Crane scheduling for a Plate Storage in a Shipyard: Experiments and Results

Jesper Hansen[*]   Torben F. H. Kristensen[§]

April 23, 2003

Abstract

This document is the third in a series of three describing an investigation of possible improvements in methods for handling the storage of steel plates at Odense Steel Shipyard (OSS). Steel ships are constructed by cutting up plates and afterwards welding them together to produce blocks. These blocks are again welded together in the dock to produce a ship.

Two gantry cranes move the plates into, around and out of the storage when needed in production. Different principles for organizing the storage and also different approaches for solving the problem are compared. Our results indicate a potential reduction in movements by 67% and reduction in time by 39% compared to current practices. This leads to an estimated cost saving by approx. 1.0 mill. dkr. per year.

This paper describes the experiments and achieved results based on the model and the approaches to solve the problem described in Hansen and Kristensen [2] and [3].

[*] Informatics and Mathematical Modelling, Technical University of Denmark, 2800 Lyngby, Denmark, email:jha@imm.dtu.dk

[§] Department of Production, Aalborg University, 9220 Aalborg Ø, Denmark, email: tfhk@wanadoo.dk

1 Introduction

This paper is the third in a series of three papers describing an investigation of possible improvements in methods for handling the storage of steel plates at Odense Steel Shipyard (OSS). Steel ships are constructed by cutting up plates and afterwards welding them together to produce blocks. These blocks are again welded together in the dock to produce a ship.

In Hansen and Kristensen [2] we described the physical environment on the plate storage and how it fits into the overall process of building steel plate ships. Based on the problem description, a model was developed including both the physical entities of the system and the planning and processing aspects. In this paper we will discuss experiments and achieved results, but we will first briefly repeat the main modelling concepts from [2].

Our model of the storage consists of 8 × 24 plate stacks and two gantry cranes can move plates between the stacks. 8 additional stacks are used for arriving plates. Each plate has a due date specifying at which date it must leave the storage and enter the production line. When a plate leave the storage, it is placed on a conveyer belt referred to as the exit-belt. A maximum of 8 plates can be at the belt at the same time and a plate is drawn from the exit-belt at certain intervals. The two gantry cranes share tracks and can hence not cross each other. The adressed problem is to develop approaches for scheduling the crane operations better than the current practices. Three different storage principles are considered:

The block storage is the current storage principle. Two stacks are associated to a specific block of the ship and all plates to be used to produce that block are placed in these designated stacks.

The due date storage was suggested by the shipyard management as an alternative to the block storage. Here each stack is assigned a due date interval and a plate with a due date in that interval can be placed on that stack. The storage is divided into zones and each zone is divided into a number of due date intervals. Several stacks are assigned each due date interval. The user can determine the layout of the storage by specifying the different parameters regarding size of the intervals, stacks per interval, number of intervals and overlap in due dates between zones. Refer to [2] for more details.

The self-regulating storage principle is the last alternative. No specific purpose or due date is assigned to the stacks. The organization of the storage is determined by the planning procedure.

In Hansen and Kristensen [3] different approaches to solving the problem of scheduling cranes at plate storage was discussed. The problem is hard to solve since it is both a question of placing the plates on stacks in order to minimize future movements, but also scheduling the crane movements to minimize moved distance and at the same avoiding collisions. Two approaches are investigated. The first approach is an *on-line algorithm* or more precisely a *heuristic discrete event feedback control system* where an operation is chosen in on-line. The other approach uses the on-line algorithm off-line as a greedy construction heuristic to get a good initial solution, which is then improved by local search heuristics. The off-line algorithm makes a schedule for the controller to follow. The schedule specifies the order in which the movements are to be executed. A control module dispatches the operations in the sequences and adjusts the schedule if the initial schedule becomes infeasible caused by the underlying stochastic nature of the system.

The major goal of the investigation reported in this and earlier papers is to compare the three proposed ways of managing the steel plate storage: The block, due date and self-regulating storage. Experiments reported in Hansen and Clausen [1] shows that a change from the block principle to any of the two others would result in a saving of approximately 50% in number of movements and 40% in total makespan. No further experiments have been conducted for the block storage since it is clearly inferior. Instead we will focus on a comparison of the due date and the self-regulating principles.

The paper is organized as follows: In section 2 the different local search methods described in Hansen and Kristensen [3] are compared. In section 3 we describe the experimental setup and report on the results comparing the different storage principles and solution approaches. Finally in section 4 we conclude on the findings.

2 Comparison of Search Methods

The complexity of the problem makes it impossible to evaluate exactly feasibility and change in cost for neighbour solutions without simulating the entire sequence. Experiments on estimating the change instead were reported in [3] showing the difficulty of estimating the cost function. Before picking a neighbour as a new solution when using estimation, the correct objective value is found. If the value is improving the current solution, we pick the neighbour solution and otherwise the next best estimate is evaluated. This is repeated until all neighbours are evalutated. At that stage we select the neighbour with the best correct value which is then worse than the current solution. For the Steepest Descent algorithm and Tabu Search, estimation was still better than a complete evaluation since complete evaluation was too time consuming.

Here we compare the performance of the different local search methods that we have implemented. More precisely a Simulated Annealing, a Steepest Descent algorithm and a Reactive Tabu Search. The last two with estimation and evaluation of only a subset of the neighbours. Unless a subset is chosen, the search will be too time consuming. Initially the subset has size 100, but it is gradually increased when the estimates deteriorate as explained in [3].

The experiments are divided into two parts. First we discuss the comparison for the due date principle only. The results are shown in section 2.1. Following that, the conclusion is tested on the self-regulating storage principle. This experiment is described in section 2.2.

2.1 Tests for the Due Date Storage Principle

The test instance has 1856 movements on one day performed by one crane. The potential number of solutions are 1856! not including destination stack decisions. All three algorithms are run for 2 hours on the same machine.

In figure 1 on the following page the runs of the different search methods are depicted. For the Simulated Annealing the objective value seems to converge nicely to 65402. Looking at the Tabu Search it converges much faster and reaches the same value after just 443 seconds ending at 63434. Finally the simple descent algorithm rather surprisingly reaches a similar value and at that stage still finds improving solutions i.e. a local optimum is not found within two hours. The Descent and Tabu Search only iterates through approximately 1800 solutions.

In some cases, in the end over 500 neighbour solutions are evaluated. This is an indication of the difficulty of estimating the change in objective function. Apparently 499 solutions were estimated better than the chosen one, but all of them were non-improving.

Further investigating the supposedly good performance of the descent algorithm, we exactly evaluate all 100 neighbours in the subset before picking the best of them – a form of steepest descent. The search reached a local optimum of the sub neighbourhood after only 70 iterations with an objective

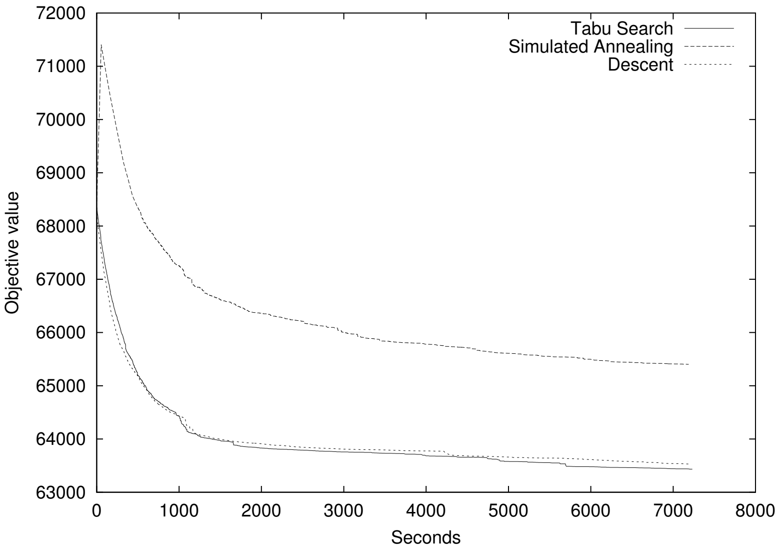

Figure 1: Comparison of different search methods.

value of 67621. The "more gentle" descent, estimating the objective instead of exact evaluation gives a better convergence, since the estimation error allows the search to avoid local optima.

The descent algorithm is then run for more than two hours. After almost 5 hours and 2256 iterations a solution with value 62916 is achieved. In the final iteration 1560 random neighbour solutions were evaluated taking 1.5 minutes without finding an improving solution. There is of course no guarantee that this is actually a local optimum, but it is still surprising how much better the performance is compared to the Simulated Annealing.

To check the possible variation of quality over different runs, 5 runs are performed and the results plotted in figure 2 on the next page. The longer running times are to compensate for a slower machine. Note that the time actually runs out without a local optimum being reached. The variation is quite small with a spread of only 42 indicating a robust search method.

We now return to the Simulated Annealing. In the beginning the temperature may be too high, since the search is not able to recover the large increase in objective value. We decrease the initial probability of accept to 0.01 even though in figure 1 it is only 0.1. This results in a lower start temperature and

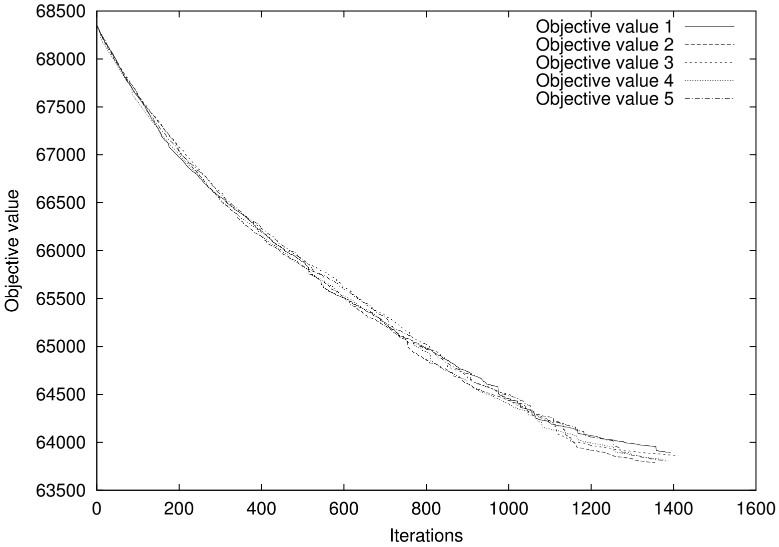

Figure 2: Steepest Descent with neighbourhood subset.

therefore be closer to the descent behavior. In figure 3 on the following page we have plotted the original run and 4 new runs with $p_0 = 0.01$ together with the Descent and Tabu Search runs. The Simulated Annealing is now clearly competitive within the given amount of runtime.

The runs are performed for a quite large instance measured in number of movements. To verify the results similar tests are carried out on a smaller instance with 753 movements. The initial solution will most likely be closer to a local optimum. The hypothesis is that the Descent Algorithm will sooner get caught in a local optimum, while the Simulated Annealing and Tabu Search will be able to escape and end up with better solutions. 5 runs are executed with a 18000 CPU seconds available for all runs in total. Figure 4 on the next page shows the descent runs where all the runs reach a local optimum using 9538 CPU seconds in total. The average objective value is 23212. Note that the local optima are for the limited subset size mentioned earlier.

For the Tabu Search in figure 5 on page 8 the average is 23152 using all 18000 seconds.

For the Simulated Annealing in figure 6 on page 9 the average is 23826 with initial probability of accept $p_0 = 0.1$. In some cases the search gives

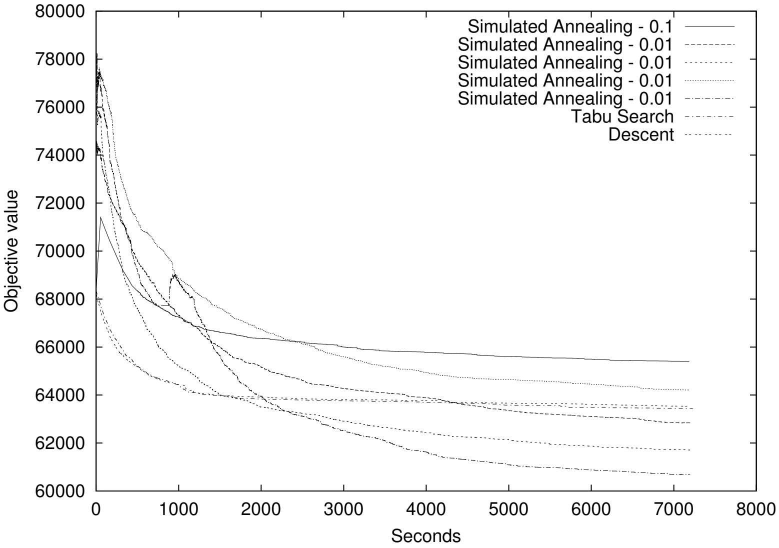

Figure 3: Simulated Annealing with $p_0 = 0.01$ compared to runs in figure 1.

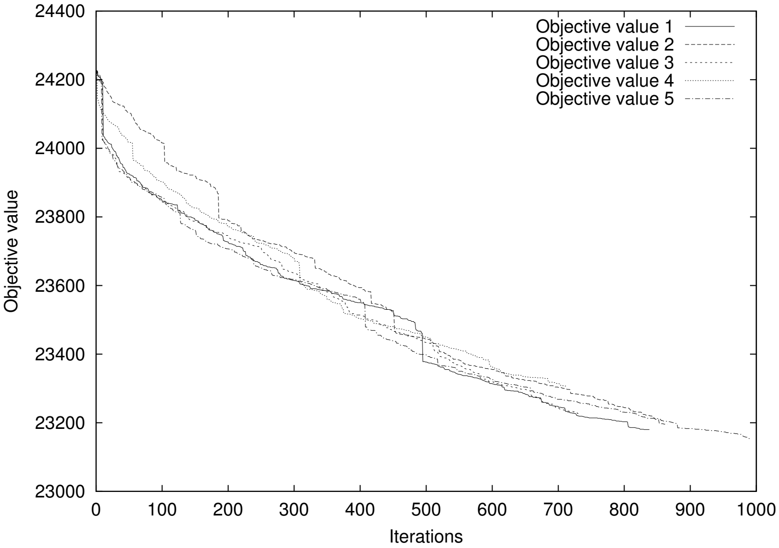

Figure 4: Descent on a smaller instance.

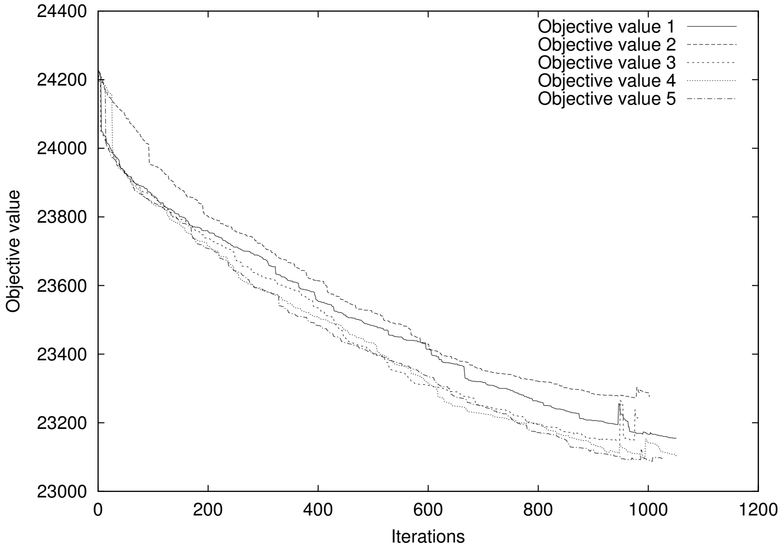

Figure 5: Tabu Search on a smaller instance.

solutions not far from the Tabu Search, but in other cases it gets completely off track. We again tried to set the accept probability to 0.01 resulting in the plot in figure 7 on the following page. The average is now down to 23653, but we still observe runs with very poor performance.

Using Simulated Annealing, all considered neighbours are evaluated exactly, so estimation can not be blamed for the (in some cases) bad performance indicated in figures 6 and 7. The plot in figure 8 on page 10 perhaps gives another hint. Here the objective values for all tried neighbours are depicted. It is clear that the "small" and "local" changes in the solution resulting in neighbour solutions have a significant and not "small" impact on the solution. This is not surprising, since we have earlier discussed the difficulty of estimating these local changes. They *do* have a global impact. Together with the notoriously slow convergence, Simulated Annealing seems inferior on this problem type.

In conclusion Tabu Search seems to have the most robust behaviour, since we do not observe runs with very poor performance as is the case with Simulated Annealing. If the planning time is very limited Tabu Search and the Descent Algorithm are superior, since they are much more intensive in the

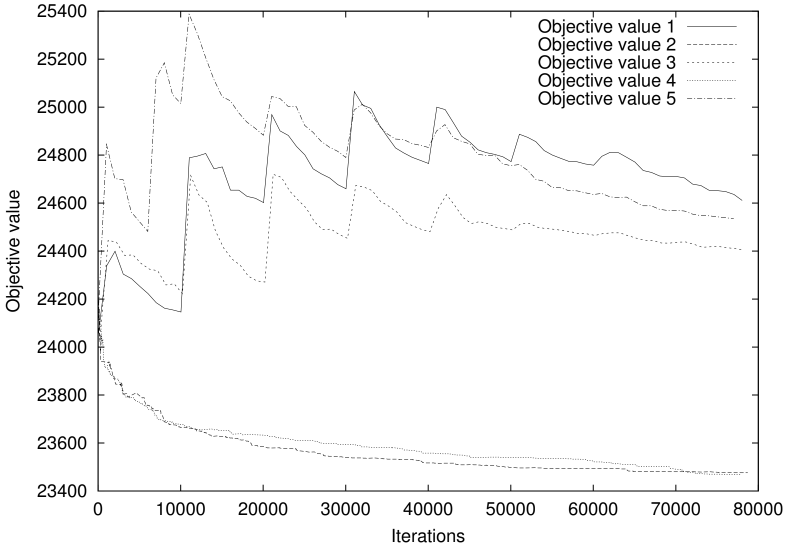

Figure 6: Simulated Annealing on a smaller instance with $p_0 = 0.1$.

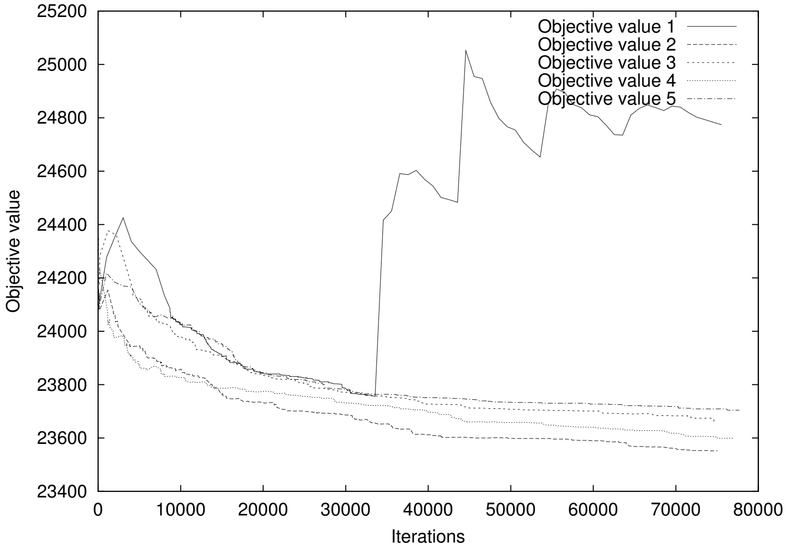

Figure 7: Simulated Annealing on a smaller instance with $p_0 = 0.01$.

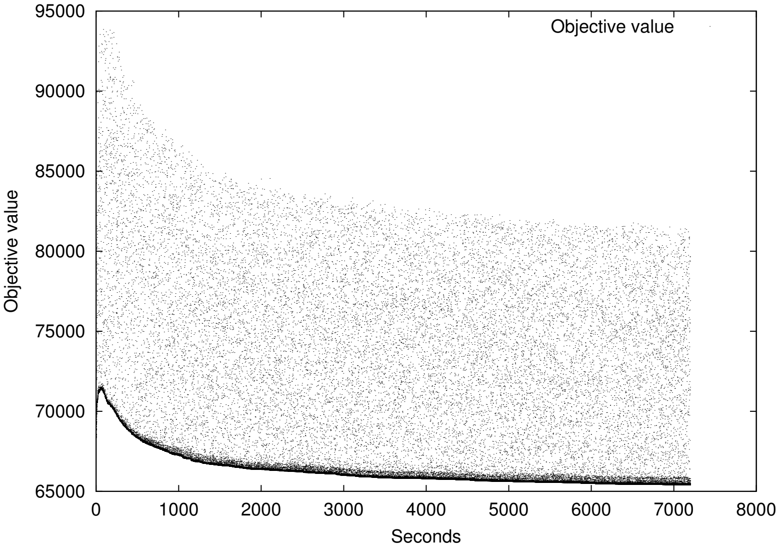

Figure 8: Simulated Annealing.

beginning of the search while Simulated Annealing is much less aggressive. Tabu Search is generally better than the Descent Algorithm if more time is available.

2.2 Tests for the Self-Regulating Storage Principle

We now investigate whether the results for the due date principle also applies for the self-regulating storage principle. We have limited ourselves to a comparison between Simulated Annealing and Tabu Search. For the self-regulating principle more degrees of freedom exist since all stacks are potential destinations for all but the exit-movements. Hence more iterations are needed for finding good quality solutions. 6 hours of computation time is available for both methods. Figure 9 on the following page shows the objective values for the two methods and Tabu Search is clearly superior.

Running the Simulated Annealing and Tabu Search over 50 working days on the storage gives the picture shown in figure 10 on page 12. Only the first day a better solution is achieved with Tabu Search. The remaining days are significantly better with Simulated Annealing. This result is quite surprising

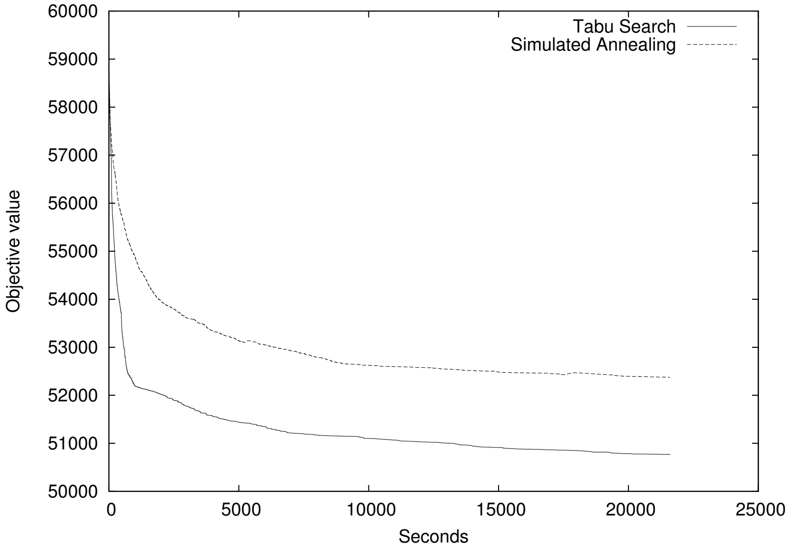

Figure 9: Objective value for the self-regulating principle for 1 day.

since supposedly Tabu Search is better.

Recall that the objective value was better the first day with Tabu Search, we observe that Tabu Search actually creates better solutions with regard to total time (makespan) and movement time (total moved distance for both cranes in time) as indicated in figure 11 on the following page.

Investigating matters further reveals that the difference lies in the ability to leave a well-sorted storage. In figure 12 on page 13, the stack sorting is plotted for the two methods. Clearly Simulated Annealing reaches much better solutions with regard to stack sorting than Tabu Search. This is the reason that Simulated Annealing the following 50 days consistently performs better than Tabu Search. The objective function seems to be too short-sighted focusing on time spent today rather than in the future. Another possible reason is that Tabu Search is using cost estimation, which might favour time improvements over stack sorting. We performed an experiment with the stack sort cost multiplied by 10. The stack sort was naturally improved, but still worse than Simulated Annealing with the original cost and more time was spent as well. In the following experiments, we will hence use Simulated Annealing only for the self-regulating storage principle.

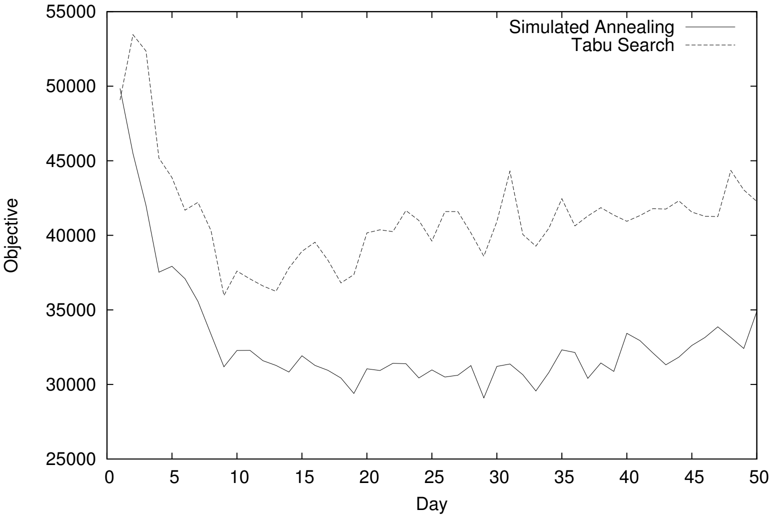

Figure 10: Objective value for the self-regulating principle over 50 days.

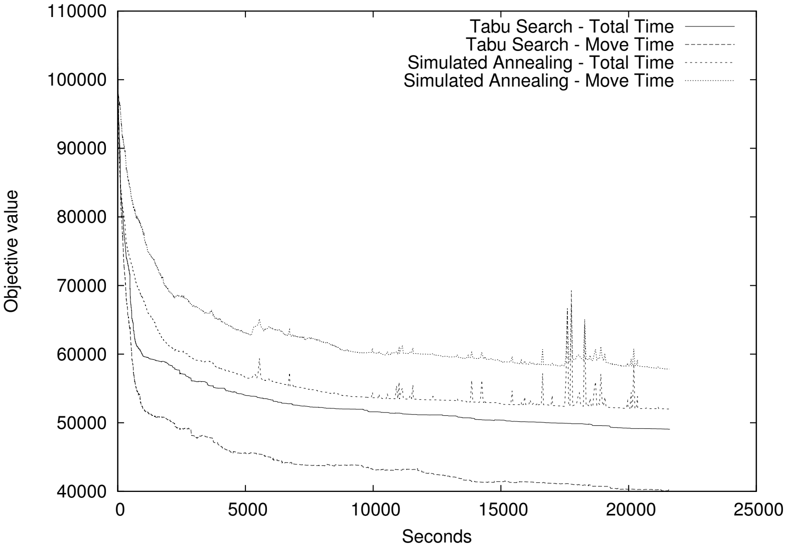

Figure 11: Time for the self-regulating principle for 1 day.

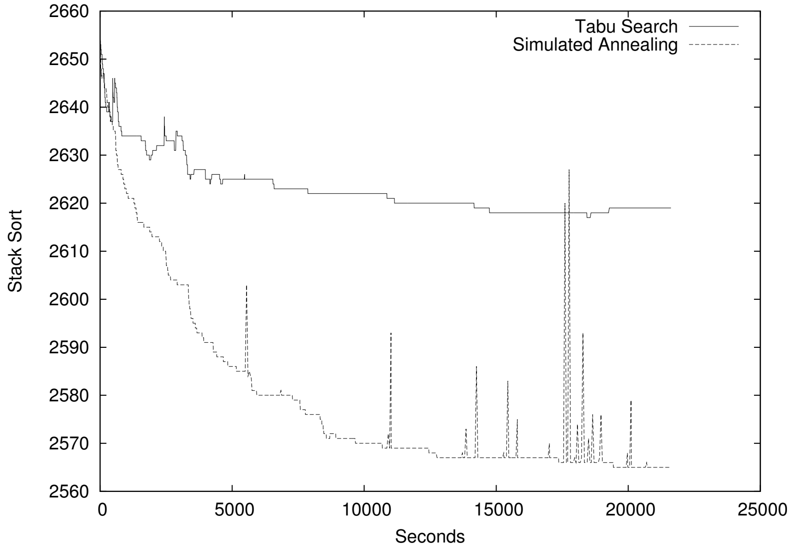

Figure 12: Stack sort for self-regulating principle for 1 day.

3 Experiments and Results

In section 2 the performance of the different local search methods were investigated. The best approach was shown to be Tabu Search for the due date principle and Simulated Annealing for the self-regulating principle. In the following we will therefore only use these combinations of methods and principles. The focus will be on comparing the on-line control algorithm and the approach of combining planning and control.

The layout of zones for the due date principle has an impact on the performance. Another issue is the periodically large peaks of work when using the due date principle. In section 3.1 we show the gain in performance when smoothing out peaks of work. The performance of the self-regulating storage is adjusted by the objective function, which is discussed in section 3.2. Finally in section 3.4 we compare the different solution approaches on the different storage principles to reach a final conclusion.

In all the experiments 121 plates arrive to the storage and leave the storage every day over 50 working days. The number of days where all plates have arrived is 10 days and the period where plates are accumulating on the storage is adjusted by the accumulation parameter, which is set to 2 (See Hansen and Kristensen [2]). This leads to a total of 3630 plates placed in 199 stacks – 18.24 on average in each stack. Every 5 days 500 plates change due date; all 500 plates have a due date from 10 to 20 days into the future. A uniformly random number of plates between 0 and 20 with due date today and tomorrow swap due date with other plates having due date no more than 20 days into the future. Disturbances on lift/drop times and exit-belt are only enabled for the experiments in section 3.4, when comparing the different control systems.

For the due date storage the plates are initially placed according to that principle, while for the self-regulating principle the plates are placed according to the block storage. This initially gives an advantage to the due date principle, but by measuring the performance as the average of the days from 20 to 50 for both storage types, the unfairness is removed. Figure 10 on page 12 confirms our hypothesis for the self-regulating storage.

For the planning module 1200 CPU seconds (20 minutes) are available for each day for the due date storage and 10800 seconds (3 hours) for the self-regulating storage. The search space for the self-regulating storage is much larger than the due date storage, which justifies the increased run times. When comparing solutions we consider only the 3 most important cost factors: The exit and total makespan and number of movements.

3.1 Due Date Storage Layouts

Finding the best storage layout for the due date principle is a cumbersome task. An obvious idea is to construct an automatic procedure to search the space of different layouts and pick the best of the visited – a meta problem of the crane scheduling problem. Considering the required runtime for such a procedure makes it next to impossible. Instead we make a less ambitious attempt simply trying a few different layouts using our intuition. Doing this will most likely not result in the best layout, but a good one is sufficient.

The first layout is a slight modification of the one originally suggested by the yard management. The parameters are shown in table 1 on the next page. The yard suggested that zone 1 consists of one week of one day stacks closest to the current day. We have changed it to 8 days, since the storage has 8 rows. Zone 2 would consist of several weeks of stacks with 5 day intervals. We have again chosen 8 intervals of 5 days. The last zone would have 2 intervals of 1 month each, which in our version is 10 days, since no more due dates into the future are required to place all plates on the storage.

Parameter	Zone 1	Zone 2	Zone 3
Number of days in each interval	1	5	10
Number of due-date intervals	8	8	2
Number of stacks per interval	6	13	20
Overlap between zones	2	5	

Table 1: Originally suggested Due Date storage layout.

The original suggestion leads to huge fluctuations in the number of movements over time. This is mainly because of the quite large intervals in zone 3, but also in zone 2. At given intervals the due dates of the stacks are changed and the stacks must be emptied. The plates are typically moved to several different intervals in the next zone. If the intervals were smaller resulting in fewer plates per interval, then the due dates would change more often, but less plates had to be moved per change. This is the motivation for the alternative suggestion where the intervals in zone 2 and 3 have been reduced to 2 and 4 days and the number of intervals in zone 3 has increased to 8, shown in table 2.

Parameter	Zone 1	Zone 2	Zone 3
Number of days in each interval	1	2	4
Number of due-date intervals	8	8	8
Number of stacks per interval	6	10	8
Overlap between zones	1	2	

Table 2: Alternative Due Date storage layout

In a perfect world, 4 moves are necessary to move a plate from arrival through the 3 zones and further on to the exit belt. Clearly by reducing the number of zones the number of necessary movements would decrease. Using only one zone with 1 day intervals would lead to too long travel distances and crane conflicts on days where a stack far from the exit belt is emptied. Using two zones makes more sense. In our suggestion zone 2 has 16 intervals of 3 days, shown in table 3 on the next page.

The result of the 3 suggested layouts are shown in table 4 where “Exit” is the makespan of exit movements and “Total” is the makespan of all movements

Parameter	Zone 1	Zone 2	Zone 3
Number of days in each interval	1	3	0
Number of due-date intervals	8	16	0
Number of stacks per interval	6	9	0
Overlap between zones	1	0	

Table 3: Due Date storage layout with 2 zones.

both in hours. “MovesD” and “MovesP” are the average number of movements per day and per plate respectively. On average the original suggestion by the yard is better than our alternative. The run with two zones has a somewhat smaller number of movements per day, but needs more time on average to perform the moves. More time is spent on unproductive waiting and moving away from the other crane. Basically the cranes are more in conflict. In the following we use only the two-zone layout, since the primary objective is number of movements.

Due Date Layout	Exit	Total	MovesD	MovesP
Original	6.0	9.0	552.0	4.6
Alternative	6.0	9.3	595.7	4.9
Two zones	6.1	9.6	524.4	4.3

Table 4: Comparing different due date layouts.

In the above runs, there were a maximum of 14.4 hours available each day for both cranes and the fluctuations in work were reduced by changing the layout. Another way of reducing the fluctuations or smoothing out the workload is simply to reduce the available amount of work time for each day. Any remaining movements, after the available time has been used, are not executed, but delayed until the next day. Figure 13 on the following page illustrates the result when smoothing out the movements by reducing the available time. The fluctuations with 10 hours are much smaller than with 14.4 hours.

For 2 cranes we made two runs with 9, 10 and 14.4 hours and for 1 crane 14.4, 16 and 21.4 hours. 14.4 and 21.4 are 2 and 3 work shifts. The results are shown in table 5 on the next page.

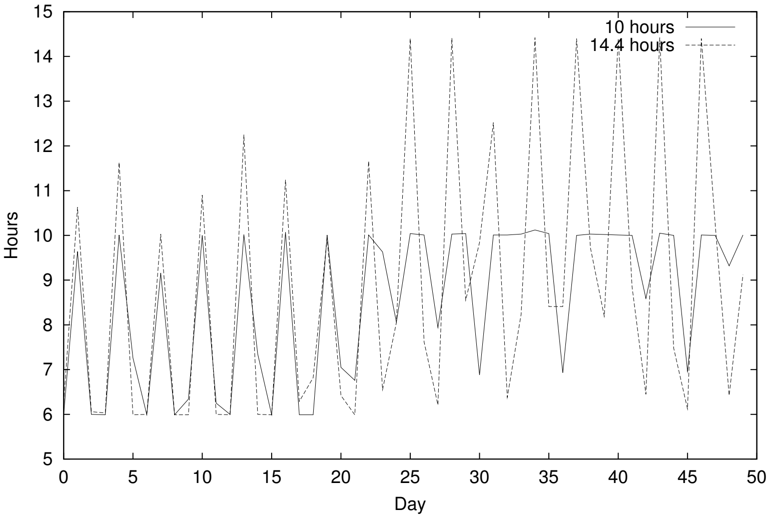

Figure 13: Total Makespan with 10 and 14.4 hours available time.

Max Time of Day		Exit	Total	MovesD	MovesP
2 cranes,	14.4 hours	6.1	9.6	524.4	4.3
2 cranes,	10.0 hours	6.1	9.3	516.2	4.3
2 cranes,	9.0 hours	6.4	8.8	530.4	4.4
1 crane,	21.4 hours	6.4	16.5	529.8	4.4
1 crane,	18.0 hours	6.3	16.9	559.0	4.6
1 crane,	16.0 hours	6.7	15.5	508.0	4.2
2 cranes,	10.0 hours				
no due date disturbance		6.0	8.5	363.3	3.0

Table 5: Smoothing out movements.

When lowering the available time for the two cranes, the average time decreases. The average number of movements is reduced as well, because movements of some plates are moved directly from the arrival stacks to zone 1 or from zone 2 directly to the exit-belt. When comparing 10 and 9 hours, the average total time still reduces, but the number of movements and exit time increases. The reason for this is illustrated in figure 14. The remaining movements to be executed at the end of the day is plotted over the 50 days. In the first half of the period there are peaks, but the remaining number of movements are regularly reduced to zero. This is not the case in the second half with only 9 hours, resulting in increasing numbers of remaining movements.

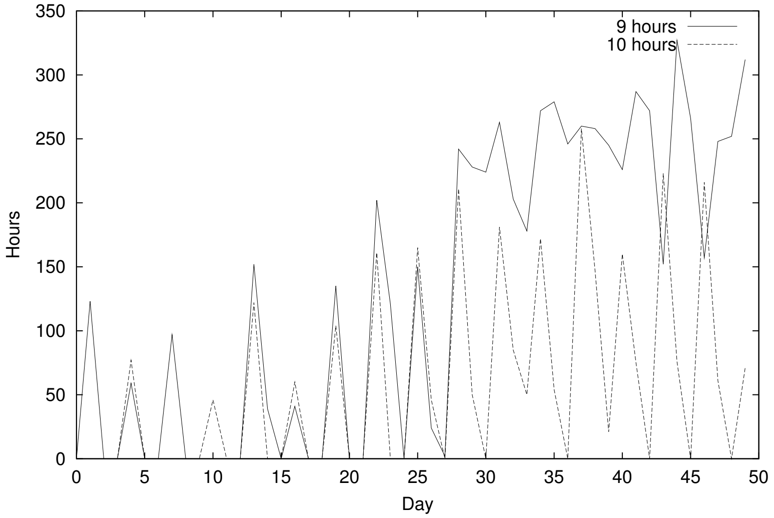

Figure 14: Remaining moves each day after 9 and 10 hours available time.

We conclude that lowering the available time can be useful for reducing the average working time, but only to a certain degree. It must be possible regularly to finish more or less all movements. In practice for instance once a week. The conclusions are however quite fragile since the due date disturbances introduce noise in the comparison. The results for 1 crane in table 5 illustrate this fact. By decreasing the available time from 21.4 to 18 hours the average time actually increase caused by additional movements, however reducing the exit time. Decreasing again to 16 hours reduces the amount of possible movements, significantly aggravating the exit time.

Table 5 also shows the result of a run without due date disturbances. We see that the theoretical limit of 3 movements per plate is actually achievable. Another way of reducing the number of movements is hence to reduce the magnitude of changes on due dates, i.e. make better or more robust plans for the entire yard.

Now let us compare the above results with the results in table 6 achieved with the control module alone. For two cranes with 10 hours, the results are approximately the same while for 14.4 hours the time is reduced by 8.5% when using Tabu Search. Generally the exit time is improved with approximately 5% as well. The number of movements are also the same, which is expected. In the case of one crane the planning module however is significantly better measured both in time and movements.

Parameters	Exit	Total	MovesD	MovesP
Control module				
2 cranes, 14.4 hours	6.3	10.5	534.3	4.4
2 cranes, 10.0 hours	6.4	9.3	518.3	4.3
1 crane, 21.4 hours	7.1	16.6	583.0	4.8
Planning module				
2 cranes, 14.4 hours	6.1	9.6	524.4	4.3
2 cranes, 10.0 hours	6.1	9.3	516.2	4.3
1 crane, 21.4 hours	6.4	16.5	529.8	4.4

Table 6: Control vs. Planning module.

In conclusion, not much is gained for the due date storage by using more complex approaches such as Tabu Search in case of two cranes. In case of one crane the conclusion is the opposite.

3.2 Adjusting the Self-Regulating Storage

The performance of the self-regulating storage is adjusted by the weights in the objective function. There are four main weights to set. The weights on exit and total time reflect the hourly wages for the crane operators and operators on the machines following directly after the storage. The machines are manned by 6 people and the exit weight is hence significantly more important than the weight on total time. The costs on lifting and dropping plates are quite high and perhaps the most important cost factor. The last cost factor is the cost of moving the cranes. It is insignificant compared to the other three factors. In the following experiment we have kept the costs on time and crane movement constant and only changed the cost on movements. In fact the parameter we are adjusting in the objective function is not directly corresponding to the number of movements, but the number of future dig-up movements. We refer to the parameter as the stack sort parameter. The default value is 7.2 and trials with 0.72 and 72.0 have been made as shown in figure 7. Only one trial with 7.2 have been made with only one crane. The experiments clearly show a dependency between the sort parameter and the average number of movements. Increasing the weight on sorting significantly reduces the number of movements, but the cranes travel longer distances and spend more time to place the plates on "good" stacks. Reducing the weight instead increases the number of movements, since the objective function favours reducing the time spend by the cranes today. Unfortunately the shortsighted focus increases the average time spend in the future since more movements are necessary.

Stack Sort		Exit	Total	MovesD	MovesP
2 cranes,	7.20	6.0	8.0	540.6	4.5
2 cranes,	0.72	6.0	8.5	675.9	5.6
2 cranes,	72.00	6.1	8.2	488.1	4.0
1 crane,	7.20	6.9	15.5	536.4	4.4

Table 7: Adjusting the self-regulating storage.

We have illustrated that it is possible for the management to prioritize between cost factors by adjusting the weights in the objective function. Note that the 1 crane solution might also be interesting since it is competitive both in total time and number of movements. Specially of the exit time is not critical.

Comparing these results with the results so far for the due date principle, we can conclude that it is possible to reduce the total time by approx. 12% and at the same time reduce the number of movements by approx. 5%. The results are repeated in table 8 on the next page.

Storage principle	Exit	Total	MovesD	MovesP
Self-regulating	6.1	8.2	488.1	4.0
Due-date	6.1	9.3	516.2	4.3

Table 8: Comparison of self-regulating and due date storage.

3.3 Comparison without Due Date Disturbances

In this section we discuss the results achievable in case of no due date disturbances. Even though this is an unrealistic case, it is still interesting also from a practical point of view. It gives an indication of the potential gains from the different storage principles. 3 experiments have been made: The due date storage with 2 cranes and the self-regulating storage with both 1 and 2 cranes. The results are shown in table 9 on the following page. In the table the "Construction" rows are the results achieved with the control module constructing the initial solution and the following row indicates the result achieved with Tabu Search or Simulated Annealing. The plates moved are in all cases only moved once, hence we see no change in number of movements. Clearly, the control module performs reasonably well on the due date storage, which we have seen earlier as well, but on the self-regulating principle significant improvements in both exit and total time is possible. In fact we do not even consider the improvement in stack sort. The last row of the table is the result when using only the control module. Hence when using Simulated Annealing, the number of movements is decreased by 35%, which is caused by the better sorting of the stacks.

Comparing the due date and self-regulating principles with 2 cranes, gives a potential gain from using the self-regulating principle with 21% in total time, but with an increase in movements of 5%.

3.4 Comparison with Time Disturbances

Finally comparisons are made when both due date and time disturbances are included. In table 10 on page 23 the results are shown. First the on-line control approach is used for the due date principle. Note that the time disturbances have no impact on the performance of the on-line control module. The reason is that the next decision taken by the algorithm is done after the actual time of the previous operation has been observed. This is confirmed by looking at

Methods	Exit	Total	MovesD	MovesP
Due date, 2 cranes				
Construction	6.0	9.8	363.3	3.0
Tabu Search	6.0	8.5		
Self-regulating, 2 cranes				
Construction	6.4	10.5	382.0	3.2
Simulated Annealing	6.0	6.7		
Self-regulating, 1 crane				
Construction	7.1	16.2	370.0	3.1
Simulated Annealing	6.2	13.3		
Self-regulating, 2 cranes				
Control	6.7	10.3	590.6	4.9

Table 9: Without due date disturbances.

table 6 on page 19.

The control module working on the basis of a plan is inferior to the on-line control approach for the due date storage. This is not surprising when considering the very simple control procedure deployed together with the fact that insignificant improvements are possible by using off-line planning.

Finally we consider control on the basis of a plan for the self-regulating storage. The total time is increased by 10% and the exit time by 8.3% compared to the case without time disturbances for two cranes. Even with this increase the time is still comparable with the due date principle. The exit time is similar while the total time is approximately 5% better, but with 3% more movements.

For one crane the total time and exit times are only increased by 4.5% and 1.5% respectively compared to the case without time disturbances. The reason is that the plan is not disrupted because of occurring crane collisions, but only because of time disturbances on the exit belt. Plans with only one crane is hence more robust than with two cranes. If the increase in exit time when using one crane is not critical, it could be beneficial to use only one crane operator, since two crane operators in 8.8 hours is in total 17.6 man hours, while one crane operator alone could do the work in 16.2 hours.

Method & Principle	Exit	Total	MovesD	MovesP
On-Line control Due date storage				
2 cranes, 14.4 hours	6.3	9.7	523.5	4.3
2 cranes, 10.0 hours	6.4	9.3	520.9	4.3
1 crane, 21.4 hours	7.3	16.2	583.2	4.8
Control with a plan Due date storage				
2 cranes, 14.4 hours	6.7	10.6	531.6	4.4
Control with a plan Self-regulating storage				
2 cranes	6.5	8.8	537.2	4.4
1 crane	7.0	16.2	535.1	4.4

Table 10: On-Line control and control with a plan.

4 Conclusion

We have in this paper reported on a number of experiments conducted to compare different approaches and principles for improving the current practices on a steel plate storage. First Simulated Annealing and Tabu Search were compared leading to a conclusion depending on the storage principle at hand. Tabu Search was seemingly superior to Simulated Annealing, but the Tabu Search used an estimation of the cost function, which was not sufficiently precise for the self-regulating principle. It was hence optimizing in the “wrong” direction leading to solutions, which were inferior in the long run compared to Simulated Annealing.

Then we turned our attention to comparing the different storage principles, storage layouts and objective weights. The experiments indicate that significant improvements can be achieved. No final conclusion was reached on best layouts or weights. These decisions should be taken by the management and users of the system reflecting the overall goals of the organization.

To conclude, the on-line control approach is more robust to time disturbances and is performance-wise superior to control based on a plan for the due date principle.

The self-regulating principle on the other hand is dependent on improving the plan initially achieved with the on-line control approach. Better solutions time-wise can be achieved with the self-regulating storage compared to the due date storage, but the saving of 12% is reduced to 5% when introducing time uncertainties. Approximately the same number of movements is required. The user of the software can however adjust the behavior of the self-regulating storage to focus more on reducing the number of movements, reducing the number of movements down to 4 per plate. Compared to current practices this is a reduction by 67% and at the same time a reduction in time by 39% leading to a saving of approx. 1.0 mill. dkr. per year.

The development of the system and these investigations should not end here. Significant improvements can be made to all modules: The on-line control module, control module based on a plan and the planning module itself. One must however take into consideration whether the more complex approach of control based on a plan is sufficiently more promising than the more simple on-line control approach, which might be sufficient for the purposes of the yard.

References

[1] J. Hansen and J. Clausen, *Crane scheduling for a plate storage*, Informatics and Mathematical Modelling, Technical University of Denmark, IMM-TR-1 (2002).

[2] J. Hansen and T. F. H. Kristensen, *Crane scheduling for a plate storage in a shipyard: Modelling the problem*, Informatics and Mathematical Modelling, Technical University of Denmark, IMM-TR-4 (2003).

[3] ——, *Crane scheduling for a plate storage in a shipyard: Solving the problem*, Informatics and Mathematical Modelling, Technical University of Denmark, IMM-TR-11 (2003).