Crane scheduling for a Plate Storage in a Shipyard: Modelling the Problem

Jesper Hansen* Torben F. H. Kristensen§

April 23, 2003

Abstract

This document is the first 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 modelling aspects of the investigation. Hansen and Kristensen [8] and [9] describe the aspects of solving the model, experiments conducted and the achieved results.

*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 document is the first 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 cranes are used for moving the plates on the storage. Two crane operators control these cranes. A simulator and control systems have been developed to take the decisions normally taken by the crane operators. These control systems are designed such that they can be implemented in the real system and be used as decision support system for the crane operators.

The paper is organized as follows. First we describe the problem, including the physical characteristics of the plate storage and the operation of the storage. Section 3 gives an overview of related research. In section 4 we describe our model of the plate storage including the assumptions made for simplifying the model and in section 5 we conclude. The aspects of solving the problem based on the model described here is reported in Hansen and Kristensen [9] and experiments and results are given in Hansen and Kristensen [8].

2 Problem Description

2.1 The purpose of the plate storage

The plate storage contains the plates used for building blocks. The storage is a buffer between the suppliers of the plates and the production at OSS. The flow of plates through the storage can be considered as a dynamic process where new plates are added to the storage and replace plates that are removed from the storage. The plate storage is accumulating plates for a period of time to ensure that the plates needed in the production at a certain time can be delivered. Since steel works from around the world supply the plates, the accumulation of plates for one day's production can run over several weeks. Some plates are delivered a long time before the day and some are delivered close to it. A diagram of the plate storage is shown in figure 1 on the following page. The ellipse in the figure indicates the area which has been zoomed in and shown in figure 2 on page 4.

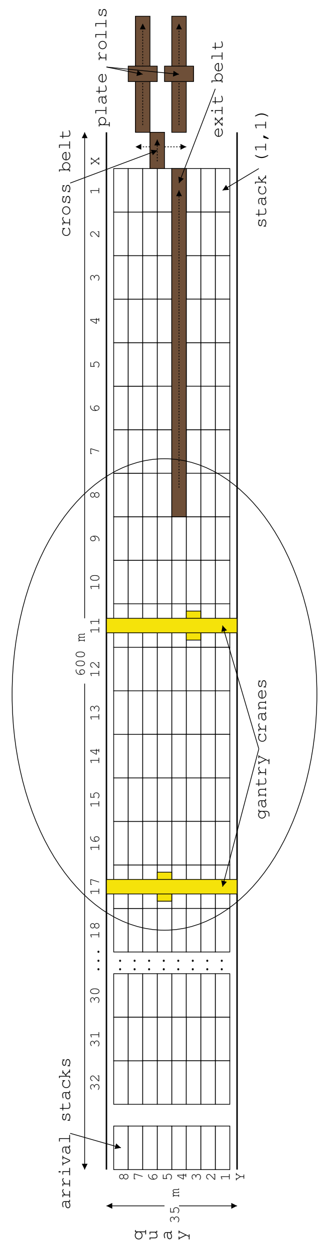

Figure 1: Illustration of the storage layout.

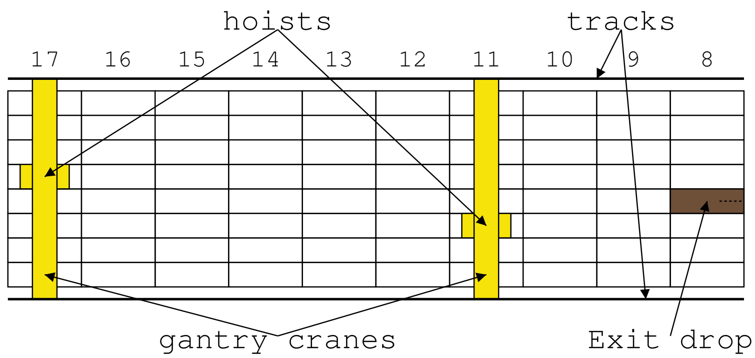

Figure 2: Zoomed illustration of the storage layout.

2.2 Stacks of plates

The plate storage is organized in 32 times 8 stacks of plates. X and Y directions indicate this in figure 1 on the preceding page. Note that the Y-axis is placed to the left, because of lack of space on the right. Stack $(1, 1)$ is indicated in the figure. The size of the storage is 600×35 meters. The storage contains 5.000 steel plates on average – approximately 20 plates are stored in each stack on average. One quarter of the stacks are used for *special purpose plates* and *surplus plates*, for example stacks with identical plates. The rest of the storage is used for plates, which for the main part have different sizes. Each plate is ordered for a specific purpose and the date on which it will be required in production is known. Changes in the production plan, can however influence the due dates of the plates in the storage, which means that the due date of a plate can change several times before it is required in production.

2.3 Gantry cranes

Two identical gantry cranes carry out the movements of plates. They can move with a speed of 3.3 m/s in both the X and Y directions. I.e. if t_1 is the movement time of a movement in one direction and t_2 in the other direction, the resulting movement time is $\max\{t_1, t_2\}$. A crane operator operates each of the cranes. The cranes share tracks and hence cannot pass each other. For each of the 32 rows the hoist of the cranes can reach each of the 8 stacks. The cranes use electro magnetics to lift the plates. Because of the use of magnetism, it sometimes occur that two plates are lifted at the same time because the magnetism "runs through" the upper plate. When this happens some seconds are lost while the crane operator gets rid of the plate that was not meant to be moved. Since the cranes are gantry cranes there is a limit on the height of the stacks.

2.4 Arrival of new plates

Most steel plates are delivered by sea from the steel works. Trucks also deliver a small number of plates. The new plates are moved from the ships into arrival stacks in the quay area by a tower crane. The identities of the plates are not known before they are manually measured and an identification code is painted on them. When a plate is identified and registered it is moved to an appropriate stack in the storage by one of the gantry cranes. The company policy concerning arrival of new plates to the storage says that plates should be delivered 18 working days before they are requested in production. In reality they are not for various reasons:

- Delays occur because of production and transportation problems.
- Big bulks of plates arrive early because of cheap deals are possible when buying large quantities of plates.
- Special types of plates are only delivered once a year.

2.5 Exit-belt

Each day a set of plates must leave the storage to be processed. The size of the daily set varies from 70 to 150 plates, but to smooth out the production about 100-120 plates are to leave the storage every day. The production planner chooses the set of plates on the basis of needs of plates in the production and how the plates are placed in the stacks. The plates are put on a conveyer belt called the exit-belt, which has a capacity of 8 plates. The plates can be dropped at any vacant slot on the exit-belt from (1,1) to (8,1). The exit-belt is the start of a production line. On the production line several processes are performed. Plates are rolled, sandblasted, painted and finally bar-coded. The exit-belt works as a queue and plates are drawn from the end of the queue. A plate can be drawn from the queue when one of the plate-rollers following the exit-belt is idle. Constraints on the rollers and following processes determine the time interval between plates being drawn. The constraints refer to the dimensions of the plates and required time to adjust the machines according to the processed plates. The time interval between plates being drawn is in other words depending on the order and dimensions of the plates on the belt.

2.6 Operation of the Plate Storage

As mentioned earlier ships are constructed by cutting up plates and afterwards welding them together to produce blocks. Finally the blocks are welded together in the dock to build the ship. Currently the stacks in the plate storage are assigned to different blocks, and plates are placed in stacks according to the blocks for which they will be requested. As indicated in figure 3 on the next page the two stacks (27,7) and (27,8) are used for block 705. A block is produced over several weeks, which means that the cranes often have to get relatively few plates from particular block stacks. Often the requested plate is not on top of the stack and a lot of unproductive movements are required to find the plate. The plates requested for the production of the next day are put in the sort stacks near the exit-belt. The next day plates due for production can be put on the exit-belt with little delay. Plates arriving a long time before their due date are put in stacks in the waiting area. Finally standard and surplus stacks are used for identical special purpose plates and plates with no assigned purpose. We call this layout the *Block-storage*. The common opinion is that the storage layout delivers a poor performance. Therefore we introduce two alternative layouts, which have been tested and compared to the performance of the Block-storage described in Hansen and Kristensen [9]. The principles of the two alternative layouts are briefly described in the following.

The *Due-date storage* organizes the storage in three zones as shown in figure 4 on page 9. All stacks have been assigned a specific due date interval given by a start and end date. A Plate is now supposed to be located in stacks where the due date of the plate is in the interval of the stacks. The intervals can be set by the user, but are usually increasing from zone 1 to 3. Zone 1 is closest to the exit-belt and contains plates that have a due date close to the current day, which means that plates due today can be put on the exit-belt with little delay. In zone 1 each stack has an interval of 1 day. Zone 2 consists of plates with a due date from a few days up to some weeks into the future. In zone 2 each stack has an interval around 2-5 days. Stacks in Zone 3 have the longest interval and most of the new plates arriving at the quay are moved

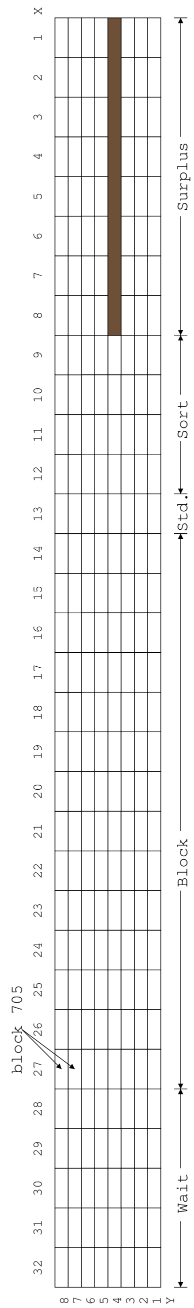

Figure 3: Illustration of the block storage layout.

into this zone. In figure 4 on the next page an example of a due date layout is given. The numbers in the fields indicate the due date from today. The 8 stacks (9,8) to (16,8) have the plates due today and stacks (25,1) to (32,1) for example have the plates with a due date in the interval 21 to 24. We will later discuss how these due dates are set up and updated during simulation.

The *Self-regulating storage* uses no stacks which are dedicated to a specific purpose or due date. The storage is organized by use of *objective-functions*. The objective-functions are designed to place plates on stacks where unproductive moves are less likely to be introduced in the future. In other words, the ideal situation is to place plates on stacks where all other plates have a due date later than or equal to the given plate. Off course this is not always possible and may result in long inefficient travelling distances by the cranes. The optimal solution is found by minimizing costs.

3 Related Research

This work is an extension of earlier work of Stidsen et. al. [17] where a simplified problem was solved with only one crane. Hansen and Clausen [7] describe the results of an earlier version of the planning procedure also part of Hansen and Kristensen [9].

Much work has been done on similar or related problems within scheduling and routing. We will distinguish between off-line problems where all information is available before the planning starts and online where decisions have to be done before all information is available. Available information can be deterministic or stochastic. Real-time online problems are problems where the decisions must be computed within tight time bounds. How tight the time bounds are depend on the specific application. Grötschel et.al. [6] gives a good introduction to the subject specifically on evaluation of online algorithms. Often online algorithms are referred to as dynamic and off-line as static. For some problems the achieved plan are to be executed in a cyclic fashion. Everything must then be deterministic.

A similar problem to the problem we are considering is the well known *travelling salesman problem* (TSP) where a number of customers must be visited exactly once minimizing the travelled distance by the salesman. A variant of this is *the stacker crane problem* first introduced by Fredrickson et. al. [3] where a crane must move given items from their origin to some given destination. With multiple salesmen or cranes we get *the multiple travelling*

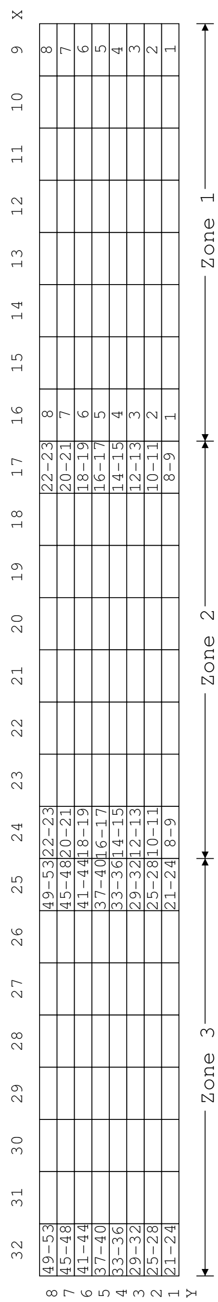

Figure 4: Illustration of the due-date storage layout.

salesman problem or *the vehicle routing problem* (VRP). In our case we have additional precedence constraints representing the order in which plates can be lifted from the stacks. Much research has been done on these problems and recently more attention has been put on the dynamic versions as in Larsen [11].

Avoiding conflicts between cranes or other machinery is often an issue in production and logistic systems. Lee et. al. [12] gives a state-of-the-art overview on deterministic scheduling including robotic cells, automated guided vehicles (AGVs) and cyclic hoist scheduling. AGVs are computer-controlled vehicles without drivers and can hence be considered as a generalization of the above routing problems where congestion, conflicts and deadlocks must be avoided. Refer to Qiu and Hsu [14] for an in depth survey on AGVs.

In the *hoist scheduling problem* one or several hoists are to perform movements of items with associated time windows and precedence relations. Bloch et.al. [2] gives an overview of work done on this and related problems. Of specific interest to our case are Lamothe et.al. [10] that solves a real time multi-hoist problem.

Some of the earliest work we have found on similar "real life" applications are Fujita et. al. [4]. They are scheduling multiple overhead cranes moving around coils. They have implemented a simulator and a very simple control system. Another interesting area are container terminals where combinations of cranes and AGVs are extensively used. Alicke [1] uses constraint programming for solving the scheduling problem at a container terminal. Steenken et.al. [16] solves a problem composed of planning stowage of containers on ship and transport of the containers to the quay by AGVs. The quality of the stowage determines how many unproductive container movements will be necessary in the future to unload the ship. Gantry cranes stacking containers at terminals also resembles our planning problem, even though fewer containers are stacked than steel plates.

4 Model

A Java program has been developed to test the control and planning systems in different scenarios. To perform these tests the program includes an object-oriented model of the plant to be used in simulations. The model is parameterized such that different scenarios can be tested. All data in the model are generated by methods based on specifications and on statistical information. Process times are varied by probability distributions. Furthermore the due date of the plates are perturbed by methods based on rules that emulate the phenomena in the storage. These changes are caused by the dynamical adjustment of the production plans. The simulation model and is also extensively described in Range and Yde [15] who have participated in the development of the simulator. The simulation model is a discrete event simulation model as described in Pidd [13].

4.1 Facilities

The model consists of the following facilities: Two cranes, 8 stacks for new plates delivered at the quay, exit-belt and 191 stacks.

4.1.1 Stacks of plates

The number of stacks is reduced to 191 (8×24-exitbelt). The stacks excluded are in columns 1-8 containing plates that are not assigned to a specific block; therefore these stacks are not taken into account when simulating. 8 additional stacks are dedicated to contain new plates delivered to the shipyard.

4.1.2 Gantry cranes

The acceleration/deceleration time of the cranes is not taken into account. The lift/drop time is set to 15 seconds but varied by an exponential distribution. This variation emulates the variations that occur when lifting a plate. As described in section 2, it sometimes occur that two plates are lifted at the same time because the magnetism "runs through" the upper plate. This is one of the reasons for the variation in the lift/drop time.

4.1.3 Exit-belt

The exit-belt is modelled with a maximum capacity of 8 plates. The average time interval between two plates being removed from the queue is set to 192 seconds. This was found from statistical data. The interval time is varied by an exponential distribution. This variation is emulates the variations that occur on the production line.

4.2 Plate generation

The plates in the model are generated by the program instead of retrieving real data from the shipyards Production Management System (PMS). The method is a simplified model of the flow of plates through the plate storage, but the characteristics of the real plate storage is remained. The generation method is used for two purposes: When the simulation is started it is used to generate plates initially in the storage, and when the simulation is running it is used for generating delivered plates. In the model the plates only have three attributes, due date, block and ID. The plate generator method uses 4 parameters, which are set by the user of the system. We will first describe the generation of delivered plates and afterwards generation of the plates initially on the storage.

4.2.1 Generation of new plates

The new plates are generated before the start of a new day of the simulation. The plates are placed in the arrival stacks. This is partly a simplification of the real life situation described in section 2.4. When the plates are identified and marked we assume that they are moved to designated arrival stacks and the decision where to move the plates into the storage is planned for the next day and not done online the same day.

We have chosen what could be called a *triangular distribution* of arriving plates. In figure 5 is shown an example for the parameter value $SRP = 3$. SRP are the number of new plates with the median due date of the arriving plates. By construction the number of arriving plates is exactly SRP^2. This explains the acronym SRP as **S**quare**R**ootof**P**lates.

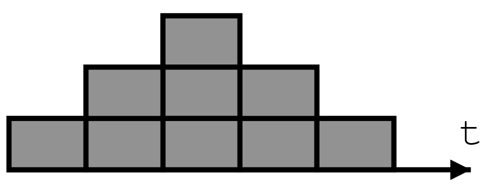

Figure 5: Distribution of plate due dates for $SRP = 3$

The number of days between two plate deliveries is determined by the parameter $DBSA$ (**D**ays**B**etween**S**hip**A**rrival). In case the number of days between two deliveries of new plates is set to $DBSA$ days, then the number of new plates is $SRP^2 \times DBSA$, but only every $DBSA$th day. The white rectangles in figure 6 represent the plates already on the storage and we will later explain the parameters BD and ACC. The leftmost column are the plates due today and for each column going right the due date is one day into the future. The grey rectangles represent arriving plates with $DBSA = 1$. In figure 7 on the following page the change in generation of plates are shown when $DBSA = 3$. New due dates are introduced on the storage every time new plates are generated. The number of new due dates is equal to the value of $DBSA$.

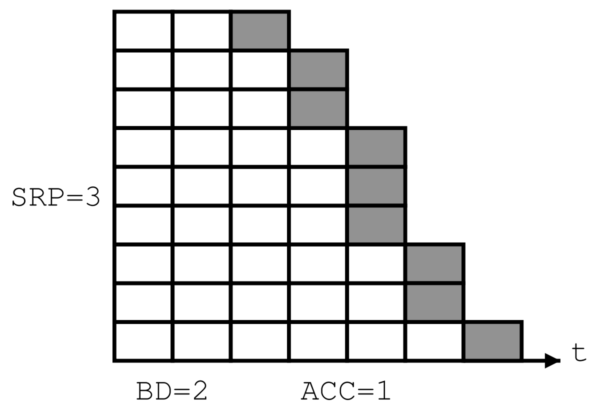

Figure 6: Example of generation of plates with $DBSA = 1$.

4.2.2 Generation of initial plates

The user of the system can control the size of the problem to be solved, using the parameters mentioned above and in the following. The user can hence test different scenarios by creating different due date profiles of the plates.

In our model, the number of plates removed from the storage each day is constant. This is justified by the fact, that the production planner selects a more or less constant number each day in order to avoid large fluctuations in the production level. The number of plates is specified by the variable SRP. The square of this parameter determines the number of plates removed from the storage. For instance if the value of the parameter is set to 3 then the number of plates removed will be 9 as in figure 6. In this way the number

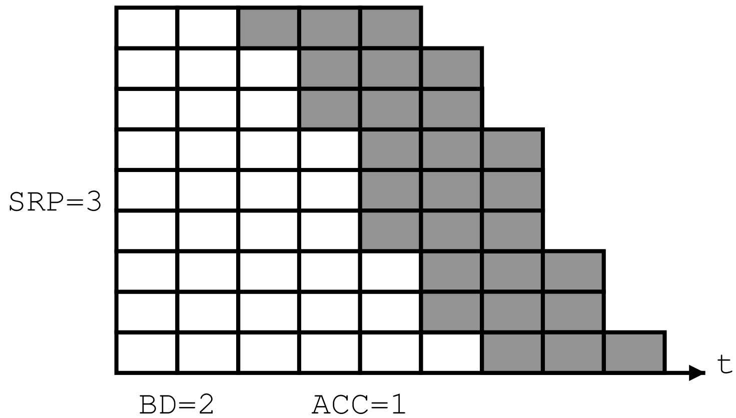

Figure 7: Example of generation of plates with $DBSA = 3$.

of outgoing plates will match the number of incoming plates. The number of plates removed can then assume the following values: 1, 2, 4, 9, 16 etc.

As described in section 2.4 the company policy regarding the storage says that plates should be delivered 18 working days before they are required in production. The length of this period can be controlled by the parameter BD (**B**uffer**D**ays). In figure 7 for instance all plates have arrived for the next two days, $BD = 2$.

The time beyond the BD days is the period where plates are arriving to the storage – or what we have called *accumulating*. This period can be adjusted by another parameter, ACC (**ACC**umulating). Increasing the parameter increase the period of days where plates arrive to the storage. Two examples are shown in figure 6 with $ACC = 1$ and figure 8 on the next page with $ACC = 3$. Again the grey boxes are arriving plates. With $ACC = 3$ the arrival of plates are over a longer period of days resulting in more plates on the storage. In figure 8 we see that the due dates of the arrival plates are exactly 3 days apart.

The number of plates in the initial storage is $((SRP-1)ACC+BD)SRP^2$.

4.3 Storage Types

As described in section 2.6 three different storage-types are considered: Self-regulating, block and due- date. The selected storage type can be set by a

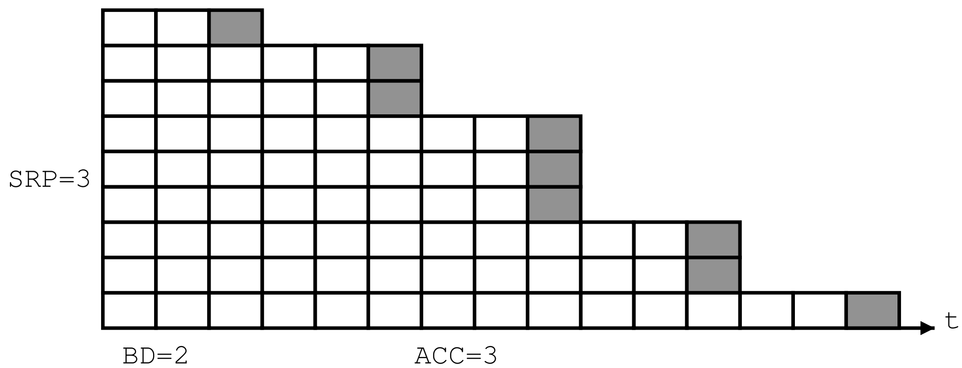

Figure 8: Example of generation of plates with $ACC = 3$.

parameter.

4.3.1 Initialization of the storage

An available parameter defines how the plates are placed initially on the storage. Selecting Block or due-date storage places the plates as expected in the stacks corresponding to the characteristics of the plates. If more than one stack corresponds to the characteristics of the plate, the particular stack is chosen randomly. For the block storage, a plate fitting the sort stack due-date is placed on such a stack. Otherwise it is placed in a random stack according to the block number.

When choosing the self-regulating principle the plates are placed randomly on the stacks. The stack- height is checked as well as any upper limit on the number of plates in a stack.

4.3.2 Self-regulating storage

Defining the storage layout or organization of the self-regulating principle is straightforward: All stacks are identical and not dedicated to specific purposes or dates. The user can only adjust the maximum stack height and the maximum number of plates in the stacks. The storage is organized by use of objective functions that express how well the storage is organized. These functions are discussed in detail in Hansen and Kristensen [9].

4.3.3 Due-date storage

The due-date storage is divided into three zones as seen in figure 9. Each stack, s, has a dedicated due-date interval, $I_s = [b_s, e_s]$, where $b_s \leq e_s$. When only one value is depicted in the figure, $b_s = e_s$. In the ideal situation the stack only contains plates with a due-date $d_p \in I_s$. Generally the size of the interval varies from one and up to many days depending on the zone.

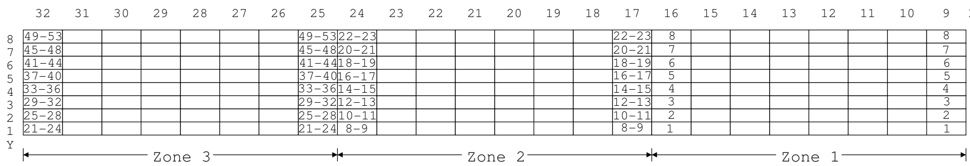

Figure 9: Setting the due-date storage layout.

The due-date intervals for the stacks in the storage can be controlled by 3 parameters for each zone, $z \in Zones$:

Number of days in each interval defines how many due dates are included in each stack. The interval for the stacks in zone 1 is typically 1. The interval for zone 2 is chosen to be between 2 - 5 days and for zone 3 the interval is typically 4 - 20 days. In figure 9 the chosen values are 1, 2 and 4 for zones 1, 2 and 3. Let $Days_z$ be the associated parameter.

Number of stacks per interval specifies the number of stacks included in each interval in the zone. For example if the number of plates requested in production each day is 100 and the number of stacks per interval in zone 1 is equal to 5 and the number of days in each interval is 1, then the number of plates in each of the 5 stacks will be 20 on average. In figure 9 the chosen values are 8 stacks per interval for all 3 zones. Let $Stacks_z$ be the mentioned parameter.

Number of due-date intervals specifies the number of different due-date intervals in each zone. Consider the case of zone 2 where the number of days per interval could be set to 2 days and the number of intervals is set to 5, then 10 due dates are included in zone 2. Typical values are 8 since the storage has 8 rows, which is also the choice in figure 9. Let $Intervals_z$ be the mentioned parameter.

Overlap between zones are introduced in order to avoid large fluctuations in plate movements from day to day when plates are moved between zones. Overlap between zone 1 and 2 defines how many due-dates are included in both zone 1 and zone 2 and similarly for zone 2 and 3. In figure 9 we have intervals corresponding to days 1 up to 8 in zone 1 and an interval in zone 2 with the days 8 and 9. Plates with due date 8 can hence be moved from zone 2 to 1, but plates with due date 9 cannot. Let $Overlap_z$ be the overlap between zones z and $z+1$.

The optimal setting of the parameters is depending on the number of plates on the storage and the distribution of due-dates, since generally the state of the storage is better if the plates are distributed as evenly as possible on the stacks.

When assigning due-date intervals to the stacks this is done with the procedure shown in algorithm 1 on the next page. First the stacks are placed in a list and sorted primarily in the horizontal direction (X) and secondarily in the vertical direction (Y). Next, in a nested loop is iterated over zones, number of intervals per zone and number of stacks per interval. Let us consider zone 1. When assigning due dates to stacks for an interval we pick the first stack in the list starting with stack $(1,1)$ and assign due dates to stacks according to the number of stacks in the interval. In figure 9 8 stacks are assigned to each interval, which results in the stack $(1,1),(2,1),...,(8,1)$ being assigned the due date 1. A stack is removed from the list when assigned a due date. For the next interval we again pick the first stack in the list, which now is $(1,2)$, etc. until the last interval in zone 1 starting at $(1,8)$. After zone follows the same procedure for zone 2 and 3.

If the assignment of stacks were performed in the opposite direction from $(1,1)$, to $(1,2)$ up to $(1,8)$ instead, it would be difficult for the cranes to work together to empty and fill up stacks in the same interval. The distance to the exit-belt from stacks in zone 1, between zones and from the quay and to zone 3 will vary more with this layout as well.

4.3.4 Updating the due-date storage

Since the due date of stacks reflect time, due dates for stacks change over time. Each day we need to update the due dates of several stacks. Plates due today is being removed from their stacks in zone 1. These stacks will then get a new due-date interval equal to the stacks with the latest due-date in zone 1

Algorithm 1 Assignment of due dates

```
b = today, e = today.
Put stacks in a list, L.
Sort in increasing distance from the exit-belt in the Y direction.
Sort in increasing distance from the exit-belt in the X direction.
for all z ∈ Zones do
  for i = 1 to Intervals_z do
    Remove first stack s from L.
    e = e + Days_z − 1.
    b_s = b, e_s = e.
    x = x_s, y = y_s.
    for j = 1 to Stacks_z − 1 do
      x = x + 1.
      if ∃s′ ∈ L where x_s′ = x and y_s′ = y then
        Remove s′ from L.
      else
        Remove first stack s′ from L.
      end if
      x = x_s′, y = y_s′.
      b_s = b, e_s = e.
    end for
    b = e + 1, e = b.
  end for
  b = b − Overlap_z, e = b.
end for
```

plus the number of days per interval in zone 1.

Plates in zone 2 with a due date in that particular interval can now be moved from zone 2 to 1. The same holds for a given interval in zone 2. If all due dates in an interval in zone 2 are present in zone 1 then the due date interval is updated as well. Plates can now be moved from zone 3 in the same way.

For example, in figure 9 plates with due date 1 will be removed from the storage and since the latest due date in zone 1 is 8, these stacks will get the new due date 9. It is now possible to move plates with due-dates 8 and 9 from zone 2 to 1. Stacks in zone 2 with due dates 8 to 9 will hence get the new interval 24-25. In zone 3 stacks with interval 21-24 can now get the new interval 54-58.

The update of due dates is done automatically. The user only specifies the 3 parameters for each zone and the two overlap parameters discussed earlier.

4.3.5 Block storage

In the model of the block storage we ignore waiting stacks and only have block stacks and sort stacks. The number of sort stacks is defined in the same way as zone 1 for the due-date storage. The rest of the stacks are used for block stacks. It is assumed that each block is stored in exactly two stacks, but this could easily be generalized. When generating plates a random block is assigned to each plate. Assigning blocks to stacks are done first in the Y-direction and secondly in the X-direction for example like the following (ignoring sort stacks): Block 1 is $(1, 1)$ and $(1, 2)$, block 2 is $(1, 3)$ and $(1, 5)$ since $(1, 4)$ is the exit-belt. Block 3 is $(1, 6)$ and $(1, 7)$, block 4 is $(1, 8)$ and $(2, 8)$, block 5 is $(2, 7)$ and $(2, 6)$, etc. Note that this is a simplification of the current operation where more than two stacks can be assigned to blocks with many plates. In other cases blocks with few plates are pooled together sharing stacks. We have instead chosen to distribute the necessary number of plates and hence stacks evenly between block.

4.4 Generating Movements

All movements of plates are generated before the simulation of the current day is started. Some movements are generated to deliver the plates ordered by the production for the current day while others are generated to improve the organization of the storage. A strong characteristic of the system is that all plates to move can be identified before the execution starts. Note however that a plate may be moved more than once if it is put on a stack from which plates are going to be moved later on the same day. This will dynamically introduce extra movements to schedule. A movement, m, is defined by the plate to be moved, p_m, the source stack, s_m and the destination stack, d_m. The movement types are defined as follows.

Exit and Dig-up movements

An exit movement is a movement of a plate, which has a due-date that matches the current day. In real life the user of the system selects plates to be removed from the storage. For example in figure 10, today is 1 November 01 and exit-movements of the plates p_2 and p_7 are therefore generated.

A dig-up movement is a movement of a plate, that has to be moved because an exit movement is generated for a plate below it in the stack, or that has to be emptied the current day because the due date interval of the stack is changed. Again in figure 10, a dig-up movement of plate p_1 is necessary since p_2 is going to be moved.

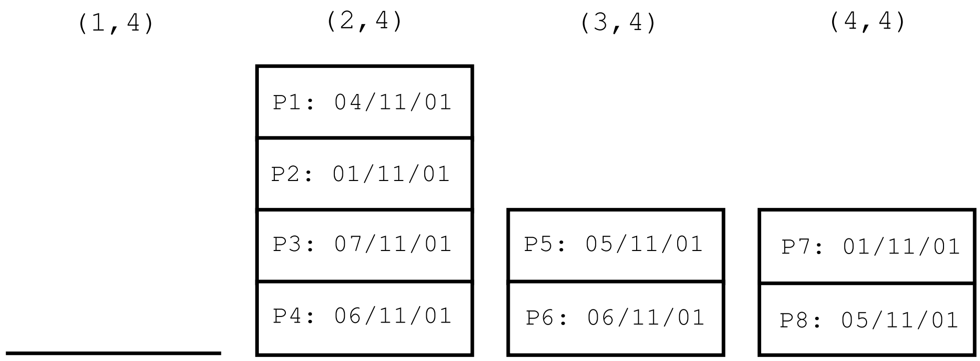

Figure 10: Example of generating movements

Sort movement

In order to improve the state of the storage some sorting of the stacks are performed. The reason for doing sorting is that in the future less dig-ups will be necessary, and less time will be spent removing plates from the storage. However if the due-date of the plates in the storage change frequently there is a risk of doing a lot of sorting that later turn out to be wasted.

The chosen sorting strategy is depending on the storage principle. For the block-principle we identify plates that are due the next day. These plates are moved to special due-date stacks near the exit-belt.

For the due-date principle sort movements can be split into three main categories:

1. A movement of a plate in one of the stacks that contain new plates delivered to the quay area.

2. Movements of plates that are immediately accessible and are located in a stack where the due date of the stack does not include the due date of the plate.

3. if a given percentage, e.g. 75%, of the plates in a stack do not have a due date that match the due date of the stack, then a sort movement for each of the plates in the stack is generated. To avoid generating too many movements the fraction should be set close or equal to 100%.

Optionally the user can choose to extend the sorting in zone 2 and 3: Plates that are to leave the storage within b days from today are to be moved. b is usually set to 1. The above strategy seems a bit complicated, but basically makes sure that plates are moved from zone 3 to 2 to 1 when the due date of the stacks changes. The rest is just trying to move plates as soon as possible before the due date changes if it does not create much extra work.

For the self-regulating principle experiments show that sorting in the following way is appropriate: Move plates away from stacks such that plates that are due tomorrow are on top of the stacks. In that way no dig-ups will be necessary the day after, if the due dates have not changed. Note that the plates that are due tomorrow are not collected in stacks near the exit-belt as for the block storage. This should in principle save one movement per plate.

4.5 Disturbances

A lot of decisions are made on the basis of expectation, but often these expectations are not fulfilled and may lead to poor or even infeasible decisions. This is also the case at the plate storage. The processes performed at the plant are somewhat stochastic and as mentioned above these stochastic disturbances are included in the model. We have earlier discussed disturbances concerning the time of the lift/drop process of the cranes and the interval between two plates being removed from the exit-belt. Other stochastic phenomena concerning the due dates of the plates are included in the model as well. As mentioned in section 2.2 the due date on the plates are changed during their stay at the storage. The due dates are changed because of adjustments to the production plan. The adjustments are caused by different phenomena. In the simulation model three of these phenomena are emulated. In the following the disturbances on the facilities and due dates are described.

4.5.1 Disturbances on facilities

The processes performed by the two cranes and the exit-belt are influenced by stochastics. The process times for both processes are modelled by an exponential function. The process times can never drop below a certain level, i.e. they have a minimum process time. Theoretically the process times have no maximum.

Cranes

As described earlier, lift and drop times vary because of difficulties in lifting the plates. The process time for a lift/drop has a lower limit. From statistical data a mean value of the time has been established. The time used for the simulations is 15 seconds. The process time has a lower limit of 5 seconds.

Exit-belt

The bottleneck on the production line sets the speed by which plates can be removed from the queue at the exit-belt. The bottleneck on the line is the painting station. From statistical data a mean value of the time has been established. The time used for the simulations is 192 seconds. The process time has a lower limit, which is found to be 64 seconds.

4.5.2 Disturbances on due dates

The ideal situation would be a situation where the plates are given only one due-date through their "lifetime". But as explained earlier this is not the case at the plate storage. Plates can have several due-dates during their stay at the storage. Sometimes the change is only a single day, but can also be several days or weeks. Some of the changes are performed to smoothen out production workload. Other changes are performed because plates have been used for other purposes than planned, i.e. they have been used for substitution of other plates. In the simulation model three methods are implemented to change the due date of the plates, they are described in the following. The user of the system can control the methods by parameters.

Rush jobs

In the model plates that have a due date that matches the current day, are supposed to be delivered at the exit-belt the current day. In the real system an employee prepares the movement-list for each day. Not only does he select the plates with a due date matching the current day, furthermore he selects plates, which are supposed to be delivered on other days. He determines the need for plates at the machines where the first operation will be performed on the plates. This behaviour should actually result in a change of the due date on the plates. This change in due dates is not done because it does not affect the real plant, but to emulate this behaviour in the model a random number of plates swap due dates. The swaps are done before the workday starts.

The method uses two sets of plates. The first set contains plates with due date matching the current day and the second set contains plates with a due date later than the current day, but not later than a date specified by the user. In the method the changing is done by swapping the due date of a plate from the first set with a plate from the second set. Both plates are randomly chosen from both sets. Stacks that before the swap were well sorted can afterwards incur additional dig-up or sort movements. An example of the swap procedure can be seen in figure 11 where the grey boxes indicate plates with swapped due-dates.

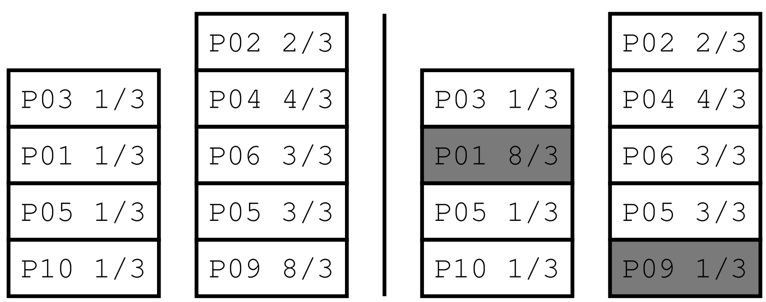

Figure 11: Two stacks of plates before and after swapping due-dates.

In the example the current date is 1/3. Plate $P01$ is swapped with plate $P09$. Before the swap 4 dig-ups were going to be performed to deliver the 4 plates with due-date 1/3. After the swap this number is increased to 9 moves. As can be seen from this example the swapping of plates is a very “effective” way of increasing the complexity of the problem to solve.

An extension of the above rush jobs has been included in the system, which also allows plates due the day after to be swapped.

Weekly change of due-dates

Each week a re-planning is run on the company's production management system, resulting in major changes in the due dates of the plates. Blocks being hurried or postponed cause the changes. When blocks are postponed other blocks are hurried and the other way around because of the limited capacity in the production. The due-date is changed on approximately 500 plates. This means that the sorting of the plates in the stacks will be spoiled and on the following days a lot of dig-up movements will be necessary. This phenomenon is also emulated in the model.

The number of days between the rescheduling is run on the shipyards production management system can be set by the user. Usually the parameter is set to 5, which is equivalent to once a week.

A random number of plates between a lower and upper bound change due-date by the following procedure, which is essentially a cyclic shift (cf. figure 12 on the following page):

- A randomly chosen plate, p_1, is chosen from plates with a due-date in the interval from some minimum change horizon from the current day to some maximum change horizon from the current day. From statistical data these values have been set to 10 and 20 days from the current day.
- The value corresponding to the change in due-date are picked from a normal distribution with mean and variance specified by the user. After the change the due-date, d^{new}, must still be in the above due-date interval. The spread was found to be 6.4 days from statistical data.
- A randomly chosen plate p_2 with due-date d^{new} is selected. p_2 now gets a due-date one day closer the original due-date of plate p_1. A new plate is randomly found with that due-date, etc. This procedure is continued until a plate is assigned the original due-date of plate p_1.
- The above procedure is repeated until a random number of changed due-dates has been reached.

In figure 12 on the next page is shown an example of the above procedure. Plate p_1 changes due-date from 1/3 to 4/3. A plate p_2 is chosen with due-date 4/3 which is then changed to 3/3, a plate p_3 is chosen with due-date 3/3, which is changed to 2/3 and finally the due-date of p_4 is changed to the original due-date of p_1.

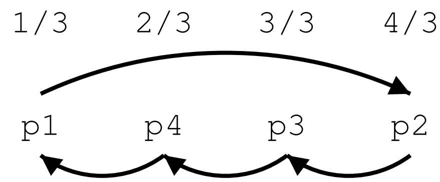

Figure 12: Example of weekly change in due-dates.

The changes of the due dates described above are done before the start of a new day in the simulation. Therefore these changes can be included in the optimized schedule. The disturbances on the facilities are obviously generated runtime and can therefore not be included in the optimization of the schedule.

5 Conclusion

In this paper we have 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 has been developed including both the physical entities of the system and the planning and processing aspects. Specially different storage principles have been introduced as well as different ways of designing the layout of the storage.

Methods for solving the problem are described in Hansen and Kristensen [9] and the validity of the model is shown by simulation experiments reported in both Hansen and Clausen [7] and Hansen and Kristensen [8].

References

[1] K. Alicke, *Modeling and optimization of the intermodal terminal mega hub*, OR Spectrum, 24 (2002), pp. 1–17.

[2] C. Bloch, A. Bachelu, C. Varnier, and P. Baptiste, *Hoist scheduling problem: State-of-the-art*, in IFAC Intelligent Manufacturing Systems, 1997, pp. 127–133.

[3] G. N. Fredrickson, M. S. Hecht, and C. E. Kim, *Approximation algorithms for some routing problems*, SIAMM Journal of Computing, 7 (1978), pp. 178–193.

[4] F. Fujita, M. Okada, and T. Kaita, *Automatic scheduling system for an overhead multi-crane yard*, in IFAC Control Science and Technology 8th Triennial World Congress, 1981.

[5] M. Grötschel, S. O. Krumke, and J. Rambau, eds., *Online Optimization of Large Scale Systems*, Springer Verlag, 2001.

[6] M. Grötschel, S. O. Krumke, J. Rambau, T. Winter, and U. Zimmermann, *Combinatorial Online Optimization in Real Time*, in Grötschel et al. [5], 2001, pp. 679–704.

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

[8] J. Hansen and T. F. H. Kristensen, *Crane scheduling for a plate storage in a shipyard: Experiments and results*, Informatics and Mathematical Modelling, Technical University of Denmark, IMM-TR-12 (2003).

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

[10] J. Lamothe, C. Thierry, and J. Delmas, *A multihoist model for the real time hoist scheduling problem*, in Symposium on Discrete Events and Manufacturing Systems. CESA'96 IMACS Multiconference. Computational Engineering in Systems Applications, 1996.

[11] A. Larsen, *The Dynamic Vehicle Routing Problem*, PhD thesis, Technical University of Denmark, 2000.

[12] C.-Y. Lee, L. Lei, and M. Pinedo, *Current trends in deterministic scheduling*, Annals of Operations Research, 70 (1997), pp. 1–41.

[13] M. Pidd, *Computer Simulation in Management Science*, John Wiley & Son, 4. ed., 1998.

[14] L. Qiu and W. Hsu, *Scheduling and routing algorithms for AGVs: a survey*, Tech. Rep. CAIS-TR-99-26, "Centre for Advanced Information Systems, School of Applied Science, Nanyang Technological University, Singapore", 1999.

[15] T. M. Range and S. Yde, *Storage management at odense steel shipyard. simulation, product placement and control.*, Master's thesis, University of Southern Denmark, 2002. (In Danish).

[16] D. Steenken, T. Winter, and U. Zimmermann, *Stowage and transport optimization in ship planning*, in Grötschel et al. [5], 2001, pp. 731–745.

[17] T. K. Stidsen, J. Hansen, and J. Clausen, *Large steel plate storage optimization*, in The 1st International EuroConference on Computer Application and Information Technology in the Marine Industries 2000, COMPIT'2000, 2000, pp. 449–462.