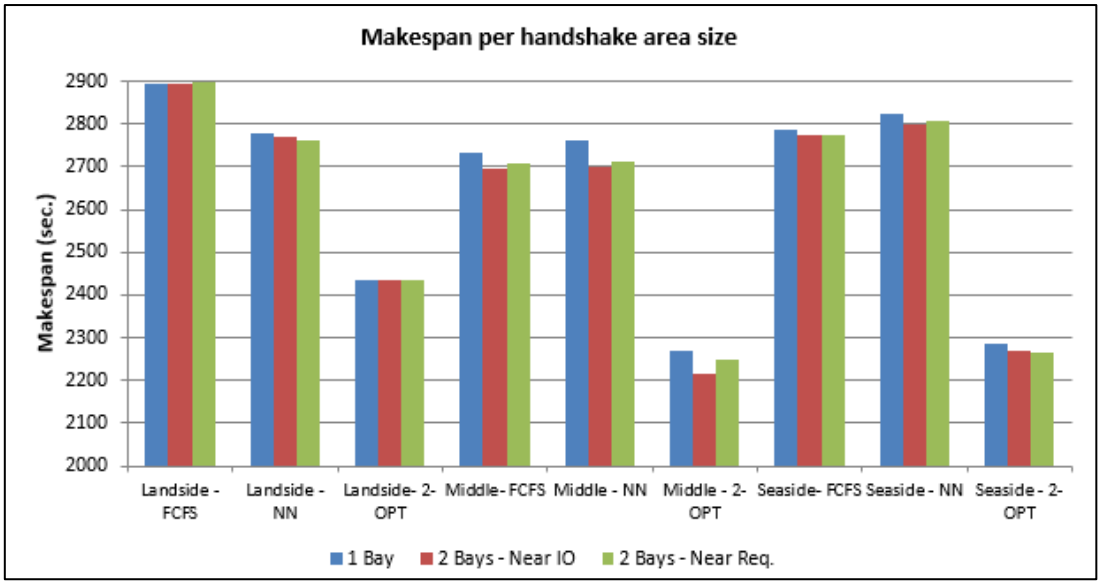


Figure 10. Makespan of different handshake area size vs. handshake area location and sequence

12.1.5 Selecting the number of handshake areas

Table 10 presents the results for settings with two handshake areas. *DTZ* is the number of non-exchange bay(s) in between the handshake areas. In line with the previous findings, settings with two handshake areas also performs best with the 2-OPT scheduling heuristic. In addition, placing the container in the handshake area closer to the I/O point results in a less blocking time and a lower makespan. When comparing the handshake areas, the makespan is the best with 1 bay in between. For

each combination of sequence and storage strategy, 1 bay difference results in a lower makespan compared to a difference of 3 bays. When the difference increases beyond 1 bay, the makespan increases due to two reasons. First, the chance of delay due to waiting for a container to be deposited in the handshake area increases. Second, the chance of a split request increases.

Table 10. Summary table for the results (in Sec.) of settings with two handshake areas

DTZ	*SL*	*SH*	Z_{SH}	G_{SH}	B_{SH}	G_{SH}
1 bay	Near IO	FCFS	2599.65	18.09%	1239.20	60.79%
		NN	2585.47	17.64%	1401.61	65.33%
		2-OPT	2129.34	0.00%	485.94	0.00%
	Near Req.	FCFS	2668.50	20.20%	1330.74	63.48%
		NN	2664.64	20.09%	1467.57	66.89%
		2-OPT	2224.15	4.26%	617.03	21.25%
3 bays	Near IO	FCFS	2655.46	19.81%	1312.69	62.98%
		NN	2659.08	19.92%	1536.41	68.37%
		2-OPT	2191.46	2.83%	548.50	11.41%
	Near Req.	FCFS	2706.37	21.32%	1399.56	65.28%
		NN	2716.30	21.61%	1570.30	69.05%
		2-OPT	2253.00	5.49%	625.22	22.28%

12.1.6 Comparing the results for settings with one and two handshake areas

In Figure 11, the various systems with one and two handshake areas are plotted against their blocking time for an instance with N=15, PL=50% and R=50%. In this figure, the settings with two handshake areas (all the red marks in dashed and shaded ovals) outperform settings with one handshake area. Even a system where the difference between the handshake areas is 3 bays, performs better in terms of both makespan and blocking time. Thus, it seems that the increase of split requests (as is the case with two handshake areas) has a less impact on the makespan than the increase of interferences (as is the case with one handshake area).

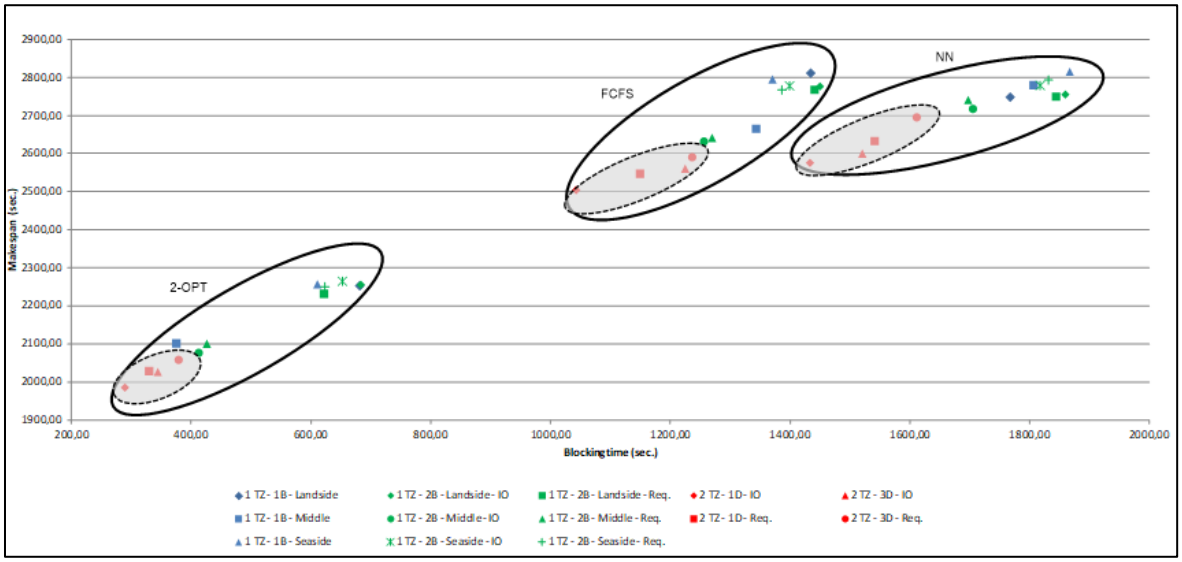


Figure 11. Makespan vs. blocking time (in Sec.) for settings with one and two handshake areas (N=15, PL=50%, R=50%)

13. Comparing settings with and without a handshake area

In this section, we first discuss our general findings and then focus on managerial insights.

13.1.1 Findings and discussions

The main objective of simulation is finding a set of robust rules for each decision variable such that the total makespan is minimized. For settings without a handshake area, we found that the scheduling rule 2-OPT with priority rule 2 results in minimum makespan. For settings with a handshake area, the best performance is achieved for a system with 2 handshake areas, with 1 bay in between, and a storage strategy which places the container closest to the I/O point of the request. On average, the settings without a handshake area slightly perform better than the settings with a handshake (compare Table 7 with Table 9). Below, we summarize the impact of each decision variable on the makespan.

Scheduling. 2-OPT results in the best performance. Among the decision variables, scheduling is the most influential variable on the makespan (up to 44% difference between the different scheduling heuristics). This implies that the choice of scheduling rule should be prevalent over other decision variables. A two sample t-test with equal sample sizes ($H1: Z_{\text{2-OPT}} \neq Z_{\text{FCFS or NN}}$) for each scenario shows that almost in all cases, the 2-OPT policy results in a shorter makespan compared to the FCFS or NN policy at a 5% level significance level (e.g., $p - value < 0.05$). 2-OPT fails to provide better results for an instance with $N = 5$, $R = 75\%$, and $PL = 25\%$ (see Table 7). In this instance, most requests are retrieval requests to the seaside and thus there is little room to improve the makespan.

Storage location in the handshake area. The results show that the storage location within the handshake area has no significant effect on the performance. A two sample t-test with equal sample sizes ($H1: Z_{\text{Near IO}} \neq Z_{\text{Near Req}}$) for each scenario shows the makespan of the Near IO policy and the makespan of the Near Req policy do not differ significantly at a 5% level significance level (e.g., $p - value > 0.05$).

Location of the handshake area. The results show that the location of the handshake area, the distribution of requests between the cranes, and the scheduling heuristic are related. The t-test results differ per instance, which shows that the location depends on each specific instance. However, on average, when there is an equal split of requests between the cranes, for all scheduling heuristics, the best handshake area location is in the middle of the stack. When there is an uneven distribution of requests, for FCFS and NN, the best location is in the middle of stack. For 2-OPT, the best location is the side of the stack which has the I/O point of the crane with the majority of requests. In addition, the choice of location has a much lower effect on the makespan than the choice of scheduling heuristic. This suggests that the location decision should be based on the distribution of requests and the scheduling heuristic.

Size of the handshake area. The bigger the size of handshake area is, the more requests will be split. At the same time fewer interferences occur. The t-test result differs per instance, so no general rule can be driven. However, the results show that a handshake area of 2 bays on average results in a better performance than a handshake area of only 1 bay. When the handshake area is located in the middle of the stack, the difference can be up to 2.5%.

Number of handshake areas. The results show that on average, settings with two handshake areas and a difference of 1 bay have the best performance. The outperformance compared to settings with two handshake areas and a difference of 3 bays is mainly due to the increased number of split requests in the latter settings which need extra rehandlings and more collaboration. Furthermore, settings with two handshake areas outperform settings with one handshake area. A two sample t-test with equal sample sizes ($H1: Z_{2\text{ HS}} \neq Z_{1\text{ HS}}$) for these scenarios show that two handshake areas results in a shorter makespan as compared to one handshake area at a 5% level significance level (e.g., $p - value < 0.05$).

13.1.2 Managerial insights

An automated container terminal can be divided into three main areas namely seaside, stacking area and landside. At the seaside, quay cranes load and unload containers from ships berthed along the quay. Quay cranes pick up or drop off containers on automated guided vehicles which transport containers from the seaside to the stacking area where ASCs take over. At the landside, straddle carriers or terminal trucks transport containers either between the ASCs and trucks or trains. From the practical point of view, it is beneficial to implement a system of twin ASCs in many instances (e.g., Ng, 2005, Saanen & Valkengoed, 2005, Vis & Carlo, 2010, and Gharehgozli et al., 2015). One advantage is that the landside and seaside operations can be decoupled. Therefore, in contrast with the single ASC configuration, handling trucks and trains at the landside does not impact the turnaround time of a ship at the seaside whose containers are stacked in the same stack. However, despite its benefits, it seems that container terminals are not willing to implement such a system yet. Euromax Container Terminal is among the few terminals who has implemented this configuration. Supposedly, the main reason is the complexity involved in scheduling the cranes and dealing with the interferences. Nevertheless, the complexity can be addressed by using automation and handled by computer software. We cover this gap by proposing methodologies that can be plugged into the automation process.

Our results show that settings without a handshake area outperform settings with a handshake area. In addition, the results show that two factors have a major effect on the makespan: the scheduling heuristic and the method used to determine the split request. More sophisticated scheduling methods might result in better results for settings with a handshake area compared to settings without a handshake area. Such methods can specifically improve the stacking operations in the following two cases: 1) in the setting with handshake area, the additional waiting times for the ASC crane imposed due to safety distance constraints are noteworthy, 2) the additional times due to double handling in the handshake area take up a big portion of total storage and retrieval time. Indeed, in unrealistic cases where the number of bays is significantly large (i.e. the container block is too long) those two effects can die out. Moreover, in practice, the container terminal operators might prefer to use handshake area in a twin ASC seaport terminal while sacrificing the performance. This is due to the fact that when the seaside crane need to access a container at landside I/O point, it might be unsafe for a landside crane to leave the block. A handshake area will avoid such situations.

Last but not lease, note that a handshake area gives schedulers flexibility to re-sequence the handling of containers to improve productivity. For example, a one-bay handshake area can stack multiple containers (the number of containers equals the number bays times the number of rows; with one unburied container on top of each pile). This extra buffer can be used to smoothen the operations and deal with the interferences. Such an area can be also used for reshuffling containers.

14. Conclusion and future research

Containerized transportation has become an essential part of world trade during the past decades. A large terminal annually handles millions of containers, which must be stored for a certain length of time in multiple container stacks in the stacking area. Therefore, an efficient stacking operations can significantly affect the overall terminal performance. Terminal operators always look for new technological and methodological advancements to improve the stacking operations. This paper focus on both aspects by studying the effect of implementing a handshake area in a container stack with twin collaborating ASCs. Settings with and without a handshake area are tested. In settings without a handshake area, each request is only handled once and by one crane. In settings with a handshake area, requests can be split among the two cranes. In such a case, one crane will deliver the container to the handshake area and the other crane will retrieve the container from the handshake area and finish the request. Various decision variables which affect both settings are studied. These decision variables are

(1) how to schedule the requests, (2) how to handle interference of cranes, (3) the size and location of the handshake area, (4) the number of handshake areas and, (5) how to select a storage location in the handshake area. For each decision variable, a set of rules are tested with respect to their influence on the performance (total makespan and total blocking time). The results show that for each decision variable, the combination of rules influence the performance of settings with and without handshake area. The terminal managers can use our results to make more informed decisions.

Being among the first studies on the impact of a handshake area on container handling operations, several additions to the study could provide more insights. In this study, the inter-row movements and reshufflings are not considered. In reality, cranes cannot immediately acquire the desired container and need to make extra movements by moving other containers. It would be interesting to study how reshuffling affects the performance of settings with a handshake area. Future work may also focus on understanding the process of how to split requests and assign them to cranes. In this study it is assumed that all requests which cross the handshake area are split. In reality, there might be more sophisticated methods. The other addition which may impact the makespan of the cranes is the size of containers. In this study, we only consider 20-foot containers, whereas 40-foot containers are also stacked in container stacks. The assumption on the availability of containers is also worthwhile investigating. This means that not all containers are available at the terminal in advance. So it needs to be studied how this impacts the schedules. Another important research topic is to study the effect of more advanced scheduling heuristics and priority rules on the makespan. Finally, it is interesting to see how the assignment of storage locations affects settings with and without a handshake area.

Acknowledgement
The authors would like to thank the editor and anonymous reviewers whose constructive comments and suggestions helped us to improve the quality of this paper.

References

Angeloudis, P., & Bell, M. G. (2011). A review of container terminal simulation models. *Maritime Policy & Management*, *38*(5), 523–540.

Bierwirth, C., & Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, *202*(3), 615-627.

Bortfeldt, A., & Forster, F. (2012). A tree search procedure for the container pre-marshalling problem. *European Journal of Operational Research*, *217*(3), 531-540.

Boysen, N., & Emde, S. (2016). The parallel stack loading problem to minimize blockages. *European Journal of Operational Research*, *249*(2), 618–627.

Boysen, N., & Stephan, K. (2016). A survey on single crane scheduling in automated storage/retrieval systems. *European Journal of Operational Research*, *254*, 691–704.

Boysen, N., Briskorn, D., & Meisel, F. (2016). A generalized classification scheme for crane scheduling with interference. *European Journal of Operational Research*, 1-15.

Briskorn, D., & Angeloudis, P. (2016). Scheduling co-operating stacking cranes with predetermined container. *Discrete Applied Mathematics*, *201*, 70-85.

Briskorn, D., Emde, S., & Boysen, N. (2016). Cooperative twin-crane scheduling. *Discrete Applied Mathematics*, *211*, 40-57.

Carlo, H. J., & Vis, I. F. (2012). Sequencing dynamic storage systems with multiple lifts and shuttles. *International Journal of Production Economics*, *140*, 844-853.

Carlo, H. J., Vis, I. F., & Roodbergen, K. J. (2014). Storage yard operations in container terminals: Literature overview, trends, and research directions. *European Journal of Operational Research*, *235*(2), 412–430.

Carlo, H., & Martínez-Acevedo, F. (2015). Priority rules for twin automated stacking cranes that collaborate. *Computers & Industrial Engineering*.

Carlo, H., & Vis, I. (2009). New initiatives in stacking crane configurations. *Port Technology International*, 32-36.

Carlo, H., & Vis, I. (2012). Sequencing dynamic storage systems with multiple lifts and shuttles. *International Journal of Production Economics*, *140*, 844-853.

Caserta, M., Voß, S., & Sniedovich, M. (2011). Applying the corridor method to a blocks relocation problem. *OR Spectrum*, *33*, 915-929.

Chen, J. H., Lee, D. H., & Cao, J. X. (2012). A combinatorial benders' cuts algorithm for the quay-side operation problem at container terminals. *Transportation Research Part E: Logistics and Transportation Review*, *44*(1), 266-275.

Choe, R., Park, T., Seung, M., & Kwang, R. (2007). Real-Time Scheduling for Non-crossing Stacking Cranes in an Automated Container Terminal. *Lecture Notes in Computer Science*, *4830*, 625-631.

Choo, S., Klabjan, D., & Simchi-Levi, D. (2010). Multiship crane sequencing with yard congestion constraints. *Transportation Science*, *44*(1), 98-115.

Cordeaua, J. F., Legato, P., Mazza, R. M., & Trunfio, R. (2015). Simulation-based optimization for housekeeping in a container transshipment terminal. *Computers & Operations Research*, *53*, 81-95.

Dell, R., Royset, R., & Zyngirdis, I. (2009). Optimizing Container Movement using One and Two Automated Stacking Cranes. *Journal of Industrial and Management Optimization* , *5*(2), 285-302.

Dorndorf, U., & Schneider, F. (2010). Scheduling automated triple cross-over stacking cranes in a container yard. *OR Spectrum*, *32*, 617-632.

Dragović, B., Tzannatos, E., & Park, N. K. (2016). Simulation modelling in ports and container terminals: literature overview and analysis by research field, application area and tool. *Flexible Services and Manufacturing Journal*, 1-31.

Eilken, A., & Fliedner, M. (2016). Scheduling cooperating stacking cranes under strict processing priorities. *Working paper*.

Exposito-Izquierdo, C., Melian-Batista, B., & Moreno-Vega, M. (2012). Pre-marshalling problem: Heuristic solution method and instances generator. *Expert Systems with Applications*, *39*(9), 8337-8349.

Forster, F., & Bortfeldt, A. (2012). A tree search procedure for the container relocation problem. *Computers & Operations Research*, *39*(2), 299-309.

Fransoo, J. C., & Lee, C. Y. (2013). The Critical Role of Ocean Container Transport in Global Supply Chain Performance. *Production and Operations Management*, *22*(2), 253-268.

Gharehgozli, A. H., de Koster, R., & Jansen, R. (2017a). Collaborative solutions for inter terminal transport. *International Journal of Production Research*, In Press.

Gharehgozli, A. H., Y. Yu, Y., de Koster, R., & Udding, J. T. (2014a). An Exact Method for Scheduling a Yard Crane. *European Journal of Operational Research*, *235*(2), 431-447.

Gharehgozli, A. H., Yu, Y., de Koster, R., & Udding, J. T. (2014b). A decision-tree stacking heuristic minimising the expected number of reshuffles at a container terminal. *International Journal of Production Research*, *52*(9), 2592-2611.

Gharehgozli, A., Laporte, G., Yu, Y., & de Koster, R. (2015). Scheduling Twin Yard Cranes in a Container Block. *Transportation Science*, *49*(3), 686-705.

Gharehgozli, A., Mileski, J., & Duru, O. (2017b). Heuristic Estimation of Container Stacking and Reshuffling Operations under the Containership Delay Factor and Mega-Ship Challeng. *Maritime Policy & Management*, In Press.

Gharehgozli, A., Roy, D., & de Koster, R. (2016). Sea container terminals: New technologies and OR models. *Maritime Economics & Logistics*, *18*(2), 103-140.

Goerigk, M., & Knust, S. L. (2016). Robust storage loading problems with stacking and payload constraints. *European Journal of Operational Research*, *253*(1), 51-67.

Gorman, M., Clarke, J. P., Gharehgozli, A. H., Hewitt, M., de Koster, R., & Roy., D. (2014). State of the Practice: Application of OR/MS in Freight Transportation. *Interfaces*, *44*(6), 535-554.

Huang, S. H., & Lin, T. H. (2012). Heuristic algorithms for container pre-marshalling problems. *Computers & Industrial Engineering*, *62*(1), 13-20.

Ji, M., Guo, W., Zhu, H., & Yang, Y. (2015). Optimization of loading sequence and rehandling strategy for multi-quay crane operations in container terminals. *Transportation Research Part E: Logistics and Transportation Review*, *19*, 1-19.

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.

Kemme, N. (2012). Effects of storage block layout and automated yard crane systems on the performance of seaport container terminals. *OR Spectrum*, *34*(3), 563-591.

Kim, K., & Park, Y. (2004). A crane scheduling method for port container terminals. *European Journal of Operational Research*, *156*(3), 752-768.

Ku, D., & Arthanari, T. S. (2015). On the Abstraction Method for the Container Relocation Problem. *Computers & Operations Research*, *in press*.

Lee, D. H., Wang, H. Q., & Miao, L. (2008). Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E: Logistics and Transportation Review*, *44*(1), 124-135.

Lehnfeld, J., & Knust, S. (2014). Loading, unloading and premarshalling of stacks in storage areas: Survey and Classification. *European Journal of Operational Research*, *239*(2), 297-312.

Li, W., Goh, M., Wu, Y., Petering, M., De Souza, R., & Wu, Y. (2012). A continuous time model for multiple yard crane scheduling with last minute job arrivals. *International Journal of Production Economics*, *136*(2), 323-343.

Li, W., Wu, Y., Petering, M. E., Goh, M., & de Souza, R. (2009). Discrete time model and algorithms for container yard crane scheduling. *European Journal of Operational Research*, *198*(1), 165-172.

Lim, A., Rodrigues, B., & Zhou, X. (2007). A m-parallel crane scheduling problem with a non-crossing constraint. *Naval Research Logistics*, *54*(2), 115-127.

Lim, A., Rodrigues, B., Xiao, F., & Zhu, Y. (2004). Crane scheduling with spatial constraints. *Naval Research Logistics*, *51*(3), 386-406.

Liu, J., Wan, Y. W., & Wang, L. (2006). Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. *Naval Research Logistics*, *53*(1), 60-74.

Meisel, F. (2011). The quay crane scheduling problem with time windows. *Naval Research Logistics*, *58*(7), 619-636.

Meisel, F., & Bierwirth, C. (2013). framework for integrated berth allocation and crane operations planning in seaport container terminals. *Transportation Science*, *47*(2), 131-147.

Moccia, L., Cordeau, J. F., Gaudioso, M., & Laporte, G. (2006). A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal. *Naval Research Logistics*, *53*(1), 45-59.

Ng, W. C. (2005). Crane scheduling in container yards with inter-crane interference. *European Journal of Operational Research*, *164*, 64-78.

Park, T., Choe, R., Ok, S. M., & Ryu, K. R. (2010). Real-time scheduling for twin RMGs in an automated container yard. *OR Spectrum*, *32*, 593-615.

Petering, M. E., Wu, Y., Li, W., Goh, M., & de Souza, R. (2009). Development and simulation analysis of real-time yard crane control systems for seaport container transshipment terminals. *OR Spectrum*, *31*(4), 801-835.

Saanen, Y., & Valkengoed, M. (2005). Comparison of Three Automated Stacking Alternatives By Means of Simulation. *Processings of the 2005 Winter Simulation Conference*, 1567-1576.

Speer, U., John, G., & Fischer, K. (2011). Scheduling Yard Cranes Considering Crane Interference . *Lecture Notes in Computer Science*, *6971*, 321-340.

Stahlbock, R., & Voß, S. (2010). Efficiency considerations for sequencing and scheduling of double-rail-mounted gantry cranes at maritime container terminals. *International Journal of Shipping and Transport Logistics*, *2*, 95-123.

Vis, I., & Carlo, H. (2010). Sequencing Two Cooperating Automated Stacking Cranes in a Container Terminal. *Transporation Science*, *44*(2), 169-182.

Vis, I., & Roodbergen, K. (2009). Scheduling of container storage and retrieval. *Operations Research*, *57*(2), 456-467.

Yu, M., & Qi, X. (2013). Storage space allocation models for inbound containers in an automatic container terminal. *European Journal of Operational Research*, *226*(1), 32-45.

Zaerpour, N., Yu, Y., & de Koster, R. (2014). Storing fresh produce for fast retrieval in an automated compact cross-dock system. *Production and Operations Management*, *24*, 1266–1284.

Zhang, C., Tao Wu, T., Zhong, M., Zheng, L., & Miao, L. (2014a). Location assignment for outbound containers with adjusted weight proportion. *Computers & Operations Research*, *52*(A), 84-93.

Zhang, C., Wu, T., Kim, K. H., & Lixin Miao, L. (2014b). Conservative allocation models for outbound containers in container terminals. *European Journal of Operational Research*, *238*(1), 155–165.

Appendix A: Makespan for settings with one handshake area

Makespan for settings with one handshake area

				SZ_1			SZ_2					
							SL_{IO}			SL_{Req}		
N	***PL(%)***	***R(%)***	***LC***	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$
5	25	25	Land	1008.60	966.70	961.40	1065.75	1035.1	983	1074.95	1037.2	993.55
5	25	50	Land	970.45	874.30	868.40	999.35	896.45	893.85	1004.9	900.85	903.7
5	25	75	Land	999.00	903.90	877.74	992.1	908.52	887.22	1000.68	901.86	893.52
5	50	25	Land	974.80	881.95	821.80	981.3	878.15	823.3	977.8	879.85	822
5	50	50	Land	980.75	947.60	861.05	981	926.15	854.5	981.55	933.5	857.9
5	50	75	Land	992.04	924.96	833.46	931.02	873.84	817.38	965.1	928.26	842.1
15	25	25	Land	3024.95	2936.00	2620.05	3014.55	2888.95	2610.55	3027.4	2928	2623
15	25	50	Land	2837.90	2669.00	2330.40	2921.65	2672.55	2393.95	2932.4	2669.25	2407.15
15	25	75	Land	2992.74	2845.62	2560.86	2969.16	2778.96	2485.32	2923.98	2698.98	2487
15	50	25	Land	2872.95	2746.05	2335.70	2844.25	2743.9	2322.8	2839.45	2758.7	2315.4
15	50	50	Land	2811.95	2748.80	2253.25	2776	2754.85	2255.2	2767.2	2749.25	2231.35
15	50	75	Land	2964.96	2814.54	2496.60	2963.1	2796.84	2487.06	2929.92	2678.88	2448.96
25	25	25	Land	4985.85	4723.55	4298.50	4915.7	4690.65	4226.45	4943.65	4776.35	4239.5
25	25	50	Land	4811.95	4451.35	3983.35	4785.55	4432.35	3987.65	4814.45	4442.25	4004.45
25	25	75	Land	4885.08	4633.50	4176.42	4920.3	4660.08	4224.06	4907.04	4555.62	4229.7
25	50	25	Land	4763.70	4746.20	3958.30	4677.7	4609.75	3881.8	4661.35	4656.85	3854.05
25	50	50	Land	4615.95	4599.15	3789.70	4679.1	4693.95	3802.5	4665.75	4717.4	3788.3
25	50	75	Land	4604.58	4591.20	3831.84	4666.86	4602.9	3886.08	4725.18	4505.04	3893.88
			Sub-average	*2894.34*	*2778.02*	*2436.60*	*2893.58*	*2769.11*	*2434.59*	*2896.82*	*2762.12*	*2435.31*
5	25	25	Middle	980.60	980.05	887.40	961.7	954.55	850.65	972.35	970.7	867.3
5	25	50	Middle	918.95	903.15	805.40	947.95	908.35	812.55	957.35	917.25	827.25
5	25	75	Middle	931.44	881.04	792.60	944.1	882.96	794.22	963	901.38	835.56
5	50	25	Middle	953.35	968.10	824.35	931.05	928.6	798.3	933.8	953.95	806.1
5	50	50	Middle	902.00	875.75	767.85	918.45	913.25	790.5	922.7	935.8	796.7
5	50	75	Middle	886.14	847.20	756.24	926.04	868.68	775.92	925.08	887.94	794.4
15	25	25	Middle	2847.40	2808.20	2420.90	2757.95	2704.7	2335.7	2791.6	2777.8	2368.85
15	25	50	Middle	2733.55	2748.95	2221.60	2623.85	2674.2	2132.2	2648.85	2677.2	2164.8
15	25	75	Middle	2765.94	2740.68	2341.56	2702.52	2669.64	2233.62	2760.24	2674.56	2308.56
15	50	25	Middle	2652.00	2839.40	2159.90	2688.45	2789.6	2167.7	2707.3	2866.8	2187.05
15	50	50	Middle	2664.70	2779.10	2101.15	2631.4	2717.05	2076.2	2640.7	2740.15	2100.2
15	50	75	Middle	2790.90	2722.74	2339.46	2731.44	2668.8	2246.76	2686.5	2595.18	2256.3
25	25	25	Middle	4598.70	4548.00	3884.90	4509.4	4420.95	3765.6	4559.45	4513.25	3833.8
25	25	50	Middle	4517.35	4644.15	3742.15	4522.85	4565.35	3700.7	4568.05	4602.8	3761.35
25	25	75	Middle	4610.40	4523.94	3894.54	4481.94	4365.42	3774.66	4513.44	4318.26	3842.34
25	50	25	Middle	4467.00	4762.70	3659.60	4503.15	4615.95	3634.85	4533.55	4742.8	3684.35
25	50	50	Middle	4469.00	4682.90	3584.00	4290.8	4490.45	3405.15	4309	4545.05	3428.75
25	50	75	Middle	4469.22	4447.02	3634.92	4416.54	4423.68	3609.3	4385.16	4205.88	3586.14
			Sub-average	*2731.04*	*2761.28*	*2267.70*	*2693.87*	*2697.90*	*2216.92*	*2709.90*	*2712.60*	*2247.21*
5	25	25	Sea	992.80	949.15	840.15	975.4	894.1	816.35	977.9	948.05	827.4
5	25	50	Sea	965.65	976.70	827.50	987.2	943.3	823.75	987.3	971.6	826.5
5	25	75	Sea	975.72	987.96	837.30	969.36	980.16	838.8	995.7	966.66	822.78
5	50	25	Sea	983.05	983.45	846.25	959.55	954.8	819.3	958.05	976.25	815.3
5	50	50	Sea	959.40	947.05	826.25	980.8	956.35	832.95	958.05	976.25	815.3
5	50	75	Sea	970.14	943.74	827.82	1019.34	1005.72	868.26	974.16	969.36	829.74
15	25	25	Sea	2777.85	2751.85	2288.25	2732.9	2667.05	2213.35	2731.7	2730	2234.4
15	25	50	Sea	2914.35	3024.25	2322.40	2787.55	2871.85	2202.75	2784.3	2879.75	2197.5
15	25	75	Sea	2821.26	2864.82	2283.18	2706.72	2765.88	2171.52	2745.9	2755.32	2232.9
15	50	25	Sea	2839.95	2874.55	2336.90	2826.2	2795	2298.3	2817.2	2838.9	2295.65
15	50	50	Sea	2794.50	2814.75	2256.75	2778.45	2779.4	2265.1	2767.7	2793.85	2250.85
15	50	75	Sea	2729.46	2772.12	2195.16	2784.96	2881.44	2255.46	2784.36	2802.24	2235.48
25	25	25	Sea	4448.15	4523.30	3677.70	4561.3	4571.45	3716.45	4563.35	4601.6	3732.15
25	25	50	Sea	4608.90	4779.15	3704.55	4471.5	4771.65	3604.65	4468.75	4793.2	3618.1
25	25	75	Sea	4569.90	4747.50	3721.74	4613.52	4763.88	3766.14	4663.68	4714.62	3773.82
25	50	25	Sea	4671.25	4767.40	3914.10	4610.75	4628.05	3814.4	4600.1	4694.25	3809.45
25	50	50	Sea	4612.10	4557.05	3750.00	4587.05	4566	3776.3	4574.25	4588.55	3754.85
25	50	75	Sea	4545.48	4572.3	3710.94	4574.16	4590.78	3765.24	4585.26	4518.48	3705.48
			Sub-average	*2787.77*	*2824.28*	*2287.05*	*2773.71*	*2799.27*	*2269.39*	*2774.32*	*2806.61*	*2265.43*
	Average over all situations			*2804.38*	*2787.86*	*2330.45*	*2787.05*	*2755.43*	*2306.97*	*2793.68*	*2760.44*	*2315.98*

Appendix B: Blocking time for settings with one handshake area

Blocking time for settings with one handshake area

				SZ_1			SZ_2					
							SL_{IO}			SL_{IReq}		
N	***PL(%)***	***R(%)***	***LC***	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$
5	25	25	Land	420.88	519.66	322.10	488.95	590.15	347.1	505	622.85	362.25
5	25	50	Land	280.39	320.20	175.32	309.5	353.55	212.75	313.5	374.9	225.5
5	25	75	Land	394.39	384.88	262.68	346.74	385.62	259.68	378.54	393.24	269.7
5	50	25	Land	353.56	327.37	174.29	309.7	314.15	145.95	315.5	334	161.2
5	50	50	Land	452.78	501.37	333.66	353.9	420.15	236	356.35	432.3	238.85
5	50	75	Land	408.88	393.95	221.41	322.08	333.42	198.54	337.56	345.36	173.88
15	25	25	Land	1957.61	2198.49	1268.49	1856.35	2015.2	1252.85	1884	2123.7	1299.75
15	25	50	Land	1350.59	1725.66	934.39	1571.25	1846.05	1044.85	1581.95	1854	1028.1
15	25	75	Land	1665.80	1533.95	1084.83	1599.36	1550.64	1075.38	1764.84	1405.56	1170.84
15	50	25	Land	1399.02	1516.10	640.68	1390.9	1629.3	678.65	1394.9	1639.8	674.05
15	50	50	Land	1433.85	1766.78	681.07	1449.85	1859.45	683.6	1440.85	1843.95	622.1
15	50	75	Land	1758.73	1793.56	1219.17	1621.44	1553.76	1123.86	1678.98	1430.64	1094.58
25	25	25	Land	3385.76	3406.10	2332.39	3270.05	3353.65	2213.6	3319.5	3472.8	2250.75
25	25	50	Land	3007.02	3637.46	1947.22	2992.6	3501.95	2123.2	3037.65	3536.4	2169.4
25	25	75	Land	3120.59	2845.46	2131.76	2967.06	2855.28	2159.52	3010.62	2508	2245.14
25	50	25	Land	2888.05	3350.78	1522.39	2614.6	2995.1	1388.8	2593.4	3007.8	1324.7
25	50	50	Land	2589.80	3277.90	1535.71	2752.95	3553.55	1377.2	2735.25	3527.2	1314.1
25	50	75	Land	2688.44	3084.73	1425.80	2691.3	2968.38	1476.54	2795.64	2757.96	1421.34
			Sub-average	1642.01	1810.24	1011.85	1606.03	1782.19	999.89	1635.78	1756.14	1002.57
5	25	25	Middle	444.88	542.49	306.73	412.65	463.9	237.35	433.85	505.75	255.5
5	25	50	Middle	396.29	482.34	222.59	393.95	487.3	225.15	407.55	508.95	245.4
5	25	75	Middle	364.98	354.00	169.02	409.56	407.46	211.5	423	414.84	227.64
5	50	25	Middle	383.71	450.29	217.90	372.35	407.25	202.8	384.2	436.9	215.05
5	50	50	Middle	350.93	384.29	186.88	360.25	439.2	187.8	369.65	469.4	187.3
5	50	75	Middle	246.88	256.10	113.56	365.22	345.96	189.36	353.22	341.46	149.22
15	25	25	Middle	1629.66	1558.68	831.80	1485.4	1468.5	737.55	1549.55	1542.6	788.3
15	25	50	Middle	1500.15	2042.93	689.12	1353.8	1876.95	610.3	1391.4	1904.3	657.1
15	25	75	Middle	1434.00	1585.61	714.44	1393.02	1493.46	592.62	1448.52	1439.52	662.52
15	50	25	Middle	1296.00	1652.93	452.93	1382.2	1490.75	431	1423.45	1560.6	442.8
15	50	50	Middle	1343.27	1806.44	375.51	1256.25	1705.1	412.95	1269.7	1697.15	426.55
15	50	75	Middle	1452.00	1599.95	625.76	1364.22	1460.7	550.14	1388.04	1379.52	689.76
25	25	25	Middle	2710.54	2633.12	1408.39	2537.5	2525.7	1221.35	2631.25	2575.6	1314.65
25	25	50	Middle	2635.90	3709.32	1306.54	2592.35	3467.95	1183.65	2668.15	3484.6	1296.2
25	25	75	Middle	2665.90	2781.51	1430.49	2485.68	2541.96	1197.84	2425.32	2361.42	1252.92
25	50	25	Middle	2352.73	2692.54	885.07	2474.9	2493.9	797.7	2540.8	2614	880.85
25	50	50	Middle	2381.27	3172.68	728.63	2179.45	2937.55	606.55	2213.85	2938.5	626.25
25	50	75	Middle	2280.00	2659.32	726.29	2211.96	2600.46	753	2253.36	2294.46	809.82
			Sub-average	1437.17	1686.92	632.87	1390.60	1589.67	574.92	1420.83	1581.64	618.21
5	25	25	Sea	505.17	464.78	256.54	409.85	364.15	228.7	422.75	442.45	232.35
5	25	50	Sea	482.93	587.85	275.71	494.8	515.8	222.15	495.15	550.45	228.3
5	25	75	Sea	476.63	595.76	287.27	460.32	520.74	249.6	508.14	524.82	219.3
5	50	25	Sea	432.44	495.66	222.29	386	455.9	219.4	394.7	472.3	220.5
5	50	50	Sea	427.90	515.41	259.46	451.5	503.3	251.1	451.35	521.9	239.3
5	50	75	Sea	439.76	458.20	217.17	491.22	547.32	282.06	436.5	496.98	208.02
15	25	25	Sea	1426.83	1385.41	493.02	1323.6	1329.6	402.65	1347.8	1327.4	450.35
15	25	50	Sea	1714.39	2210.20	527.71	1564	1976.25	501.55	1552.9	1971.6	443.35
15	25	75	Sea	1411.17	1707.66	486.44	1395.9	1678.5	408	1394.82	1598.88	418.68
15	50	25	Sea	1530.00	1894.68	738.44	1515.55	1708.95	647.55	1512.55	1750.55	645.35
15	50	50	Sea	1370.49	1866.73	610.98	1398.5	1816.75	652.2	1385.7	1830.55	622.95
15	50	75	Sea	1502.63	1732.68	463.90	1453.14	1802.7	452.64	1509.48	1699.68	436.38
25	25	25	Sea	2122.39	2376.88	834.00	2316.75	2659.25	778.45	2336.85	2503.8	781.75
25	25	50	Sea	2624.78	3522.29	829.46	2477.75	3490.5	742.7	2464.1	3516.8	760.45
25	25	75	Sea	2554.98	3260.34	806.49	2573.76	3216.78	885.54	2581.5	2993.22	780.42
25	50	25	Sea	2604.15	3227.56	1313.27	2597.3	2980.3	1234.25	2591.35	2950.25	1176.15
25	50	50	Sea	2492.78	3117.22	1176.88	2539.55	3259.5	1231.15	2521.95	3251.9	1189.8
25	50	75	Sea	2349.95	3040.39	1153.32	2488.86	3029.52	1233.42	2548.68	2790.72	1139.22
			Sub-average	1470.52	1803.32	608.46	1463.24	1769.77	590.17	1469.79	1733.01	566.26
	Average over all situations			1516.57	1766.83	751.06	1486.62	1713.87	721.66	1508.80	1690.26	729.01

Appendix C. Makespan and blocking time for settings with two handshake areas

				SZ_1 (Makespan)						SZ_1 (Blocking time)					
				SL_{IO}			SL_{Req}			SL_{IO}			SL_{Req}		
N	***PL(%)***	***R(%)***	***DTZ***	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$
5	25	25	1	952.50	928.20	827.46	1008.35	989.15	893.80	399.84	393.36	178.20	469.50	549.45	303.65
5	25	50	1	903.78	866.22	769.08	939.80	898.45	823.80	351.96	372.06	179.10	381.60	464.80	225.75
5	25	75	1	976.50	911.22	800.58	938.30	867.90	812.65	437.88	412.98	221.16	339.45	369.10	196.25
5	50	25	1	919.80	889.68	783.48	928.75	978.05	800.35	302.22	324.48	135.54	372.05	450.05	175.75
5	50	50	1	902.88	858.48	751.26	914.85	915.65	784.90	339.06	346.38	166.92	329.95	413.30	173.60
5	50	75	1	871.74	815.94	735.78	898.55	878.70	778.90	308.88	286.20	151.14	303.20	312.80	143.15
15	25	25	1	2740.56	2666.82	2279.82	2760.75	2826.55	2368.30	1430.70	1371.54	652.86	1535.15	1656.70	844.20
15	25	50	1	2625.66	2574.72	2109.78	2734.20	2611.15	2242.80	1308.18	1616.16	529.02	1365.05	1724.50	724.95
15	25	75	1	2714.28	2596.38	2197.20	2768.00	2561.65	2324.80	1315.80	1279.98	497.04	1401.95	1155.65	704.15
15	50	25	1	2603.40	2645.82	2107.98	2676.50	2877.75	2157.40	1214.70	1278.00	384.60	1387.60	1513.10	477.30
15	50	50	1	2503.80	2575.20	1985.28	2546.80	2632.30	2028.15	1042.98	1433.10	289.80	1149.50	1541.15	330.05
15	50	75	1	2540.16	2487.06	2043.30	2620.55	2440.80	2119.55	1154.70	1170.54	342.54	1124.60	942.85	372.95
25	25	25	1	4403.94	4323.24	3686.10	4543.85	4515.85	3866.15	2329.68	2421.48	1128.54	2702.60	2645.60	1544.40
25	25	50	1	4340.94	4278.60	3549.36	4413.55	4388.10	3707.65	2370.12	3042.78	1090.86	2411.15	3199.60	1384.40
25	25	75	1	4341.90	4321.44	3670.14	4598.15	4311.90	3929.90	2178.90	2437.86	1065.30	2552.55	2239.65	1418.25
25	50	25	1	4165.08	4277.82	3391.68	4203.00	4680.10	3448.00	1920.30	2182.14	668.04	2043.50	2536.75	738.00
25	50	50	1	4133.04	4320.42	3277.92	4286.35	4513.85	3432.55	1964.58	2638.74	518.76	2098.80	2787.95	618.45
25	50	75	1	4153.74	4201.20	3361.86	4252.70	4075.65	3515.10	1935.06	2221.20	547.44	1985.20	1913.20	731.30
		Sub-average		2599.65	2585.47	2129.34	2668.50	2664.64	2224.15	1239.20	1401.61	485.94	1330.74	1467.57	617.03
5	25	25	3	946.92	954.96	841.44	994.55	984.65	895.45	396.06	456.24	217.80	465.00	557.30	300.35
5	25	50	3	955.74	924.36	814.80	967.25	918.80	836.15	439.86	470.64	195.24	435.30	512.75	252.35
5	25	75	3	969.24	917.34	823.56	977.90	925.10	840.80	420.78	449.76	219.42	449.95	498.90	262.45
5	50	25	3	897.66	898.50	775.62	932.55	953.20	796.80	323.88	375.36	159.18	349.20	442.45	192.35
5	50	50	3	917.70	894.24	779.76	919.45	899.60	784.70	347.76	389.04	191.88	365.25	422.55	168.65
5	50	75	3	930.66	889.62	788.34	937.20	909.90	806.55	379.86	356.04	173.28	363.65	380.75	197.35
15	25	25	3	2735.82	2670.54	2323.74	2819.60	2823.65	2414.05	1444.80	1463.40	748.56	1612.95	1628.80	890.50
15	25	50	3	2673.42	2702.16	2166.24	2766.00	2699.65	2277.35	1323.00	1910.28	595.38	1493.30	1901.75	717.00
15	25	75	3	2707.68	2674.86	2238.36	2741.35	2625.25	2314.30	1349.82	1476.42	606.36	1289.10	1274.10	651.15
15	50	25	3	2692.74	2745.72	2160.06	2645.85	2851.10	2167.00	1328.76	1412.28	408.60	1311.65	1504.80	432.75
15	50	50	3	2560.38	2599.56	2026.08	2590.15	2694.80	2057.85	1224.84	1520.52	344.58	1236.50	1611.30	379.35
15	50	75	3	2649.72	2594.70	2136.84	2703.75	2599.50	2209.95	1239.06	1316.82	372.78	1257.60	1202.70	383.80
25	25	25	3	4492.86	4404.36	3786.60	4545.70	4570.15	3859.00	2407.80	2495.40	1173.24	2685.20	2629.50	1428.15
25	25	50	3	4417.38	4440.72	3621.36	4475.40	4567.75	3754.90	2471.28	3320.76	1191.72	2548.25	3419.00	1324.95
25	25	75	3	4481.94	4400.88	3740.34	4583.95	4349.80	3897.10	2374.08	2664.96	1222.44	2598.40	2452.45	1444.85
25	50	25	3	4283.76	4440.06	3513.54	4420.20	4748.80	3627.95	2155.32	2316.48	750.90	2318.95	2662.30	849.70
25	50	50	3	4174.62	4431.84	3390.48	4275.80	4532.50	3430.30	1968.36	2877.06	635.40	2130.25	2927.30	661.65
25	50	75	3	4310.04	4279.08	3519.12	4418.00	4239.15	3583.80	2033.04	2383.98	666.30	2281.65	2236.70	716.65
		Sub-average		2655.46	2659.08	2191.46	2706.37	2716.30	2253.00	1312.69	1536.41	548.50	1399.56	1570.30	625.22
Average over all situations				2627.56	2622.28	2160.40	2687.43	2690.47	2238.58	1275.94	1469.01	517.22	1365.15	1518.93	621.13

Appendix D. Makespan and blocking time for different ASC speeds

In the following tables, the impact of higher speeds is investigated. The results show that a higher speed results in a shorter makespan. However, the other findings discussed in the paper still hold.

Makespan and blocking time for settings with no handshake area

				$Z_{2\text{-}opt}$			$B_{2\text{-}opt}$		
N	***PL(%)***	***R(%)***	***Speed***	**Prio1**	**Prio2**	**Prio3**	**Prio1**	**Prio2**	**Prio3**
5	50	50	1	733.79	727.50	767.48	40.10	41.60	94.04
5	50	50	2	541.72	542.2	542.96	96.68	93.54	92.64
5	50	50	3	427.09	497.20	497.01	67.67	193.07	193.20
5	50	50	4	375.39	368.56	369.28	53.02	33.65	33.38
15	50	50	1	1665.75	1652.71	1664.25	103.85	142.27	122.19
15	50	50	2	1268.1	1223.64	1216.94	265.4	181.66	173.74
15	50	50	3	1016.75	994.95	992.41	197.60	128.24	121.33
15	50	50	4	882.79	882.79	882.79	155.27	106.73	99.27
25	50	50	1	2767.21	2713.85	2752.44	251.19	272.54	228.92
25	50	50	2	2062.28	1935.16	1942.8	205.96	113.9	202.62
25	50	50	3	1694.91	1607.65	1601.35	439.48	231.28	214.39
25	50	50	4	1466.25	1416.73	1414.07	292.96	185.37	174.91

Makespan and blocking time for settings with one handshake area

				SZ_1 (makespan)			SZ_1 (Blocking time)		
N	***PL(%)***	***R(%)***	***Speed***	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$	B_{FCFS}	B_{NN}	$B_{2\text{-}OPT}$
5	50	50	1	902.00	875.75	767.85	350.93	384.29	186.88
5	50	50	2	634.26	638.06	531.8	280.9	333.02	134.76
5	50	50	3	550.84	551.73	467.49	262.32	293.76	125.76
5	50	50	4	488.57	496.14	416.56	223.16	258.21	108.67
15	50	50	1	2664.70	2779.10	2101.15	1343.27	1806.44	375.51
15	50	50	2	1803.76	1945.88	1449.72	929.54	1269.52	290.14
15	50	50	3	1553.95	1702.19	1235.67	814.85	1119.77	218.64
15	50	50	4	1404.5	1552.56	1120.03	738.08	1015.83	222.72
25	50	50	1	4469.00	4682.90	3584.00	2381.27	3172.68	728.63
25	50	50	2	2976.14	3294.24	2394.32	1588.92	2238.64	457.16
25	50	50	3	2546.64	2835.32	2040.2	1377.4	1912.72	405.1867
25	50	50	4	2302.12	2600.06	1851.01	1241.73	1785.39	390.82

Makespan and blocking time for settings with two handshake areas

				SZ_1 (Makespan)						SZ_1 (Blocking time)					
				SL_{IO}			SL_{Req}			SL_{IO}			SL_{Req}		
N	***PL(%)***	***R(%)***	***Speed***	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$	Z_{FCFS}	Z_{NN}	$Z_{2\text{-}OPT}$
5	50	50	1	917.70	894.24	779.76	919.45	899.60	784.70	347.76	389.04	191.88	365.25	422.55	168.65
5	50	50	2	605.97	608.04	518.67	602.57	628.57	521.47	247.05	284.76	128.28	248.10	314.93	121.37
5	50	50	3	514.62	515.82	448.02	552.47	563.02	475.38	196.86	235.26	119.92	243.60	297.82	126.00
5	50	50	4	504.44	494.39	423.92	477.98	485.73	402.63	235.49	240.47	105.66	207.53	245.48	92.62
15	50	50	1	2560.38	2599.56	2026.08	2590.15	2694.80	2057.85	1224.84	1520.52	344.58	1236.50	1611.30	379.35
15	50	50	2	1775.82	1868.1	1423.08	1732.23	1848.30	1401.23	876.87	1160.91	263.76	804.90	1087.93	264.03
15	50	50	3	1531.86	1615.46	1209.68	1474.16	1614.27	1180.13	774.56	1006.2	231.74	722.73	990.02	212.56
15	50	50	4	1383.06	1515.74	1093.61	1367.63	1503.63	1092.45	701.48	969.96	182.72	704.92	953.10	207.62
25	50	50	1	4174.62	4431.84	3390.48	4275.80	4532.50	3430.30	1968.36	2877.06	635.40	2130.25	2927.30	661.65
25	50	50	2	2991.39	3198.96	2369.88	2943.73	3236.83	2384.00	1622.79	2159.52	442.29	1521.87	2111.60	440.60
25	50	50	3	2475.94	2731.58	1974.4	2478.96	2757.47	2007.18	1298.88	1792.46	371.26	1284.44	1795.07	387.82
25	50	50	4	2235.18	2537.03	1795.29	2256.95	2545.07	1805.02	1148.81	1702.22	326.63	1202.30	1677.55	360.40

Appendix E. Makespan and blocking time for different safety distance sizes

The following table shows the impact of different safety distance sizes. As explained in section 4, in our simulation study, if interference occurs, the crane without priority stops at its current location. Therefore, the safety distance is always equal to or larger than 1 bay. As a result, larger safety distance sizes does not influence the findings. Comparing the results with the settings with one or two handshake areas show that the conclusion drawn in section 5.3 still holds.

Makespan and blocking time for settings with no handshake area and different safety distance sizes

				$Z_{2\text{-}opt}$			$B_{2\text{-}opt}$		
N	***PL(%)***	***R(%)***	***Size***	**Prio1**	**Prio2**	**Prio3**	**Prio1**	**Prio2**	**Prio3**
5	50	50	1	733.79	727.50	767.48	40.10	41.60	94.04
5	50	50	2	734.7	755.34	736.14	64.32	109.44	57.18
5	50	50	3	737.16	775.38	790.56	60.72	172.14	175.74
15	50	50	1	1665.75	1652.71	1664.25	103.85	142.27	122.19
15	50	50	2	1725.78	1702.44	1708.44	155.82	148.50	138.60
15	50	50	3	1711.74	1697.22	1709.28	136.08	143.28	134.52
25	50	50	1	2767.21	2713.85	2752.44	251.19	272.54	228.92
25	50	50	2	2808.00	2721.84	2761.20	270.42	309.6	285.66
24*	50	50	3	2668.5	2599.32	2628.9	235.5	305.4	244.44

* When one crane is at an I/O point, the other crane can stack or retrieve a container 4 bays further away. Therefore, in a block with 30 bays, only 24 can be used to stack containers. Since we generate unique locations, we cannot generate enough locations for 25 containers. As a result we run the simulation for 24 requests. This assumption is considered to satisfy the safety distance constraint. We can simply relax this constraint. It will increase the number of locations available and will not impact the conclusions made.