Heuristics for the integration of crane productivity in the berth allocation problem

Jonathan Bogerd

414304

July 7, 2019

Abstract

Due to growth in the world trade, the number of containers and container vessels has been rising for decades. This challenges container terminals to make more efficient berthplans. In particular assigning berth locations and quay cranes to arriving vessels is important to run the container terminal as efficiently as possible. In this paper the berth allocation problem and the quay crane assignment problem are studied simultaneously. The results show that the studied heuristics provide good solutions for the berth and quay problems within a reasonable computation time.

Erasmus University Rotterdam

Erasmus School of Economics

Bachelor Thesis Double Degree programme Econometrics & Economics

Name of Supervisor: T.A.B. Dollevoet

Name of Second Assessor: N. Milovanovic

Contents

1 Introduction

Several logistical processes take place in a container terminal. This paper studies two of these processes, namely the assignment of quay cranes (QCs) to vessels and the assignment of quay space (QS) to vessels. Both QCs and QS are limited resources that have to be distributed between several vessels that use the terminal. Factors like the minimum number of cranes, arrival time, the required number of cranes and the productivity loss by using several cranes for one ship have to be taken into account for assigning the cranes to the vessels. For assigning quay space to the vessels, other factors like length, preferred berth location and arrival time of a vessel play a role.

2 Problem and Research Question

In this paper, the Berth Allocation and Crane Assignment Problem (BACAP) is studied. It combines the previously mentioned assignment of QCs and the assignment of QS in one model. The main objective is to find the solution to the BACAP problem that has the lowest cost. The first goal of this paper is to replicate the results of Meisel and Bierwirth (2009). It compares the four heuristics to find the best heuristic for the BACAP, based on the quality of the solution and the computation time. Second, this paper adds a new heuristic that combines the discussed heuristics and algorithms to find with better solutions in both quality and running time. The results of this new heuristic are compared to the heuristic of Meisel and Bierwirth (2009). Last, the consequences of more flexible planning options for the quay cranes are investigated.

3 Theoretical Background

The berth allocation problem (BAP) has been studied for many years. Both the discrete case, which is studied in this paper, and the continuous case were studied in the previous century. Imai, Nagaiwa, and Tat (1997) study a typical BAP in the context of an Asian container terminal. In their paper, a solution method is provided for a BAP problem in which some berth interference coefficient is used. They introduce the notion that the berthing time is dependent on the berthing location and show the importance such a coefficient. Even earlier, Lai and Shih (1992) study the berth allocation problem by means of heuristics and computer simulation for a container terminal in Hong Kong. They find 3 allocation strategies that provide good solutions for the BAP problem. Up to then, vessels were assigned a berth location by means of a standard berth location, the so-called home berth. Then, if a new berthplan had to be made, vessels were first assigned their home berth, based on historical berth locations. Lai and Shih (1992) state that this results in unnecessary queues. Their allocation strategies are based on classifying the arriving vessels in groups, based on size. Then, their policies give a priority ranking and the vessels are assigned a berth location in order of the priority. This concept is similar to the construction heuristic that will be described in this paper. However, in this heuristic, vessels are not prioritized based on size but on arriving time.

The BAP does not only arise in large container terminals. Imai, Nishimura, and Papadimitriou (2001) study the BAP for smaller harbors in Japan, due to their high berthing costs. Similar to Lai and Shih (1992), the initial berth allocation in principle is based on first come, first serve. They study a dynamic case of the BAP and provide an solution strategy that is not based on first come first serve. In order to solve their problem, a Langrangian relaxation is used, similar to for instance Park and Kim (2005), in combination with the subgradient method. In their paper, they claim to find near optimal solutions for smaller container terminals, as found in Japan. In the literature there are many examples of studies using other solution methods, for instance by F. Wang and Lim (2007), Ting, Wu, and Chou (2014), Moon (2000) and Ramani (1996).

The quay crane assignment problem (QCAP) has been extensively studied. In Sammarra et al. (2007), a TABU heuristic is used to schedule quay cranes for loading and unloading vessels. In Lee, H. Q. Wang, and Miao (2008) the QCAP is studied by means of a genetic algorithm and CPLEX, because they prove that their QCAP is NP-hard. They find that there algorithm obtains near optimal solutions for their QCAP. In most QCAP's, the travel time of the QCs is ignored. However, this can of course play an important role in finding the optimal QC assignment. In Al-Dhaheri, Jebali, and Diabat (2016) re-positioning time of the QCs is taken into account. Another important aspect of loading and unloading vessels is vessel stability. It is not possible to start unloading a ship from the front and end at the rear of the vessel. Therefore, more recently, QCAPs have been extended to incorporate this feature as well, for instance in Al-Dhaheri, Jebali, and Diabat (2016) and J. Wang, Hu, and Song (2013). In this paper however, both QC re-positioning time and vessel stability are not taken into account. Especially not adding the re-positioning time is important, because one of the research questions is the effective of a more flexible QC planning. Not taking into account the effect of re-positioning will influence the obtained results.

So far, both the BAP and QCAP are independently of each other. However, QC assignment influences the optimal berthing location and vice versa. Therefore, the next step is to combine these two problems into the BACAP. Park and Kim (2005) show that the BACAP can be solved without taking into account which crane serves which vessel specifically. Therefore, the BACAP does not keep track of individual QCs. Many solution techniques are proposed and studied for the BACAP. Vacca, Salani, and Bierlaire (2013) use a method based on column-generation, Liang, Huang, and Yang (2009) propose an evolutionary algorithm and Zhang et al. (2010) use a subgradient technique to name a few. Elwany, Ali, and Abouelseoud (2013) study the BACAP problem in its continuous form. Also, constraints regarding the depth of the water around quays have been added to the BACAP. This paper is particularly interesting for this research, as they use similar methods to Meisel and Bierwirth (2009). In Elwany, Ali, and Abouelseoud (2013), an crane interference component is included, similar to this paper, and once a priority list is generated, both resource leveling and spatial and temporal shifting are applied. Using these similar methods, they find high quality solutions for the continious case of the BACAP.

This paper will add 3 things to the already existing literature. First, it replicates and verifies the results found by Meisel and Bierwirth (2009). Second, it provides a new heuristic for the BACAP, based on existing heuristics. Last, it studies the effect of more flexible QC planning.

4 Model

In this section, a model for the BACAP is provided and explained. First, the notation of the model is introduced, after which the model itself is given and the constraints are explained.

4.1 Notation

The following input variables are used in the BACAP model (Meisel and Bierwirth (2009)):

V	The set of all vessels that have to be served in the given time frame, $V = \{1, 2, ..., n\}$
Q	The number of quay cranes (QCs) on the quay

L	The length of the quay in berth segment of 10 meters
T	The set of time periods of 1 hour, $T = \{0, 1, ..., H-1\}, H$ the planning horizon
l_i	The length of vessel $i \in V$, per 10 meter segment
b_i^0	The prefered berthing position of vessel i
m_i	The number of QC-hours that are required to serve vessel i
r_i^{min}	The minimum number of QCs that have to servel vessel i at each time period t
r_i^{max}	The maximum number of QCs that have to servel vessel i at each time period t
R_i	The allowed range of number of QCs that can be assigned to vessel i, $R_i = [r_i^{min}, r_i^{max}]$
ETA_i	Expected arrival time of vessel i
EST_i	Earliest possible arrival time of vessel i, if vessel i is speeded up, $EST_i \leq ETA_i$
EFT_i	Expected finishing time of serving vessel i
LFT_i	Latest finishing time of serving vessel i for which there are no extra costs
c_i^1, c_i^2, c_i^3	Costs for serving vessel i in 1000 USD per hour
c^4	Cost for operating QCs in 1000 USD per QC-hour
α	Exponent that determines the rate of interference if several QCs serve a single vessel
β	Coefficient that determines the extra demand for QCs due to the distance between the optimal and the provided berthing location
M	A large positive number

The decision variables of this model are as follows (Meisel and Bierwirth (2009)):

b_i	Berthing location of vessel i, integer
s_i	Starting time of handling vessel i, integer
e_i	Ending time of handling vessel i, integer

r_{it} Variable to keep track of the serving of vessel i at time t, 1 if al least one QC is assigned to vessel i at time t, 0 otherwise, binary

r_{itq} Variable to keep track of the number of QCs that are assigned to vessel i at time t, 1 if q QCs are assigned to vessel i at time t, 0 otherwise, binary

Δb_i The distance between the optimal and the provided berthing location of vessel i, $\Delta b_i = \mid b_i^0 - b_i \mid$, binary

ΔETA_i The speed up of vessel i that is required to reach the berthing time, $\Delta ETA_i = \max\{ETA_i - s_i, 0\}$, integer

ΔEFT_i The delay of finishing handling vessel i, $\Delta EFT_i = \max\{e_i - EFT_i, 0\}$, integer

u_i A variable to track if vessel i has received a penalty for finishing too late, set to 1 if finishing time of vessle i exceeds LFT_i, 0 otherwise, binary

y_{ij} A variable to track the relative berthing position of vessel i and j, set to 1 if vessel i is berthed below vessel j, $b_i + l_i \leq b_j$, 0 otherwise, binary

z_{ij} A variable to track the relative handling time of vessel i and j, set to 1 if vessel i is served and finished before s_j

4.2 BACAP model

The BACAP model then becomes (Meisel and Bierwirth (2009)):

$$min\ Z = \sum_{i \in V} (c_i^1 * \Delta ETA_i + c_i^2 * \Delta EFT_i + c_i^3 * u_i + c^4 \sum_{t \in T} \sum_{q \in R_i} q * r_{itq}) \tag{1}$$

$$\sum_{t \in T} \sum_{q \in R_i} q^{\alpha} * r_{itq} \geq (1 + \beta * \Delta b_i) * m_i,\ \forall i \in V \tag{2}$$

$$\sum_{i \in V} \sum_{q \in R_i} q * r_{itq} \leq Q,\ \forall t \in T \tag{3}$$

$$\sum_{q \in R_i} r_{itq} = r_{it},\ \forall i \in V,\ \forall t \in T \tag{4}$$

$$\sum_{t \in T} r_{it} = e_i - s_i,\ \forall i \in V \tag{5}$$

$$(t+1) * r_{it} \leq e_i,\ \forall i \in V,\ \forall t \in T \tag{6}$$

$$t * r_{it} + H * (1 - r_{it}) \geq s_i,\ \forall i \in V,\ \forall t \in T \tag{7}$$

$$\Delta b_i \geq b_i - b_i^0,\ \forall i \in V \tag{8}$$

$$\Delta b_i \geq b_i^0 - b_i,\ \forall i \in V \tag{9}$$

$$\Delta ETA_i \geq ETA_i - s_i,\ \forall i \in V \tag{10}$$

$$\Delta EFT_i \geq e_i - EFT_i,\ \forall i \in V \tag{11}$$

$$M * u_i \geq e_i - LFT_i,\ \forall i \in V \tag{12}$$

$$b_j + M * (1 - y_{ij}) \geq b_i + l_i,\ \forall i, j \in V,\ i \neq j \tag{13}$$

$$s_j + M(1 - z_{ij}) \geq e_i,\ \forall i, j \in V,\ i \neq j \tag{14}$$

$$y_{ij} + y_{ji} + z_{ij} + z_{ji} \geq 1,\ \forall i, j \in V,\ i \neq j \tag{15}$$

$$s_i, e_i\ \in \{EST_i, ..., H\},\ \forall i \in V \tag{16}$$

$$b_i\ \in \{0, 1, ..., L - l_i\},\ \forall i \in V \tag{17}$$

$$\Delta ETA_i, \Delta EFT_i \geq 0,\ \forall i \in V \tag{18}$$

$$r_{itq}, r_{it}, u_i, y_{ij}, z_{ij}\ \in \{0, 1\},\ \forall i, j \in V,\ \forall t \in T,\ \forall q \in R_i \tag{19}$$

Solving the model provides a solution on when vessels are served, by how many QCs vessels are served and the location of the vessels. The objective (1) is to minimize the total costs of the use of the QCs and the handling of the vessels. The first element of the objective is the cost of speeding up vessel i so that it will arrive earlier than the expected arrival time. The second part of the objective resembles the cost of leaving later than expected. In the paper of (Meisel and Bierwirth 2009) it is unclear how the expected leaving time of a vessel is calculated. In this paper it is set to the fastest possible leaving time, given that the handling of vessel i started at the expected arrival time. To achieve the fastest possible leaving time, the vessel is assigned the preferred berth location and assigned the maximum number of cranes in each period of the handling time. The next element of the objective is a cost incurred if vessel i leaves later than a given time LFT_i. The last component of the objective function is the cost of the assigned cranes. Constraint (2) guarantees that there are enough QCs to unload/load the vessel. The exponent α is the exponent of interference between cranes. Thus, this constraint takes into account the reduced productivity of using more QCs. Constraint (3) fixes the maximum amount of cranes to be used in one time period to at most the total number of cranes. In both constraints, the binary variable r_{itq} is used. Constraint (4) ensures that this variable is set to 1 for only one value of q. Therefore, for each vessel there cannot be two different numbers of cranes working in the same time period. To link the starting and ending time to the crane variables, constraint (5) states that for each vessel the handling time is the time that at least one crane is working on that vessel. The next constraint (6) ensures that a vessel cannot leave as long as there are cranes working on loading or unloading the vessel. Note that $t+1$ is used to take into account that a ship can only leave the time period after QCs were working on the ship. Constraint (7) fixes the starting time to be before a ship is handled and, if a vessel is not handled, fixes the starting time to be before the end of the planning horizon. This guarantees all ship handling is started before the end of the planning horizon. To set the distance between the ideal and actual berth location Δb_i correctly, constraints (8) and (9) are used. By using 2 constraints, the absolute value of the distance is taken into account, without the constraints being non-linear. The correct required speedup to reach the new arrival time for each vessel is guaranteed by (10). Note that equality is guaranteed by minimizing (1) and setting $\Delta ETA - i$ to at least 0 in (18). In the same way (11) handles the delay time of a vessel. To keep track of the acquired penalties due to too late finishing times, constraint (12) is used. Constraints (13), (14) and (15) prevent overlapping between ships. Table 4.2 shows the correctness of the statements. In this table, the values for y_{ij}, y_{ji}, z_{ij} and z_{ji} are given. Note that there is overlap at the **bold** entries in the tablet The sum of the variables at the given coordinates are equal to 0, hence constraint (15) does not hold. In all other cases the sum is at least 1.

berth/time	i before j, no overlap	i before j, overlap	j before i, no overlap	j before i, ov
i above j, overlap	(0,0,1,0)	(0,0,0,0)	(0,0,0,1)	(0,0,0,0)
i above j, no overlap	**(0,1,1,0)**	(0,1,0,0)	**(0,1,0,1)**	(0,1,0,0)
j above i, overlap	(0,0,1,0)	(0,0,0,0)	(0,0,0,1)	(0,0,0,0)
j above i, no overlap	**(1,0,1,0)**	(1,0,0,0)	**(1,0,0,1)**	(1,0,0,0)

The remaining constraints ensure that all variables are selected within their range.

Meisel and Bierwirth (2009) find that solving this model is very time consuming and only possible for small instances with a low number of vessels. Therefore, heuristics are used to find solutions. The model provides lower bounds to test the quality of the heuristics. In this paper the BACAP is revisited using both a more modern computer (3.1 GHz instead of 2.4 GHz) and a newer version of CPLEX (12.8 instead of 9.1). The results therefore will most likely show better performance of the BACAP and better lower bounds to compare the heuristics to.

5 Methods

In the paper of Meisel and Bierwirth (2009), a heuristic is described to find a solution to the BACAP. Vessels are inserted in a schedule one by one in a predetermined order. Meisel and Bierwirth (2009) start by ordering the vessels based on their arrival time. Then the heuristic procedure is performed to find a solution. The solution of this heuristic is compared to a solution in which the heuristic is paired with refinement procedures. Meisel and Bierwirth (2009) provide two extra procedures to improve solutions, based on their heuristic. These so-called meta-heuristics only change the order in which the vessels are scheduled and call the heuristic several times until no better solution is found. The construction heuristic performs eight steps and a few checks, which will briefly be discussed next.

5.1 Construction Heuristic

The construction heuristic starts by setting the contribution of vessel i to the total cost of the berth plan to ∞. An important observation is that (1) can be split over i, a specific vessel. The total costs are only the individual costs of all vessels combined. Therefore, vessels can be scheduled one by one in a berth plan and their contribution to the total costs can easily be identified. The next step is to set the start handling time to the expected arrival time, that is $s_i = ETA_i$. This is a good starting point, because no extra costs are made to speed up the vessel and there is the maximum amount of time before a penalty arises for being delayed. Afterwards, the same is done for the berth position, so $b_i = b_i^0$. Given a starting time and location, the number of QCs can be determined in the next step.

The QCs are assigned in a uniform manner. This means that in each period of the handling of vessel i, the same amount of cranes are assigned to the vessel, if these cranes are available in that period. Then cranes are added in each period, until the crane demand is satisfied or the maximum allowed number of cranes is assigned to the vessel. Assigning QCs in a uniform way is done because of the interference coefficient α. Adding a crane in a period with k cranes is more efficient than adding a crane in a period with $k + 1$ cranes. There are two possible outcomes of the QC assignment. Either a feasible QC assignment was found, in that case overlapping is checked, or no feasible QC assignment was found, in which case a new starting time is determined. Overlapping is checked by constraint (15).

If vessels do overlap, a new berth position that has not yet been checked is selected. If this is not possible, a new starting time is selected in an iterative way. That is, the next starting time is the first, not yet selected from the list $ETA_i - 1, ETA_i + 1, ETA_i - 2...$ up to EST_i and H respectively. If a vessel is not overlapping with previously scheduled vessels, its contribution to the total costs Z_i is calculated and the procedure starts again with a new starting time to find a better solution. When two consecutive solutions are more expensive than Z_i, the procedure is stopped and vessel i is inserted in the best found location. Two consecutive solutions are used, because then, both the earlier and later starting times do not give better solutions. Note that this heuristic will not always yield a feasible solution for the overall problem, although such a solution might exist. This is because of the fact that once a ship is inserted, it cannot be moved in the berthplan. If this is the case, this can be solved by either solving the MIP model or by using a different heuristic. One of those heuristics will be discussed in this paper.

5.2 Refinement Procedures

Meisel and Bierwirth (2009) provide two refinement procedures. The first of them, QC resource leveling, fixes the number of used cranes to a value lower than or equal to r_i^{max}, to give vessels that are scheduled earlier less of an advantage by assigning them less cranes. The second refinement procedure, spatial and temporal shifts, shifts blocks of vessels to a position in which the total cost of that block of ships is lower than in the previous position. These refinement procedures will be discussed in more detail in the following sections.

5.2.1 QC Resource Leveling

The QC Resource Leveling procedure (QCRL) is a procedure that tries to reduce the advantage of vessels that are high in the priority list. This is done by creating several berthplans, one for each resource level r. The procedure chooses each value from r_i^{min} up to and including r_i^{max} and adds vessel i to the berthplan with this new r_i^{max}. Then all other ships that are lower on the priority list are added in the standard way. In order to improve the procedure, vessel i is then removed again from the berthplan and then reinserted. In this way, vessel i can make use of the remaining berth space and cranes that are not used by other vessels, without having a higher priority than these vessels. The total costs of all berthplans are calculated and the berthplan with the lowest cost is selected. All vessels lower on the priority list than vessel i are removed, after which i is set to $i + 1$ and the procedure continues in the same way, up to $n - 1$ vessels. The last vessel is inserted in the normal way, because it does not have to take into account other vessels. The advantage of this refinement procedure is that for each vessel, the crane assignment is chosen in such a way that it results in the lowest total cost of the berthplan, instead of the lowest cost for only that vessel. However, this refinement procedure still does not allow to shift ships or relocate them even if this would result in a berthplan with a lower total cost.

5.2.2 Spatial and Temporal Shifts

The Spatial and Temporal Shifts procedure (STS) is a procedure that relocates blocks of vessels, to find a berthplan with lower total cost. The previous refinement procedure considers the crane assignment of vessels, this procedure focuses on the berth position of vessels. This refinement procedure is particularly useful if two vessels have similar arrival times and similar preferred berth locations. If that is the case, the first of these vessels on the priority list, claims its preferred spot, whereas the second vessel will be placed further from its preferred location or at a different time period.

For the STS procedure, first a berthplan is constructed using the QCRL procedure. Then, all blocks, called clusters, of vessels are identified. There are two types of clusters: a spatial cluster, that is a cluster of vessels that share at least one handling period and are scheduled directly next to each other in the berthplan, and a temporal cluster, that consists of vessels that share at least one berth segment and are scheduled directly after each other in the berthplan. After a list of these clusters is constructed, the shifting part of the algorithm starts. First, all spatial clusters are shifted one berth segment up (if they are not already at the border). Then, for each vessel in the cluster, the crane assignment is re-evaluated. If a crane can be removed without violating the crane demand, it is removed in the period in which the most cranes are assigned to that vessel. If the maximum number of cranes is assigned in more time periods, the first of these periods is selected. If the original crane assignment violates the crane demand in the new position, the vessel is removed from the cluster and added later on in the procedure. The violation of the crane demand is possible, because of the berth penalty coefficient β. After the berthplan is updated and all freed up cranes are set available in the berthplan, the vessels that were removed in the previous stage are added to the berthplan. Then, the cost of this new berthplan is calculated. The same procedure is repeated, but then shifting all vessels in the spatial cluster one berth segment down from their original location. If shifting the ships up results in lower total costs, all ships are shifted up and the next phase of the procedure starts. If shifting the vessels down results in lower or equal total costs, all ships are shifted down. If both shift directions increased the cost of the berthplan, this berthplan is deleted from the cluster list.

After the shift procedure is applied to all spatial clusters once, the same procedure is applied to the temporal clusters, only this time shifting the clusters one time segment. If both temporal shift directions do not result in a better berthplan, that cluster is no longer considered. The STS procedure terminates if the set of both clusters is empty or if the solution did not improve in 200 iterations. Note that it is possible to shift vessels, without improving the solution. This is the case if both a down shift and an up shift do not change the solution. The vessels in the cluster are then still shifted, because it might be possible to improve the solution by shifting a cluster for instance down twice, without improving the solution in the first shift.

5.3 Meta Heuristics

The order in which vessels are inserted in the berth plan by the construction heuristic is very important. Meisel and Bierwirth (2009) start by using a first come first serve ordering (FCFS). That is: they construct a priority list based on the arrival times of the vessels. Meisel and Bierwirth (2009) provide two heuristics that change the order of the priority list in order to find a better berthplan and to construct solutions that cannot be constructed by the given heuristics, provided the FCFS priority list. First, the squeaky wheel optimization heuristic (SWO) is provided, in which the priority list is changed after each call of the construction heuristic. Second, a TABU search algorithm is provided, that evaluates all priority lists that can be created using one swap from the original priority list.

5.3.1 Squeaky Wheel Optimization

A requirement for the SWO heuristic is that the costs of a berthplan can be split into the individual costs of the vessels (Meisel and Bierwirth (2009)). From (1) it can be seen that that is the case for this problem. Therefore, the total costs of a berthplan can be split in costs per ship. In calculating the individual cost of a vessel, Meisel and Bierwirth (2009) ignore the cost of using QCs, because the QC demand is different for different vessels. In this paper however, these costs are taken into account as well. In order to compensate for the different QC demand, the QC cost is divided by the crane demand of the vessel. The SWO procedure then is relatively straight forward: given an initial priority list, all ships are inserted using the construction heuristic combined with the refinement procedures and the individual costs per vessel are calculated. Then, starting at the top of the priority list, a vessel is swapped with the next vessel in the list if the cost of the vessel is lower than the cost of the next vessel. In following iterations, the total cost of the vessel so far are considered, instead of the cost of the vessel in that particular berth plan. That is, the total cost of vessel i is the sum of the cost of vessel i in all previously obtained berthplans The procedure is repeated until no better solution is found in 200 iterations. If the SWO heuristic is in a loop, the construction heuristic without refinement procedures is used until a new solution is found. A loop is detected by comparing the order of the priority list to all previously used orders in the SWO heuristic. It is found that loops occur quite frequently, even if the construction heuristic is used without refinement procedures.

5.3.2 TABU Search

The TABU search heuristic also changes the order of the priority list. First a solution is calculated using the given priority list and the refinement procedures. Then all priority lists that can be constructed using only one swap of 2 vessels from the original priority list are saved. For all these priority lists, the construction heuristic without refinement procedures is used to find the best solution among these priority lists. Then, this best solution is added to the TABU list, the list of priority lists that will not be searched again. For this best solution, also the refinement procedures are used to improve the solution. Afterwards, the current solution is updated. In the paper of Meisel and Bierwirth (2009), the stopping condition of this heuristic is set to 200 iterations without any improvement. However, it is found that this can take quite long, because of the large number of explored priority lists in one iteration, and the heuristic usually does not find an improvement after a few iterations. Therefore, the number of iterations is set to 20 in this paper.

5.4 Hybrid Heuristics

In this section, a heuristic is provided that combines several of the already mentioned heuristics. All provided heuristics share the property that, once a vessel is inserted, it cannot be moved (only in a spatial/temporal cluster). However, in many cases this can result in better solutions. Therefore a hybrid heuristic is provided, that removes and moves vessels if that can yield a better solution

5.4.1 Hybrid Construction Heuristic

The hybrid construction heuristic (HC) is a combination of the the construction heuristic and the MIP model. The aim of the heuristic is to use the construction heuristic if placing the vessel in the berthplan is easy, and the MIP model if placing the vessel is difficult. The HC heuristic starts with placing the vessels with the construction heuristic. If a vessel i overlaps with an already placed vessel, detected by using (15), the construction heuristic is terminated, the overlapping vessel(s) are removed from the berthplan and vessel i and the vessel(s) that overlap are placed using the MIP model in an empty berthplan. The number of available cranes in each period is set to $Q-$ the used cranes by the already placed vessels in order to avoid crane capacity issues. If the resulting placement of these vessels overlaps with vessels that are already in the berthplan, these ships are removed and the MIP model is solved again with the newly overlapping vessels. If there is no new overlap, the vessels are placed in the berthplan, and the construction heuristic continues with a new vessel. By using an empty berthplan, the MIP model does not take into account already placed vessels. This aids finding better solutions and grants the ability to move all already placed vessels, thereby making the order of the priority list irrelevant. The already used cranes are subtracted from the available cranes in the MIP model. It is possible to set the number of available cranes to Q. Then, if a solution of the MIP model cannot be placed in the the berthplan, the vessels that use cranes in the period(s) of a shortage of available cranes are removed from the berthplan and added in the MIP model. In this paper, it was chosen not to do is, because this can increase the running time of the algorithm substantially. When all vessels are placed, the crane assignment problem is solved, given the positions of the vessels.

The solution found by an HC is not necessarily the optimal solution. because of the already assigned cranes in the berthplan. An idea to improve the heuristic is to run an CAP model after all vessels are placed. However, this also does not guarantee an optimal solution, because it might be effective to change the position of vessels, given their new crane assignment. In this heuristic, several MIP models are solved. In order to reduce the running time, the CPLEX solver is terminated is the optimality gap is 1 percent or the running time of the solver is more than 100 seconds. In case no solution is found by the solver in the time limit, the heuristic is terminated and does not provide a solution. For these cases other heuristics can be used, as described in this paper.

6 Planning flexibility of QCs

In this section one of the extensions of the Meisel and Bierwirth (2009) paper will be discussed: the case of more flexible planning of QCs, without changing the flexibility of planning the vessels. The idea of this case is that it is possible to move the QCs more frequently, i.e. twice in an hour. This can improve the solution of the corresponding berthplan in two ways. First, by assigning QCs per half hour instead of an hour, a more precise number of QCs can be assigned in order to just meet the crane demand requirement. Therefore it is possible to reduce the number of used QCs by 1. Second, the freed up QCs can be used for other vessels, making it possible to reduce the total handling time of other vessels and thereby reducing the need to arrive earlier of leave later than originally planned. This can reduce the cost to a larger extent than reducing the number of QCs, because the cost for speeding up and leaving later than planned outweigh the cost of one QC. In this thesis, the BACAP model is adapted for the flexible planning case and the Construction Heuristic is also used to evaluate the advantage of a more flexible planning of QCs.

The new BACAP model with flexible planning is as follows:

$$min\ Z = \sum_{i \in V} (c_i^1 * \Delta ETA_i + c_i^2 * \Delta EFT_i + c_i^3 * u_i + 0.5 * c^4 \sum_{w \in W} \sum_{q \in R_i} q * r_{iwq}) \qquad (20)$$

$$\sum_{w \in W} \sum_{q \in R_i} q^{\alpha} * r_{iwq} \geq 2 * (1 + \beta * \Delta b_i) * m_i, \ \forall i \in V \tag{21}$$

$$\sum_{i \in V} \sum_{q \in R_i} q * r_{iwq} \leq Q, \ \forall w \in W \tag{22}$$

$$\sum_{q \in R_i} r_{iwq} = r_{iw}, \ \forall i \in V, \ \forall w \in W \tag{23}$$

$$\sum_{w \in W} r_{iw} = 2 * (e_i - s_i), \ \forall i \in V \tag{24}$$

$$(w + 1) * r_{iw} \leq 2 * e_i, \ \forall i \in V, \ \forall w \in W \tag{25}$$

$$w * r_{iw} + H - Z * r_{iw} \geq 2 * s_i, \ \forall i \in V, \ \forall w \in W \tag{26}$$

$$r_{iwq}, r_{iw}, u_i, y_{ij}, z_{ij} \ \in \{0, 1\}, \ \forall i, j \in V, \ \forall w \in W, \ \forall q \in R_i \tag{27}$$

in which W is the set of half hours in the planning period H. All other constraints remain the same, because they do not affect the assignment of QCs. (20) is altered with a new cost coefficient. Because time periods are now split in half, the cost of using a crane for half an hour are also halved. In constraint (21), the crane demand is multiplied by 2, to account for the shorter time periods. Constraints (22) and (23) only changed with respect to their time index, from t to w. Constraint (24) translates time periods of an hour and half an hour, such that assigning cranes is only possible in the half hour that vessel i is handled. Constraints (25) and (26) ensure that no cranes are assigned outside of the serving hours of vessel i.

The Construction Heuristic is changed in QC assignment phase. The number of periods is doubled and the crane demand of vessel i is doubled as well. In order to benefit from the more flexible planning of QCs, a flexible planning of vessels is preferred. The Construction Heuristic will reduce the number of QCs used by the vessels, however it is less likely to use the potential of shorter handling times, because of the inflexible nature of the heuristic. Therefore, the BACAP model with flexible planning of QCs is also run in CPLEX. In the Results section, the difference in objective values between the standard BACAP and the BACAP with flexible QC planning will be evaluated for the data sets of size 10.

7 Data

In order to compare the four different heuristics, a data set is required. The dataset of Meisel and Bierwirth (2009) cannot be used, because it is not publicly available. Therefore, a data set is randomly generated as well, using the classification of vessels provided by Meisel and Bierwirth (2009). This classification can be found in Table 1. The data set is provided by the supervisor and contains 20 instances of 10, 20 and 30 vessels. For all vessels, the length l_i, the number of required QC-hours, m_i, the minimum and maximum number of cranes that can serve the vessel in one time period, r_i^{min} and r_i^{max} respectively, and the cost coefficients c_i^1, c_i^2 and c_i^3 are specified.

8 Computations and Results

In this section, the performance of the presented algorithms and heuristics is investigated. All heuristics were programmed in Java, similar to Meisel and Bierwirth (2009). CPLEX version 12.8 was used to solve the mip model. The maximum solver time was 1800 seconds and the solver terminated if the relative gap was 0.01 for the data set containing 30 vessels and 0.005 for the other data sets. All computations were run on a 3.1 GHz laptop. For all heuristics, $\alpha = 0.9$ and $\beta = 0.01$.

8.1 Solving MIP Model

In Tables 2, 3 and 4 the results of the CPLEX solver can be found. From this table it is clear that the CPLEX solver performs well for smaller data sets, but for some larger data sets it struggles to find a solution or to converge the lower and upper bound. This is particularly the case for 'difficult' data sets, that are data sets in which the expected arrival time and preferred berth location are similar for several ships. In this paper, the lower bounds obtained by the CPLEX solver are used to measure the quality of the solutions of the heuristics. Therefore, for all data sets lowerbounds are obtained, although CPLEX can not find solutions for most cases in which n is 30. A suggestion to improve the CPLEX solver can be to provide a warm start for CPLEX or to obtain an upper bound by means of one of the heuristics. Then, CPLEX can use this solution or lower bound to reduce the search tree. In the cases for which no solution is found, the search tree gets very large and slows the computer down.

8.2 Construction Heuristic

In Tables 5, 6 and 7 the solutions obtained by means of the construction heuristic can be found. Compared to the CPLEX solver, running the construction heuristic is done considerably faster. The running time is not specified in the table, because it was always less than a second. For larger data sets, the relative gap of the construction heuristic is considerable. This is particularly the case in data sets in which large ships form clusters. That is, in data sets for which several larger ships have similar arrival times and preferred berthing locations. Because the construction Heuristic places the ships in order of arrival, it can not change the position of an earlier arriving vessel for the benefit of a later arriving vessel. However, the construction heuristic provides a good foundation for next heuristics. Note that the construction heuristic is unable to find a solution for 2 cases with n is 30. Therefore, for all other heuristics, only the data sets of size 10 and 20 will be considered.

8.3 Refinement Procedures

In Table 8 and 9 the solutions obtained by both refinement procedures can be found. As in Meisel and Bierwirth (2009), only the results for Spatial and Temporal shift heuristic are provided. However, this heuristic uses the first refinement procedure, the resource leveling heuristic. Compared to the construction heuristic, the obtained results have improved. The average gap in the case of n is 20 has reduced from 0.378 to 0.162, effectively halving the gap between the obtained solution and the lower bound that is found by CPLEX. In the next section, the results of the meta-heuristics are discussed.

8.4 Squeaky Wheel Optimization

In Table 8 and 9 the running time and solutions for the SWO heuristic are shown. SWO clearly outperforms the construction heuristic combined with refinement procedures. This was to be expected, because the first solution obtained by SWO is the same solution as in the refinement procedures. After finding this solution, SWO tries to improve the solution by changing the order of the priority list. It is important to note that the run time of the SWO heuristic is hard to forecast. In most cases, the run time is relatively short, but in other cases, the run time can get rather long. In order to shorten the run time of the SWO heuristic, it is advised to reduce the number of iterations without improvement before terminating the heuristic. In the paper of Meisel and Bierwirth (2009) this is set at 200. However, after a few iterations without improvement, the algorithm will not find better solutions in most cases. The SWO heuristic often creates a loop which it will not escape. Therefore a TABU heuristic might be a good approach for this problem.

8.5 TABU

In Table 8 and 9 the solutions of the TABU heuristic are provided. From the tables it can be seen that the TABU and SWO heuristic perform similarly for n is 10, but for n is 20 the solutions are slightly worse. For n is 20, the SWO heuristic has an average gap of 0.08, whereas the TABU heuristic has an average gap of 0.114. The disadvantage of TABU is the large number of checked priority lists. In each step all single swaps within the lists are investigated. Therefore the running time of the TABU heuristic is longer than the running time of the SWO heuristic. Because of the longer running time, the TABU heuristic can search less iterations than the SWO heuristic and it performs the search by means of the construction heuristic instead of the construction heuristic with refinements. This results in solutions that are on average worse.

8.6 Hybrid Heuristic

The Tables 10 and 11 contain the solutions obtained by the hybrid heuristic. This heuristic clearly outperforms the best of the rest SWO heuristic. For n is 20, the average gap is 0.059 compared to 0.08 for the SWO. However, there are several downsides of this heuristic. The running time is on average exactly the same (both 39), but for several data sets, the running time of the hybrid heuristic is very long. For one data set, the hybrid heuristic was not able to find a solution, because the running time of CPLEX was exceeded. For that case it is advised to use the SWO heuristic. For larger data sets it is expected that the hybrid heuristic will not always find solutions, unless the running time of CPLEX is changed.

8.7 Flexible QC planning

The final result that is provided in this paper can be found in Table 12. This table compares the CPLEX solutions and the solutions obtained by the construction heuristic for the case of flexible QC planning. QCs can be moved after half an hour instead of a full hour. The results show an improvement in the obtained solution. This is to be expected, because the solution space of the problem is extended, but all previously feasible solutions remain feasible in this problem. On average, the solutions improve by around 2 percent. However, to draw stronger conclusions the CPLEX procedure has to be repeated, with longer running times to find more accurate solutions. What the consequences of a more flexible QC planning are for larger data sets is a subject for further research.

9 Conclusion

In this paper the research of Meisel and Bierwirth (2009) is replicated. The main conclusions are similar to the results of Meisel and Bierwirth (2009). First, CPLEX was used to find solutions for the BACAP problem. Due to a newer version of CPLEX and a faster computer, it was possible to find solutions more often. Then construction heuristic was used to find solutions. The performance of this heuristic is similar to that found by Meisel and Bierwirth (2009). Another important aspect of this paper is the performance of the provided meta-heuristics. For the BACAP problem, SWO outperform TABU in most cases.

Also, a new heuristic was provided, the so-called hybrid heuristic. This heuristic combines the CPLEX solver and the construction heuristic and outperforms all heuristics provided by Meisel and Bierwirth (2009). A drawback of this heuristic is that it will not always yield a solution, which was the case in one of the test data sets. The heuristic can be improved by adding and solving a CAP after the initial QC and QS assignment.

The final conclusion of this paper is that a more flexible QC planning can result in an improvement of around 2 percent. However, it is unclear what the improvement can be for larger data sets. Further research can be done in this topic, by for instance using the hybrid heuristic to solve the BACAP with flexible QC planning faster. Then, it is possible to gain an insight in the effect of more flexible QC planning in busier container terminals.

References

Al-Dhaheri, Noura, Aida Jebali, and Ali Diabat (2016). “The quay crane scheduling problem with nonzero crane repositioning time and vessel stability constraints”. In: *Computers & Industrial Engineering* 94, pp. 230–244.

Elwany, Mohammad Hamdy, Islam Ali, and Yasmine Abouelseoud (2013). “A heuristics-based solution to the continuous berth allocation and crane assignment problem”. In: *Alexandria Engineering Journal* 52.4, pp. 671–677.

Imai, Akio, Ken’Ichiro Nagaiwa, and Chan Weng Tat (1997). “Efficient planning of berth allocation for container terminals in Asia”. In: *Journal of Advanced transportation* 31.1, pp. 75–94.

Imai, Akio, Etsuko Nishimura, and Stratos Papadimitriou (2001). “The dynamic berth allocation problem for a container port”. In: *Transportation Research Part B: Methodological* 35.4, pp. 401–417.

Lai, KK and Katharine Shih (1992). “A study of container berth allocation”. In: *Journal of advanced transportation* 26.1, pp. 45–60.

Lee, Der-Horng, Hui Qiu Wang, and Lixin Miao (2008). “Quay crane scheduling with non-interference constraints in port container terminals”. In: *Transportation Research Part E: Logistics and Transportation Review* 44.1, pp. 124–135.

Liang, Chengji, Youfang Huang, and Yang Yang (2009). “A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning”. In: *Computers & Industrial Engineering* 56.3, pp. 1021–1028.

Meisel, Frank and Christian Bierwirth (2009). “Heuristics for the integration of crane productivity in the berth allocation problem”. In: *Transportation Research Part E: Logistics and Transportation Review* 45.1, pp. 196–209.

Moon, KC (2000). “A mathematical model and a heuristic algorithm for berth planning”. In: *Brain Korea* 21.

Park, Young-Man and Kap Hwan Kim (2005). “A scheduling method for berth and quay cranes”. In: *Container terminals and automated transport systems*. Springer, pp. 159–181.

Ramani, KV (1996). “An interactive simulation model for the logistics planning of container operations in seaports”. In: *Simulation* 66.5, pp. 291–300.

Sammarra, Marcello et al. (2007). “A tabu search heuristic for the quay crane scheduling problem”. In: *Journal of Scheduling* 10.4-5, pp. 327–336.

Ting, Ching-Jung, Kun-Chih Wu, and Hao Chou (2014). "Particle swarm optimization algorithm for the berth allocation problem". In: *Expert Systems with Applications* 41.4, pp. 1543–1550.

Vacca, Ilaria, Matteo Salani, and Michel Bierlaire (2013). "An exact algorithm for the integrated planning of berth allocation and quay crane assignment". In: *Transportation Science* 47.2, pp. 148–161.

Wang, Fan and Andrew Lim (2007). "A stochastic beam search for the berth allocation problem". In: *Decision support systems* 42.4, pp. 2186–2196.

Wang, Junjia, Hao Hu, and Yuanbin Song (2013). "Optimization of quay crane scheduling constrained by stability of vessels". In: *Transportation Research Record* 2330.1, pp. 47–54.

Zhang, Canrong et al. (2010). "The allocation of berths and quay cranes by using a sub-gradient optimization technique". In: *Computers & Industrial Engineering* 58.1, pp. 40–50.

10 Appendix

Table 1: Dataset specifiactions

Class	l_i	m_i	r_i^{min}	r_i^{max}	c_i^1	c_i^2	c_i^3
Feeder	U[8,21]	U[5,15]	1	2	1	1	3
Medium	U[21,30]	U[15,50]	2	4	2	2	6
Jumbo	U[30,40]	U[50,65]	4	6	3	3	9

Table 2: CPLEX with N=10

N=10				
#	Z	LB	GAP	Running Time
0	38.2*	38.2	0.0	8
1	11.0*	11.0	0.0	1
2	27.1	27.0	0.4	27
3	29.5	29.4	0.4	7
4	21.5	21.4	0.5	11
5	44.7	44.5	0.4	7
6	25.8	25.7	0.4	6
7	43.4	43.2	0.5	19
8	30.5*	30.5	0.0	9
9	20.1*	20.1	0.0	4
10	12.4*	12.4	0.0	1
11	37.4	37.2	0.5	15
12	30.3	30.2	0.5	19
13	38.3	38.1	0.4	8
14	14.3*	14.3	0.0	3
15	27.6*	27.6	0.0	7
16	22.8*	22.8	0.0	1
17	30.2	30.1	0.5	3
18	32.2	32.0	0.5	2
19	30.5	30.4	0.4	16

Table 3: CPLEX with N=20

N=20	Z	LB	GAP	Running Time
0	47.3	47.06	0.4	55
1	52.9	52.65	0.4	19
2	48.6	48.40	0.4	47
3	32.8	32.66	0.4	23
4	-	60.35	-	1800
5	-	57.98	-	1800
6	53.3	53.0	0.4	182
7	-	111.59	-	1800
8	100	90.12	9.0	1800
9	64.1	63.78	0.4	487
10	35.6	35.45	0.4	487
11	50	49.76	0.4	312
12	56.5	56.2	0.4	284
13	72.3	67.89	6.0	1800
14	-	82.78	-	1800
15	-	82.58	-	1800
16	55.9	55.6	0.4	405
17	42.9	42.72	0.4	316
18	56.3	56.02	0.4	431
19	-	65.92	-	1800

Table 4: CPLEX with N=30

N=30	Z	LB	GAP	Running Time
0	60.6	59.99	0.01	1655
1	-	109.98	-	1800
2	64.8	63.03	0.02	1800
3	-	107.82	-	1800
4	99.30	98.31	0.01	1330
5	-	81.08	-	1800
6	-	79.68	-	1800
7	-	103.93	-	1800
8	-	147.34	-	1800
9	48.5	48.01	0.01	416
10	-	133.01	-	1800
11	-	135.19	-	1800
12	-	94.01	-	1800
13	-	184.81	-	1800
14	-	78.39	-	1800
15	-	91.76	-	1800
16	-	99.34	-	1800
17	-	101.71	-	1800
18	-	129.88	-	1800
19	63.2	62.5	0.01	1240

Table 5: Construction Heuristic with N=10

N=10	Z	LB	GAP
0	40.2	38.2	0.05
1	11.0	11.0	0
2	39.2	27.0	0.45
3	41.2	29.4	0.40
4	26.2	21.4	0.22
5	55.8	44.5	0.25
6	25.8	25.7	0.01
7	51.4	43.2	0.18
8	41.4	30.5	0.36
9	20.2	20.1	0.01
10	12.4	12.4	0
11	53.1	37.2	0.42
12	74.8	30.2	1.47
13	42.3	38.1	0.11
14	16.4	14.3	0.15
15	28.4	27.6	0.03
16	22.8	22.8	0
17	39.9	30.1	0.33
18	36.5	32.0	0.14
19	32.0	30.4	0.05

Table 6: Construction Heuristic with N=20

N=20	Z	LB	GAP
0	48.4	47.06	0.03
1	60.4	52.65	0.15
2	59.0	48.4	0.21
3	36.5	32.66	0.11
4	109	60.35	0.81
5	72.6	57.98	0.25
6	65.0	53.3	0.21
7	182.8	111.59	0.64
8	169.4	90.12	0.87
9	76.5	63.78	0.20
10	52.6	35.45	0.48
11	72.2	49.76	0.45
12	67.2	56.2	1.47
13	143.8	67.89	0.20
14	148.0	82.78	0.79
15	139.1	82.58	0.69
16	63.1	55.6	0.13
17	46.0	42.72	0.08
18	83.6	56.02	0.48
19	103.7	65.92	0.58

Table 7: Construction Heuristic with N=30

N=30	Z	LB	GAP
0	69.9	59.99	0.17
1	-	109.98	-
2	88.5	63.03	0.40
3	599	107.82	4.55
4	114	98.31	0.16
5	195	91.04	1.14
6	197.9	76.68	1.58
7	339.1	103.93	2.29
8	397.9	147.34	1.70
9	52.7	48.01	0.09
10	273.5	113.01	1.41
11	567.5	135.19	3.19
12	179.7	94.01	0.90
13	-	184.81	-
14	220.5	78.39	1.82
15	173.4	91.76	0.89
16	223.7	99.34	1.25
17	368.5	101.71	2.65
18	212.7	129.88	0.64
19	83.4	62.5	0.33

Table 8: Results for RP, SWO and TABU with N=10

N	LB	RP	GAP	SWO	Runtime	GAP	TABU	Runtime	GAP
0	38.2	40.2	0.05	38.2	6	0	38.2	3	0
1	11.0	11.0	0	11.0	0	0	11.0	0	0
2	27.0	39.2	0.45	29.0	5	0.07	28.0	4	0.04
3	29.4	41.2	0.40	30.1	4	0.02	30.3	5	0.03
4	21.4	26.2	0.22	21.5	4	0.01	21.5	3	0.01
5	44.5	44.7	0.01	44.7	13	0.01	44.7	10	0.01
6	25.7	25.8	0.01	25.8	6	0.01	25.8	1	0.01
7	43.2	46.4	0.07	43.4	13	0.01	43.4	10	0.01
8	30.5	41.4	0.36	30.5	6	0	30.5	3	0
9	20.1	20.1	0	20.1	1	0	20.1	0	0
10	12.4	12.4	0	12.4	1	0	12.4	0	0
11	37.2	53.1	0.43	37.4	11	0.01	37.4	6	0.01
12	30.2	45.4	0.50	30.3	7	0.01	30.3	14	0.01
13	38.1	41.9	0.10	38.3	7	0.01	38.3	12	0.01
14	14.3	16.4	0.15	14.3	1	0	14.3	3	0
15	27.6	28.4	0.03	27.6	1	0	27.6	2	0
16	22.8	22.8	0	22.8	1	0	22.8	0	0
17	30.1	37.9	0.26	31.1	4	0.03	31.1	1	0.03
18	32.0	36.5	0.14	32.2	19	0.01	32.2	5	0.01
19	30.4	32.6	0.06	30.5	1	0.01	30.5	0	0.01

Table 9: RP, SWO and TABU with N=20

N=20	LB	RP	GAP	SWO	Runtime	GAP	TABU	Runtime	GAP
0	47.06	48.4	0.03	47.3	29	0.02	47.4	35	0.03
12	52.65	60.3	0.14	55.6	27	0.05	56.7	19	0.08
2	48.4	57.6	0.19	49.0	41	0.01	49.0	20	0.02
3	32.66	34.4	0.05	33.0	45	0.01	33.6	25	0.02
4	60.35	108.8	0.80	76.8	39	0.27	76.0	19	0.25
5	57.98	73.4	0.27	67.1	0.47	0.17	70.3	52	0.20
6	53.0	65.0	0.23	54.4	41	0.02	54.4	38	0.02
7	111.59	166.7	0.49	156.1	104	0.4	158.9	61	0.41
8	90.2	225.9	1.51	110.4	38	0.22	116.9	27	0.19
9	63.78	74.2	0.16	64.1	16	0.01	64.1	18	0.01
10	35.45	46.6	0.32	35.6	10	0.01	35.6	7	0.01
11	49.76	71.9	0.47	50.1	36	0.01	50.0	25	0.01
12	56.2	62.8	0.11	58.5	19	0.03	58.9	13	0.04
13	67.89	142.9	1.12	75.2	65	0.11	78.3	47	0.16
14	87.78	135.9	0.55	105.4	42	0.20	106.8	31	0.22
15	82.58	129.7	0.55	109.9	53	0.33	113.4	29	0.35
16	55.6	58.8	0.05	57.7	52	0.02	57.7	11	0.02
17	42.72	45.9	0.07	44.7	17	0.03	44.7	23	0.03
18	56.02	80.4	0.43	58.1	39	0.02	58.2	23	0.02
19	65.92	103.6	0.56	77.4	51	0.18	78.2	35	0.19

Table 10: Hybrid with N=10

N=10	LB	Hybrid	Run Time	GAP
0	38.2	38.2	0	0
1	11.0	11.0	0	0
2	27.0	27.1	1	0.01
3	29.4	29.4	1	0
4	24.4	24.5	1	0.01
5	44.5	44.5	3	0
6	25.7	25.8	1	0.01
7	43.2	43.3	1	0.01
8	30.5	30.5	3	0
9	20.1	20.1	1	0
10	12.4	12.4	0	0
11	37.2	37.4	2	0.01
12	30.2	30.3	3	0.01
13	38.1	38.3	3	0.01
14	14.3	14.3	0	0
15	27.6	27.6	0	0
16	22.8	22.8	0	0
17	30.2	30.1	4	0.01
18	32.0	32.2	1	0.01
19	30.4	30.5	1	0.01

Table 11: Hybrid with N=20

N=20	LB	Hybrid	Running Time	GAP
0	47.06	47.2	2	0.01
1	52.65	52.9	1	0.01
2	48.4	49.8	1	0.02
3	32.66	32.8	0	0.01
4	60.35	65.5	16	0.08
5	57.98	65.7	94	0.14
6	53.0	53.4	4	0.01
7	111.59	164.6	339	0.47
8	90.12	100	21	0.11
9	63.78	64.1	2	0.01
10	35.45	35.6	2	0.01
11	49.76	50	3	0.01
12	56.2	56.5	5	0.01
13	67.89	69	13	0.02
14	87.78	97.8	212	0.11
15	82.58	-	-	-
16	55.6	56.0	1	0.01
17	42.72	42.9	2	0.01
18	56.02	56.3	2	0.01
19	65.92	74.2	71	0.12

Table 12: Flexible QC planning, CPLEX and CH

N=10	Z	LB	Running Time	Z CH
0	38.25	36.37	6	40.1
1	10.85	10.85	1	10.85
2	27	25.6	5	38.95
3	29.45	28.4	6	40.85
4	21.55	21.0	12	26.1
5	44.95	42.7	9	46.55
6	25.5	24.36	6	25.5
7	42.1	40.6	11	51.15
8	30.35	29.0	7	41.3
9	19.85	19.75	6	19.85
10	12.15	12.15	2	12.15
11	35.4	33.6	6	52.95
12	28.3	26.9	5	74.65
13	38.3	37.1	6	42.1
14	14.05	13.95	6	16.2
15	27.4	27.2	9	28.15
16	22.55	22.4	1	22.5
17	29.95	29.11	3.25	39.65
18	32.5	31.77	4	36.35
19	30.30	28.8	5	32.55