

Effect of Introducing a Time Factor in the Container Relocation Problem

Menno Rotteveel
Master student Business Analytics
Faculty of exact sciences Vrije Universiteit van Amsterdam

July 25, 2018

Abstract

The container relocation problem (CRP) is a common problem in container terminals around the world, where containers are temporarily stored before they are retrieved in the correct order by cargo trucks. To do this efficiently, known heuristics attempt to minimize the number of realized relocations, which is a simplified way to minimize the total time necessary to relocate containers. The current paper however, attempts to focus more on the relocation times instead with the hopes of dropping some of the assumptions the CRP makes.

Three known heuristics (The Lowest Position, Reshuffle Index and MinMax) were compared against a new Quality heuristic, which is shown to be better both in number of relocations and in total relocation times as it uses predictions to minimize the expected number of necessary relocations as opposed to the more greedy approaches used by the other three heuristics.

The assumption that relocations can only occur within container bays is tested for relevance and it is shown that relocating between bays does not significantly reduce the total relocation time. To achieve this, it was estimated that the crane speed in the horizontal direction (between bays) needed to be doubled without increasing the crane speed in the vertical direction (within a bay).

Contents

1	Introduction	2
1.1	Research question	2
2	Definitions	2
2.1	Literature review	2
2.2	Definitions and assumptions	3
3	Addition of a time factor to known heuristics	5
3.1	Instance creation	5
3.2	Known Heuristics	6
3.2.1	TLP	6
3.2.2	Reshuffle Index	6
3.2.3	MinMax	6
3.3	Results	7
3.3.1	Relocation time calculation	7
3.3.2	Resulting values	7
4	The Quality heuristic	10
4.1	Quality definitions	10
4.2	Scoring function	11
4.3	Performance	11
5	Relaxation of two-dimensional assumption	12
5.1	results	13
5.2	Improvement of horizontal crane speed	15
6	Conclusion and Discussion	17
6.1	conclusion	17
6.2	discussion	17
A		20

1 Introduction

Container yards in terminals, such as the ECT Euromax container terminal in the Rotterdam harbour, are used to temporarily store containers that arrive by ship, to be picked up by cargo trucks on the following day. However, these containers are not placed in the container yard in the order that the trucks are expected to arrive, which results in the necessity of reshuffling the bays to pick up the correct container. This problem is called the container relocation problem.

Several algorithms and heuristics exist in literature to solve this problem. Those algorithms attempt to optimally relocate blocking containers (containers that are on top of the container that needs to be picked up) to another stack such that the number of necessary relocations is minimized. This focus-point is chosen to generalize and simplify the problem that needs to be solved.

Because these algorithms focus on minimizing the number of relocations as opposed to minimizing the required relocation times, it is interesting to investigate the effect of adding a time factor, as moving a container to the adjacent stack will take less time than moving it to the stack furthest away in the bay. The assumption made by the algorithms that relocations can only occur within a bay as opposed to relocating between bays, becomes unnecessary as the inclusion of such a time factor would give an appropriate preference to moving within a bay, without dismissing the possibility of a better relocation to a stack in another bay.

The impact of this added time factor is analyzed in the present paper, which also discusses a new heuristic (the quality heuristic) that defines the quality of the placement of containers and the possible relocations.

1.1 Research question

What is the influence of the inclusion of the time-cost of moves between stacks and bays on known algorithms that solve the deterministic container relocation problem and how well does the quality heuristic compare to these known algorithms.

2 Definitions

2.1 Literature review

Relocating containers within a container yard is named the container relocation problem (CRP) or the block relocation problem (BRP). This problem is a critical issue for the importance of container terminals such as those in Rotterdam and Hong Kong (Kim and Kim (1999)) and reducing the time it takes

to relocate containers is an important way for terminals to gain a competitive advantage (Chan and Chang (2011)), thus minimizing the number of required relocations is crucial.

The CRP however is defined in mathematics as NP-hard (Avriel *et al* (1998), meaning that the problem cannot be solved within polynomial time and a working solution cannot be checked for within polynomial time. This sprouted many greedy-algorithms, which make decisions based on the current state of the yard. Those algorithms attempt to find near optimal solution by using several choices(heuristics). Zhang (2000) came up with one of the first simple greedy algorithms that attempted to solve the CRP using heuristics, this was called the Lowest Position heuristic (TLP). Other algorithms include Beam Search which resulted in the Reshuffle Index (Murty *et al*) and the Corridor Method which resulted in the MinMax heuristic (Caserta *et al* (2011)). In literature, many other heuristics exist that are mainly extensions of those algorithms, as well as implementations of these for a similar problem where the container yard is reshuffled without retrievals (pre-marshalling). An example of a heuristic that combines the two is the heuristic as proposed by Expósito-Izquierdo *et al*, where some badly placed containers are relocated beforehand, resulting in a layout that is more easily solvable.

All these algorithms focus on minimizing the required number of relocations and make the assumption that relocating between different bays takes up a significantly greater amount of time as the crane moves considerably slower between bays than between stacks. However little research has been done to find out whether or not this assumption is valid and if not allowing horizontal movement is actually undesirable in every situation.

2.2 Definitions and assumptions

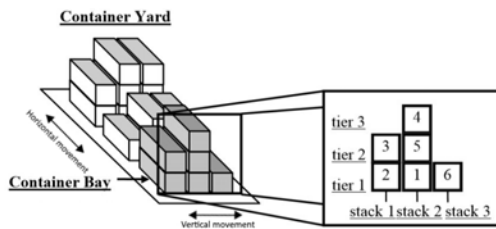


Figure 1: Illustration of stacking in a container yard by [6]

The container relocation problem is defined as the process of retrieving all the containers in a container yard in the correct order. The goal of the problem is to minimize the amount of moves necessary to empty the yard.

In the present paper, bays, stacks, horizontal movement and vertical movement are defined as shown in Figure 1. A priority container is defined as the container from a stack that needs to be retrieved first (has highest priority in the stack),

the target container is the container with the highest priority in the container yard, a blocking container is a container that is on top of a container with a higher priority, and the problem container for a container is the container with the highest priority out of the containers below the current container (in *Figure 1* the problem container for container 4 is container 1).

To simplify the complex realistic situation, the known algorithms make some assumptions:

1. The maximum number of stacks, -bays and -tiers are given.
2. Every bay has the same maximum size.
3. The initial placement of the containers and the order in which they have to be retrieved is known in advance.
4. Containers cannot be relocated to a different bay, as it is assumed that this takes up too much time (the three-dimensional problem is seen as several individual two-dimensional problems). This assumption will be referred to by the present paper as the two-dimensionality assumption.
5. New arrivals of containers are not allowed during the retrieval process.
6. Relocations only occur on containers above the target container, which implies that pre-marshalling in between retrievals is not allowed in this problem.
7. Timeframes of containers are unique, meaning that two containers are not allowed to be retrieved in the same timeframe.
8. All containers in the yard have the same size.

The relevance of the fourth assumption is investigated in the present paper as shifting the goal of minimizing the number of relocation moves to minimizing the relocation time makes the assumption, that relocating between bays is unfavourable, unnecessary. The algorithms should check if a resulting chosen stack in another bay is closer than the chosen stack in the same bay.

Because of the added timefactor, it is important that some assumptions are made to define which times are included in the calculation of the relocation time:

1. The horizontal speed (for moving between bays) and the vertical speed (for moving between stacks) of the crane are given.
2. The time needed to pick up and raise a container is not included in the relocation time.
3. The time it takes for the crane to return to the stack with the target container after relocation is included in the relocation time and is equal to the time it took to relocate the container.
4. The size of the containers is given.

5. The distance between adjacent containers is negligible and is hence not included in the calculation of the relocation time.
6. The time it takes to move the crane to the next target container after a retrieval is not included in the relocation time.
7. The speed of the crane is constant.

3 Addition of a time factor to known heuristics

Three known algorithms that attempt to solve the container relocation problem using heuristics to minimize the number of relocation moves are the lowest position (TLP), Reshuffle Index(RI) and MinMax which are used to solve a randomly generated instance of a container bay. These heuristics solve the three-dimensional problem as several individual two-dimensional problems as they assume relocating between bays is not profitable.

In the present paper, two different sizes of container yards are considered with two different types of containers (still yielding to the assumption that all containers in the same yard have the same size). In reality container yards all over the world have different sizes, temporarily storing different amounts of containers, however the current paper uses two example yards:

1. A container yard with ten bays, ten stacks per bay and a maximum height of five containers per stack. Which is defined here as the large container yard. (Most large container yards like the ECT Euromax container terminal in the Rotterdam Harbour can have more bays, but for the current problem ten bays is sufficiently large as relocating over more than three bays takes up more time than relocating from one side of a bay to the other side (for the relocation times calculated in section 3.3))
2. A container yard with six bays, four stacks per bay and a maximum height of four containers per stack. Which is defined here as the small container yard.

There are two common types of containers: The 20 feet container is 6.06 meters long and the 40 feet container is 12.2 meters long. Both containers have a width of 2.44 meters.

The present papers considers two different fill rates: 67% and 75%, which is the percentage of the container yard to be filled up.

3.1 Instance creation

All the instances are randomly generated for the given size of the container yard and the given fill rate, by first calculating the number of containers in the yard and then adding them to a random stack in the yard that does not yet reach the maximum height. Since this results in the stacks all storing the containers with ascending timeframes, all the stacks are shuffled such that the yard is fully

randomized. The algorithm determines a maximum amount of containers per bay, by subtracting the amount of containers equal to one full stack from the maximum amount possible in a full bay (height * bays * stacks). This is done to prevent bays becoming so full that containers cannot be relocated within the bay, which makes the instance infeasible.

This is done a thousand times per yard size and fill rate, because the resulting relocation moves are very dependent on the setup of the container yard.

These instances are saved such that all algorithms run on the same instances. An example of one of the instance files used is shown in *Table 7* and *Figure 7* in the appendix.

3.2 Known Heuristics

3.2.1 TLP

The heuristic that prefers to place a blocking container on top of the lowest stack was first introduced by Zhang (2000). The idea behind this TLP heuristic is that number of blocking containers is expected to be lower for an arbitrary container, hence possibly resulting in the reduction of the required number of relocations.

In *Figure 2*, the TLP heuristic relocates the blocking container to the lowest stack, which is stack 1 in the example. In literature ties are broken arbitrarily, however the present paper gives priority to the stack that is nearest.

3.2.2 Reshuffle Index

This heuristic, as introduced by Murty *et al* (2005), calculates the number of containers that have a higher priority in a stack, this number is defined as the reshuffle index. The algorithm prioritizes stacks with a lower reshuffle index, with the hopes of decreasing the required number of relocations.

In *Figure 2*, the Reshuffle Index heuristic relocates the blocking container to the stack with the fewest containers with a higher priority: Let $nhpc_i$ be the number of containers in stack i with a higher priority than the blocking container, then the chosen stack j is the stack for which $nhpc_j$ is minimized. This is the third stack in the example. Like the TLP heuristic, ties are broken arbitrarily in the literature, but for the current paper it was chosen to break ties by choosing the nearest stack.

3.2.3 MinMax

This heuristic is similar to the Reshuffle Index, but rather prioritizes stacks where the priority container is the highest. This heuristic was first introduced by Caserta *et al* (2011). This was chosen as containers blocking the priority container in a stack have to eventually be relocated anyway so should not be an influencing factor. In *Figure 2*, the MinMax heuristic relocates the blocking container to the stack with the priority container with the lowest priority: Let

hp_i be the highest priority in stack i , then the chosen stack is j for which hp_j is the lowest. This is the fourth stack in the example with the container with timeframe 4 as the priority container. Because timeframes are unique, there are no ties to break with this heuristic.

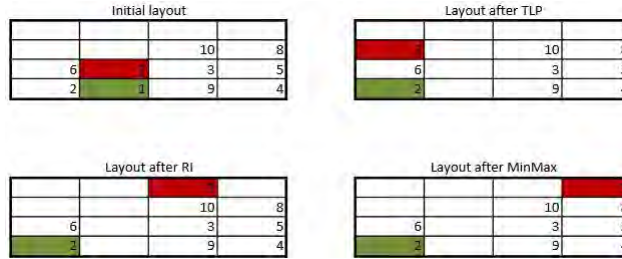


Figure 2: Example relocations by TLP, RI and MinMax

3.3 Results

3.3.1 Relocation time calculation

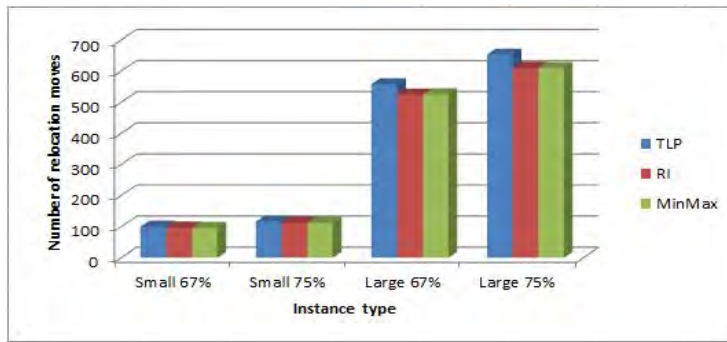
The relocation time from moving a container from stack A to stack B is estimated by using the speed of the crane. The speed in horizontal direction of a crane is 100 meters per minute and in vertical direction is 180 meters per minute as provided by a technical manager of a container terminal. The time it takes to move a container between stacks can be calculated by width of container / vertical speed = $2.44/180 = 0.0136$ minutes (or 0.81 seconds) and the time it takes to move a container between bays is length of container/horizontal speed = $6.06/100 = 0.0606$ minutes (or 3.64 seconds) or $12.2/100 = 0.122$ minutes (or 7.32 seconds) depending on the type of container. Changing the speeds of the crane changes these times. The time it takes to move a container from A to B is then $|stack_A - stack_B| * vertical_time + |bay_A - bay_B| * horizontal_time$ and the relocation time is twice this value as the crane needs to move back to the target stack. The total relocation time to be minimized is the sum over all these relocation times.

3.3.2 Resulting values

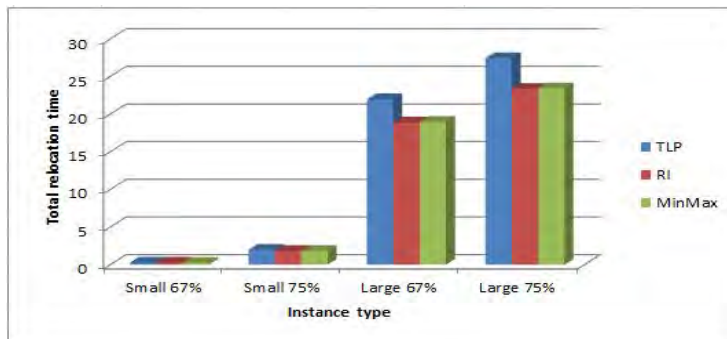
The average number of relocation moves and the average total relocation times are shown graphically in [Figure 3a](#) and [3b](#) respectively, and the actual average values and the standard deviation are shown in [Table 1](#)

From the graphs in [Figure 3](#) it can easily be concluded that the Reshuffle Index heuristic and the MinMax heuristic perform significantly better than the TLP heuristic for the Large container yard instances. This conclusion is supported further by the 95% confidence interval estimated from [Table 1](#) as the mean values are outside of the confidence intervals for the Large instances of

the Reshuffle Index- and the MinMax heuristics as shown in *Table 2*. However, these conclusions cannot be made between the Reshuffle Index- and the MinMax heuristic, so the similarity of the results cannot be rejected.



(a) average number of relocations



(b) average total relocation time

Figure 3: Barplots of resulting values of the heuristics for different instance types with a thousand simulations

To get an indication which of the heuristics should be preferred however, *Table 3* shows how well the heuristics perform compared to each other, with the percentage of the 1000 instances the heuristic solved the best out of all three heuristics. These percentages add up over 100% as heuristics can have similar scores for the same instances (as the relocation moves are not required to be different). From this table, it follows that the TLP heuristic rarely performs better than the other two heuristics and that the MinMax heuristic might be slightly more preferable to the RI heuristic for smaller container yards (especially considering the number of relocation moves), whilst the Reshuffle Index heuristic might be slightly more preferable for larger container yards.

	Instance type	Fill Rate	TLP		RI		MinMax	
	Yard size		Mean	SD	Mean	SD	Mean	SD
Moves	Small	67%	99.80	5.57	97.18	5.34	96.63	5.09
	Small	75%	116.25	6.44	112.96	6.20	112.42	5.94
	Large	67%	557.99	13.39	524.73	12.45	525.05	11.48
	Large	75%	653.17	15.73	610.53	14.55	609.83	13.17
Time	Small	67%	1.607	0.279	1.483	0.258	1.470	0.252
	Small	75%	1.977	0.324	1.829	0.308	1.824	0.302
	Large	67%	21.971	1.663	18.881	1.473	18.960	1.468
	Large	75%	27.443	1.955	23.349	1.751	23.404	1.668

Table 1: Resulting average number of relocations and average total relocation time of the heuristics for different instance types

Value	Fill rate	TLP	RI		MinMax	
		Mean	Min	Max	Min	Max
Moves	0.67	557.99	499.83	549.63	502.09	548.01
	0.75	653.17	581.43	639.63	583.49	636.17
Time	0.67	21.971	15.935	21.827	16.024	21.896
	0.75	27.443	19.847	26.851	20.068	26.740

Table 2: Mean of TLP Heuristic along with 95% confidence intervals of the RI and MinMax heuristics for Large container yards

Heuristic	Yard size	Fill rate	% best moves	% best time
TLP	Small	0.67	7.5%	13.3%
	Small	0.75	3.3%	8.9%
	Large	0.67	0.0%	0.0%
	Large	0.75	0.0%	0.1%
Reshuffle Index	Small	0.67	51.3%	47.2%
	Small	0.75	50.5%	49.6%
	Large	0.67	57.1%	54.3%
	Large	0.75	51.1%	54.0%
MinMax	Small	0.67	77.7%	52.1%
	Small	0.75	73.3%	50.6%
	Large	0.67	49.8%	46.6%
	Large	0.75	57.6%	46.2%

Table 3: Percentage of instances (total 1000) the heuristic solved the best out of all three heuristics for the different instance types

4 The Quality heuristic

In an attempt to improve the retrieval process, the current paper introduces a new retrieval heuristic for the deterministic container relocation problem. The quality heuristic defines certain qualities and assigns those qualities to the location of the containers. In this way, the blocking containers can be moved to a location where their respective quality is the best. Locations with equal resulting qualities are given priority based on a scoring function.

This retrieval is based on the heuristics proposed in Rotteveel *et al* (2018), where defining qualities was used for pre-marshalling purposes. Inspiration for defining qualities was obtained from Expósito-Izquierdo *et al* (2014), which defines containers as either well-located or non-located.

4.1 Quality definitions

The quality heuristic defines three different types of qualities: Good, Okay or Bad. Good quality means that the container's location does not require any relocations, an Okay quality implies that the expected number of relocations is one, and a Bad quality means that the container is expected to require two or more relocations.

The assignment process for these qualities follows these steps from the bottom up per stack:

1. All containers are given a Bad quality.
2. If the container does not block a container with a higher quality, the container is given a Good quality.
3. If the container can be relocated to a stack where the quality will become Good, it is given an Okay quality. An example of this is the container with timeframe 4 in the example in *Figure 4*.
4. If the current container is on top of a container with a Good quality and if it can be relocated to another stack where it will not to block another container, at the time the good container below needs to be retrieved (i.e. a stack can be expected to be empty or only contain lower priority containers at the time the problem container has to be picked up), then the current container is given an Okay quality. Examples of this are the containers with timeframes 3 and 11 in the example in *Figure 4*.
5. If the current container is on top of a container with an Okay quality and if the priority of the current container lower than that lower container and the current container does not negatively influence the quality of the intended location for the lower container, then the current container is given an Okay quality (Relocating the current container to the stack intended for the Okay quality container below, does probably not restrict that container moving there too with Good quality). An example of this is the container with timeframe 7 in the example in *Figure 4*.

		7	
4		3	5
1	12	2	11
10	6	9	8

Figure 4: Example definition of qualities for the Quality heuristic: The container with timeframe 4 can be moved to the fourth stack to become of Good quality. The container with timeframe 3 can be moved to the first stack when the 2 needs to be retrieved, because 1 is already retrieved. The container with timeframe 7 above the 3 can be moved to the first stack before the 3 and the resulting stack will still be of Good quality. The container with timeframe 11 can be moved to the second stack when the 8 needs to be retrieved as it is likely to be empty by then.

4.2 Scoring function

The algorithm prioritizes relocations that result in a better quality. This however often results in multiple options with the same resulting quality, therefore a scoring function is added to prioritize between those options. This consists of a location score and a time score.

Let hpt_i be the highest priority timeframe in stack i , ef an *errorfactor* that punishes bad placement, tf the timeframe of the moving container, hf a factor to prioritize lower stacks, and hs_i the height of stack i . The location score of stack i is calculated by $\max(0, hpt_i - tf) * ef + \max(0, tf - hpt_i) + hf * hs_i$. If a stack is empty it is given a score of *emptyfactor* to attempt to leave the stack empty for a container that is harder to relocate.

The timescore is calculated by a *timefactor* * the time it takes to relocate to the stack. The total score is then the timescore + locationscore. The stack with the lowest score is prioritized.

For the calculations for the results in the present paper, the values of the factors were chosen to be 20, 2, 10, and either 0 or 1000 for the *errorfactor*, *heightfactor*, *emptyfactor* and the *timefactor* respectively. These factors are estimated intuitively and are not optimal. However the influence of these scores is less than the scores of heuristics such as the Reshuffle index as the resulting scores are only used to break ties when multiple relocation options have the same quality. An example of this scoring algorithm is shown in *Figure 5*.

4.3 Performance

To analyze how well the quality heuristic performs, it was chosen to test it both including the timefactor and without it (thus setting *timefactor* for the calculation of the timescore to 0).

When comparing the results from *Table 4* with the confidence intervals shown previously in *Table 2*, it can be concluded that the quality heuristic that uses the timefactor is significantly faster than the Reshuffle Index- and the MinMax

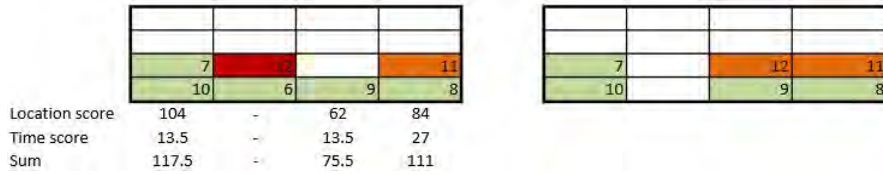


Figure 5: Example of scores for the quality heuristic

	Instance type		Timefactor 0		Timefactor 1000	
	Yard size	Fill Rate	Mean	SD	Mean	SD
Moves	Small	0.67	95.32	4.72	95.99	4.95
	Small	0.75	110.86	5.46	111.63	5.62
	Large	0.67	507.45	9.63	514.63	10.60
	Large	0.75	587.18	11.44	597.67	12.72
Time	Small	0.67	1.420	0.240	1.181	0.209
	Small	0.75	1.749	0.283	1.482	0.252
	Large	0.67	17.268	1.246	9.983	0.932
	Large	0.75	21.176	1.428	12.993	1.124

Table 4: Resulting average number of relocations and average total relocation time for the instances of different instance types with timefactor = 0 and timefactor = 1000

heuristic. It cannot be concluded from these tables that the quality heuristic for either timefactors uses significantly less relocations, however from *tables 5 and 8* it becomes clear that the quality heuristic should have the preference especially for the large container yards. From *Table 8* in the appendix it follows that the decrease in relocations also makes the algorithm more preferable for minimizing the total relocation time.

These resulting values are logical as the heuristic makes some estimation on the required number of relocations based on the position of a container in the yard. This makes the heuristic less greedy as it uses an expectation to look ahead.

5 Relaxation of two-dimensional assumption

Because it is now possible to calculate the time it takes to relocate a container from one location to another rather than assuming that fewer relocations results in a lower relocation time, it is now possible to drop the assumption that containers cannot relocate between bays as all options can be open for consideration.

Since the existing heuristics only focus on moving within a bay, they will not work as well when the two-dimensional problem is expanded to a three-dimensional problem as relocating to the other side of the container yard might

Heuristic	Yard size	Fill rate	% best moves	% best time
TLP	Small	0.67	3.8%	0.1%
	Small	0.75	1.3%	0.0%
	Large	0.67	0.0%	0.0%
	Large	0.75	0.0%	0.0%
Reshuffle Index	Small	0.67	29.2%	1.0%
	Small	0.75	27.3%	0.5%
	Large	0.67	5.1%	0.0%
	Large	0.75	3.5%	0.0%
MinMax	Small	0.67	42.5%	0.9%
	Small	0.75	38.9%	1.3%
	Large	0.67	1.7%	0.0%
	Large	0.75	2.9%	0.0%
Quality (1000)	Small	0.67	75.2%	99.4%
	Small	0.75	72.3%	99.0%
	Large	0.67	95.3%	100.0%
	Large	0.75	95.9%	100.0%

Table 5: Percentage of instances (total 1000) the heuristic solved the best out of all of the four heuristics for the different instance types with the quality heuristic using a timefactor of 1000

be the chosen option, even though it costs a significantly large amount of time, especially when there are multiple bays to consider. Hence it was chosen to implement these heuristics to choose a stack per bay according to their respective algorithms and then prioritize the stack with the lowest relocation time. This results in the algorithms giving a solution that is at least as good as they did before the relaxation of the two-dimensional assumption (as the stack within the same bay is also considered as a possibility and other bays will only be prioritized if the stack is closer).

5.1 results

The resulting values as shown in *Table 6* after the relaxation of the two-dimensionality assumption do not suggest a significant decrease in the total relocation time. As stated before, the relaxation does not result in an increase of the total relocation time for the TLP-, RI and MinMax heuristic, however the quality heuristic was not changed in a similar way as it already has a dependence on the relocation time. This resulted in a significant increase in the total relocation time for most types of instances, which is unfavourable.

From the table, it follows that in the small instances the horizontal relocations did not result in any improvement. The large instances with the containers of 20 feet did result in improvements however, with about 30 horizontal relocations per instance. This is only the case for the 20ft containers, as for the 40 feet

Heuristic	Yard size	Fill Rate	Size	Mean	SD	Horizontal relocations
TLP	Small	0.67	20ft	1.607	0.279	0.0
	Small	0.75	20ft	1.977	0.324	0.0
	Large	0.67	20ft	21.364	1.611	31.2
	Large	0.75	20ft	26.773	1.920	39.9
	Small	0.67	40ft	1.607	0.279	0.0
	Small	0.75	40ft	1.977	0.324	0.0
	Large	0.67	40ft	21.984	1.666	0.4
	Large	0.75	40ft	27.455	1.967	0.5
	Reshuffle Index	Small	0.67	20ft	1.483	0.258
Small		0.75	20ft	1.829	0.308	0.0
Large		0.67	20ft	20.608	1.678	29.8
Large		0.75	20ft	22.643	1.694	30.2
Small		0.67	40ft	1.483	0.258	0.0
Small		0.75	40ft	1.829	0.308	0.0
Large		0.67	40ft	18.895	1.479	0.3
Large		0.75	40ft	23.357	1.755	0.5
MinMax		Small	0.67	20ft	1.470	0.252
	Small	0.75	20ft	1.824	0.302	0.0
	Large	0.67	20ft	18.547	1.388	26.9
	Large	0.75	20ft	23.026	1.668	33.9
	Small	0.67	40ft	1.470	0.252	0.0
	Small	0.75	40ft	1.824	0.302	0.0
	Large	0.67	40ft	18.967	1.470	0.3
	Large	0.75	40ft	23.418	1.670	0.4
	Quality	Small	0.67	20ft	2.000	0.593
Small		0.75	20ft	2.679	0.728	9.6
Large		0.67	20ft	13.351	1.915	33.4
Large		0.75	20ft	19.088	2.565	49.0
Small		0.67	40ft	3.162	1.149	6.9
Small		0.75	40ft	4.350	1.407	9.6
Large		0.67	40ft	18.503	3.484	26.4
Large		0.75	40ft	28.033	4.761	40.9

Table 6: Resulting mean and standard deviation of the total relocation times and the average number of realized horizontal relocations after the relaxation of the two-dimensionality assumption

containers the horizontal speed of the crane is too slow to result in a meaningful amount of horizontal relocations.

5.2 Improvement of horizontal crane speed

The results from *Table 6* did suggest that the relaxation of the two-dimensionality assumption did not result in significant change. However it is interesting to find out how fast the crane should become for the resulting average total relocation time to be below the 95% confidence intervals for the three heuristics. These confidence intervals are (18.645;25.297), (15.935;21.826) and (16.025;21.895) for the TLP-, Reshuffle Index- and the MinMax heuristic respectively for the large container yards with a fill rate of 0.67 and (23.532;31.354), (19.846;26.852) and (20.067;26.741) respectively for the large container yards with a fill rate of 0.75 (estimated from the resulting values in *Table 1*).

The resulting mean values for the 20 feet and the 40 feet containers for the large bays with a 75% fill rate are plotted in the graphs in *Figure 6*. A similar graph for the 67% fill rate instances and a table containing the realized means are shown in *figures 8,9 and 10*

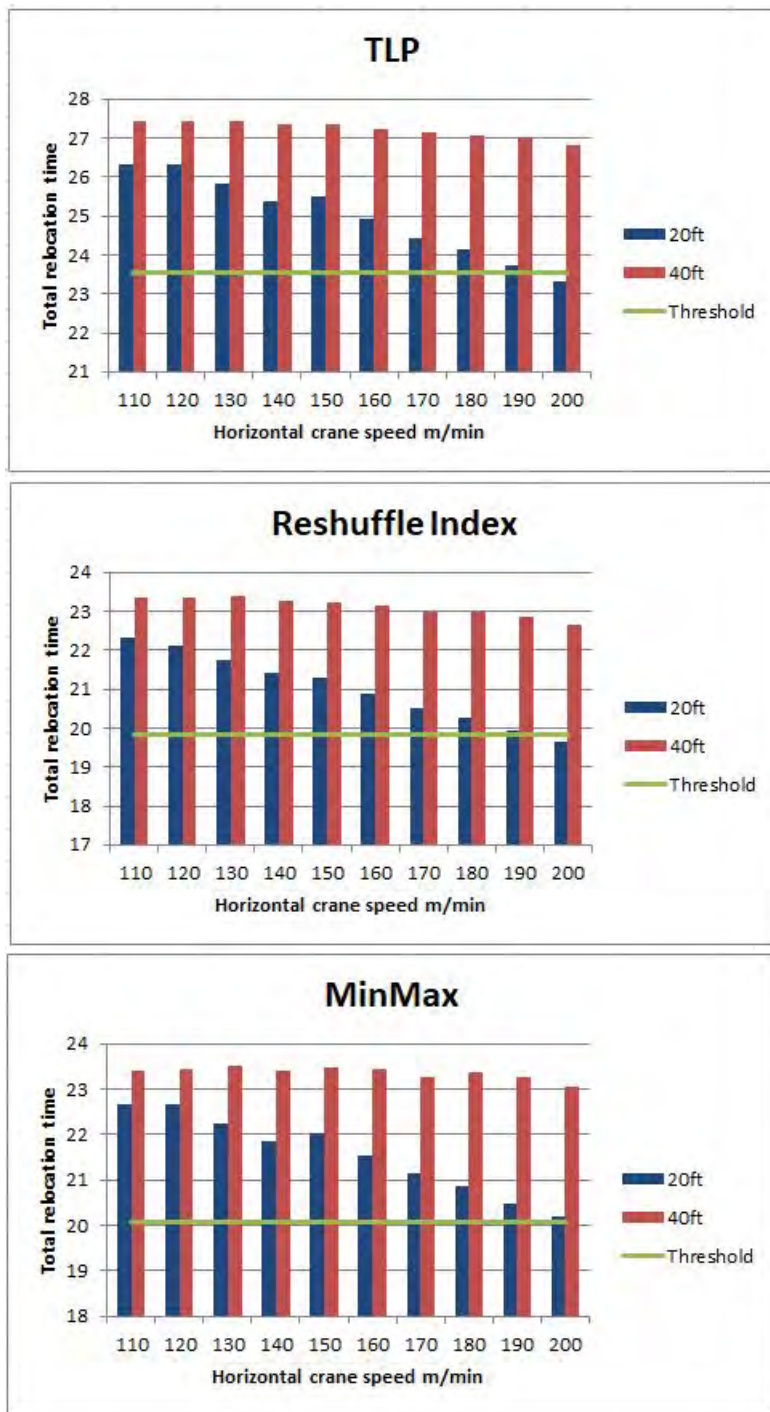


Figure 6: Resulting total relocation times after increasing the horizontal crane speed for the large container yards with fill rate 75% for the TLP, RI and MinMax heuristics

From the graphs in *Figure 6* can be concluded that the 40 feet containers are not relocated significantly faster when the two-dimensionality assumption is dropped and the 20 feet containers are only relocated significantly faster when the horizontal speed of the crane is doubled (and the vertical speed is still the same).

6 Conclusion and Discussion

6.1 conclusion

The resulting values of the implementation of the three known heuristics (TLP, RI and MinMax) for the deterministic container relocation problem suggest that the results found by TLP heuristic are significantly worse than the results from the Reshuffle Index- and the MinMax heuristic. Between the RI- and the MinMax heuristic, no significant difference was found in terms of relocation moves. The introduction of a timefactor did not result in any meaningful additional information as no well supported conclusion can be taken from the results, other than that the two heuristics have fairly similar results.

However, the introduced quality heuristic was proven to be significantly faster in terms of total relocation time when the timefactor was used, and delivered better results in terms of number of relocation moves when the timefactor was zero.

The proposed relaxation of the two-dimensionality assumption did not yield any significant improvements as the crane did not appear to be sufficiently fast in the horizontal direction, however for the large container yards with 20 feet containers, some horizontal relocations were realized as they yielded some expected improvement. This is likely thanks to the crane being sufficiently fast in some cases. However for the average total relocation time to be significantly lower, the horizontal crane speed should at least be twice as fast. The required funds to make this improvement is likely just as well spent on increasing the vertical speed of the crane.

From this can be concluded that the relaxation of the two-dimensionality problem only slightly improves the process for the TLP, RI and MinMax heuristics, and often does not improve the process for the quality heuristic.

6.2 discussion

The present paper focused on the relaxation of one of the assumption made for the deterministic container relocation problem. This was done by adding a timefactor which required further assumptions. Some of those assumptions, like the assumption that speeds are constant, should be investigated in further research to get a better indication of reality as opposed to the simplified version under the assumptions.

As this research investigates the effect on the deterministic CRP, the effect of a

timefactor on the stochastic variant is yet to be determined and requires further research, especially since the quality heuristic can be extended to work on the stochastic CRP.

The introduced quality heuristic uses several chosen factors to calculate a scoring function to break ties, as opposed to breaking ties arbitrarily. This scoring function has been determined intuitively and can be optimized like the scoring function in Rotteveel *et al.*

References

- [1] Caserta, M.; S. Schwarze; and S. Voß 2012. A mathematical formulation and complexity considerations for the blocks relocation problem. *European journal of operational research* 219(1):96-104
- [2] Zhang, C. 2000. *Resource planning in container storage yard*, PhD Dissertation, Department of Industrial Engineering, The Hong Kong University of Science and Technology
- [3] Murty, K.G. ; J. Liu; M. M. Tseng; E. Leung; K-K. Lai; and W.C. Chiu 2005. Filtered beam search in scheduling. *International journal of production research* 26:35-62
- [4] Caserta, M.; S. Voß; and M. Sniedovich 2011. Applying the corridor method to a blocks relocation problem. *OR Spectrum* 22:915-929
- [5] Expósito-Izquierdo, C.;B. Melián-Batista; and J. Marcos Moreno-Vega 2014. Optimization model and heuristic approach for blocks retrieval processes in warehouses. *Advancement of Artificial Intelligence*
- [6] Wu, K-C.; and C-J. Ting. A beam search algorithm for minimizing reshuffle operations at container yards. Department of industrial engineering and management, Yuan Ze University
- [7] Jovanovic, R; and S. Voß 2014. A chain heuristic for the blocks relocation problem. *Computers & Industrial Engineering* 75: 79-86
- [8] Kim, K.H.; and H.B. Kim (1999) Segregating Space Allocation Models for Container Inventories in Port Container Terminals. *International Journal of Production Economics* 59: 415-423
- [9] Chan, F. T.; and H. Chan (2011). Improving the productivity of order picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage. *Expert Systems with Applications* 38(3):2686-2700
- [10] Avriel, M.; M. Penn; N. Shpirer; S. Witteboon (1998). Stowage planning for container ships to reduce the number of ships. *Annals of operations research* 76: 55-71
- [11] Galle, V. (2018). Optimization models and methods for storage yard operations in maritime container terminals, *partial fulfillment of requirements for the degree of Doctor of Philosophy in Operations Research*, Massachusetts Institute of Technology.
- [12] Rotteveel, M. ; M. Koopman; R. Vrinzen; W. Falkena; and N. Blitz 2018. Container relocation, *partial fulfillment requirement for the degree of Master of Science in Business Analytics*, Vrije Universiteit van Amsterdam

Appendix A

bay
34,39,15
9,13,11
35,40,36,12
21,0
bay
47
63
10,26,58
bay
28,5,37,24
54,14
8,1
7,32,22,2
bay
23,29,56
41,30,19,3
46,20
53,52,16
bay
59,55,6,49
4,57,51,18
50,62
33,43
bay
42,25,17,48
44,27
38,61,60,31
45

Table 7: example csv file for instance with 6 bays, 4 stacks, max height 4, fill rate 67%

		12	
Bay 1	15	11	36
	39	13	40
	34	9	35
		58	
Bay 2		26	
	47	63	10
	24		2
Bay 3	37		22
	5	14	1
	28	54	8
	3		
Bay 4	56	19	16
	29	30	20
	23	41	46
	49	18	
Bay 5	6	61	
	55	57	62
	59	4	50
	48		31
Bay 6	17		60
	25	27	61
	42	44	38

Figure 7: Yard representation of Table 7

Heuristic	Yard size	Fill rate	% best moves	% best time
TLP	Small	0.67	2.8%	7.7%
	Small	0.75	0.4%	3.9%
	Large	0.67	0.0%	0.0%
	Large	0.75	0.0%	0.0%
Reshuffle Index	Small	0.67	22.2%	27.3%
	Small	0.75	19.7%	27.3%
	Large	0.67	0.0%	6.4%
	Large	0.75	0.0%	4.2%
MinMax	Small	0.67	28.6%	28.1%
	Small	0.75	22.2%	24.7%
	Large	0.67	0.0%	5.4%
	Large	0.75	0.0%	1.9%
Quality (0)	Small	0.67	92.3%	53.6%
	Small	0.75	88.6%	57.1%
	Large	0.67	100.0%	88.7%
	Large	0.75	100.0%	94.3%

Table 8: Percentage of instances (total 1000) the heuristic solved the best out of all of the four heuristics for the different instance types with the quality heuristic using a timefactor of 1000

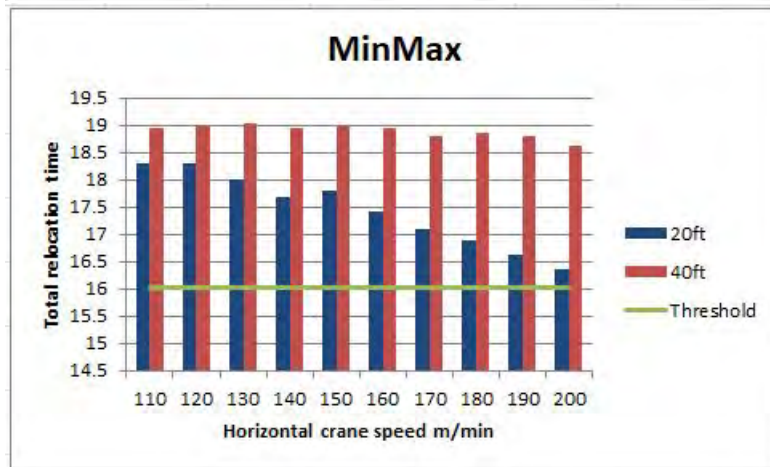
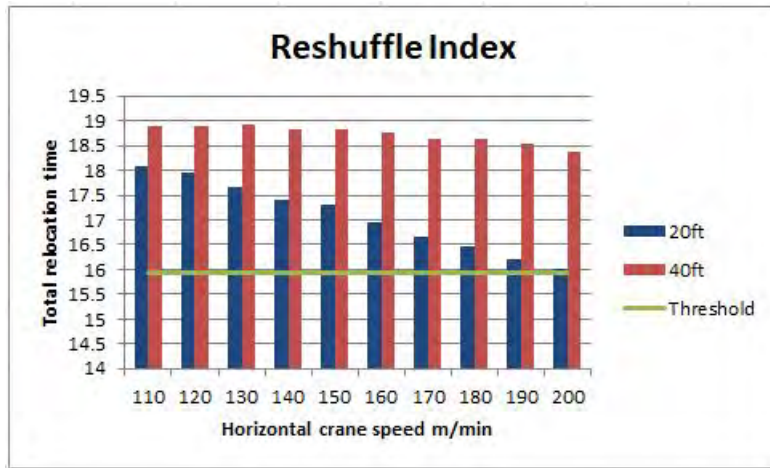
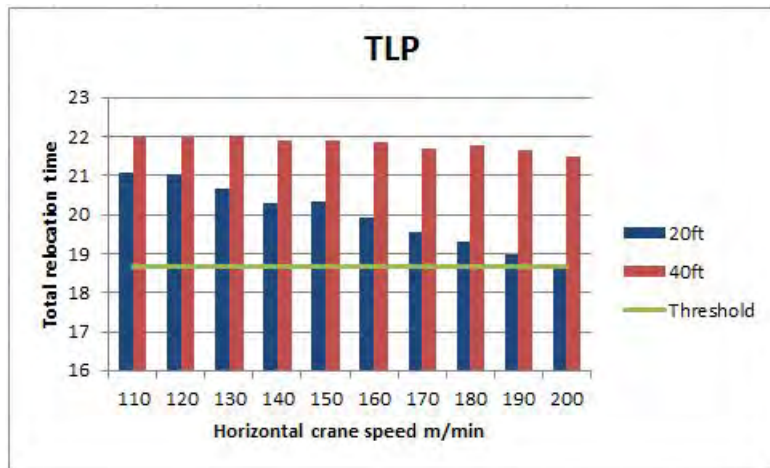


Figure 8: Resulting total relocation times after increasing the horizontal crane speed for the large container yards with fill rate 67% for the TLP, RI and MinMax heuristics

67%											
TLP				RI				MinMax			
	20ft	40ft	Threshold		20ft	40ft	Threshold		20ft	40ft	Threshold
110	21.089	21.967	18.645	110	18.095	18.877	15.935	110	18.312	18.955	16.025
120	21.049	21.978	18.645	120	17.960	18.892	15.935	120	18.305	18.974	16.025
130	20.665	22.022	18.645	130	17.671	18.925	15.935	130	17.997	19.040	16.025
140	20.292	21.912	18.645	140	17.390	18.838	15.935	140	17.675	18.946	16.025
150	20.349	21.905	18.645	150	17.294	18.819	15.935	150	17.793	18.987	16.025
160	19.914	21.851	18.645	160	16.962	18.757	15.935	160	17.416	18.931	16.025
170	19.540	21.699	18.645	170	16.676	18.640	15.935	170	17.092	18.801	16.025
180	19.307	21.764	18.645	180	16.473	18.628	15.935	180	16.901	18.844	16.025
190	18.974	21.660	18.645	190	16.218	18.528	15.935	190	16.610	18.801	16.025
200	18.683	21.466	18.645	200	16.004	18.376	15.935	200	16.363	18.632	16.025

Figure 9: Resulting means after increasing the horizontal crane speed for large bays with 67% fill rate

75%											
TLP				RI				MinMax			
	20ft	40ft	Threshold		20ft	40ft	Threshold		20ft	40ft	Threshold
110	26.329	27.443	23.532	110	22.308	23.342	19.846	110	22.649	23.403	20.067
120	26.337	27.437	23.532	120	22.127	23.334	19.846	120	22.657	23.434	20.067
130	25.842	27.426	23.532	130	21.754	23.375	19.846	130	22.253	23.504	20.067
140	25.368	27.365	23.532	140	21.398	23.268	19.846	140	21.855	23.389	20.067
150	25.493	27.353	23.532	150	21.310	23.242	19.846	150	22.014	23.466	20.067
160	24.936	27.243	23.532	160	20.891	23.120	19.846	160	21.544	23.428	20.067
170	24.441	27.151	23.532	170	20.527	22.974	19.846	170	21.139	23.263	20.067
180	24.133	27.052	23.532	180	20.243	22.987	19.846	180	20.855	23.379	20.067
190	23.708	27.036	23.532	190	19.920	22.852	19.846	190	20.491	23.262	20.067
200	23.336	26.795	23.532	200	19.651	22.661	19.846	200	20.186	23.044	20.067

Figure 10: Resulting means after increasing the horizontal crane speed for large bays with 75% fill rate