Internal Delivery Planning of Materials at the Tile Plant of Struyk Verwo Infra

A Master Thesis by S. Fontijn

Student Industrial Engineering & Management
Production & Logistic Management Track

Master thesis Industrial Engineering and Management

Internal Delivery Planning of Materials

at the Tile Plant of Struyk Verwo Infra

- public version –

S. Fontijn

University of Twente

Examination Committee:

Dr. ir. J.M.J. Schutten (University of Twente)

Dr. P.C. Schuur (University of Twente)

A. Bijl (Struyk Verwo Infra)

A. Kaddouri (Struyk Verwo Infra)

University of Twente
Drienerlolaan 5
7522NB Enschede
The Netherlands

Struyk Verwo Infra – Location Dordrecht
Merwedestraat 40
3313CS Dordrecht
The Netherlands

Management Summary

The tile manufacturing plant of Struyk Verwo Infra in Dordrecht wants to improve the Operation Equipment Effectiveness (OEE) of multiple tile producing machines in order to stay competitive. The OEE is a performance indicator for an individual machine and is based on the ratios: speed, availability, and quality. The indicator shows opportunities in different areas for improving the efficiency of a machine. The reduction of downtime of the machines is a big opportunity for Struyk Verwo Infra (from now on referred to as 'SVI'). A major cause of downtime is waiting for materials at the machines. The goal of this research project is to improve the OEE by minimising the delays caused by waiting for materials. We do this by analysing the current situation at the plant of SVI in Dordrecht. The analysis shows the total downtime caused by waiting for material adds up to 321.7 hours for all machines in the year 2015. This is equivalent to a total cost of € xx,xxx.xx for that year. This shows a large opportunity for improving the OEE.

The waiting for materials at the machines has various causes. One of these causes is the use of simple dispatching rules for the mixing orders. Another cause is that future orders are not considered during the dispatching of mixing orders. This may lead to poor scheduling of the mixing orders and cause orders to wait for each other. A delayed order may cause starvation of material which leads to downtime of the machine. Besides a delayed order, orders should not be delivered too early. An early delivery may cause poor quality of the tiles. Therefore, we measure the performance of a delivery with the weighted earliness and tardiness (lateness) function.

We propose a planning algorithm that automatically generates good schedules for mixing orders. To be able to create good schedules, the algorithm forecasts expected mixing orders within a set planning horizon. By forecasting the mixing orders, they can be considered early in the planning algorithm. The planning algorithm creates a new planning every planning cycle. We performed an experiment to determine the best values for the parameters of the planning cycle. This resulted in a planning horizon of 26.5 minutes, a planning interval of 100 seconds, and a fixed zone of 250 seconds. These settings result in good schedules with a good performance in terms of earliness and tardiness, while computation times stays low.

We estimated that our proposed planning cycle is able to save up to 128.4 hours of waiting time or € xx,xxx.xx compared to the year 2015. However, the current tool for measuring waiting times at SVI makes no distinction between waiting time caused by breakdowns or poor logistics planning. This may result in the actual savings to be lower. Waiting times lower than 60 seconds are not currently not measured by SVI, the addition of these short waiting times may increase the actual savings.

In order to determine the impact of assigning machines to different mixers, we experimented with different possible combination of assignments of machines to the mixers. This resulted in a list of favourable and unfavourable assignments given a production setup of tile heights for each machine. We concluded that assigning two machines with a high demand for the bottom-layer to the same mixer causes a lot of waiting time.

At request of SVI we performed an experiment where we simulated a day where a mixer was broken down. The experiment showed that when the machines would need to get the bottom-layer mixtures from just one mixer, the capacity of the that mixer was not sufficient. This may lead to a lot of waiting time. We experimented with lowering the mixing time for this mixer and concluded that when the mixing time was lowered with only 15

second the total waiting time for material at the machines for this day would be reduced by almost 95.62%. This means a reduction from almost 3 hours to less than 8 minutes

TMS, the software supplier for the mixing process for SVI, is able to implement the planning algorithm into the current software. A few hardware adaptions are required, such as installing laser measurement at each silo. The total investment for implementing the solution is estimated between € xx,xxx.- (best case scenario) and € xx,xxx,- (worst case scenario). As the actual savings cannot be determined because the waiting time caused by breakdowns and planning reasons cannot be distinguished, we calculated the payback period for different scenarios in investment and actual savings. This results in a payback period varying from only 0.8 years (best case scenario and 100% of savings realised) up to 5.02 years (worst case scenario and only 25% of savings realised).

Besides our planning algorithm, we propose the following recommendations:

1. Improve the registration of breakdowns at the machines in order to improve the analysis of breakdowns.
2. Improve the consistency of the moisture levels of the mixtures in order to maintain a consistent quality of the mixtures.
3. Flexibly assigning mixing orders to Mixer 4 and 5 in order to benefit the full capacity of the mixing department and reduce waiting times even further.
4. Add a buffer for plateaus between the curing chamber and the machine in order to reduce the waiting time for plateaus.

We also recommend to further research the generalisation of our solution for other, similar, plants and research the impact of breakdowns on our proposed solution by simulating these breakdowns.

Preface

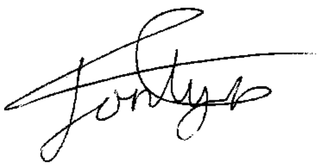
This master thesis is the end result and final work of my life as a student at the University of Twente. This final work was not possible without the help of several persons and organisations. Therefore, I would like to use this preface to give a very big thank you to these persons and organisations that helped me during my research.

First, I would like to thank Struyk Verwo Infra for letting me perform the research and providing all the resources I required during the research. I would especially like to thank Arjen Bijl and Abdelwahed Kaddouri for supervising me and helping me with the research. The sharp and critical look of Arjen towards problems greatly helped me to look deeper into situations. I greatly value the help of Abdelwahed on showing me the different people, plants, and processes of Struyk Verwo Infra. He really helped me by giving a complete look as possible on the company. I would like to thank all other employees of the plant in Dordrecht for their time, help, and input.

Secondly, I would like to thank my supervisors at the University of Twente, Marco Schutten and Peter Schuur. I want to thank Marco for the critical look and great feedback on my thesis and helping me pushing the quality of the thesis to a higher level each time. I want to thank Peter for his great insights and providing a different perspective on the thesis during each feedback session. His unique look on problems helped me in trying different approaches.

I also want to thank Martin Jansens from Canigou Consult who was around at SVI several times a week and always was available for giving his insights in experimental design and change management in organisations. I want to thank Bart de Kluijver from TMS for taking the time to help with the implementation plan and cost calculation of the software.

Lastly, I want to thank my family and friends for their unconditional support during my study and graduation. This was a great time which I will remember for the rest of my life.



S. Fontijn

Rotterdam, June 2016

Table of Contents

Table of Figures

Abbreviations

ATC	Apparent Tardiness Cost
SBP	Shifting Bottleneck Procedure
E/T	Earliness Tardiness
FCFS	First Come First Serve
IIT	Inserted Idle Time
JSSP	Job Shop Scheduling Problem
OEE	Overall Equipment Effectiveness
SVI	Struyk Verwo Infra B.V.
TWET	Total Weighted Earliness Tardiness

Chapter 1: Introduction

This report describes a research conducted at Struyk Verwo Infra B.V. (SVI) to improve the process of material replenishment. In this chapter we introduce the company and the research setup. First, Section 1.1 introduces the company. Next, Section 1.2 describes the research motivation. Section 1.3 briefly discusses the problem at Struyk Verwo Infra B.V. (SVI). Finally, Section 1.4 contains the research objective, research questions, and structure of the report.

1.1 Introduction to Struyk Verwo Infra

SVI is part of CRH Plc., which is the world market leader in the manufacturing of concrete building materials. SVI is specialized in manufacturing concrete public landscaping materials and has several manufacturing plants across the Netherlands. One of them is located in Dordrecht, which is specialized in the manufacturing of washed tiles.

Washed tiles are mostly used as paving material for squares, sidewalks, and roads. The difference between washed tiles and regular tiles is the top-layer of the tiles. The top-layer of washed tiles is made from a material consisting of more expensive aggregates. The concrete between the aggregates in the top-layer is washed away during the production process to create a better looking tile.

The process to create these special type of tiles is different and more complex than creating a regular tile. The variations offered in washed tiles are significantly larger than the variations in regular tiles. The higher variability asks for a more flexible production process than used for regular tiles.

1.2 Research Motivation

Several years ago, CRH launched a program to improve the measurement of the machine effectiveness. For this program, a software package was developed to measure the Overall Equipment Effectiveness (OEE). The OEE is a performance indicator for the effectiveness of machines. The OEE of recent years show that SVI has a lot of room for improvement. Therefore, SVI has initiated several projects in order to improve the OEE of several machines.

One of the problems of SVI is the many downtimes that occur on the machines. These downtimes are caused by many different factors. One of the causes is the starvation of input material for a machine. During operation, a machine constantly requires two types of raw material: a mixture for the top-layer and a mixture for the bottom-layer. If one of these is unavailable, the machine will stop working until it is replenished. The replenishment of the machines is done by autonomous transporters guided by a rail that is fixed to the ceiling. These transporters bring the material from the mixing machines to the production machines. The current procedures for planning the mixing of material and transporting it cause many delays in the production process.

1.3 Problem Description

Currently the transporters are controlled by a system that uses simple dispatching rules such as First Come First Served (FCFS). This system does not anticipate on other aspects such as processing times, or possible current or future scenarios. For example, a waiting transporter may get priority over a second waiting transporter while the second transporter has a higher urgency of delivering the material. As a result, machines can starve on input material until the transporter arrives. The purpose of this thesis is to prevent such situations and to minimize the waiting times for materials, caused by the current control procedures for transportation.

1.4 Research Objective and Methodology

The purpose of this research is to find and test a solution that improves the current situation. Therefore, the research objective of this thesis is as follows:

Minimize the waiting time for raw material at the machines by implementing an improved planning procedure for the mixing and transportation process.

To achieve the objective, we formulated five research questions. We list each of the research questions below with a short description of the methodology on how we answer the research question.

1. What is the current situation for the production process at SVI?

We analyse the production process, material handling, and performance of the machines. First, we analyse the production process to understand the requirements and constraints for the material handling. For the material handling we analyse the processes from the mixing to the transportation of the material to the machines. Finally, we analyse the performance of the current system. We use qualitative interviews with managers and operators, our own analysis of the material flows, and a data analysis. We answer this research question in Chapter 2.

2. What methods can we use from literature to optimise the planning procedure of the mixing and transportation of material and what data do the methods require?

We perform a literature research to find methods for optimising the planning procedure of the mixing and transportation of material at SVI. Ultimately, the methods will result in a solution for minimising the waiting time for material. We answer this research question in Chapter 3.

3. What methods can SVI use to minimise the waiting times for material at the machines?

We use the literature research from research question 2 as input for possible methods SVI can use to minimise waiting times for materials. We also develop our own methods, inspired by the methods from literature, to minimise the waiting times for material. We answer this research question in Chapter 4.

4. What is the best method for SVI, from the methods given in research question 3, based on selected criteria?

We evaluate the methods from research question 3 using experiments for selected scenarios. We base the experiments on historical scenarios, possible future scenarios, and extreme scenarios. The criteria are selected on interviews with managers and our own findings. We select the best method based on these criteria. We answer this research question in Chapter 5.

5. How can our proposed method be implemented at SVI Dordrecht?

We analyse the requirements for implementing the solution at SVI in Dordrecht. We also analyse the required investments and Net Present Value of implementing the solution at SVI. We answer this research question in Chapter 6. Finally, Chapter 7 contains our conclusions, recommendations, and further research.

Chapter 2: Current Situation

In this chapter we analyse the current situation at SVI. First, Section 2.1 describes the production process at the plant of SVI in Dordrecht. Section 2.2 contains the analysis of the ordering of material. Section 2.3 describes the analysis of the mixing process for the raw materials. Section 2.4 discusses the process for the transportation of material. Section 2.5 describes the analysis of the performance of current processes. Finally, Section 2.6 contains our conclusions for the current situation and processes.

2.1 Production Process

The plant of SVI in Dordrecht is specialised in the manufacturing of washed, concrete tiles (see Figure 2.1). The plant is set-up to be very flexible; it is able to manufacture many different types of tiles on the same day. This section describes the production process in more detail.

2.1.1 Product

SVI is specialised in creating many different type of tiles with different sizes, aggregates, and colours. Each tile consists of two layers: a bottom-layer and a top-layer. The bottom-layer is made of the same concrete mixture in every tile, while the top-layer concrete mixture differs in aggregates and colours. This depends on the type of tile that is produced.

Figure 2.1: Tiles After Washing

Figure 2.2: Tiles after Curing with the Retarding Paper on Top

The hardening of the tile is done by a chemical reaction between cement and water called the *curing* process. SVI makes use of three types of cement: blast-furnace cement, white, and Portland. The different types of cement have different colours and curing properties. Blast-furnace cement (Dutch: ‘hoogovencement’) is a light grey cement and Portland is a dark coloured cement. All bottom-layers consists of blast-furnace cement. The cement used for top-layers depends on the colour or curing properties that are required for the mixture.

The curing process of white and Portland cement is faster than that of blast-furnace cement. The quality of mixtures containing blast-furnace cement start declining 30 minutes after mixing, while the quality of mixtures containing white or Portland cement already start declining 15 minutes after mixing. When the quality of a mixture starts declining, the quality of the tiles may become unacceptable.

Most of the tiles at SVI are washed, which means that a small amount of the top-layer is washed away. To be able to wash away the concrete between the aggregates, a retarding paper is used on the top-layer (see Figure 2.2). The retarding paper slows down the curing process on top of the top-layer. After washing the top-layer, the aggregates on top of the tile are revealed (see Figure 2.1).

2.1.2 General Process

The process for producing a tile consists of six main steps, namely: mixing, transportation, pressing, curing, washing, and packaging (see Figure 2.3). At the mixing departing the raw materials are mixed in the right proportions to create a top-layer or bottom-layer mix (Section 2.2.3 describes the mixing process in more detail). When the mix is ready, a transporter takes it to the right machine (Section 2.3 describes the transportation process). At the machine the mixture is stamped into tiles and taken to a curing area for 24 hours. After curing, the tile is washed and packed. Section 2.1.3 describes the machines and production steps in more detail. For this research we focus on the mixing and transportation processes; we describe the other aspects of the process briefly.

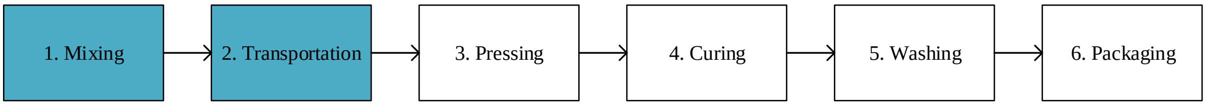


Figure 2.3: Six Main Production Steps

Figure 2.4 shows a sketch of the facility with the mixing department, transportation system, and presses. SVI has four operational presses and one press that is not in use anymore. Each press has two silos: one for the top-layer and one for the bottom-layer. The silos are named with the press number and an indicator for the top-layer 'D' (from the Dutch term 'deklaag') or the bottom-layer 'K' (from the Dutch term 'kernbeton'). For example, the silo for the top-layer of Press 1 is called 'Silo 1D'. The top of Figure 2.4 shows the five mixers (3 for the top-layer and 2 for the bottom-layer). Each mixer has its own transporter that replenishes the silos. The transporter can travel to all presses using multiple rail sections. A carousel rotates the transporters from one rail section to another. First, the transporter travels from the mixer to the carousel using a rail section. At the carousel it turns towards the right rail section for delivering the material to the silo. Next, it travels towards the silo using the rail section. When the transporter is above the silo, it empties the content into the silo. Next, it travels back to the carousel, rotates towards the right rail section, and travels back to the mixer. Press 1 and Press 2 share the same rail section (Rail Section 8) for the delivery of both the bottom-layer and top-layer. Press 3 and Press 4 also share the same rail section (Rail Section 6) for the delivery of both layers. Although Press 4 is not operational anymore, there are plans for replacing Press 3 and 4 with a press with a capacity similar to Press 5. Press 5 has separate rail sections for each layer (Rail Section 9 for the bottom-layer and Rail Section 10 for the top-layer).

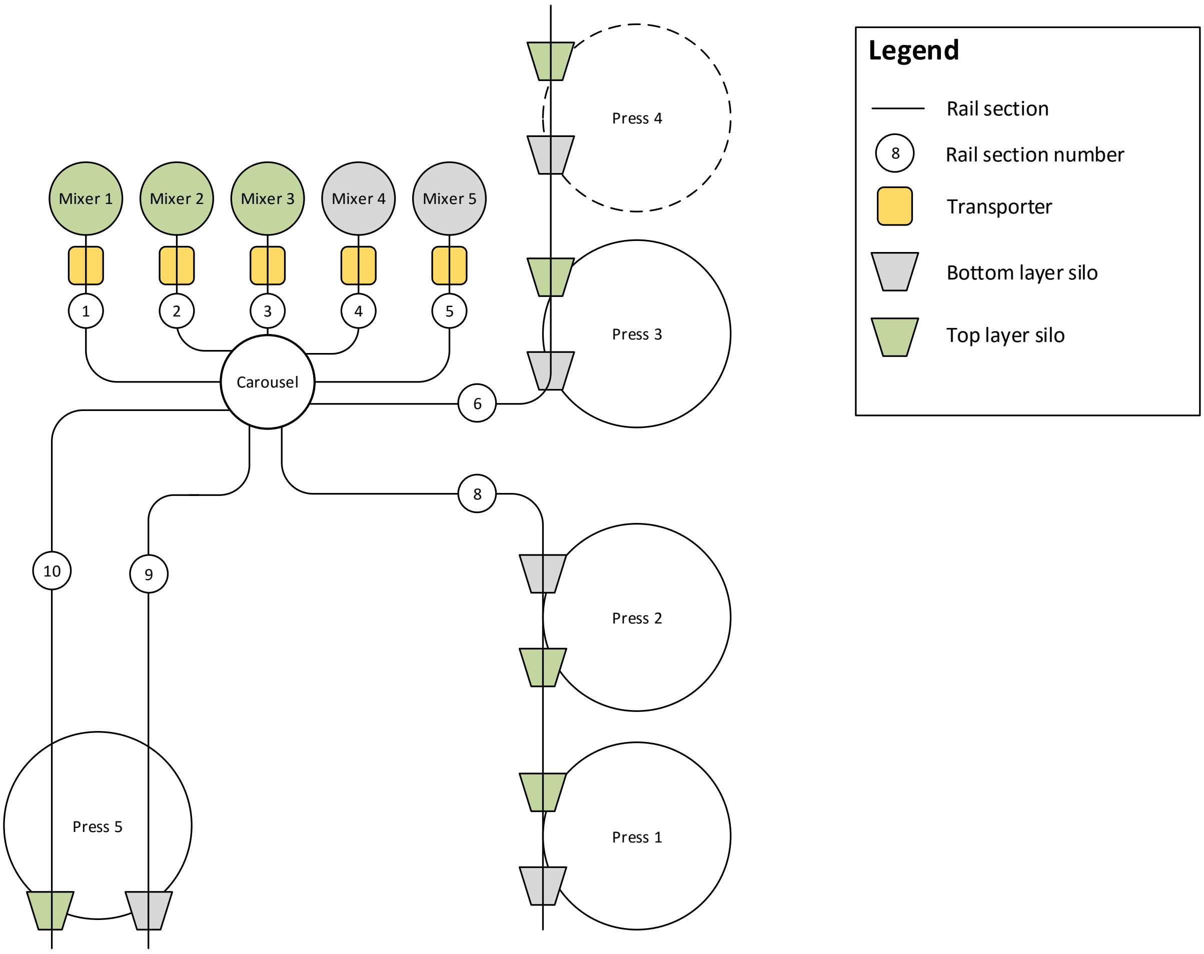


Figure 2.4: Sketch of the Layout of the Facility

2.1.3 Presses

Figure 2.5 shows the layout of a press with the production process in 11 steps. Steps 1-8 of the process are performed at a rotating table of the press (see Figure 2.6). This rotating table consists eight sections. In each section a mould is placed. The table rotates one eighth of a full rotation every 5-10 seconds; the actual duration depends on the type of tile that is produced. After each partial rotation, production steps are performed at each section (see Figure 2.5). (1) First the mould is cleaned. (2) A retarding paper is inserted. This is a sheet of paper that slows the curing (the hardening of concrete) on top of the tile in order to wash away material. (3) The top-layer is put into the mould. (4) The top-layer is stamped. (5) The bottom-layer is put into the mould. (6) The mould is in an empty waiting position. (7) The entire tile is stamped. (8) The tile is taken from the turning table onto a platform that is transported to the curing area. Now the mould is empty and the process can start over from (1). When the tile is taken from the mould and put onto a platform (9), the platform is taken to the curing area (10) when it is full; the curing takes 24 hours. After the curing, the platform is retrieved from the curing area and the tiles are taken from the platform. In the last step (11) the tile, which is now taken from the platform, is turned. By turning the tile, the top-layer comes on top. The retarding paper is then removed and a water jet washes some of the concrete from the top-layer away. After inspection the tile is stacked and packed. Next, the tile is stored outside for 7-28 days for additional curing before it will be shipped to one of the customers, which are mostly municipalities or contractors who work for municipalities.

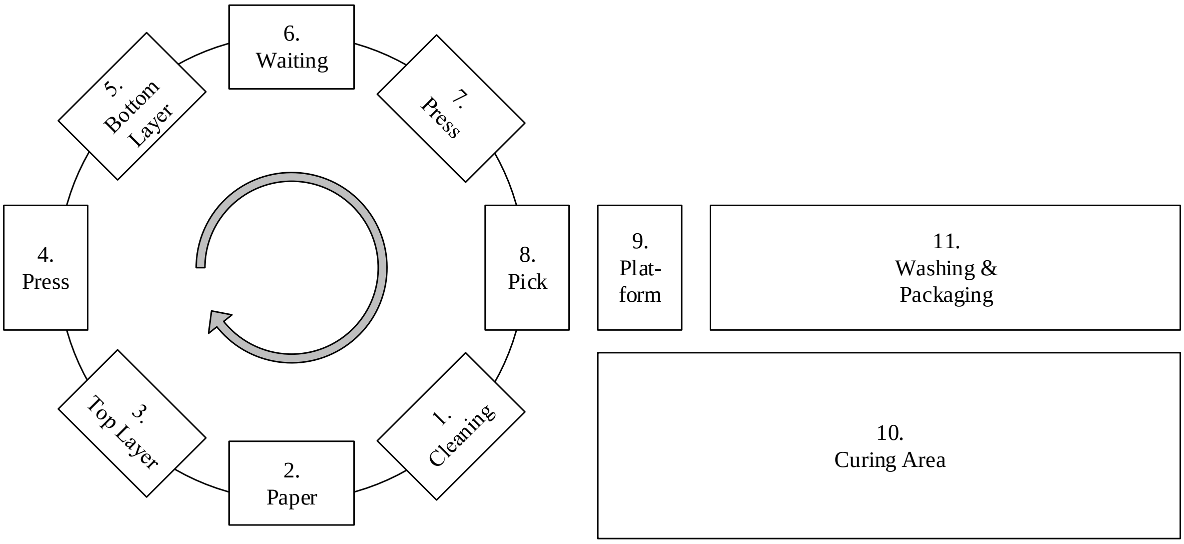


Figure 2.5: Different Steps of the Production Process for a Washed Tile

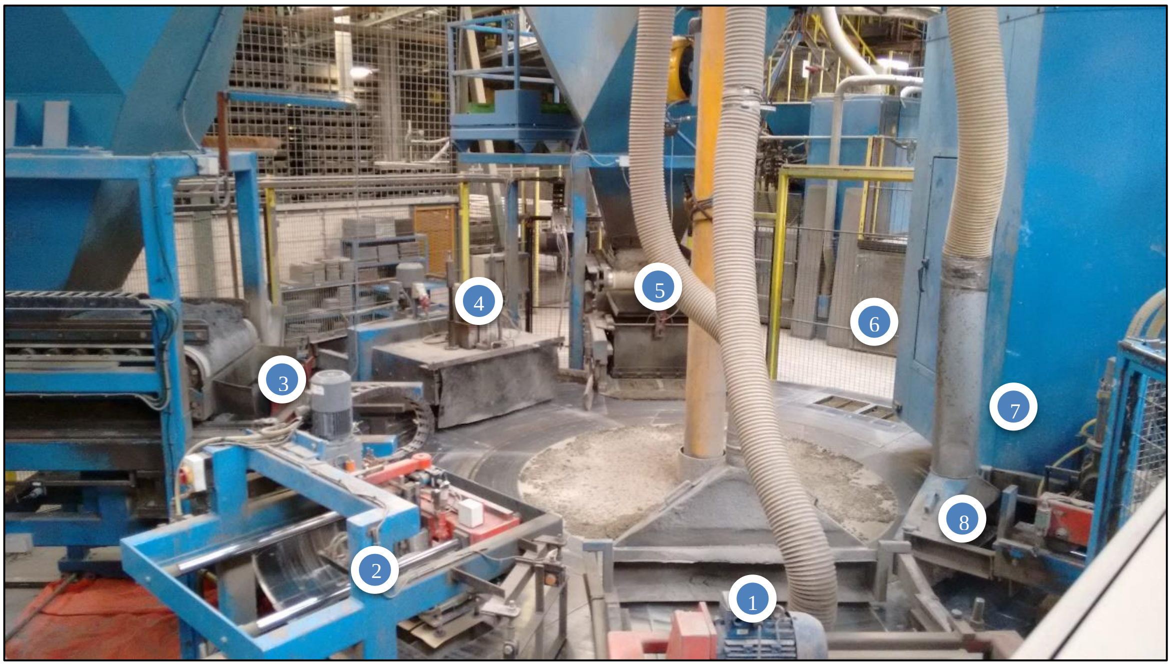


Figure 2.6: Picture of Press 2 with Production Steps 1-8

Table 2.1 shows the height restrictions for tiles each press can produce. Tiles can vary in height from 4.5 to 8 cm. Table 2.2 shows the different dimension restriction for tiles of each press. Presses 1 and 3 always produce tiles of 30x30cm; these are the most common type of tiles. Press 1 is able to produce sizes smaller than 30x30cm with some modifications and Press 3 is only able to produce 30x30cm tiles. Press 2 produces tiles varying in sizes up to 30x30cm. Press 5 produces larger tiles up to 60x40cm. The size and height of the tile have an impact on the material requirements; a larger tile requires more input material.

Table 2.1: Height Restrictions for Tiles of Each Press

Press	4.5 cm	5.0 cm	6.0 cm	7.0 cm	8.0 cm
Press 1	•	•	•	•	•
Press 2	•	•	•	•	•
Press 3		•	•		
Press 5		•	•	•	•

Table 2.2: Dimension Restrictions for Tiles of Each Press (Width x Length in cm)

Press	20x20	15x30	30x30	20x40	20x50	40x40	40x60	50x50
Press 1			•					
Press 2	•	•	•					
Press 3			•					
Press 5				•	•	•	•	•

The production rate differs for each press at SVI. The rotating table of the press consists of multiple moulds. During each rotation the mould is taken to the next production step. All moulds can hold 2 tiles, except for Press 2. The amount of tiles that can be produced in the moulds of Press 2 depends on the size of tiles it produces. If the 20x20cm tile is produced, the mould can hold 6 tiles. If the 15x30cm tile is produced the mould can hold 4 tiles. Finally, if the 30x30 cm tile is produced, the mould can hold 2 tiles (this is similar to the other presses).

Table 2.3 shows the production rate for the different presses. To compare the production rates of Press 2 with the other presses, the production rate is shown for 30x30cm tiles (where the mould also holds 2 tiles). Table 2.3 shows that Press 1-3 have the same production capacity per hour. However, Press 1 has the highest actual average production and Press 3 the lowest. Press 5 has a lower capacity than Press 1-3.

Table 2.3: Press Capacity and Actual Production per Day

Press	Capacity (tiles/hour)	Average Actual Production (tiles/hour)
Press 1	1,200	777
Press 2	1,200	702
Press 3	1,200	534
Press 5	500	350

For the delivery of material to the presses, multiple resources are used. Table 2.4 shows the resource types that are used for creating and transporting mixtures to the press. It shows the type of resource, the number of resources in each category, and a short description of the resource.

Table 2.4: Resources Used For Creating and Transporting Mixtures to the Press

	Resources	Count	Description
1	Raw Material Supply	2	The raw material supply system retrieves the required materials for a mixture from different silos and transports these towards a vertical lift using conveyor belts.
2	Vertical lift	5	The vertical lift carries the materials from floor level towards the mixer one level above the floor. When it reaches the mixer it empties the contents into the mixer.
3	Mixer	5	The mixer creates a concrete mixture from the raw materials. After the vertical lift emptied the contents in the mixer, the mixer adds: cement, water, and a dye colour and mixes these for several minutes.
4	Transporter	5	The transporter carries the mixture from the mixer towards the silo of one of the presses.
5	Rail Section	9	The rail sections are used by the transporters to move towards the presses. The rail section is attached to the ceiling.
6	Carousel	1	The carousel turns the transporter from one rail section towards another.

Table 2.5: Operations for Mixing and Delivering a Mixture

	Operation	Description
1	Raw Material Supply	The required raw materials are delivered by the raw material supply to the vertical lift.
2	Vertical lift	The vertical lift moves from the floor level towards the mixer.
3	Emptying Lift	When the vertical lift arrives at the mixer and the mixer is idle, the vertical lift empties the contents into the mixer. If the mixer is busy, however, the vertical lift waits until the mixer becomes idle.
4	Dry mixing	The mixer adds cement to the raw materials and starts mixing all the contents.
5	Wet mixing	The mixer adds water and continuously mixes all the contents. The curing process starts as soon as the cement reacts with the water.
6	Emptying the mixer	The mixer empties the contents into the transporter as soon as it is finished mixing.
7	Travel towards the carousel	The transporter travels towards the carousel using a rail section.
8	Turn on the carousel	The carousel turns the transporter towards the right rail section.
9	Travel towards the press	The transporter travels towards one of the silos of the press.
10	Empty contents into the silo	The transporter empties the contents into the silo.
11	Travel back to the carousel	The transporter travels back to the carousel.
12	Turn on the carousel	The carousel turns the transporter towards the rail section to the mixer.
13	Travel back to the mixer	The transporter travels back to the mixer.

Table 2.5 shows the operations required for creating the mixture and delivering it to the press. Each order requires these consecutive steps. The type of resources required for these operations is the same for each job. However, the actual resource may differ per job.

2.1.4 Production Setup

All presses at SVI require two employees: one for operating the machine and one for inspecting the tiles. The production at SVI normally takes place from Monday to Friday between 6:00 a.m. to 4:00 p.m. When overtime is required, production may continue after 4:00 p.m. The Saturday may also be used for overtime.

The plant produces many different types of tiles in a week. Tiles are produced in batches of several hundreds to more than 10,000 tiles. When a batch ends, a changeover or cleaning may be required before the next batch can start. If the dimensions of tiles of the new batch are different, a changeover is done. If the colour or material of the tile changes, a cleaning is done. At the end of every day the presses are cleaned as well.

2.2 Ordering & Mixing Process

For the production of a tile, two types of mixtures are required: one for the top-layer and one for the bottom-layer. These mixtures are created at the mixing department. The mixing department has five mixers for creating the mixtures: three for the top-layer and two for the bottom-layer. Section 2.2.1 describes the ordering process of these mixtures. Section 2.2.2 contains the process of the raw material supply. Section 2.2.3 describes the mixing process.

2.2.1 Ordering of Mixtures

When a replenishment is needed, it is requested with an automatic or a manual order. Automatic orders are created by a trigger of the press. Manual orders are created by a trigger of the operator of the press. Each silo has a sensor that is triggered as soon as no material is touching it (see Figure 2.8). The sensor is hanging at a certain level in the silo, however, they are all hung at a high level. An immediate trigger by the sensor for a new order would result in a delivery that is done too early. Therefore, a time delay is set before placing an order after the immediate trigger. The time delay can be adjusted by the operator of the press and mostly varies between 0-10 minutes. When the delay is set too high it can result in late deliveries, which may result in starvation of material. When the delay that is set too low it can result in an overloaded silo and poor quality of the tiles. The poor quality results from the curing process that already starts during the mixing process. When a delivery is too early, it is dropped on top of the mix that is already in the silo and it might take too long before it is processed by the press. Once the mix is finally ready to be processed, it might be too far in the curing process to produce a good quality tile.

Figure 2.7-Figure 2.10 show the process of automatic requests. Figure 2.7 shows an empty silo with the sensor and the rail above the silo, which is used by the transporter. Figure 2.8 shows the situation where the level of material drops below the sensor and triggers a signal for an order. After a set delay an order will be placed in the mixing department. Figure 2.9 shows the situation where the transporter arrives with the requested material. Figure 2.10 shows the situation where the transporter has emptied the material into the silo and the sensor is now touched again by material so that it does not trigger a new order.

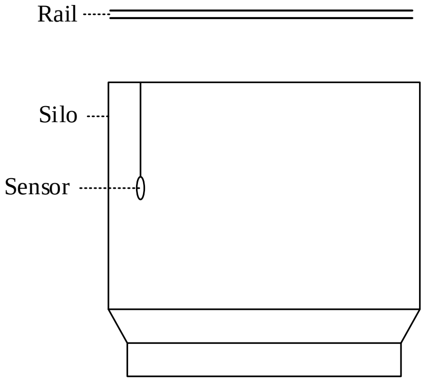


Figure 2.7: Empty Silo

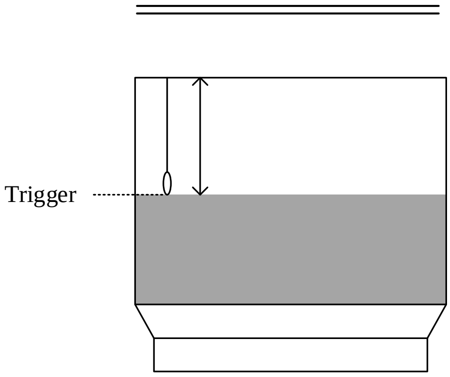


Figure 2.8: Sensor is Triggered

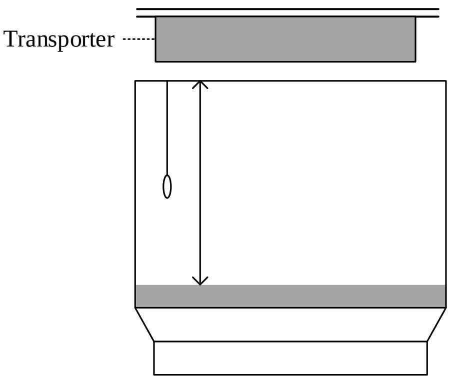


Figure 2.9: Transporter Arrives at the Silo

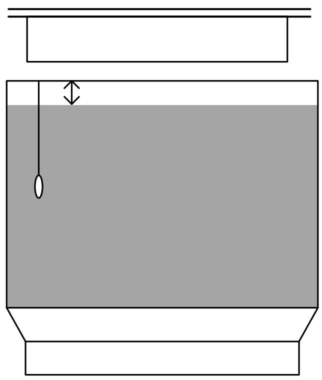


Figure 2.10: Transporter Emptied in Silo

Manual orders by the operator occur in situations such as starting up at the beginning of the day, changeover in batches, or various other situations. A manual request can be done in various degrees of automation. For example, the most automated manual request is just the manual placing of an order. After this the process will be handled automatically. It is also possible to manually control every single step of the process: from the request of raw materials to the delivery at the press.

The orders for a top-layer and a bottom-layer differ in size. A mix for the bottom-layer is approximately 2,200 kg while a mix for the top-layer is approximately 1,200 kg. A single delivery takes between 5-20 minutes to be processed by the press. The actual time depends on the dimensions of the tile, the actual weight of the delivered mix, the production speed, the number of tiles per production cycle, if a tile will be washed or not, and possible delays during production.

2.2.2 Raw Material Supply

When the mixing department receives an order it will check if it can request the materials from the different raw material silos. This can only be done if the required conveyors and vertical lift are available. If they are not available, the order will be placed in a queue First Come First Serve (FCFS). When the required conveyors and vertical lift are available, the required raw materials are delivered using the conveyors to a vertical lift. Each mixer has one vertical lift that carries the materials from the floor-level to the mixer. Once the vertical lift has reached the mixer it empties the materials into the mixer, if available. If not, the vertical lift will wait until the mixer is available.

A mixture contains the following raw materials: sand, cement, and one or multiple aggregates. These are delivered from several silos using a conveyor and a vertical lift. There are two systems, one for the top-layer and one for the bottom-layer. Both systems work in the same way, however, the one for the top-layer can deliver materials to Mixer 1-3 and one for the bottom-layer can deliver materials to Mixer 4-5. The bottom layer delivery system delivers material from four large silos, while the top-layer delivery system delivers from multiple smaller silos. The materials are deployed from the silo onto a conveyor belt. The conveyor belt delivers the material to a vertical lift. Each mixer has its own vertical lift that caries the materials from the factory floor level up towards the mixer. Once the vertical lift has reached the mixer and the mixer is available, it empties the contents into the mixer. If the mixer is not available, however, the vertical lift waits until the mixer is available again. After the vertical lift has emptied itself, it travels back down to the shop floor. The vertical lift is than available again.

2.2.3 Mixing Process

The mixing process is done by one of the mixers (see Figure 2.11). After the vertical lift is emptied in the mixer, the cement is added to the mixer using an automatic tube system. If a dye (colouring material) is required, it is also added to the mixer. Now the mixing starts. After approximately one minute of mixing, water is added to the mixture and it is mixed for another minute. The water causes a chemical reaction with the cement to start the curing process. The mixing is finished when the mixture has reached a certain moisture level. The mixture can now be transported to the right press (see Section 2.3 for the transportation process). When the mixer is emptied, it starts processing the next order in queue, if available.

Figure 2.11: A Picture of a Mixer for a Top-Layer

Orders of a press are dedicated to a certain mixer during the whole processing time of a batch. This is done for two reasons. The first reason is to minimise contamination of different colours in the mixes; processing two different colours in the same mixer consecutively would contaminate the tiles. It is possible to process different colours on the same mixer during a day, however, in that case cleaning of the mixer is required. Cleaning of the mixers takes more than 20 minutes. Therefore, the mixers are scheduled to process the same colour during the day as much as possible. The second reason is that the software that is used for controlling the mixing process, has no option for autonomously switching between mixers during the production of an entire batch. This may cause unnecessary delay when orders are waiting to be processed in a mixer while another mixer is available. For example, Mixer 4 and 5 make the same type of bottom-layer mixtures. However, a press cannot switch to another mixer during the processing of a batch. This may result in one mixer causing a lot of waiting time while the other mixer is idle.

2.3 Transportation Process

The transportation of a mixture to the right press is an important part of the material handling process. The transportation is done by five transporters that travel on a ceiling rail system. Section 2.3.1 describes the transportation system. Section 2.3.2 contains the dispatch rule currently used for controlling the transporter movements. Section 2.3.3 discusses scenarios on the transportation system that may cause waiting times.

2.3.1 Transportation System

Each mixer has its own transporter that delivers the mixture to the right press. A transporter (see Figure 2.12) can travel from the mixer to the carousel and from the carousel (see Figure 2.13) to the press using rail sections. There are five rail sections leading from the mixers to the carousel and four rail sections leading from the carousel to the presses. The carousel rotates the transporter from one section to another. The carousel makes the system very flexible as every press can be reached by every transporter. However, this also creates a bottleneck as all transporters need to cross the carousel each time. As not all mixers can create all types of layers, creating a direct rail section between a mixer and a press would reduce the production possibilities of the press and reduce the flexibility of the system.

Figure 2.12: A Picture of a Transporter

Figure 2.13: A Picture of the Carousel

The transportation time of each section differs and can vary from a few seconds to a minute. Turning the transporter at the carousel takes 5-10 seconds and emptying the transporter takes 20 seconds. We list all the processing times of possible transportation jobs in Appendix A.

The transportation system has several technical constraints, namely:

1. *A transporter can be sent to any direction on the rail system, however, it is dedicated to one mixer.*
   This means that, for example, all mixes from Mixer 1 are transported by Transporter 1. Mixes created by Mixer 1 cannot be transported by another transporter.

2. *Each rail section can host at most one transporter at a time.*
   A rail section cannot host more than one transporter as this would cause a short circuit. A short circuit would cause the whole transportation system to break down. Therefore, a transporter is only dispatched when all the required rail sections are available.
3. *Some machines share the same rail section for the delivery of materials.*
   For example, Press 1 and 2 share the same rail section (see Figure 2.4); when both presses require two mixtures, this can result in up to four transporters that require the same rail section. This may cause a lot of waiting time.

The technical constraints mentioned above cause rail sections 8 to a bottleneck in the system as the rail section serves two presses with both two silos: the silos of Press 1 and 2. The carousel is also a bottleneck in the system as each transporter needs to cross it for delivering an order and when returning to the mixer.

2.3.2 Dispatch Rule

Dispatching the transporters is currently done using a First Come First Serve (FCFS) prioritisation rule. This means the transporter that was first ready to transport is dispatched before the other transporters. However, if not all required rail sections are free the next transporter in queue whose required rail sections are free is dispatched. After the transporter has delivered the material, it will immediately travel back to the mixer when the carousel is available. If the carousel is not available, the transporter will wait at the press. As soon as the carousel becomes available, the transporters that are full are prioritised over the transporters that are empty and waiting to return. The dispatching method at SVI does not consider any other elements, such as possible starvation of presses or other jobs with earlier due dates.

2.3.3 Scenarios that Cause Delay

This section contains three possible scenarios that may cause waiting time for material. The scenarios display the weaknesses of the current dispatching rule at SVI.

Scenarios 1: An order with an early due date becomes tardy due to FCFS

This scenario may occur when a mixer receives two orders shortly after each other. The order that is received first is also processed first according to the FCFS rule. However, the order that arrived later may have an earlier due date than the first order. Table 2.6 shows an example of such a scenario where four orders arrive shortly after each other at the mixing department. Order 3 and Order 4 are the orders that arrive shortly after each other at Mixer 4. Order 1 and 2 arrive before Order 3 and 4. Order 1, 2, and 3 can immediately start mixing as the mixers are available and they do not share the same mixer. Order 4 has to wait before Order 3 is finished as they both require the same mixer. Table 2.6 also shows the required rail section for the delivery of the mixture, the time of arrival of the orders, and the ranking of the orders. The ranking of the orders is based on the FCFS rule; the earlier the order arrived, the higher it is ranked on the list.

Table 2.6: List of Orders for Scenario 1

Order	Mixer	Required Rail Section	Time of arrival (sec.)	Ranking	Due Date (sec.)
Order 1	Mixer 1	Rail section 8	50	1	600
Order 2	Mixer 2	Rail section 8	60	2	700
Order 3	Mixer 4	Rail section 8	70	3	800
Order 4	Mixer 4	Rail section 10	80	4	550

Figure 2.14 shows a disjunctive graph of the precedence relations of the operations of the four orders. The disjunctive graph is a representation introduced by Roy & Sussman (1964) and is a method of displaying precedence relations in scheduling problems. It displays the operations for each order, a source node, a sink node, and the precedence relations between operations and machines. An activity can only start as soon as all the activities preceding a node are completed. Each order has a consecutive sequencing of operations to process, these precedence relations are marked with conjunctive arcs (normal line). The precedence relations between orders on resources are marked with a disjunctive arcs (dotted line). For example, the transportation of Order 3 can only start after the transportation of Order 1 and 2 is finished. The transportation of Order 4 can only start after the transportation of Order 3 is finished. In the graph Mixing 1 represent the mixing operation of Order 1 and Transportation 1 represents the transportation operation of Order 1.

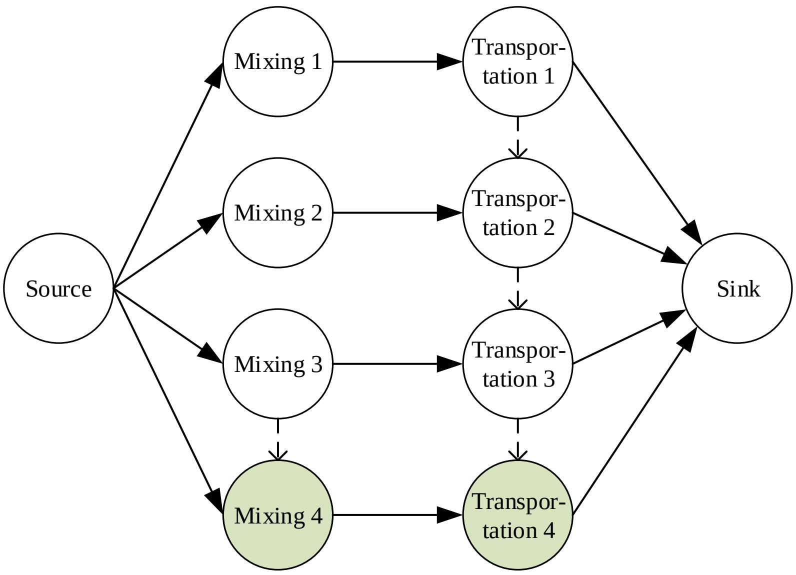


Figure 2.14: Disjunctive Graph Representation of Precedence Relations Scenario 1

Figure 2.15 shows the resulting schedule from the disjunctive graph shown in Figure 2.14. Each bar represents the activity on a specific resource with the number of the order. Each activity can only start after all precedence constraints are met. This means that, for example, the transportation of Order 3 can only start when the mixing of Order 3 is completed and the transportation of Order 2 is completed. This results in a schedule where the transportation of Order 4 starts after the transportation of Order 3. However, the due date of Order 4 is earlier than the due date of Order 3. Thus Order 4 is pushed backwards in the schedule, although it has an earlier due date. This results in tardiness (an order that is delivered too late) of Order 4 and thus waiting time at the press. Figure

2.16 shows a schedule where the tardiness could be prevented by only changing the sequence of the orders. Order 4 is not tardy anymore, while the completion time of all other orders remain the same.

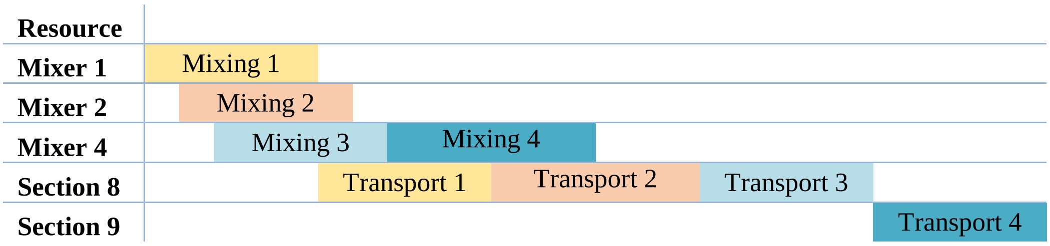


Figure 2.15: Schedule of Orders of Scenario 1

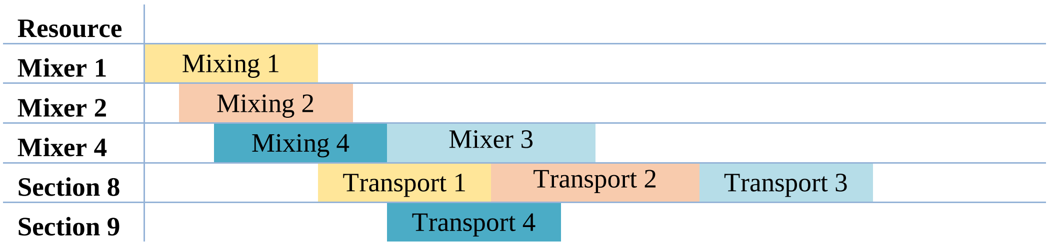


Figure 2.16: Improved Planning of Orders of Scenario 1

Scenario 2: A lot of orders within a small time window

With the current ordering system, it may occur that the Mixing department receives a lot of orders within a small time window while in other time windows there are very few to none orders. For example, the mixing department can wait for 5 minutes and have no orders, after which 8 orders arrive within several minutes. This large amount of orders within a short time window may cause delays. Table 2.7 shows an example where many orders arrive within the same time window while before these orders arrive the mixing department is idle in time window 0-300.

Table 2.7: List of Orders for Scenario 2

Order	Mixer	Required Rail Section	Time of arrival (sec)	Ranking	Due date (sec)
Order 1	Mixer 1	Rail section 8	300	1	600
Order 2	Mixer 2	Rail section 8	320	2	650
Order 3	Mixer 3	Rail section 6	330	3	700
Order 4	Mixer 4	Rail section 9	340	4	680
Order 5	Mixer 5	Rail section 8	360	5	710
Order 6	Mixer 1	Rail section 10	380	6	650
Order 7	Mixer 4	Rail section 8	400	7	730

Figure 2.17 shows a schedule that arises from this arrival. Many orders become tardy because there are a lot of orders that require the same resources within a short time window. This occurs while the mixing department was idle for a long time before these orders arrived. The current system of ordering using triggers causes this to happen occasionally. By shifting orders to an earlier position these tardy orders could be prevented.

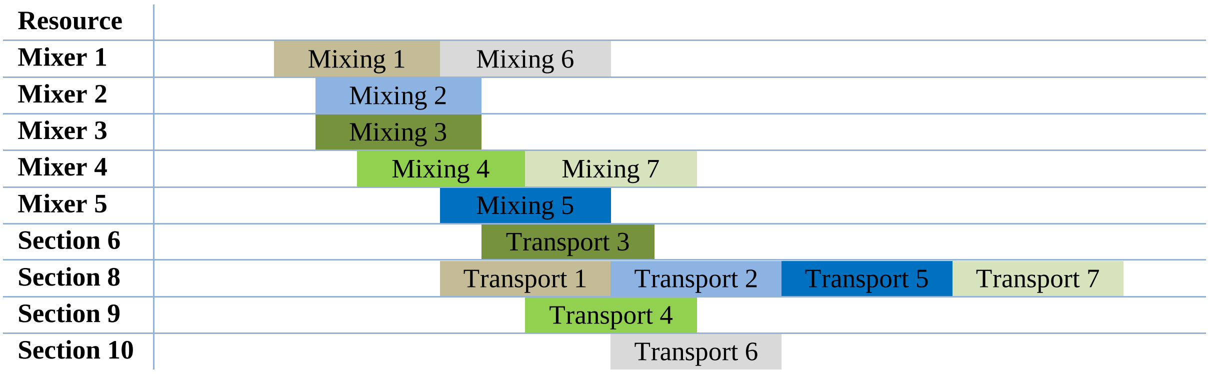


Figure 2.17: Schedule of Orders of Scenario 2

Scenario 3: Multiple transporters require Rail Section 8 within a short time period

This scenario is similar to scenario 2, but more specific for rail section 8. When multiple orders arrive within a short time window that all require Section 8 for the delivery of material this can cause a lot of delay. For example, when the mixing department receives four orders at once that are processed by four different mixers, the transporters will be scheduled FCFS for dispatching based on the mixer that was first ready. This may result in waiting time and thus delay at the presses. The waiting transporters are also blocking newly finished mixes to be transported.

2.4 Performance of the System

This section contains an analysis of the performance of the current material delivery system. Section 2.4.1 describes how SVI measures the performance of the system. Section 2.4.2 contains the analysis of the performance using historical data.

2.4.1 Measurement

A delivery of a mix can be delayed due to several reasons. A delay may result in a delivery that is tardy (too late) and cause starvation of material at the press. The press constantly requires the input of two types of mixes. Starvation of one of the mixes will cause a press to stop. Therefore, the performance of a delivery can be measured by the tardiness of the delivery. Tardiness is the amount of time a delivery is done too late. If a delivery is done on time or before the due date, the tardiness is equal to zero. Besides a delivery being tardy, a delivery should not arrive too early. An early delivery can result in an overloaded silo or poor quality of the tiles (see Section 2.2.1). Therefore, the performance of a delivery can also be measured by the earliness, which is the amount of time a mix is delivered too early. We discuss earliness and tardiness further in Section 3.1.1.

SVI has no measurement tool for the earliness and tardiness, however, they use a system to register delay times at the machine. This system measures time windows in which no production occurs. The operator of the machine then assigns the cause of the delay for each time window. The causations linked to late delivery of materials are described as ‘waiting for top-layer’ and ‘waiting for bottom-layer’. There are multiple causes for waiting for material. For this research we only focus on resource planning causes, such as poor scheduling of transporters. During production many small delays may occur. Delays with a duration lower than 60 seconds are not given a causation. These short delays are automatically marked as ‘micro stops’. Micro stops may be caused by late delivery of materials, however, the micro stops are not assigned with a causation as this requires too much effort of the operator.

2.4.2 Analysis of the Historical Performance

Table 2.8 shows the total waiting time for material per machine for the years 2012-2015. The machines are numbered from P1 to P5 (the "P" stands for press). For multiple reasons the waiting time of P3 is not measured by SVI and is therefore not shown in the table. P4 is not in use anymore and is therefore also not shown in the table. As the total production time can differ per year for each machine, we divide the waiting time with the total number of manned hours as '% of Manned Time'. This makes the waiting times comparable with other machines and other years. The table shows that P5 has almost twice the amount of delay compared to P1 and P2. According to SVI this is due to the larger sizes of the tiles at P5 as more material is required. A higher usage of material results in a higher frequency of replenishment of material and may therefore cause more delays in the mixing and transportation. P5 has two separate rail sections leading towards its silos, so the rail section does not cause the delays. However, other orders processed by the same transporter may delay the orders for P5 (see Scenario 1 in Section 2.3.3 for a more detailed description of such scenarios).

Table 2.8: Waiting times of machines pear year with the cause waiting for material

Year	Machine	Waiting Time (hours)	% of Manned Time
2012	P1	76.5	3.1 %
	P2	61.8	2.5 %
	P5	57.1	5.4 %
	Total	*195.4*	
2013	P1	127.0	4.3 %
	P2	109.8	3.9 %
	P5	173.7	10.5 %
	Total	410.5	
2014	P1	99.3	3.6 %
	P2	85.5	3.2 %
	P5	122.5	7.4 %
	Total	*307.3*	
2015	P1	102.7	3.7 %
	P2	110.7	4.2 %
	P5	108.4	5.9 %
	Total	*321.7*	

Figure 2.18 shows the relative downtime caused by a delay in *bottom-layer* for P1, P2, and P5. The graph shows P5 has far more downtime caused by a delay in bottom-layer than P1 and P2. The downtime also increases when the tile height increases. The higher the tile the more bottom-layer is used; this corresponds with the opinion of management that a significant part of the downtime is caused by the thick layers at P5. Figure 2.19 shows the relative downtime caused by a delay in *top-layer* for P1, P2, and P5. The differences with machines are not as high as with the bottom-layer. The downtime does not seem to increase when the tile height increases. When the tile height increases, the usage for the top-layer does not increase, only the usage for the bottom-layer increases. Thus the large amount of delays for bottom-layer at P5 is caused by the large tiles that require more bottom-layer material per tile.

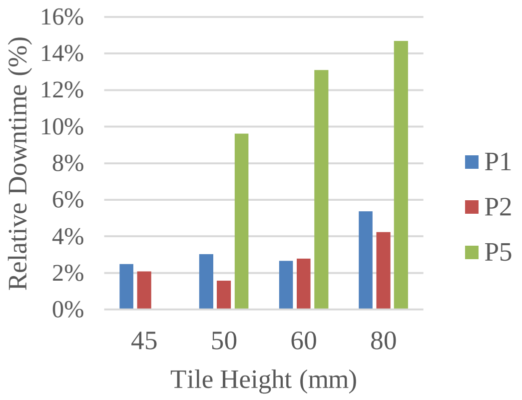


Figure 2.18: Average Relative Downtime Bottom-Layer

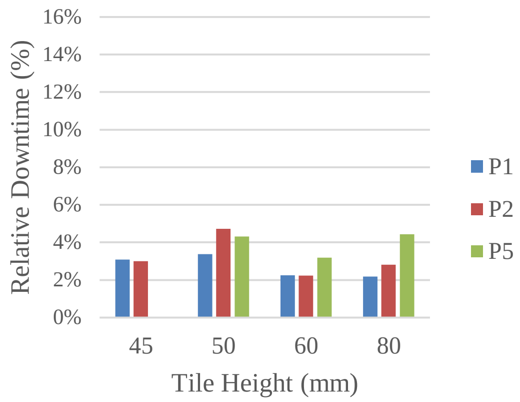


Figure 2.19: Average Relative Downtime Top-Layer

2.5 Conclusions

In this chapter we answered research question 1 regarding the analysis of the current production and material handling processes. We gave a more detailed description of the current processes and system regarding the ordering, mixing, and transportation of mixes. We concluded that at the moment simple dispatching rules are used (First Come First Serve) that do not take due dates or other factors into account. We also concluded that the carousel and rail section 8 are bottlenecks in the system and cause waiting times for the transporter. We named the tardiness of deliveries as a measurement tool of the system and deliveries should not be done too early as this can result in poor quality of the tiles. In order to measure the current performance of the system in terms of tardiness we used data from the OEE tool. This tool has missing data for waiting times below 60 seconds and Press 3. The data also contains waiting times with other causes. Finally, we concluded that P5 has the most waiting time for material and that the tile height has a large impact on the waiting time for the bottom-layer. In Chapter 3 we discuss methods for improving the planning procedure for the delivery of mixtures and reduce waiting times.

Chapter 3: Literature Review

In Chapter 2 we analysed the current production process, mixing process, transportation processes, and performance of the material delivery at SVI. This chapter answers research question 2 by discussing several methods from literature to optimise the scheduling procedures of the mixing and transportation of material. First, Section 3.1 discusses the Job Shop Scheduling Problem and a Mixed Integer Linear Programming model. Section 3.2 contains approximation methods for creating schedules. Section 3.3 describes rescheduling methods. Finally, Section 3.5 contains our conclusions of the literature research.

3.1 Job Shop Scheduling Problem

Literature describes several classifications for production environments with multiple *machines* and multiple *jobs*. Machines are a resource, often with limited capacity, that perform *operations* for a job. The operation takes a certain amount of *processing time*. A job can contain multiple operations that can be processed by different machines. Pinedo (2012) describes three common classifications of scheduling for jobs and machines, namely: the job shop, the flow shop, and the open shop. In a job shop each job has its own route for visiting machines. In a flow shop each job is processed on each machine and the route to visit these machines is the same for all jobs. In an open shop the routing of operations within the job is not predetermined. The situation at SVI best fits with the job shop, as each job has a predetermined route on different machines (see Section 4.1.3 for a further analysis of the requirements for a planning model at SVI). The job shop is modelled as a set of *m* machines and *i* jobs. Each job has its own fixed sequence of operations on one or more machines (Pinedo, 2012).

Literature describes many different characteristics for modelling job shop. The characteristics are required to be able to model the standard job shop problem and create feasible schedules. Most of the job shops consist of the following characteristics (Haupt (1989); He, Yang, & Tiger (1996); De Bontridder (2005)):

1. A job consists of multiple, sequential operations.
2. Each operation is processes by one machine with a known and fixed processing time.
3. An operation cannot be split.
4. Once an operation is started it must finish, it cannot be paused or stopped.
5. A machine can only process one operation at a time.
6. There are no machine breakdowns.
7. All jobs and their operations are known and released at time zero.

3.1.1 Objective Function

When comparing different schedules for a JSSP, the performance of each schedule is determined by the objective function. The goal of solving a JSSP is minimising the objective function, which is mostly a function of the completion time C_i of all jobs *i* (Pinedo, 2012).

The goal of this research is minimising the waiting times for deliveries. However, when a delivery is done too early, it may result in poor quality of the tiles. Thus the objective function requires minimisation for earliness and tardiness. Earliness is the amount of time a job is completed before its due date d_i, while tardiness is the amount of time a job is completed after its due date. Both earliness and tardiness are always positive or zero. Baker & Scrudder (1990) give an adaption for the Total Weighted Earliness/Tardiness (TWET) objective function with tolerances. These tolerances create a due window, instead of a due date, in which a delivery would not receive a penalty. Equation 3.1 shows the calculation of the earliness per job with a tolerance u_i. Equation 3.2 shows the calculation of the tardiness per job with a tolerance v_i.

$$E_i = \max(d_i - C_i - u_i, 0) \tag{3.1}$$

$$T_i = \max(C_i - d_i - v_i, 0) \tag{3.2}$$

The TWET problem may have different weights depending on the job. Therefore, a weight factor is assigned to both the earliness E_i and tardiness T_i. Equation 3.3 shows the TWET objective function with a weight factor α_i for earliness and weight factor β_i for tardiness.

$$\sum_{i=1}^{n}(\alpha_i E_i + \beta_i T_i) \tag{3.3}$$

Figure 3.1 shows a graph of this objective function for a single job with equal weight factors for earliness and tardiness. Figure 3.2 shows the penalty function for the TWET objective function of one job, with an earliness tolerance u_i, in a graph. A penalty is given if the completion time of the job is either before $d_i - u_i$ (earliness) or after $d_i + v_i$ (tardiness). In this graph the weight factor for earliness α_i is lower than the weight factor for tardiness β_i.

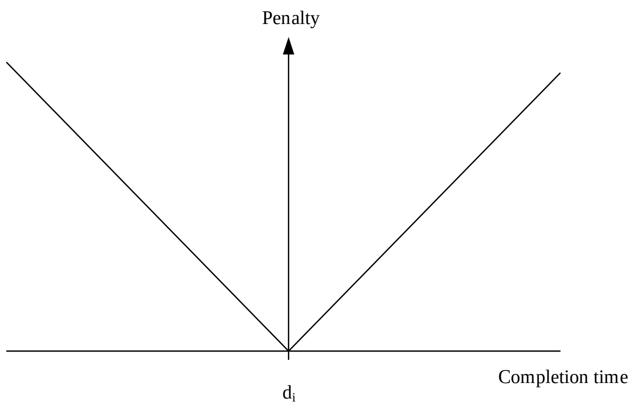


Figure 3.1: Penalty Function for One Job With Penalties for Earliness and Tardiness

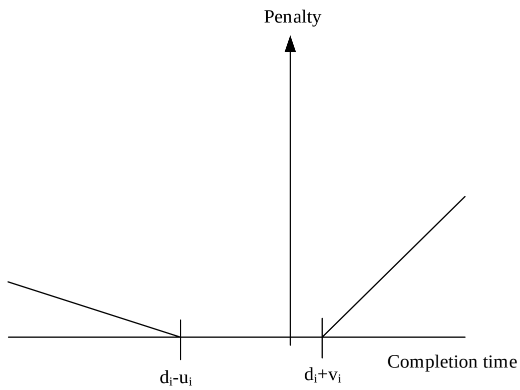


Figure 3.2: Penalty function with Tolerances and Unequal Weights by Baker & Scudder (1990)

3.1.2 Mixed Integer Linear Programming Model

The determination of the sequence of jobs on a machine is an important part of scheduling. In order to model the sequence on a machine, often integer variables are used to model the precedence relations for the sequence of jobs. Therefore, scheduling problems are often formulated using a Mixed Integer Linear Programming (MILP) model.

Zhu & Heady (2000) describe a MILP formulation for minimising the TWET objective function (Equation 3.6) for an identical, parallel multi-machine problem with distinct due dates and penalties for each job. Equation 3.7 shows how the model of Zhu & Heady (2000) determines the earliness E_i, tardiness T_i, and completion time C_i

for job i. To simplify the model, a dummy job is added as the first job. The dummy job ensures all other jobs have exactly 1 preceding job on the machine, which allows the simplification of the model. The first actual job on a machine is always preceded by the dummy job. All operations of the dummy job are processed on all machines with a processing time of 0. The precedence relations are modelled by using a binary variable Y_{ijm}. The value of Y_{ijm} is 1 when job i directly precedes job j on machine m, otherwise the variable is zero (see Equation 3.5). Jobs can be processed on each of the M identical machines; the constraint of Equation 3.8 ensures that the job is assigned to exactly one machine. The constraints of Equations 3.9-3.10 ensure a job is directly preceded by exactly one other job and directly followed by at most one other job. The constraints require variable Z_{im} which is 1 if job i is processed on machine m and 0 otherwise (see Equation 3.4). Finally, the constraint of Equation 3.11 is used to ensure the jobs on a machine do not overlap. In this constraint, B represents a large number. This ensures that when Y_{ijm} is set to 1 (meaning job i precedes job j directly on machine m) the left hand side of the constraint is forced to be larger or equal than the right hand side by placing the completion time C_i of job i at least p_{im} apart from the completion time C_i of job i. When Y_{ijm} is set to 0 the left hand side will always be larger than the right hand side as b is not subtracted from the left hand side. The model of Zhu & Heady (2000) represents many of the requirements for the scheduling model of SVI. However, to meet all requirements of the scheduling problem some adaptions are required. Section 4.1 describes all adaptions required for the scheduling problem at SVI.

Parameters

d_i	due date of job *i*
e_i	earliness cost per period for job *i*
t_i	tardiness cost per period for job *i*
p_{im}	processing of job *i* on machine *m*
B	a large number

Variables

C_i	completion time of job *i*	
T_i	tardiness of job *i*	
E_i	earliness of job *i*	
$Z_{im} =$	$\begin{cases} 1 & , if\ job\ i\ is\ processed\ on\ machine\ m \\ 0 & , otherwise \end{cases}$	3.4
$Y_{ijm} =$	$\begin{cases} 1 & , if\ job\ i\ directly\ precedes\ job\ j\ on\ machine\ m \\ 0 & , otherwise \end{cases}$	3.5

Objective Function

$$\min \sum_{i=1}^{N} (e_i E_i + t_i T_i) \quad 3.6$$

Constraints

$$C_i - T_i + E_i = d_i \qquad \forall i = 1,2,\dots,N \qquad 3.7$$

$$\sum_{m=1}^{M} Z_{im} = 1 \qquad \forall i = 1,2,\dots,\mathrm{N} \qquad 3.8$$

$$\sum_{j=2,\mathrm{j}\neq \mathrm{i}}^{N} Y_{ijm} \leq Z_{im} \qquad \forall i = 1,2,\dots,\mathrm{N}, m = 1,2,\dots,M \qquad 3.9$$

$$\sum_{i=1,i\neq j}^{N} Y_{ijm} = Z_{jm} \qquad \forall j = 2,3,\dots,\mathrm{N}, m = 1,2,\dots,M \qquad 3.10$$

$$C_j - C_i - B(Y_{ijm} - 1) \geq p_{jm} \qquad \forall i = 1,2,\dots,N, j = 2,3,\dots,N, i \neq j, \mathrm{m} = 1,\dots,\mathrm{M} \qquad 3.11$$

$$C_i, T_i, E_i \geq 0 \qquad \forall i$$

$$Z_{im} \in \{0,1\} \qquad \forall i, m$$

$$Y_{ijm} \in \{0,1\} \qquad \forall i, j, m$$

3.2 Approximation Methods for solving a Job Shop Scheduling Problem

Solving a Job Shop Scheduling Problem (JSSP) to optimality using MILP approaches mostly require many computational steps. They often require so many steps that it is impossible to solve within a few minutes or even within years. JSSPs are classified as NP-hard (Pinedo, 2012); finding an optimal solution for these problems is often very hard and requires a high amount of computation time for large problems. This section contains methods from literature for solving a JSSP (approximately). Approximation algorithms give (near) optimal solutions within short computation time. Most methods for solving a JSSP are focused on minimising the makespan of the problem. Solving a TWET scheduling problem has received fewer attention in literature. The research done for TWET scheduling problem is mostly done towards single machine scheduling problems (Yang, Sun, Saygin, & Sun, 2012). We first discuss dispatching rules in Section 3.2.1. Next, Section 3.2.2 contains a review of Neighbourhood Search methods. Finally, we describe several construction heuristics in Section 3.2.3.

3.2.1 Dispatching Rules

Dispatching rules are simple, constructive methods that require little computation time and give jobs or operations a priority based on a predefined rule. The priority defines the ranking order in which a job is scheduled. The better the job scores on a rule, the earlier is considered for scheduling. Rajendran & Holthaus (1999) classify many different dispatching rules into five categories based on the aspect of the rule, namely:

- Processing time
- Due date
- Simple rules without processing time or due date
- Shop floor conditions
- Rules including multiple aspects

For example, First Come First Served (FCFS) is a simple dispatching rule without the use of processing time or due date. A dispatching rule with multiple aspects is, for example, the Apparent Tardiness Cost (ATC) rule. The ATC rule dispatches the job based on a combination of the Weighted Shortest Processing Time (WSPT) first and

Minimum Slack (MS) first; the higher the job is prioritised, the earlier it is dispatched. Pinedo (2012) states that simple dispatching rules with one or two parameters as input work well for single objective functions, however, objectives in the real world are often more complicated. Composite dispatching rules with several different parameters can create better schedules than when using simple dispatching rules. Ouelhadj & Petrovic (2009) state that there is not one dispatching rule that outperforms in all situations, however, a simulation study may be used to evaluate several rules at once and select the one with the best performance for the specific situation.

Haupt (1989) discusses an extensive list of basic dispatching rules, these vary from the simple rules such as FCFS to more complex rules such as the ATC rule. The FCFS rule dispatches the job ranked by the arrival time; the earlier the job arrived, the earlier it is dispatched. The Shortest Processing Time (SPT) rule dispatches the job ranked on the processing time; the shorter the processing time, the earlier it is dispatched. The Earliest Due Date (EDD) rule dispatches the job ranked on the Due Date; the earlier the due date, the earlier the job is dispatched.

Thiagrajan & Rajendran (2005) suggest dispatching rules for JSSPs with E/T objective functions, they convert the E/T penalties per time unit for each job into a factor $w_i = t_i/e_i$. The reason for this factor is the earliness and tardiness penalties are contradicting in nature and weights of tardiness are higher than the weights of earliness. The proposed new dispatching rules are common rules modified with the E/T factor.

3.2.2 Neighbourhood Search

Neighbourhood search algorithms iteratively search in the neighbourhood of an already constructed solution with the goal of improving the objective function. The iterative search continues until a stopping criterion is met. Neighbourhood search algorithms are good for quickly improving initial solutions. However, as neighbourhood search algorithms only search in the local neighbourhood, the improvement of the algorithms may not be significant. A neighbour can be defined by a predefined modification of the initial schedule (Pinedo, 2012). The entire set of neighbours obtained from all modifications of the schedule is called the neighbourhood. For example, swapping is a modification method that swaps two or more jobs in a sequence. Swapping can be restricted to only adjacent jobs in the sequence. The neighbourhood of all possible adjacent swaps consists of $(n-1)$ possible sequences, while the neighbourhood of all possible swaps consists of $\frac{n(n-1)}{2}$ sequences. For a sequence of five jobs this means 4 and 10 possible neighbours respectively. As neighbourhood search is iterative, it often requires calculating the objective function of every neighbour in each step. Keeping the number of neighbours low reduces the computation time per iteration. Figure 3.3 shows an adjacent swap in a sequence of jobs and Figure 3.4 shows a general swap, where any pairwise interchange between jobs is possible. The resulting sequence after a swap is a neighbour of the initial sequence. For each neighbour the objective value is calculated. A neighbour is accepted or rejected based on the acceptance-rejection criterion (2012). For example, when using the steepest descent method, in each iteration the neighbour with the greatest improvement in objective value is selected; this may continue until a stopping criterion is met or no further improvement can be found.

In neighbourhood search a method may run into a loop as the same swaps are made each time; the method may not escape the local neighbourhood. To escape the local neighbourhood methods such as tabu search, simulated annealing, and genetic algorithms can be used (Ouelhadj & Petrovic, 2009). These algorithms escape the local neighbourhood by accepting worse solutions or creating better initial solutions.

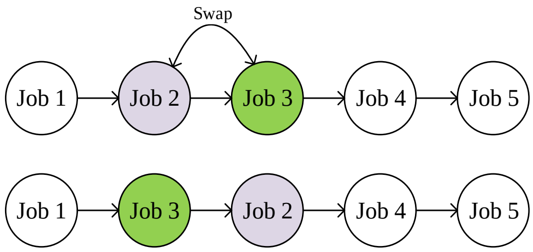


Figure 3.3: Possible Adjacent Swap in a Sequence

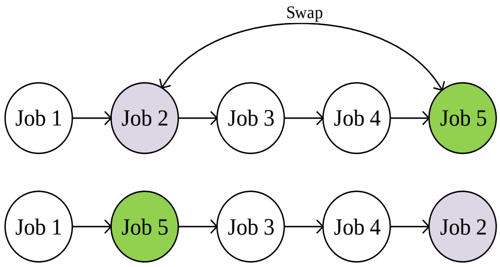


Figure 3.4: Possible Swap in Sequence

Tabu-search is a method presented by Glover (1989) for escaping a local optimum by making certain *steps* in the neighbourhood 'tabu'. When a swap is made, it is put on a tabu-list. The tabu list is a limited list in terms of the number of swaps it can contain. When performing a tabu-search, the candidates that are on the tabu-list are forbidden (tabu) and not considered in the candidate list for swapping. Each time a swap is made, the swap is added to the top of the tabu-list and the last one on the list is removed from the list. This makes the swap that was on the last position not tabu anymore, it may be put on the candidate list again (if possible). By holding a list of tabu moves the tabu-search algorithm may be able to escape the local optimum and move to a better overall objective value.

Simulated Annealing is a method that uses probabilistic acceptance-rejection criteria for escaping local optima (2012). The origins of the method lie within the metal industry in order to improve the strength of materials using an annealing process. Simulated Annealing makes use of iterations. Each iteration a schedule from the neighbourhood is selected with a certain probability. This allows the method to select schedules with worse objective values than the current one in order to escape the local optimum. However, the effectiveness depends on the neighbourhood construction. Simulated Annealing performs well when the neighbourhood construction method allows to escape the local optima. When this is not the case, Simulated Annealing is not able to efficiently move towards better optima.

The iteration process of neighbourhood search continues until a stopping criterion is reached. The stopping criterion can be reaching an iteration limit, reaching a computation time limit, or the objective value coming within an accepted range.

3.2.3 Construction Heuristics

Literature describes several construction heuristics for solving the single-machine TWET problem (approximately). Mazzini & Armentano (2001) describe a heuristic for the JSSP with unequal due dates and ready times (the moment the first operation of a job may start). Their heuristic constructs (near) optimal schedules within short computation time. The heuristic first inserts each job at its best feasible location. Next, a feasibility procedure is done in order to reposition overlapping jobs. This is done by shifting jobs to the left or right. After the feasibility procedure, an updating procedure is done that shifts blocks of jobs to the left or right. Finally, a local optimisation procedure is done to check if better solutions exist by swapping adjacent jobs (see Section 3.2.2). This method by Mazzini & Armentano (2001) provides feasible solutions within short computation time and does not require an initial sequence of jobs. However, their method makes use of due dates instead of due windows and may therefore not fit the requirements of the scheduling problem at SVI.

Wan & Yen (2002) describe a schedule construction heuristic for the single machine TWET problem with due windows for a given sequence of jobs. Their heuristic consists of three steps. Step 1 creates an initial sequence for the problem using a simple dispatching rule such as EDD or EST (see Section 3.2.1 for more dispatching rules). Step 2 generates an optimal schedule for the given sequence using the optimal timing algorithm. Step 3 generates a new sequence of jobs using a neighbourhood search. The neighbourhood is defined by adjacent swapping modification and a tabu-list. By using dominance criteria many possible swaps can be eliminated and the neighbourhood search is done faster. The optimal timing algorithm of Step 2 optimises the TWET problem with distinct due windows, given a sequence of jobs. It starts with the first job and places this, such that it finishes at the start of the due window. However, if the moment the due window starts is smaller than the processing time, it starts at 0. Next, it schedules every next job after the previous job and as optimal as possible. If the job starts at the end of the previous job, the jobs will be shifted to the left as much as possible and as long as it improves the objective function. This process is described in detail in Wan & Yen (2002).

The heuristics by Mazzini & Armentano (2001) and Wan & Yen (2002) provide good quality to optimal schedules when scheduling for the single-machine TWET problem. However, when scheduling for multi-machine TWET problems, extra procedures are required to be able to construct schedules for all operations on multiple machines. The extra procedures are required to maintain the feasibility of the schedules on all machines. These extra procedures require extra computation time and may result in worse quality schedules than the original heuristic achieves for its scheduling problem.

3.3 Rescheduling Methods for Job Shop Scheduling

During the execution of a schedule, events (such as new arriving jobs or machine breakdowns) may arise that may require a schedule to be reconsidered. These events may make the initial schedule infeasible or a better schedule may exist given the new situation. Rescheduling may be required to cope with such events. We first discuss rescheduling policies in Section 3.3.1. Next, we describe repairing methods of schedules in Section 3.3.2. Finally, Section 3.3.3 contains methods to improve robustness of schedules.

3.3.1 Rescheduling Policies

Vieira, Herrmann, & Linn (2003) distinguish three different policies for rescheduling: periodic, event-driven, or hybrid. Periodic rescheduling updates the schedule every predefined time interval. For example, if the interval is set to 10 minutes, the current schedule is checked every 10 minutes and may be updated. In event-driven policies, schedules are updated when an event occurs. The hybrid policy combines both periodic and event-driven rescheduling, thus rescheduling may occur in each period and when events occur such as the arrival of a new job. Vieira, Hermann, & Linn (2003) give a list of common events:

- Machine failure
- Urgent job arrivals
- Job cancellations
- Due date change
- Delay in the arrival or shortage of material
- Change in job priority
- Rework or quality problems

- Over- or underestimation of process time
- Operator absenteeism

3.3.2 Repairing Schedules

A schedule needs repairing if the current schedule has become infeasible or the performance has dropped below a specified level. A schedule becomes infeasible if, for example, a delay occurs and an operation cannot be executed on the scheduled window anymore. The performance may drop below a certain level if, for example, a machine has downtime and therefore change the parameters of earliness for a job, which makes the current schedule unattractive.

Viera, Hermann, & Linn (2003) suggest three methods for repairing a schedule, namely: right-shift rescheduling, partial rescheduling, and complete regeneration. In Table 3.1 we present a list of three jobs, each job has two operations that require processing on both machines. For each job, Operation 2 can only start after Operation 1 is finished and a machine can process one operation at a time.

Table 3.1: Example of jobs with scheduling on two machines

Job	**Operation 1**	**Operation 2**
Job 1	Machine 1	Machine 2
Job 2	Machine 2	Machine 1
Job 3	Machine 2	Machine 1

An example of a schedule for these jobs is given in Figure 3.5. To show how the rescheduling methods work, an event occurs during Operation 1 of Job 1 (shown by the black arrow) that causes a delay in the operation. Because the delay makes the schedule infeasible, rescheduling is required. Below the initial schedule in Figure 3.5 an example solution is shown of the three possible rescheduling methods. The *right-shift rescheduling* method shifts the complete schedule to the right from the moment the event occurs, it is an easy to apply method, however, it may result in poor objective values. The *partial rescheduling* method makes small alterations in the schedule to lower the impact of an event, it can improve the objective value within short computation time. In this example the partial rescheduling does not affect Job 2. The *complete regeneration* method creates an entire new schedule given the event. The method may result in the best objective value, however, it takes more computation time than partial rescheduling or right-shift rescheduling.

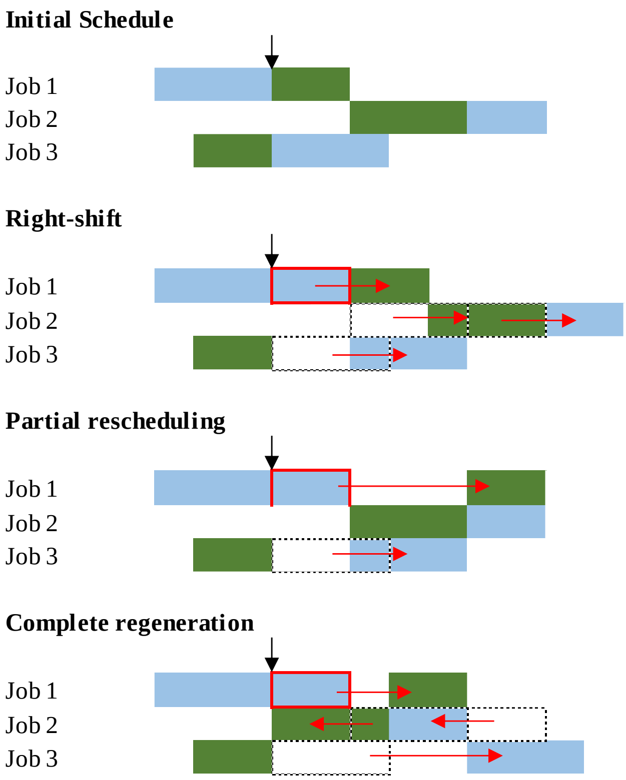


Figure 3.5: Rescheduling methods

3.3.3 Robustness of Schedules

When schedules are made robust, the actual performance after execution of the schedule may be better than when no robustness was added to the schedule (Ouelhadj & Petrovic, 2009). This while adding robustness often degrades the objective value of the schedule created. However, as events may occur during the execution of a schedule, robust schedules may perform better under these circumstances than non-robust schedules. When an operation delays due to any reason, a robust schedule may still remain feasible. This robustness can be created by, for example, deliberately increasing the processing times of operations or inserting delay time between jobs. In case variability occurs during production, it can be absorbed by the inserted time.

3.4 Performance of Methods

The performance of heuristics is often measured by speed and accuracy. Speed tells how quick a method finds a good or optimal solution. Accuracy determines how close the solution of the method comes to the optimal solution. Cordeau, Gendrea, Laporte, Potvin, & Semet (2002) state simplicity and flexibility are also good criteria for heuristics. Simplicity determines how easy to implement a method is and the ease to code. Flexibility tells how easy the method is being adapted to new constraints and other changes. In order to help decide which method from literature we use for the research, we rank each method with the four criteria. We do not have the research time to test all methods. Therefore, we rank each criterion based on our opinion.

Table 3.2 contains a ranking of all methods for creating schedules as described in this chapter. The MILP method scores low on speed as for some scheduling problems the computation time may very high. The method scores high on accuracy as it will return the optimal solution when it finishes computing. For simplicity it scores low as it requires complex coding. Though it is very flexible in adding constraints and adjusting parameters afterwards, as these can simply be added or modified without changing large pieces of code.

The heuristics by Mazini & Armentano (2001) and Wan & Yen (2002) both score the same. The speed of both heuristics is very high as the construction heuristic can always find solutions within short computation time. However, this is not certain when the heuristic is adapted for the situation at SVI. We rate the accuracy of these methods low; the method of Mazzini & Armentano (2001) requires local search to improve the accuracy, however, it may still be far from optimal. The method of Wan & Yen (2002) only finds an optimal solution for a given sequence. Both methods score high on simplicity as the methods are easy to code and apply. However, the methods are not very flexible in adding constraints or changing parameters.

We also added tabu-search and simulated annealing to the list although they are not construction methods and do not generate a schedule. We rated them both high on speed and accuracy as they can quickly converge towards the optimal solution. However, the performance also depends on the construction method used. Tabu-search and simulated annealing are both quite easy to code. They are also flexible in adding constraints or parameters for restricting certain sequences as these can be filtered out each iteration.

We conclude the local search algorithms score good for each criterion. However, the local search algorithms still require good construction algorithms. As no good construction algorithms could be found in literature that would fit the situation at SVI, the MILP method is best method for finding solutions with high accuracy. The method is also very flexible in adding constraints and changing parameters. As finding good schedules to reduce the waiting time at SVI is of high importance, we choose the MILP method for this research.

Table 3.2: Ranking of Criteria

Method	**Speed**	**Accuracy**	**Simplicity**	**Flexibility**
MILP	--	++	-	+
Mazzini & Armentano (2001)	++	-	+	-
Wan & Yen (2002)	++	-	+	-
Tabu search	+	+	+	+
Simulated Annealing	+	+	+	+

3.5 Conclusions

In this chapter we answered research question 2 by discussing methods for creating and improving schedules for the delivery of materials. We described three methods from literature for creating schedules, namely: a Mixed Integer Linear Programming approach, improved dispatching rules such as Earliest Due Date (EDD) or Apparent Tardiness Cost (ATC), and using a scheduling heuristic in combination with neighbourhood search methods. Solving using a MILP may give optimal schedules, however, the computation time for this method may run too large when the size of the problem grows. Dispatching methods are easy to implement and simple to use, however, they may give poor quality schedules. Finally, finding schedules using the heuristics and neighbourhood search provides good to optimal schedules for single-machine problems within short computation time. However, for multi-machine problems the heuristics require extensions in order to create feasible schedules and the computation time increases. The accuracy of the schedules generated may also become much worse. In order to ensure good quality solutions and great improvement from the current situation we choose the MILP method for creating schedules. We discussed a rescheduling framework for repairing schedules when rescheduling is required. This may be required when certain events occur such as the start of a new planning cycle, machine breakdown, or any other event that has an impact on the schedule. The amount of reschedules required can be lowered by increasing the robustness of the schedules. The robustness can be increased by inserting idle time between operations or increasing their processing times. In Chapter 4 we adapt the MILP method we described in this chapter to fit the situation at SVI and to be able to work with rescheduling methods that require short computation time.

Chapter 4: Solution Design

In Chapter 3 we conducted a literature review for finding methods to improve the planning of mixing and transportation. We concluded that the MILP and rescheduling method are most suitable for creating schedules with a high accuracy (close to optimal) for the problem at SVI. In this chapter we answer research question 3 and discuss methods that can be used by SVI to minimise the TWET problem. We adapt the selected methods from literature research to fit with the situation at SVI. Section 4.1 describes how we transform the practical scheduling problem at SVI into a MILP model for solving the scheduling problem. Section 4.2 introduces an order prediction model. Section 4.3 contains a rescheduling method and a planning cycle. Finally, Section 4.4 contains our conclusions regarding the solution design.

4.1 Modelling of the Scheduling Problem

In Chapter 2 we described the current situation at SVI. In Chapter 3 we introduced different methods from literature for creating schedules. In this section we model the situation at SVI in such a way that we are able to create feasible schedules using the MILP method. Section 4.1.1 describes the modelling of the different operations. Section 4.1.2 contains the modelling of the resources. Section 4.1.3 discusses the requirements for the model. Next, Section 4.1.4 contains a mathematical formulation of our model. Section 4.1.5 contains the required parameters for the MILP model. Finally Section 4.1.6 describes how the MILP can be solved using software applications.

4.1.1 Operations

In Section 2.1.3 we described all operations for delivering mixtures to the presses (from the request of raw materials, to the delivery of a mixture to the right silo). In order to simplify our model, we do not take into account all operations for scheduling. One of the operations we do not take into account is the supply of raw materials towards the mixers. The raw material supply is much faster than the mixing process; it can always have a new supply of raw material ready as soon as the mixer has finished mixing the preceding mixture. Therefore, we leave out the operations of supplying the raw material to the mixer. Some sequential operations require exactly the same resources for processing. To simplify the model we combine these operations into one single operation. The processing time of the combined operations is the sum of the processing times of these operations. We combine the following operations:

- Dry mixing and wet mixing
- The transporter traveling towards the carousel and turning on the carousel
- The transporter traveling towards the silo and emptying the material into the silo
- The transporter traveling back to the carousel and turning on the carousel

Table 4.1 lists the resulting operations we use in our model. In Section 2.1.3 we described three possible waiting moments for the transporter. During these waiting moments resources are still being used and cannot be assigned for other operations. For example, at the first waiting moment (when the mixing is complete but the transporter has not returned yet) the mixer is still in use as it has not been emptied yet. Therefore, the amount of waiting time of this waiting moment is added to the processing time of Operation 1. The second waiting moment (when the mixer is emptied into the transporter but the transporter needs to wait until it is dispatched to the carousel) is added as an extra operation, Operation 3, with a minimum processing time of 0. This extra operation is required because the waiting moment only requires the use of the transporter. The preceding operation also requires the mixer and the following operation also requires the carousel and rail section. Adding waiting time to these would unnecessarily block those resources from being used for other operations. The third waiting moment (when the transporter has emptied the contents at the silo and is waiting to return to the mixer) is added as an extra operation, Operation 6, with a minimum processing time of 0. It is not possible to add the waiting time to Operation 5 as the completion time of Operation 5 is important for the calculation of the objective function. By adding the waiting time to this operation, it would return the wrong actual completion time. The extra requirements and waiting times for the operations are not standard in job shop problems. However, these can be modelled using the MILP method by adding extra parameters, variables, and constraints (see Section 4.1.4).

Table 4.1: Resulting Operations for the Scheduling Model

Operation	**Description of Operation**
Operation 1	Dry and wet mixing of the material
Operation 2	Emptying the mix into the transporter
Operation 3	Waiting of the transporter until it is dispatched to the carousel
Operation 4	Transporting to and turning on the carousel
Operation 5	Travelling towards the machine & emptying the contents in the designated silo
Operation 6	Waiting by the transporter until it is dispatched back to the carousel
Operation 7	Travelling back the carousel and turning on the carousel
Operation 8	Travelling back to the mixer

The model uses an extra dummy job in order to simplify the model (see Section 3.1.2). The dummy job is inserted as the first job ($i=1$). All parameters of the dummy job are set to 0 except for z_{1m}, which is set to 1 for each value of m. This is done to ensure the dummy job is the first operation on each resource.

4.1.2 Resources

In Section 2.1.3 we described all resources that are used for the delivery of material towards the press. In Section 4.1.1 we described leaving out the operations of the raw material supply. Therefore, we also leave out the resources required for these operations, namely: the raw material supply and the vertical lift. We also leave out the rail sections from the mixers to the carousel as these are only used by the transporter that is assigned to the mixer; the dedicated transporter already ensures that these rail sections cannot be used by more than one transporter at a time.

We also do not model the operations of the press in the schedule as this is a continuous process. However, we use the material demand of the press as input for the scheduling problem (see Section 4.2). Table 4.2 describes the resulting types of resources required for creating schedules and contains the amount of resources available for each type. The five mixers represent Mixer 1-5, the five transporters represent Transporter 1-5, there is one carousel and the four rail sections represent Section 6, Section 8, Section 9, and Section 10.

Table 4.2: Resources Used in Scheduling Model

Type Number	**Resource type**	**Number of Resources**	**Required for Operation**
1	Mixer	5	1, 2
2	Transporter	5	2, 3, 4, 5, 6, 7, 8
3	Carousel	1	4, 7
4	Rail Section	4	4, 5, 6, 7, 8

Table 4.3: All Resources Required in the Scheduling Method

Resource Number	**Resource**	*Resource Number*	**Machine**
1	Mixer 1	***9***	Transporter 4
2	Mixer 2	***10***	Transporter 5
3	Mixer 3	***11***	Carousel
4	Mixer 4	***12***	Section 6
5	Mixer 5	***13***	Section 8
6	Transporter 1	***14***	Section 9
7	Transporter 2	***15***	Section 10
8	Transporter 3		

4.1.3 Requirements for the MILP

This section contains the requirements for adapting the model of Zhu & Heady (2000) as described in Section 3.1.2. The requirements are necessary in order to model the situation at SVI into the MILP model. We adapt the model of Zhu & Heady (2000) to incorporate the following requirements:

1. *Due window instead of due date*

   In Section 2.4.1 we described that jobs have a due date as well as an earliness date. The due date and earliness date together form a due window. A job that is completed within the due window does not get a penalty. To create a due window instead of a due date, we use the parameter u_i for each job i, which is the earliness tolerance in which a completion of the job would not cause a penalty. We also introduce the variable U_i that represents the actual window used for the earliness. The actual window must always be smaller than or equal to the parameter u_i. Therefore, we introduce the constraint as shown in Equation 4.9. Once the completion time becomes earlier than the tolerance, this means an earliness penalty arises.

2. *Operations 1, 3, and 6 may have additional waiting time*

   In Section 4.1.1 we discussed the addition of waiting time for Operations 1, 3, and 6. To allow extra waiting time for these operations in our model, we introduce an extra variable and parameter. We introduce the parameter w_{io} for the maximum allowed waiting time for operation o of job i. This parameter is set to 0 if an operation is not allowed to have additional waiting time. We also introduce the variable W_{io} for the actual added waiting time of operation o of job i. The actual added waiting time W_{io}

must always be lower than or equal to the maximum additional waiting time w_{io}. Therefore, we introduce Equation 4.10.

3. *An operation of a job starts immediately after the previous one is finished*

   Our model also requires operations of a job to immediately start after the previous one is finished. The only three moments where extra time is allowed between operations are modelled using the additional waiting time (see 2nd requirement). To ensure each operation of a job starts immediately after the previous one, we add the Equation 4.3 to the model.

4. *An operation may require multiple resources during the entire processing time of an operation*

   We add two constraints to ensure that operations of jobs do not overlap on a resource. Equation 4.7 is added for the rail sections, transporters, and mixers. Equation 4.8 is added for the carousel. Both constraints are adapted from the model of Zhu & Heady (2000) (see Section 3.1.2). The constraints ensure that the start time of an operation is higher than or equal to the completion time of the preceding operation.

4.1.4 MILP formulation

This section contains the complete formulation of the MILP model for solving the scheduling problem at SVI. The formulation is composed from the method by Zhu & Heady (2000) (as described in Section 3.1.2), the modelling of operations in Section 4.1.1, the modelling of resources in Section 4.1.2, and the requirements as described in Section 4.1.3. Table 4.4 contains the indices for the variables and parameters of the formulation. Table 4.5 contains the parameters required for the MILP formulation. Table 4.6 contains the variables for the MILP formulation.

Table 4.4: Indices for the Model

$\boldsymbol{i}$	Job index
$\boldsymbol{m}$	Resource number
$\boldsymbol{o}$	Operation number

Table 4.5: Parameters as Input for the Model

$\boldsymbol{B}$	A large number
$\boldsymbol{N}$	Number of jobs
$\boldsymbol{M}$	Number of resources
$\boldsymbol{O}$	Number of operations per job
$\boldsymbol{S}$	Set of resources
$\boldsymbol{R_m}$	Set of starting operations given resource *m*
$\boldsymbol{T_m}$	Set of finishing operations given resource *m*
$\boldsymbol{d_i}$	Due date for job *i*
$\boldsymbol{u_i}$	Earliness tolerance for job *i*
$\boldsymbol{e_i}$	Earliness cost per unit of time for job *i*

$\boldsymbol{t_i}$	Tardiness cost per unit of time for job *i*
$\boldsymbol{p_{io}}$	Processing time for operation *o* of job *i*
$\boldsymbol{r_i}$	Release time of job *i*
$\boldsymbol{w_{io}}$	Maximum additional waiting time for operation *o* of job *i*
$\boldsymbol{z_{im}}$	Binary value, 1 if job *i* is processed on resource *m*, 0 otherwise

Table 4.6: Variables as Output of the Model

$\boldsymbol{Y_{ijm}}$	Binary value, 1 if job *i* directly precedes job *j* on resource *m*, 0 otherwise
$\boldsymbol{C_{io}}$	Completion time of operation *o* of job *i*
$\boldsymbol{W_{io}}$	Additional waiting time for operation *o* on job *i*
$\boldsymbol{T_i}$	Tardiness of job *i*
$\boldsymbol{E_i}$	Earliness of job *i*
$\boldsymbol{U_i}$	Earliness within tolerance for job *i*

Objective Function

$$\min \sum_j (e_j E_j + t_j T_j)$$

Subject to

$C_{1o} = 0$	$\forall\, o = 1,2,\dots,O$	4.1
$C_{i5} - T_i + E_i + U_i = d_i$	$\forall i = 2,3,\dots,N$	4.2
$C_{i,k-1} = C_{ik} - (p_{ik} + W_{ik})$	$\forall i = 2,3,\dots,N, k = 2,3,\dots,O$	4.3
$C_{i1} \geq r_i + p_{i1} + W_{i1}$	$\forall i = 2,3,\dots,N$	4.4
$\sum_{j=2,j\neq i}^{N} Y_{ijm} \leq z_{im}$	$\forall i = 1,2,\dots,N, m = 1,2,\dots,M$	4.5
$\sum_{i=1,i\neq j}^{N} Y_{ijm} = z_{jm}$	$\forall j = 2,3,\dots,N, m = 1,2,\dots,M$	4.6
$C_{jk} - C_{io} - B \times (Y_{ijm} - 1) \geq p_{jk} + W_{jk}$	$\forall j = 2,3,\dots,N, i = 1,2,\dots,N, i \neq j, k \in R_m, o \in T_m, m \in S$	4.7
$C_{jo} - C_{io} - B \times (Y_{ij11} - 1) \geq p_{jo} + W_{jo}$	$\forall j = 2,3,\dots,N, i = 1,2,\dots,N, i \neq j, o = 4 \vee 7$	4.8
$U_i \leq u_i$	$\forall i = 2,3,\dots,N$	4.9
$W_{io} \leq w_{io}$	$\forall i = 2,3,\dots,N, o = 1,2,\dots,O$	4.10
$C_{io}, U_i, W_{io}, T_i, E_i \geq 0$	$\forall i, o$	4.11
$Y_{ijm} \in \{0,1\}$	$\forall i, j, m$	4.12

The first constraint (Equation 4.1) ensures that the completion time of all operations of the dummy job are set to 0. This ensures the operations of the dummy job are the first one in sequence on each resource and makes it easier to model precedence relations in the MILP.

The second constraint (Equation 4.2) determines the completion time, tardiness, earliness, and earliness window. The subtraction of the tardiness and addition of the earliness and earliness window to the completion time of a job must match the due date for each job. We use the completion time of Operation 5 ($o=5$) in the calculation. The completion of Operation 5 represents the replenishment of the silo and is therefore the time we compare the due date with. We added the earliness window variable U_i to the constraint in order to create a due window instead of a due date. Earliness E_i will only arise when the completion time is before the due date and the difference between the completion time and due date becomes larger than U_i.

We introduce the third constraint (Equation 4.3) to ensure operations of a job start as soon as the previous operations of that job is finished; the starting time of an operation is the completion time of the preceding operation of that job (subtracted by the processing time and waiting time of that operation). The constraint is not applied to the first operation as it has no preceding operation in the job.

The fourth constraint (Equation 4.4) ensures a job can only start after its release date. This ensures that when new jobs are added in a new planning cycle, the job can only be scheduled after it was taken into account (see Section 4.3.1 for a further description of the planning cycle). We model this by constraining the completion time of the first operation of a job to be larger than or equal to the sum of the processing time and waiting time of the first operation and the release time of that job.

The fifth and sixth constraint (Equation 4.5-4.6) ensure that on each resource, each job is followed by at most one other job and each job is preceded by exactly one job. The latter does not hold for the dummy job. The dummy job simplifies the model by ensuring the first actual job on a resource is also preceded by exactly one job (the dummy job).

The seventh constraints (Equation 4.7) ensures the start time of a job following another job on a resource is larger than the completion time of that job. This constraint only holds when the job *i* precedes job *j*. This is done by using the parameter *B* in combination with the Y_{ijm} variable. When Y_{ijm} is equal to 1 (job *i* precedes job *j* on resource *m*), the constraint ensures the start time of job *j* is greater than or equal to the completion time of job i. When Y_{ijm} is equal to 0 (job *i* does not precede job *j* on resource *m*), parameter *B* ensures the completion times for these jobs are not constrained by each other. The starting operation and the finishing operation are denoted by the parameters R_m and T_m respectively. Table 4.8 contains the starting and finishing operation for each value of resource *m*.

The eight constraint (Equation 4.8) has the same functionality as the seventh constraint (Equation 4.7), however, this constraint is added separately for the carousel (*m=11)*. The constraint ensures the completion time of preceding jobs on the carousel are before the start time of Operation 4 or Operation 7 of the job.

We added the ninth constraint (Equation 4.9) to ensure the earliness window is lower than or equal to the earliness tolerance for a job. The tenth constraint (Equation 4.10) ensures the additional waiting time for an operation is lower than or equal to the maximum additional waiting time for an operation. The eleventh constraint (Equation

4.11) ensures the completion time, earliness window, waiting time, tardiness, and earliness variables are equal or greater than 0. The twelfth constraint (Equation 4.12) ensures the variable Y_{ijm} can either be 0 or 1 (binary).

4.1.5 Parameters for the MILP model

In Section 2.2.1 we described the different costs attained with earliness and tardiness. For competitive reasons we do not publish the actual costs associated with earliness and tardiness, therefore we transformed the actual costs into factors. The factors differ for each product type and the press it is produced on. However, the differences between the costs per product on the same press are minimal. Therefore, we take the average of the product cost on the same press to simplify the model. Table 4.7 contains the earliness and tardiness factors for each job processed on a press.

Table 4.7: Earliness and Tardiness Factors

Press		**Earliness factor e_i**	**Tardiness factor t_i**
Press 1	*Top-layer*	1.4	4.0
	Bottom-layer	1.7	4.0
Press 2	*Top-layer*	1.5	5.0
	Bottom-layer	1.8	5.0
Press 3	*Top-layer*	1.4	3.8
	Bottom-layer	1.7	3.8
Press 5	*Top-layer*	1.0	4.5
	Bottom-layer	2.0	4.5

Table 4.8 contains the starting (R_m) and finishing (T_m) operations number given resource number m. The starting and finishing operations number are used in Equation 4.7 of the MILP model in order to specify the starting times, completion times, and processing times of the operation given resource m.

Table 4.8: Starting and Finishing Operations of a Resource Given the Resource Number

Resource m	R_m	T_m
1-5	1	2
6-10	2	8
11-15	4	7

4.1.6 MILP Solving

Solving a MILP can be done using a solver, such as CPLEX, which is a software application that tries to find the optimal value. However, as the scheduling problem is NP-hard it may take a long time before the software finds an optimal solution. This depends on the CPU power and problem size. The problem size increases when more variables or constraints are added. When the problem size increases the computation time may increase exponentially. Therefore, the problem size must be kept small as the scheduling problem has to be solved within a few minutes during production time (see Section 4.3.1). When it is not solved within a few minutes the scheduling problem might be changed and the process starts over again. This may result in not finding a schedule at all. Therefore, we limit the computation time of the MILP method. If the computation time is exceeded, we add all non-fixed jobs to the schedule one-by-one ranked by the Earliest Due Date (EDD) dispatch rule. We do this in order to certainly retrieve a feasible schedule. As jobs are added with the EDD rule, the jobs with the highest priority are scheduled first and have the highest flexibility in starting time. This method may lead to poorer objective values, however, it is required in order to always obtain feasible schedules.

4.2 Order Prediction Model

In Section 2.2.1 we described two modes of ordering material, either manually or automatically. Automatic orders are created using a sensor that is triggered when the material in the silo drops below a certain level. As the sensor is located high in the silo, a time delay between the trigger and the actual ordering of the mix is created to ensure orders are not delivered too early. Within this time window many things can happen that provide new information for the ordering of the material, such as a breakdown of the machine. However, in the current system, the mix will still be ordered. Also, the ordering of mixtures using a trigger can result in schedules with waiting time. To be able to improve scheduling and reduce waiting time, orders are required to be known more in advance (before the trigger). Therefore, we create a model that predicts orders within a specified planning horizon. The order prediction model forecasts the moments when the silo is expected to run empty.

Equation 4.13 shows the formula for calculating the remaining processing time for the contents in silo, where A is the weight of delivered mix, *w* is the weight of the mix used per tile, *t* is the time of one production cycle and *n* is the number of tiles produced per production cycle. For example, when a bottom-layer mix arrives of $A = 2{,}200\ kg$ and in each tile $w = 8.0\ kg$ of the mixture is used. Each production cycle takes $t = 5.5\ seconds$ and has $n = 2\ tiles/cycle$, then the total processing time before a mix is fully used by the press $p = \frac{2{,}200}{8.0*2} \times 5.5 = 756{,}25\ seconds$, or approximately *12.6 minutes*.

$$p = \frac{A}{wn} t \qquad 4.13$$

Figure 4.1 shows an example of the silo level with the actual level from *t=0* to *t=3* and a prediction after *t=3*. The prediction is based on Equation 4.13, where A is the current level in the silo. The figure shows the estimates of the due date; each vertical line represents a replenishment at the due date. The next due date is calculated in the same way with the starting point at the moment of replenishment. The silo level represents how much material the silo contains.

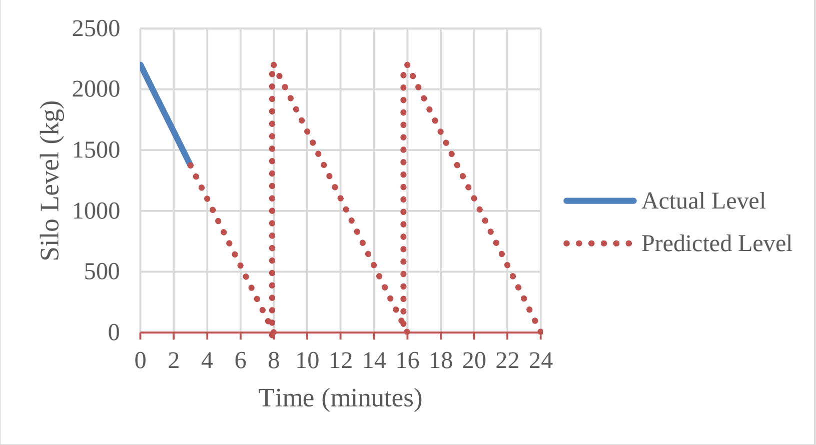


Figure 4.1: Prediction of Silo Level

Table 4.9 shows the jobs that are created using this model with their due date, earliness tolerance, and early date. We only select the jobs of which the due date is within the planning horizon *h*. If the planning horizon is (for example) 20 minutes, only Job 1 and Job 2 are selected; their due date is lower than 23, which is the sum of the current time *(t=3)* and planning horizon *(h=20)*. We do not include jobs of which the due date is outside the planning horizon, while the early date is within the planning horizon. This would create too many jobs in the planning horizon and we want to ensure not too many jobs are selected (see Section 4.3.1). When the planning horizon is longer this increases the uncertainty of the due dates as many events may occur in the meantime. In Chapter 5 we experiment with several sizes of planning horizons. At each interval or event, the job list is updated (see Section 4.3).

Table 4.9: Jobs Created From Prediction Model

Job	d_i *(min.)*	u_i *(min.)*	e_i *(min.)*
Job 1	12	15	0
Job 2	20	15	5
Job 3	28	15	13

Throughout the day new jobs keep arriving, therefore the problem and its sub-problems require an update when a new job arrives. The job is added to the current set of jobs with the corresponding due date and early date. This is done when a new planning cycle starts, we describe the planning cycle in Section 4.3.1.

4.3 Rescheduling

The scheduling method proposed in Section 4.1 creates schedules for jobs within a given planning horizon. However, during a day new jobs arrive and the jobs in the planning horizon are updated. Therefore, rescheduling is required. When new jobs arrive, breakdowns occur or delays arise, schedules may become infeasible or unattractive; better schedules may exist that improve the performance of the system. In this section propose rescheduling methods to cope with these situations. Section 4.3.1 contains the planning cycle in which parameters are updated and rescheduling is done. Section 4.3.2 contains events that may trigger a new planning cycle. Finally, Section 4.3.3 describes methods for improving the robustness of schedules.

4.3.1 Planning Cycle

During a day new jobs arrive to be scheduled. In order to handle these jobs, we introduce a planning cycle for rescheduling. The process of a planning cycle consists of five steps (see Figure 4.2), namely:

1. *New Planning Cycle*

   When a new planning cycle starts, all required data for the planning cycle is retrieved, the input parameters of the planning cycle are updated, and the start time and end time of the fixed zone (see Step 2 of the planning cycle) and planning horizon are updated.

2. *Fix Jobs in Fixed Zone*

   Jobs that are scheduled during previous cycle and whose start time fall within the new fixed zone (see Figure 4.3) are fixed. When a job is fixed, the start times of operations are fixed to the value obtained during the previous planning cycle. This ensures the job is not rescheduled during this planning cycle. This is done for multiple reasons. First, the start time of a job may be before the new schedule is created and thus the job might not start on time when it would not be fixed. Second, it reduces the computation time of scheduling as less jobs are considered for rescheduling. Finally, it makes the schedule more stable as the jobs are known to be fixed more in advance. The fixed job is still used in the schedule to prevent resources from being used by other operations at already planned moments. The longer the fixed zone, the worse the performance of the schedule will be as schedules are not as flexible anymore. Thus for determining the length of the fixed zone, a trade-off is required between computation speed and stability of the schedule on one side and the performance of the schedule on the other side.

3. *Add New Jobs and Update Parameters*

   When a new planning cycle is triggered, new jobs are generated according to the order prediction model (see Section 4.2). All new jobs are added whose due date fall within the new planning horizon (see Figure 4.3). This is a fixed time window in which jobs are taken into account and shifts forward every planning cycle. The planning horizon is limited in order to keep the problem size small; if the planning horizon is too large, too many jobs are added and the problem size becomes too large. The planning horizon is also limited as the due date becomes more uncertain when it is further in the future. When adding a new job the release time r_i is set to the current time to ensure the job is not scheduled before the planning cycle started.

4. *Reschedule*

   Rescheduling is done using the fixed jobs, newly added jobs, and updated parameters. This is the new input for the MILP. The MILP is solved using CPLEX (see Section 4.1.6) or using EDD (when no solution is found within a set computation time). The output of the CPLEX or EDD is used as the new schedule.

5. *Wait for Next Interval*

   A new planning cycle will start as soon as the interval has ended. The interval is a fixed amount of time after which a new planning cycle is triggered. The interval between planning cycles must be large enough to ensure the MILP method has finished computing the schedule for the previous cycle and is small enough in order to trigger enough rescheduling moments.

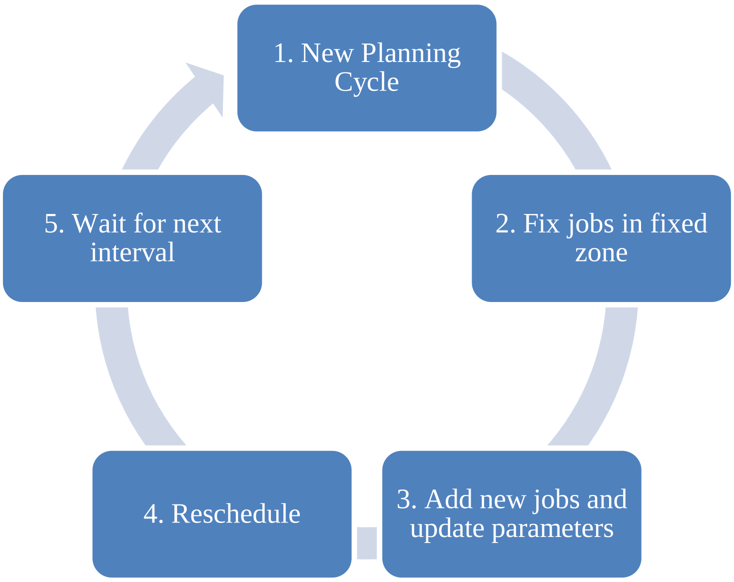


Figure 4.2: Planning Cycle for Rescheduling

Figure 4.3 shows a graph of the planning cycle with the fixed zone and planning horizon. Also three jobs are shown in the graph, namely: job *a*, *b*, and *c*. The start time of job *a* falls within the fixed zone and is therefore fixed. The due date of job *b* falls within the planning horizon and is therefore taken into account during the planning cycle. The due date of job *c* does not fall within the planning horizon and is therefore not considered during the planning cycle. It will be considered during a planning cycle in the future.

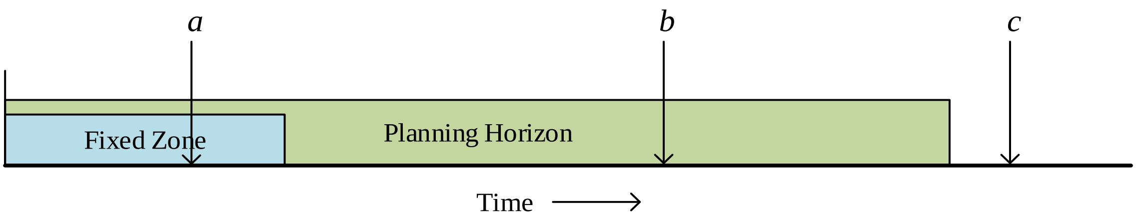


Figure 4.3: Planning Horizon and Fixed Zone

4.3.2 Events

During a production day events may occur that may cause a rescheduling. However, these events do not always require immediate rescheduling. The events can also be taken into account during the next planning cycle. Whether events require immediately rescheduling or are taken into account in the next planning cycle is assessed by the type of event and the impact of the event. The following events may have an impact on the schedule and thus trigger an assessment:

1. *Manual creation of a priority job.*
   When a job is manually added by an operator and it has a high priority it immediately triggers a new planning cycle.
2. *Delay in an operation.*
   When an operation is delayed due to any cause, it may disrupt the current planning. Therefore, the delays in operations should be monitored. If the delay becomes larger than a specified value a rescheduling is triggered. This value has to be determined with a simulation study.
3. *Breakdown of a press.*
   When a press has a breakdown, no material is being used. Therefore, all due dates of jobs on that press shift to the right exactly the amount of time the breakdown occurs.
4. *Breakdown of a mixer or transporter.*
   When a mixer or transporter breaks down, the presses that are supplied by this mixer or transporter cannot get mixtures from these resources. When the breakdown occurs, the jobs assigned for these presses need to be removed from the schedule in order not to block resources for other jobs. When the breakdown takes a long time, an operator may want to switch to another mixer and transporter. This decision requires the expert opinion of the operator. Therefore, affected jobs are removed from the scheduling problem until the mixer or transporter become available again or the operator changes to another mixer or transporter.

An assessment is required if the event is important enough to trigger immediate rescheduling. For example, if a delay occurs that does not affect the other jobs in the schedule, the event is not important enough to trigger immediate rescheduling and will wait for the next planning cycle. This ensures that not too much rescheduling is done and stabilizes the scheduling. However, when a delay occurs that does affect other jobs in the schedule, rescheduling is required in order to hold the schedule feasible.

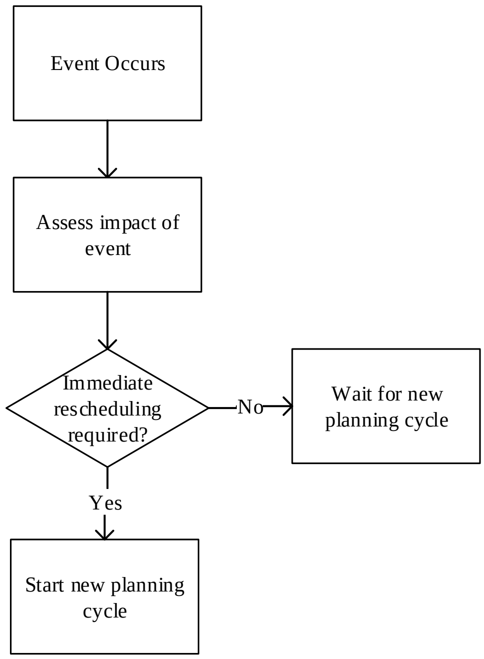


Figure 4.4: Rescheduling Procedure When an Event Occurs

As too many events within a short period may trigger rescheduling, events can be taken into account during the next planning cycle. Figure 4.4 shows the rescheduling procedure that is triggered when an event occurs. When one of the above events occurs the impact of the event on the current schedule is assessed. When the impact on the schedule is significant, it immediately triggers a new planning cycle (as described in Section 4.3.1). However, when the impact is not significant, it will not immediately trigger a new planning cycle. When a new planning cycle starts, the parameters are updated with the new information and rescheduling occurs. A simulation study is required in order to test the impact and significance of an event. However, this research is limited in time and we therefore choose not to perform a simulation study. Events are therefore not taken into account in our research.

4.3.3 Robustness

In Section 3.3.3 we discussed the robustness of schedules as described in literature. Making schedules more robust may reduce the amount of events that trigger rescheduling. To increase the robustness of the schedules we propose the following methods:

1. *Adding slack time to an operation.*
   By adding slack time to an operation the processing time of that operation may be delayed by as much as the slack time without causing problems for the schedule. The amount of slack added can be a fixed amount per operation or a certain percentage of the processing time.
2. *Inserting idle time between operations on a machine*.
   By inserting idle time between operations on a machine, the operations are allowed to finish later on the machine or start earlier than scheduled.
3. *Tightening the due windows of jobs.*
   By tightening the due windows of a job, the MILP is less flexible in finding good schedules. However, when executing a schedule and a delay occurs, the schedule is more robust as a penalty will not occur as quickly compared to the initial due window. The tightening of the due windows can be done by lowering the tolerance of the earliness on the left side and lowering the due date on the right side. Lowering the earliness tolerance will result in more robust schedule when the press the job is scheduled for breaks down. Lowering the due date will result in a more robust schedule when operations are delayed that have impact on the job.

To determine the optimal values for robustness a simulation study or real-time experiments are required. This allows to test the impact of robustness on the performance of the model. The simulation or experiments for robustness are not in the scope of our research as this requires too much research time. However, additional research can be done by another party to determine the optimal values and further improve the solution.

4.4 Conclusions

In this chapter we answered research question 3 by using MILP and rescheduling methods from literature and adapt them for the TWET problem. We transformed the practical situation at SVI in a model for creating feasible schedules. This was done by excluding unnecessary operations and machines. We also combined operations adjacent to each other when the resource requirements where equal. We adapted the MILP formulation by Zhu & Heady (2000) to match our model. We introduced a model for predicting orders within a specific planning horizon for machines. The model is based on the level of the silo and calculates the due date of upcoming orders when there is no machine breakdown. We described a rescheduling method and introduced the planning cycle. The planning cycle is triggered every time interval or when an event occurs that has a significant impact on the current schedule. Jobs that fall within a fixed zone are not considered for rescheduling. Finally, we discussed methods for increasing the robustness of the schedule by adding slack to processing times of operations, inserting idle time between operations on a machine, or tightening due windows of jobs. Chapter 5 contains experimental design and results with different parameters for providing the best results for the situation at SVI and test the solution for different scenarios at SVI.

Chapter 5: Experiments

This chapter contains the experimental design and results for our proposed planning algorithm from Chapter 4. First, Section 5.1 discusses the general setup for experimentation with our proposed solution. Next, Section 5.2 describes the experimental design and results for obtaining the best parameter settings for the planning cycle. Section 5.3 contains multiple experiments for comparing our solution with the current situation. Finally, Section 5.4 contains our conclusions from the experiments.

5.1 General Setup

We use an Intel(R) Core(TM) i5-5200 CPU with 2.20 GHz and internal memory of 8.00 GB on a Windows 10 Home x64 system. The CPU has an impact on the speed of solving the problems. A faster CPU would obtain solutions quicker. We model our planning cycle experiments in Microsoft Office Excel. In Excel each parameter of the model can be adjusted. For each experiment, multiple iterations of the planning cycle are done. During each iteration Excel generates the scheduling problem that needs to be solved. Excel then sends the problem to AIMMS, which is a software application that can solve MILP problems. AIMSS uses the CPLEX algorithm from IBM for obtaining solutions for the scheduling problem. When AIMMS has determined the optimal solution it will send it to Excel.

The system setup we use mostly finds a good feasible solution within a computation time of just seconds. However, as discussed in Section 4.1.6, for some scheduling problems it may take a long time before an optimal solution is found; when the problem size grows, the computation time may grow exponentially. The computation time must always be lower than the planning interval (see Section 4.3.1); when the computation time exceeds the interval, a new planning cycle may already start while the current one is not finished yet. In order to always obtain a feasible solution for the planning cycle, we limit the computation time of CPLEX. We choose a limit for the computation time of 1 minute. We choose 1 minute as after this computation time a new planning cycle may start and the computation could be interrupted. When no optimal solution is found after 1 minute, we add the non-fixed jobs one-by-one using the planning algorithm (see Section 4.1.6) for finding a feasible solution within short computation time.

Figure 5.1 shows how we use the planning cycle in experiments. Each iteration a planning cycle is performed. After an interval has elapsed a new iteration, and thus new planning cycle, will start. The iterations continue until the experiment is finished (when the time limit of the experiment is reached). Each planning cycle takes an unknown time to compute. Therefore, we set a time limit for the computation time of each planning cycle (see Section 5.2.2). After the planning cycle is computed, the parameter values for the scheduling problem may have changed. For example, when a delay has occurred at a press the due date of a job may have been shifted to the right. However, in order to keep the system operational, we use the obtained schedule and take the information of this delay into account during the next planning cycle. Once the planning cycle iteration is finished, it results in new jobs being fixed (see Section 4.3.1). The fixed jobs are added to the operational schedule being executed.

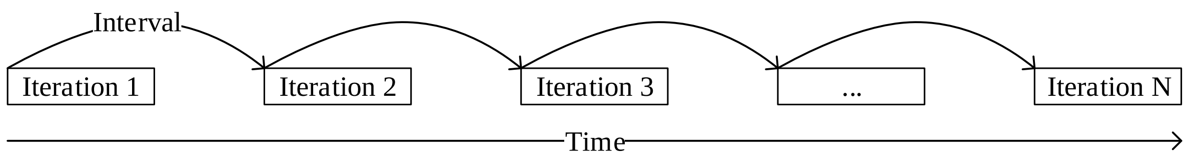


Figure 5.1: Iterations of the Planning Cycle During an Experiment

5.2 Experimental Design for Planning Cycle Parameters

This section contains the experimental design for the planning cycle parameters. The planning cycle consists of 3 parameters, namely: the planning horizon, planning interval, and fixed zone (see Section 5.2.2 for a further description of these parameters). We use an experimental design in order to determine the optimal values for the parameters. Obtaining the optimal values for the parameters is important, as a change in parameters may improve the objective value and reduce the required computation time. By using a good experimental design, we can reduce the number of experiments required to find the best values for the parameters.

We use two performance indicators for the planning cycle, namely: the TWET objective of all jobs in the experiment and the average computation time of one planning cycle. We perform experiments over a period of time in which multiple planning cycle iterations are done. To calculate the TWET objective value, we sum the TWET objective values of all fixed jobs. These objective values can be used to compare different experiments. We also compare the average computation time required for each planning cycle for all experiments.

In order to compare the outcomes of the experiments we assign weights to the objective value and the computation time. We do this in order to work with the same planning cycle parameters for all experiments. Therefore, we evaluate the importance of the objective value opposed to the computation time. As the focus of this research lies on minimising the objective value we weigh it significantly more important than the computation time. However, we also consider the computation time fairly important as the solution needs to work within short computation time during production. Therefore, we give the objective value a weight of 3 and the computation time a weight of 1. Section 5.2.3 contains a sensitivity analysis for these weights. We select the parameters that result in a good objective value and computation time. SVI may choose to use different parameters when they value the objective value or computation time at different weights.

5.2.1 Setup of Experiments

The production setup at SVI can vary in many different ways. Experimenting with all different setups would result in a large number of experiments. In order to reduce the number of experiments for determining the planning cycle parameters, we choose an extreme production setup for experimentation. The most extreme realistic setup is when all machines produce their largest tile possible (see Table 5.1). We assume that the parameters resulting from the experiments with these settings also produce good quality schedules with scenarios that are less extreme and require less material. Table 5.1 also contains the assignment of mixers for the top-layer and bottom-layer for each press. We use this assignment as Press 1 and 2 share the same rail section. This minimises the conflicts of transporters to use the rail section at the same time. For this experiment Press 5 has the highest material requirement, while Press 3 has the lowest material requirement. Therefore, Press 3 and 5 are using the same bottom-layer mixer and top-layer mixer.

Table 5.1: General Settings for Planning Cycle Parameter Experiments

Press	Tile Height (mm)	Orders top-layer at	Order bottom-layer at
Press 1	80	Mixer 1	Mixer 4
Press 2	80	Mixer 2	Mixer 4
Press 3	50	Mixer 3	Mixer 5
Press 5	80	Mixer 3	Mixer 5

In order to determine the optimal values for the parameters we make use of a 2^k factorial design. A 2^k factorial design allows measuring the effects and interactions of factors by using only 2 values for each factor (Law, 2007). A low and a high value are chosen for the 2 values for each factor. This reduces the number of experiments compared to experimenting with, for example, 5 values per factor and reduces the time it takes to run all experiments. Our experimental design consists of 3 parameters, namely: the planning horizon, planning interval, and fixed zone (see Section 5.2.2 for further description of the parameters). The experiments consist of the 8 experiments for the low and high values and a 9th experiment for the centerpoint. The centerpoint is an extra experiment in order to verify whether the regression model obtained from the experiment is linear or contains curvature (Law, 2007). The centerpoint is added by using the middle values for each factor, namely: a planning horizon of 25 minutes, a fixed zone of 200 seconds, and a planning interval of 200 seconds. Curvature means the regression model for the experiments cannot be described with a linear model only and quadratic (or higher-order) effects are present. If curvature is significant, an extra Central Composite Design (CDC) is required to test for possible quadratic effects. The CDC is added by created by adding extra design points. The new design points lie on a 'rotated surface' between the original design points.

Table 5.2 shows the experiments for all combinations of levels for the factors. The '-' sign represents the low value, the '+' sign represents the high value for a factor, and the 'o' sign represents the middle value for a factor. The response for an experiment *n* is given by R_n.

Table 5.2: Design Matrix for a 2^3 Factorial Design (Law, 2007)

Experiment	Planning Horizon	Fixed Zone	Planning Interval	Response
Experiment 1	-	-	-	R_1
Experiment 2	+	-	-	R_2
Experiment 3	-	+	-	R_3
Experiment 4	+	+	-	R_4
Experiment 5	-	-	+	R_5
Experiment 6	+	-	+	R_6
Experiment 7	-	+	+	R_7
Experiment 8	+	+	+	R_8
Experiment 9	o	o	o	R_9

5.2.2 Parameters

In Section 4.3.1 we explained the planning cycle and the required parameters for setting up the planning cycle, namely: the planning horizon, fixed zone, and planning interval. The parameters have an impact on the objective value and time for computing an optimal schedule. We define the limits for the parameters as follows:

1. *Planning horizon*

   We choose a minimum planning horizon of 20 minutes and a maximum of 30 minutes; from several single experiments we concluded that a planning horizon lower than 20 minutes results in a poor objective value. A large planning horizon results in jobs being taken into account with high uncertainty; the further away a job in the planning cycle, the more uncertain the due date is. We arbitrarily choose a maximum planning horizon of 30 minutes. We performed a few short experiments with longer planning horizons and concluded that using longer planning horizons did not give much better objective values while the computation time increased largely.

2. *Fixed zone*

   We choose a minimum fixed zone of 100 seconds and a maximum of 300 seconds. We choose a minimum of 100 seconds, a lower fixed zone would risk jobs being fixed too late. We arbitrarily chose a maximum for the fixed zone of 300 seconds. Choosing a higher maximum results in jobs being fixed too early and lowers the flexibility of the planning cycle. It would result in poorer schedules.

3. *Planning interval*

   We choose a planning interval with a minimum of 100 seconds and a maximum of 300 seconds. We choose a minimum of 100 for the planning interval as selecting a lower interval may cause conflicts with the previous planning cycle. A lower planning interval would result in more often updates of the schedule, however, we choose not to lower this as the minimum computation time and a slack for the backup heuristic is required. A larger interval than 300 seconds would result in poor objective values as the schedule is not updated as often.

Table 5.3 contains the parameters with low and high values as described above. These parameters are used for the experiments of our 2^k factorial design (see Section 5.2.3).

Table 5.3: Factors for 2^k Factorial Design

Factor	**Low**	**High**
Planning horizon	20 minutes	30 minutes
Planning interval	100 seconds	300 seconds
Fixed zone	100 seconds	300 seconds

5.2.3 Experiments and Results

Table 5.5.4 contains the resulting experiments for the 2^k factorial design (Experiment 1-8) and a centerpoint (Experiment 9). We use Minitab in order to analyse the model and test for curvature. Curvature is significant when the P-value for curvature is below 10% (Law, 2007). The analysis using Minitab shows a P-value of 6.3%, which means curvature is significant and a quadratic (or higher-order) effect is present. Therefore, we consider a central composite design (CCD) to ‘rotate’ the initial experiment and test the model for quadratic effects (see Section 5.2.1. This results in adding Experiments 10-16 in Table 5.5.4.

Table 5.5.4: Experiments for 2^k Factorial Design

Experiments	**Planning Horizon**	**Fixed Zone**	**Planning Interval**	**Objective Value**	**Computation Time**
Experiment 1	20	100	100	4,395.8	3.96
Experiment 2	30	100	100	635.1	6.51
Experiment 3	20	300	100	6,257.7	2.69
Experiment 4	30	300	100	891.9	14.13
Experiment 5	20	100	300	7,309.2	5.33
Experiment 6	30	100	300	572.7	26.92
Experiment 7	20	300	300	20,276.2	4.58
Experiment 8	30	300	300	1,401.4	15.04
Experiment 9	25	200	200	931.9	8.69
Experiment 10	25	200	100	826.4	6.03
Experiment 11	25	200	300	1,195.2	8.17
Experiment 12	25	100	200	984.8	9.94
Experiment 13	25	300	200	2,081.5	8.28
Experiment 14	20	200	200	15,971.3	4.94
Experiment 15	30	200	200	882.4	19.89
Experiment 16	25	200	200	2,1134	9.00

All experiments from Table 5.5.4 are analysed using Minitab. We do not take the objective values of Experiments 7 and 14 into account for the analysis as these are marked as outliers by Minitab. The outliers mean that the values are far away from the expected results. The reason these objective values are so high, is for some planning cycle iterations it was very difficult to find solutions for these experiments. The results may not be actual outliers but be actual representative values for the experiments. However, in order to determine good parameters for the planning cycle quickly, we mark these results as outliers; we do not take these experiments into account for the model for determining the values for the parameters. This may result in parameters that are not optimal for the system. However, we want to find fairly good values for the planning cycle quickly and take the risk of not having optimal values.

Using Minitab we perform an analysis on the significance of the interaction effects and the factors on the objective value. Interaction effects are the combined effect of two or more factors on the outcome of the experiment. From our analysis using Minitab we conclude that neither the interaction effects of the planning interval with the fixed zone, nor the interaction effect of the planning horizon with the fixed zone are significant. The quadratic effects of the planning interval and the fixed zone are both not significant. Therefore, we eliminate the non-significant effects from the model. Appendix D contains more detailed results on the significance tests using Minitab. The resulting factors are the Planning Horizon, Planning Interval, Fixed Zone, the quadratic effect of the Planning Horizon, and the interaction effect of the Planning Horizon and the Planning Interval. We use Minitab to study the effects of the parameters on the objective value and the computation time. Figure 5.2 shows the effect of the factors on the objective value. Figure 5.3 shows the effect of the factors on the computation time for each iteration. We conclude that choosing a low planning interval is beneficial for both the computation time and the objective value. The higher the fixed zone the lower the computation time and the higher the objective value. The higher the planning horizon the higher the computation time and the lower the objective value.

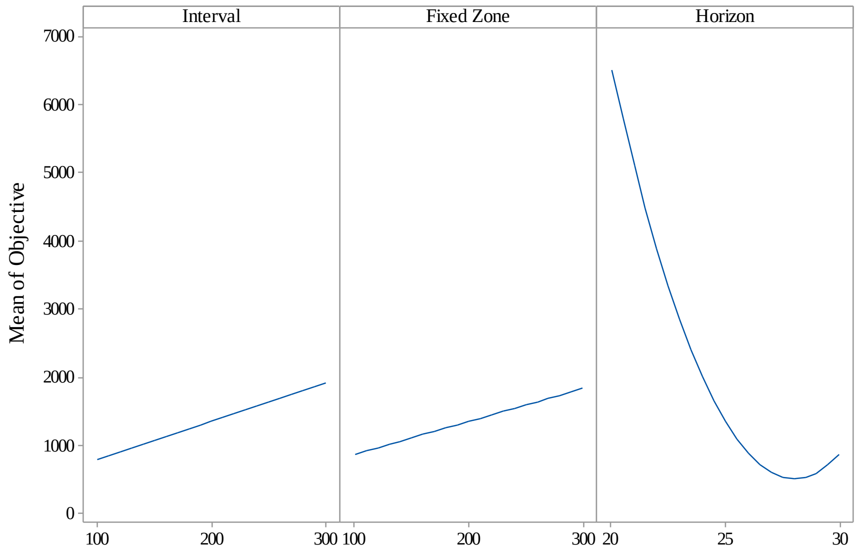


Figure 5.2: Effect of Factors on Objective Value

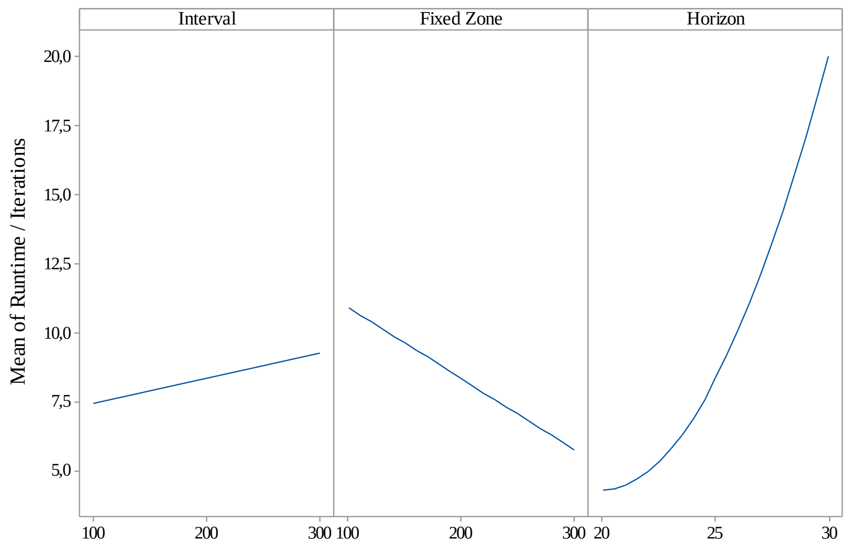


Figure 5.3: Effect of Factors on Computation Time

We use Minitab to calculate the optimal values for the parameters. We assign a weight factor of 3 to the objective value and a weight factor of 1 to the computation time. The resulting optimal values for the parameters are a planning interval of 100 seconds, a fixed zone of 247 seconds and a planning horizon of 26.36 minutes. The resulted value for the planning interval lies at the set minimum of 100 seconds. The actual optimal value might be lower than 100 seconds, however, we did not test this in the experiments. We do not consider a planning interval lower than 100 seconds as this may result in conflicts between planning cycles as these may overlap when the computation time is large.

In order to test the effect of the weights on the values for the planning cycle parameters we perform a sensitivity analysis. Table 5.5 contains the results of the sensitivity analysis of the weights of the objective value and computation time. The weights are changed in order to test the impact of the weights on the results in Minitab. The sensitivity analysis shows the interval does not change when the weight of the objective is changed. The planning horizon changes slightly but not significantly. The value for the fixed zone is impacted the most when changing the weight for the objective value. However, the resulting value for the fixed zone only changes slightly. When the weight for the objective value is set to 2.0 the fixed zone is changed the most (to 300 seconds). However, in order to keep the estimated computation time low, we keep the weights we initially used for our analysis. We round the values we initially obtained to a planning interval of 100 seconds, a fixed zone of 250 seconds, and a planning horizon of 26.5 minutes

Table 5.5: Sensitivity Analysis Weight Factors

Run	**Weight Objective**	**Weight Comp. Time**	**Interval (sec.)**	**Fixed Zone (sec.)**	**Planning Horizon (min.)**
1	2.0	1.0	100	300	26.57
2	2.5	1.0	100	237.37	26.36
3	3.0	1.0	100	247	26.36
4	3.5	1.0	100	245.45	26.16
5	4.0	1.0	100	253.54	26.16

5.2.4 Conclusion

We used a 2^k factorial design to test the effects of the parameters on the objective value and computation time per iteration required for the planning cycle. We added a centerpoint to test whether quadratic effects where present. This resulted in the use of a Central Composite Design (CCD) with additional experiments. We used Minitab to analyse the results of all experiments and concluded the effects of the planning horizon, fixed zone, and planning interval where significant on the objective value of the planning cycle. As well as the interaction effect of the planning interval with the planning horizon and the quadratic effect of the planning horizon. We used the resulting model to determine the optimal values for the parameters, given a weight factor of 3 for the objective value and a weight factor of 1 for the computation time. This resulted in a planning interval of 100 seconds, a fixed zone of 250 seconds, and a planning horizon of 26.5 minutes.

5.3 Experimental Design for the Performance of the Planning Algorithm

In Section 5.2 we obtained the optimal values for the planning cycle parameters. This section contains the experimental design for assessing the performance of the planning cycle. We divide the assessment of the performance into three categories, namely: comparison with current situation, effect of mixer assignment and tile height, and additional scenarios. First, Section 5.3.1 describes the general setup and settings of the experiments. Next, Section 5.3.2 contains the experiment for comparing the planning cycle performance to the current situation.

Section 5.3.3 describes the experiment for assessing the effects of mixer assignment and the tile height on the performance of the planning cycle. Finally, Section 5.3.4 contains an additional experiment towards the effects of the breakdown of a mixer.

5.3.1 General Setup

For running the different experiment described in Sections 5.3.2-5.3.4 we use the following general setup and assumptions:

1. *Production starts at 6 a.m. and ends at 4 p.m.*

   We assume in each experiment that the production starts at 6 a.m. This means all machines require the input material at 6 a.m. Therefore, the due date of all the first mixtures is set to 6 a.m. The mixing department may start mixing at 5.45 a.m. The production ends at 16 p.m. After 4 p.m. no new mixtures are ordered. Therefore, the runtime for each experiment is 10 hours.

2. *No breakdowns are considered*

   We do not take breakdowns of presses, mixers, or transporters into account for the experiments, except for the experiment in Section 5.3.4, where we intentionally want to assess the performance of the system when a mixer breaks down. We do not take the breakdowns into account, as these occur randomly and cannot be predicted. A simulation study with multiple replications would be required for each experiment in order to take breakdowns into account. Unfortunately, this is not possible within the research time of this project. Therefore, we leave out breakdowns and compute the performance of a schedule when no breakdowns occur.

3. *Production of the same batch throughout the day*

   For this research we assume that a press produces the same batch throughout the whole day. This means it produces the same tile size and orders at the same mixers during the whole day. Therefore, setups and changeovers are not taken into account throughout the day. In real production situations changeovers and setup time do occur on some days, however, in most situations presses produce the same tile height throughout the day. For testing the performance of our solution, we do not take changeovers and setup times into account.

5.3.2 Performance of the New Planning Algorithm Compared to the Current Situation for the Year 2015

To compare the performance of the planning algorithm with the current situation, we compare the performance of the planning algorithm with the performance of the current situation for the year 2015. In the experimentations we run, we do not take breakdowns into account. This means our experiments do not exactly represent the current situation. However, when a breakdown of the press occurs, the tardiness would become lower as the due date of a job shifts right. Therefore, if our solution performs well in terms of tardiness compared to the current situation, it would even perform better when we take breakdowns into account. However, breakdowns of mixers and transporters may have a negative effect on the tardiness. For example, when a mixer breaks down, deliveries from that mixture may become tardy. This may result in additional waiting time. We do not take these breakdowns into account as we do not have the data for the actual delay caused by breakdowns. By not taking these breakdowns into account, the actual performance of the planning cycle may be worse.

Each production day for the year 2015 has a different production setup. The current situation consists of many different setups and scenarios. As evaluating every possible scenario would take too much computation time we evaluate only a select number of scenarios that represent the current situations as much as possible. In order to lower the required number of experiments for testing the performance of our solution, we generalise the demand of a press into three categories (Table 5.6 contains an overview of the generalisation of the settings for the presses), namely:

1. *Off*

   When a press is not producing on a day we mark the demand of the press with 'Off'. The press will not issue any orders.

2. *Low*

   When the tile height at a press is 45-60 mm we set the demand of the press to 'low'. The press will issue orders for the top-layer and bottom-layer similar to when the press produces 50 mm tiles.

3. *High*

   When the tile height at a press is 70-80 mm we set the demand of the press to 'high'. Press 3 does not produce tiles of 70-80 mm so this setting is not applicable. When the demand of the press is set to high the press will issue orders for the top-layer and bottom-layer similar to when the press produces 80 mm tiles.

Table 5.6: Generalisation of Settings for the Presses

Press	**Off**	**Low**	**High**
Press 1	Press not in use	45-60 mm	70-80 mm
Press 2	Press not in use	45-60 mm	70-80 mm
Press 3	Press not in use	45-60 mm	N/A
Press 5	Press not in use	50-60 mm	70-80 m

We use the generalisation settings of Table 5.6 to create experiments for the year 2015. Table 5.7 shows the top 10 (in terms of occurrence of that setting) resulting experiments from the generalisation of the production days at SVI in the year 2015. Appendix B contains the entire table with all 28 setups for the year 2015. Table 5.7 lists the setup for each press and the frequency of days that this setup occurred in the year 2015. For each experiment the waiting time that occurred in the year 2015 for that setup and the waiting time that was obtained using the planning cycle multiplied by the frequency.

Table 5.7: Top 10 of Highest Frequency Setup in Year 2015

Experiment	Press 1	Press 2	Press 3	Press 5	Frequency	Current Situation (hrs.)	Planning Cycle (hrs.)
Experiment 1	Low	High	Off	Low	28	20.31	0
Experiment 2	Low	High	Off	Off	25	5.41	0
Experiment 3	High	Low	Off	Low	17	7.91	0
Experiment 4	Low	Low	Off	Low	16	7.97	0
Experiment 5	Low	High	Off	High	16	9.75	0
Experiment 6	High	High	Off	Low	16	9.91	0
Experiment 7	High	Low	Off	High	12	9.65	0
Experiment 8	Low	Low	Off	Off	12	4.52	0
Experiment 9	High	High	Off	Off	12	5.49	0
Experiment 10	Low	Low	On	Low	12	8.74	0

Table 5.8 shows the comparison of the current situation with the planning cycle solution as a result from the experiments. As there is no data available from the current situation of Press 3 (see Section 2.4.2), we show the total waiting time and total cost excluding Press 3. In order to get an indication of the possible extra cost including Press 3, we added an estimation of the total waiting time and cost including Press 3. We estimated the waiting time by taking the average relative waiting time (compared to the operational time) for top- and bottom-layer of Press 1, 2, and 5. The average relative waiting time is 2.5% and 3.4% for the top- and bottom layer respectively. This results in the total waiting time and total cost including the estimation of Press 3. Table 5.8 also shows the savings that result from the new solution using the planning cycle. The new planning cycle solution can result in a saving of 128.4 hours of waiting time excluding Press 3 or a saving of 137.2 hours including Press 3 for the year 2015. The savings of this waiting time result in a cost savings of € xx,xxx.xx excluding Press 3 or € xx,xxx.xx including Press 3 for the year 2015 (see Appendix B for a calculation of the cost savings for each press). The results from our new planning cycle solution do not contain the waiting times caused for other reasons, such as the breakdown of a mixer or transporter, while these breakdowns are present in the current situation. Therefore, the actual savings will be lower than our calculation. However, in the data from the OEE tool no difference is made between waiting time caused by breakdown and waiting time caused by inefficient planning. In order to determine the actual savings, a distinction should be made in the measurement system between the causes of waiting time. Section 6.2.2 contains different scenarios for the actual savings per year and the impact on the payback period for implementing the system.

Table 5.8: Comparison of the Current Situation with the New Solution for the Year 2015

Comparison	**Total Waiting Time (hours)**	**Total Cost (€)**
Current Situation (excl. Press 3)	128.7	€ xx,xxx.xx
Current Situation (incl. estimation Press 3)	137.5	€ xx,xxx.xx
New Planning Cycle	0.3	€ xx.xxx.xx
Savings (excl. Press 3)	***128.4***	***€ xx,xxx.xx***
Savings (incl. estimation Press 3)	***137.2***	***€ xx,xxx.xx***

5.3.3 Effects of Mixer Assignment and Tile Height on Performance of Planning Cycle

The decision of assigning a press to a top-layer and bottom-layer mixer has an impact on the performance of the planning cycle. SVI is interested in the effect of assigning different combinations of presses to mixers. In order to determine these effects, we experiment with multiple production setups and analyse the results. In order to keep the number of experiments low, we experiment with a low and a high tile height setting for the press. For each setting, we experiment with all combinations of assigning the presses to the mixers. This results in a list of combinations of mixer assignments that are unfavourable (given the tile height setup). Unfavourable combinations of assignments to the mixers may cause a lot of waiting time. This can support the management in the planning of daily production orders. Each possible combination receives a score based on the results from the experiments. However, when assigning a press to a mixer, the limits of using a certain mixer do still hold; the mixers are limited in the use of types of cements and dying colour (see Section 2.2.3). These limitations must be taken into account by the management when planning the production orders.

In order to reduce the number of experiments and be able to compare the results of the mixer assignment and tile height experiment we make the following assumptions:

1. *A press orders the mixture for top-layer and bottom-layer at the same mixers for the entire day*

   In Section 5.3.1 we described the assumption that we experiment with a batch run for the entire day. During the production of a batch the ordering of material at a mixer is assigned to the same mixer (see Section 2.2.1). Therefore, the press orders the mixtures at the same mixer for the entire day. This means the bottom-layer is ordered at the same mixer for the entire day and the top-layer is ordered at the same mixer for the entire day.

2. *A maximum of 2 presses are assigned to a mixer*

   We do not assign more than 2 presses to a mixer. We do this as there is enough capacity to have a maximum of 2 presses assigned to a mixer and reduce the number of combinations and thus experiments.

3. *As many top-layer mixers are used as possible*

   When 3 or more presses are in use, all top-layer mixers are used. The assignment of the presses to the mixers is done as evenly as possible. For example, when 4 presses are in use, all mixers supply 1 press and one of the mixers supplies 2 presses. This means that we eliminate the situation where 2 mixers both operate 2 mixers and one mixer does not supply any press at all.

4. *The type of mixer is not taken into account for the experiments*

   For the experiment we do not take the limitations of a mixer into account. For example, Mixer 3 is only able to produce mixtures with blast-furnace cement. However, for this experiment we do not take the limitations of the cement and dying colour into account. We are interested in the effects of assigning different combinations of presses to the mixers. Whether a combination is, for example, assigned to Mixer 1 or Mixer 2 is not relevant for the outcome of the experiment.

In Section 5.3.1 we described the requirements and setup for the different scenarios. This section contains a list of all possible scenarios for experimentation. Table 5.9 contains all 6 scenarios for the top-layer mixer assignment in the case that Press 3 is *on*. Table 5.10 contains 1 scenario for the top-layer mixer assignment in the case that Press 3 is *off*. Table 5.11 contains all 3 scenarios for the bottom-layer mixer assignment in the case that Press 3 is *on*. Table 5.12 contains all 3 scenarios for the bottom-layer mixer assignment in the case that Press 3 is *off*. Table 5.13 contains all scenarios for the mixer assignment. Each scenario consists of 6 experiments for the different tile heights. These experiment represent all combinations of the low/high demand options for Presses 1, 2, and 5.

Table 5.9: Scenarios for Top-Layer Mixer Assignment Given Press 3 is On

Scenario	**Press 1**	**Press 2**	**Press 3**	**Press 5**
A	M1	M2	M3	M1
B	M1	M2	M3	M2
C	M1	M2	M3	M3
D	M1	M1	M2	M3
E	M1	M2	M1	M3
F	M1	M2	M2	M3

Table 5.10: Scenarios for Top-Layer Mixer Assignment Given Press 3 is Off

Scenario	**Press 1**	**Press 2**	**Press 3**	**Press 5**
A	M1	M2	*Off*	M3

Table 5.11: Scenarios for Bottom-Layer Mixer Assignment Given Press 3 is On

Scenario	**Press 1**	**Press 2**	**Press 3**	**Press 5**
A	M4	M4	M5	M5
B	M4	M5	M5	M4
C	M4	M5	M4	M5

Table 5.12: Scenarios for Bottom-Layer Mixer Assignment Given Press 3 is Off

Scenario	**Press 1**	**Press 2**	**Press 3**	**Press 5**
A	M4	M4	*Off*	M5
B	M4	M5	*Off*	M4
C	M4	M5	*Off*	M5

Table 5.13: Experimental Setup for Mixer Assignment and Tile Height

Scenario	Press 3	Top-Layer	Bottom-Layer	Experiments
1	On	A	A	8
2	On	A	B	8
3	On	A	C	8
4	On	B	A	8
5	On	B	B	8
6	On	B	C	8
7	On	C	A	8
8	On	C	B	8
9	On	C	C	8
10	On	D	A	8
11	On	D	B	8
12	On	D	C	8
13	On	E	A	8
14	On	E	B	8
15	On	E	C	8
16	On	F	A	8
17	On	F	B	8
18	On	F	C	8
19	Off	A	A	8
20	Off	A	B	8
21	Off	A	C	8

We run a total of 126 experiments. For each experiment a different combination of assignments of presses to mixers are done. In order to reduce the time for running the experiments we run each experiment for 2 hours of production time. This means for each experiment 72 iterations are done of the planning cycle.

The experiments result in a list of scores that SVI can use for determining the best assignment of presses to a mixer for given tile heights each press produces. The lower the score the better the assignment of the presses to the mixers. Appendix C contains an example of the results for one tile height setup and the score for each possible combination of assigning the presses to the mixers. This combination may differ for each production setup. We conclude from all results that Press 5 should not be assigned in combination with Press 1 or 2 for the bottom-layer mixers when they are producing tiles with a large height (80 mm).

5.3.4 Effects of the Breakdown of a Mixer

During production a breakdown of a mixer may occur. This can result in not being able to use the mixer for an entire day. When this happens, a lot of waiting time may occur in the current situation. We experiment how our planning cycle would perform in such a situation. To run this experiment, we take a production day where this breakdown occurred and compare this situation with the results from our experiment. The day we simulate is the 25th of March 2016. Mixer 5 was broken down on this day. Press 1, 2, and 5 were operational and producing tiles with a height of 60 mm, 80 mm, and 50 mm respectively. We simulate this day using our planning cycle solution in order to determine whether our solution would lower the waiting time. We use the same mixing orders that are used on that day.

In the current situation the total waiting time for top- and bottom-layer was more than 2 hours. Using our planning cycle solution, no improvement is made in the waiting time. In order to determine the cause of the waiting time, we use a Gantt chart to visualise the resulting schedule from a planning cycle iteration (see Figure 5.4). The Gantt chart shows the scheduled window of processing each job for each resource. Mixer 5 has no jobs scheduled as the mixer is broken down. However, Mixer 4 is continuously processing jobs as it is the only mixer available for creating bottom-layer mixtures. Therefore, the capacity of the mixer is, in this case, the major cause for waiting time of the bottom-layer.

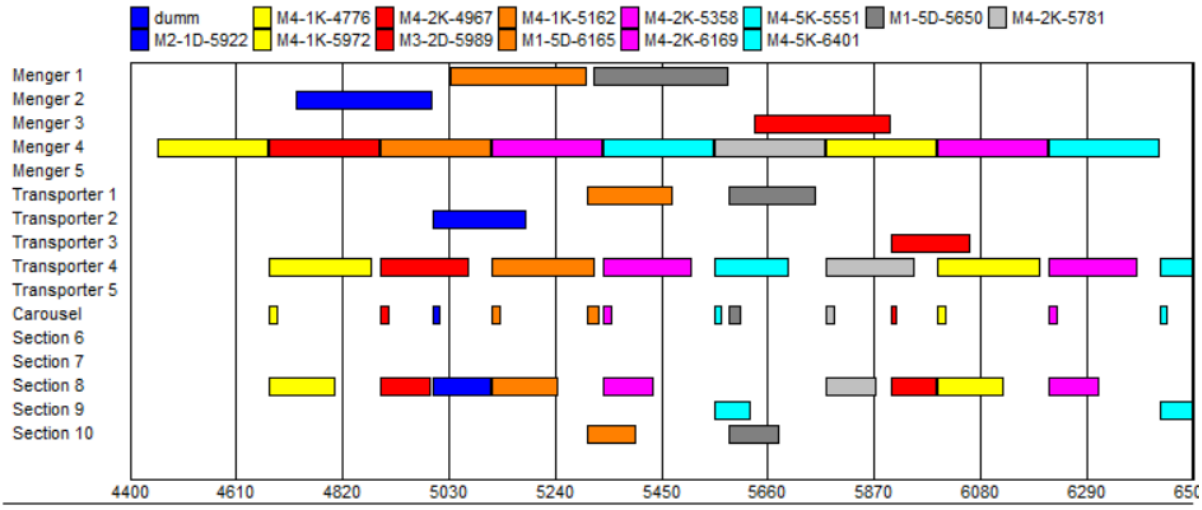


Figure 5.4: Gantt Chart of One Planning Cycle Iteration With the Breakdown of Mixer 5

SVI stated it may be possible to reduce the processing time of a job on the mixer. Therefore, it may be beneficent for SVI to perform a thorough analysis on the mixing process in order to reduce the processing time. In order to determine the effect of reducing the processing time, we perform two experiments. The current processing time of a job on Mixer 4 is 160 seconds. We perform experiments with a processing time of 145 and 130 seconds in order to analyse the effect of the processing time on the waiting time for material. Table 5.14 contains the results for the reduced processing time. This shows that by reducing the processing time on Mixer 4 by 15 seconds, the waiting time for raw materials at the press is reduced by 95.62%. By reducing the processing time even further to 130 seconds, the waiting time of almost 3 hours is reduced to just 5 seconds. This shows that reducing the processing time of Mixer 4 would largely reduce the waiting time for this scenario.

Table 5.14: Total Waiting Time with Reduced Processing Times for Mixer 4

Processing Time Mixer 4	Waiting Time (sec)	Reduction (%)
160 sec. (current situation)	10,672	-
145 sec.	467	95.62%
130 sec.	5	99.95%

5.4 Conclusions

In this chapter we performed multiple experiments in order to assess the performance of our planning cycle solution. First, we obtained the optimal parameters for the planning cycle algorithm using a 2^k factorial design. This resulted in a planning horizon of 26.5 minutes, a planning interval of 100 seconds, and a fixed zone of 250 seconds. We used the optimal parameters for assessing the performance of the new planning cycle solution. We assessed the performance using 3 experiments.

In the first experiment, we compared the planning cycle with the current situation by running experiments that represent the production days at SVI for the year 2015. This resulted in a savings of 128.4 hours of waiting time and € xx,xxx.xx in cost compared to the year 2015 excluding Press 3 and a savings of 137.5 hours of waiting time and € xx,xxx.xx in cost including an estimation of Press 3. The results from the planning cycle solution do not include the waiting time for breakdowns, while these are included in the current situation. Therefore, the actual savings would turn out lower. In order to determine the actual savings, a distinction should be made in the future in the measurement between the different causes of waiting time.

In the second experiment, we researched the effect of different mixer assignment and tile height setup for each press on the performance of the planning cycle solution. This resulted in a list of unfavorable combinations of mixer assignments given the tile height setup for a day (see Appendix C for an example of this list). This list can support the planning of production orders by the management for a day as it shows possible unfavorable order combinations. Finally, we concluded that Press 5 should never be combined with Press 1 or Press 2 on the same bottom-layer mixer when both are producing tiles with a tile height of 80 mm.

In the third experiment, we experimented with the breakdown of a mixer (Mixer 5). The breakdown of the mixer showed the planning of jobs was not the cause for the waiting time on the day Mixer 5 was broken down. The visual representation showed the capacity of Mixer 4 was the bottleneck. We experimented with the reduction of processing time of this mixer and concluded that the waiting time on this day could be reduced by 95.62% by reducing the processing time on Mixer 4 from 160 seconds to 145 seconds.

Chapter 6: Implementation & Cost Calculation

In Chapter 5 we described the experiments and results of using our new planning cycle solution at SVI. This chapter describes how the solution can be implemented at SVI and at what cost. First, Section 6.1 describes the implementation steps required for implementing the solution at SVI. Next, Section 6.2 contains a cost analysis for implementing the solution. Finally, Section 6.3 discusses the conclusions regarding the implementation of the solution.

6.1 Implementation Steps

This section contains the implementation steps required for implementing the solution at SVI. We collaborated with TMS, the supplier of the software for operating the mixing and transportation at SVI, for the requirements of implementing the solution at the plant of SVI in Dordrecht. TMS gave an indication on the hardware requirements, software requirements, and required investments. In this section we briefly describe the hardware and software requirements. Sections 6.2.1 contains the required investments for implementing the solution.

TMS advised in using laser measurement equipment at each silo at each press in order to get a more accurate and updated silo level for forecasting the remaining processing time of the contents of the silo. These laser measurements are to be installed by TMS and connected to the TMS system. This way the software can get real-time data of the silo level instead of just one measuring point as used in the current situation.

In order to use our planning cycle solution in real-time production, TMS needs to code the method into their software package instead of MS Office Excel. This allows direct communication with the hardware of the system. The software can then start new planning cycles and communicate with AIMMS for finding solutions to the scheduling problem, without the use of MS Office Excel.

6.2 Cost Analysis

This section contains an analysis of the required investment for implementing the solution (Section 6.2.1) and the payback period of the solution (Section 6.2.2).

6.2.1 Investment

This section contains the required investments for implementing the solution at SVI. The following investments are required for implementing the solution (as described in Section 6.1) and are estimated by TMS:

1. *Silo level measurement equipment - € xx,xxx.- to € xx,xxx.-*

   In order to get a real-time measurement of the silo a laser measurement needs to be installed in each silo. The total cost for installing these lasers is estimated between € xx,xxx.- and € xx,xxx.-.

2. *Software package update - € xx,xxx.- to € xx,xxx.-*

   To make the planning cycle work with the current system an update of the software package is required. The total cost for this update is estimated between € xx,xxx.- and € xx,xxx.-.

3. *Adding status signal of press to software - € xx,xxx.-*

   The software of TMS currently has no ability to check whether a machine is idle or broken down. This status needs to be added using connections between the presses and the TMS software. The estimated total cost for this is estimated at € xx,xxx.-.

The total investment costs required for the software update by TMS are estimated from € xx,xxx.- (best case scenario) to € xx,xxx.- (worst case scenario).

Besides an investment in the adaptions by TMS, an AIMMS license is required to be able to run the planning cycle. AIMMS estimates the licensing cost with CPLEX at single investment of € xx,xxx,-.

The total investment in the update by TMS and the licensing cost result in a total investment from € xx,xxx.- (best case scenario) to € xx,xxx.- (worst case scenario).

6.2.2 Payback Period

This section contains an analysis of the payback period given the investment (see Section 6.2.1) and benefit (see Section 5.3.2). We calculate the payback period by using a discount rate as provided by CRH plc, which is the holding company of SVI. The discount rate of CRH plc. is estimated at 10.15% (Gurufocus, 2016). Table 6.1 contains the best case and worst case scenario cost for implementing the solution at SVI that are provided by TMS (the supplier of the hardware and software for the mixing department at SVI).

Table 6.1: Required Investments for Implementing the Solution

Investment	**Best Case Scenario (€)**	**Worst Case Scenario (€)**
Silo Level Measurement Equipment	xx,xxx.xx	xx,xxx.xx
Software Update (TMS)	xx,xxx.xx	xx,xxx.xx
Signal Adaption	xx,xxx.xx	xx,xxx.xx
AIMMS licence	xx,xxx.xx	xx,xxx.xx
Total	**xx,xxx.xx**	**xx,xxx.xx**

Table 6.2 shows the payback period in years for different scenarios as a result from a discounted cash flow. For investments, the best case and worst case scenario are taken. For the savings realised each year, we use the scenarios where 25%, 50%, 75%, or 100% of the maximum savings possible are realised. This results in a payback period of 0.80 years (best case scenario and 100% of savings realised) up to 5.02 years (worst case scenario and 25% of savings realised).

Table 6.2: Calculation of Payback Period for Different Scenarios

Realised Savings (%)	**Savings per Year (€)**	**Payback Period Best Case (years)**	**Payback Period Worst Case (years)**
25%	€ xx,xxx.xx	3.61	5.02
50%	€ xx,xxx.xx	1.65	2.22
75%	€ xx,xxx.xx	1.06	1.43
100%	€ xx,xxx.xx	0.80	1.05

6.3 Implementing the Solution at other Plants

We designed the planning cycle solution specifically for the plant of SVI Dordrecht. However, the method may be generalised in order to be applied at other plants than SVI Dordrecht. These plants can be other plants of SVI, such as the plant in Amsterdam. However, also other customers of TMS may benefit from the generalised solution. Therefore, SVI may request a compensation when the solution is used at other plants or companies. This may reduce the total cost for implementing the planning algorithm.

6.4 Conclusions

Implementing the planning cycle solution at SVI requires the help of TMS (the software supplier for the mixing and transportation system). TMS estimated the hardware and software requirements for implementing the solution and the cost. The system requires additional laser measurement at each silo in order to measure the silo level in real time. In order to be able to function with the current system of TMS, the method that is now used in MS Office Excel needs to be recoded into the software package of TMS. The total investment is estimated between € xx,xxx.- (best case scenario) and € xx,xxx.- (worst case scenario). We provided the payback period for different scenarios and different actual savings as a result from implementing the planning algorithm. The payback period can vary from 0.80 years (best case scenario and 100% of savings realised) to 5.02 years (worst case scenario and only 25% of savings realised). The solution we provided may be generalised and be used by other customers of TMS. SVI may request a compensation if the solution is also applied elsewhere.

Chapter 7: Conclusions & Recommendations

The goal of this research is to minimise the waiting time for raw materials at the machines at the tile plant of SVI in Dordrecht by implementing an improved planning procedure for the mixing and transportation process (see Chapter 1). This chapter contains our conclusions (Section 7.1) and recommendations (Section 7.2) regarding this research. We also describe further research steps that can be taken (Section 7.3).

7.1 Conclusions

This research was initiated in order to improve the Overall Equipment Effectiveness (OEE) at the tile manufacturing plant of SVI in Dordrecht. The OEE is lowered by many causes of downtime that occur at the tile presses. One of these causes is the waiting for input material for the bottom- and top-layer of the tile. At the moment, SVI makes use of simple First Come First Serve (FCFS) dispatching rules for mixing and transportation orders (see Chapter 2). These simple rules do not take into account the orders' due date or other factors and may therefore cause starvation of material at the presses. Starvation of material leads to waiting time at the presses. We concluded the carousel and rail section 8 are bottlenecks in the system. Each transporter needs to cross the carousel and may therefore cause waiting time. Rail section 8 is required for the delivery of all mixtures to Press 1 and 2, and may therefore also cause waiting time for the transporters.

We use tardiness and earliness for measuring the performance of deliveries. Tardiness is the amount of time a delivery is done too late and earliness is the amount of time a delivery is done too early. We measure earliness, as an early delivery results in poor quality of the tiles. In order to measure the performance of the current system we used historical data from the OEE tool. Currently, the earliness of a delivery is not being measured. The OEE tool also has missing data for waiting times below 60 seconds and Press 3. The data also contains waiting times with other causes that cannot be distinguished from each other. We concluded that Press 5 has the most waiting time for material and that the tile height has a large impact on the waiting time for the bottom-layer.

In Chapter 3 we performed a literature research in order to find methods for optimising the planning procedure of the mixing and transportation of material. We concluded the Mixed Integer Linear Programming (MILP) approach has the highest accuracy for finding optimal solutions. In order to keep updating the schedule throughout the production day the rescheduling framework can be used. The performance of a schedule can be increased by adding robustness to schedules. Robustness can be added by, for example, tightening the due window or adding slack to processing times. The effect of adding robustness requires a simulation study with breakdowns, this study is outside the scope of this research.

In order to anticipate better on future orders, we propose an order prediction model (see Section 4.2) that predicts the due date of future orders. This is done by making use of the material usage by the press and the contents of the silo. The generated orders that fall within a specified planning horizon are considered for scheduling using the MILP method. To fit the MILP method to the situation at SVI, we changed the model to:

1. Use due windows instead of due dates.
2. Have additional waiting time added to the processing time for certain operations.
3. Have operations of a job start immediately after the previous one is finished.

4. Allow operations to require multiple resource during the entire processing of an operation.

The MILP method minimises the Total Weighted Earliness Tardiness (TWET) objective function. The MILP method may, however, not always find feasible solutions within short computation time. Therefore, we proposed a backup method that schedules jobs one-by-one (ranked by Earliest Due Date), when the computation time limit is exceeded. This will result in a feasible schedule within short computation time, however, the objective value of the method may become poor. In order to update the schedule throughout the day, we proposed the planning cycle method. The planning cycle updates the scheduling problem each specified interval with new jobs when they are forecasted within a specified planning horizon. When the start time of a job lies within a specified fixed zone the job is not considered for rescheduling during the next planning cycle. The planning cycle may also be triggered by events such as priority jobs issued by an operator or the breakdown of a resource.

For determining the optimal parameters for the planning cycle, we performed a 2^k factorial design. We concluded that curvature was significant in the model and added a Central Composite Design (CCD) in order to analyse the curvature effects. This resulted in a planning horizon of 26.5 minutes, a planning interval of 100 seconds, and a fixed zone of 250 seconds. These parameters were used for all further experiments.

We performed multiple experiments for assessing the performance of the new planning cycle solution. First we compared the performance of our planning cycle with the performance of the current situation for the year 2015. This resulted in a savings of 128.7 hours and a total cost savings of € xx,xxx.xx (excluding Press 3). In our experiments we did not take waiting time with the cause of breakdowns into account. Therefore, the actual savings may turn out lower than the resulting savings. However, in the current situation all waiting times lower than 60 seconds are not registered, which may increase the savings.

To determine the effects of assigning mixing orders of presses to certain mixers, we performed multiple experiments with all possible combinations of mixer assignments. This way we tested the effect of the different tile height setups at the presses and the mixer assignment on the performance of the system. The resulting performance for each setup can support the management in planning daily production orders at the plant. The results show favourable and unfavourable combinations of presses assigned to a mixer. For example, it is unfavourable to combine Press 5 with Press 1 or 2 for the bottom-layer mixers when both are producing large tiles.

It may occur that on a production day a mixer or transporter is broken down. This causes all the presses assigned to that mixer to be assigned to another mixer. One of the days was 25th March 2016 where Mixer 5 was broken down. This resulted in three presses being assigned to Mixer 4 and causing a lot of waiting time for material. We simulated this situation with our new planning cycle solution to test whether our solution would lower the waiting time. We concluded our solution does not lower the waiting time, however, we gained the insight that the capacity of Mixer 4 was the bottleneck during that day. We experimented with the effect of lowering the processing time for the mixer and concluded that lowering the processing time with 15 seconds could already lower the total waiting time for material that day by 95.62%.

We conclude that our new planning cycle model can help SVI in reducing the waiting time for input material at the presses and help increase the Overall Equipment Effectiveness (OEE). By implementing the system in cooperation with TMS the payback period of the investments is 0.80 years (best case scenario and 100% of savings realised) up to 5.02 years (worst case scenario and 25% of savings realised).

7.2 Recommendations

Section 7.1 described our conclusions regarding the research performed at SVI. This section contains our recommendations for SVI that are not directly related to our research goals.

1. *Improve registration of breakdowns of the presses*

   For some presses the breakdown data is missing or inconsistent. In order to allow better analysis of the breakdowns, the registration of these breakdowns should be improved. Currently, Press 3 has no registration of breakdowns at all. We recommend to start registering the breakdowns of this press as well in order to be able to analyse the performance. We also advise to implement extra options for registering the cause of waiting time for top-layer or bottom-layer. At the moment, it is not always clear what causes the waiting time for the top-layer or bottom-layer. Therefore, we recommend to add additional options to register whether the waiting time is caused by planning reasons or due to a breakdown. This gives better insight in performance of the system.

2. *Improve the moisture level consistency of the mixtures*

   SVI sometimes copes with the problem of keeping moisture levels at a constant rate at the mixing process. One of the causes of this problem was the variance in input material. We eliminated this cause with a project team by implementing a different process for calibrating the weighing stations of raw materials. Another cause is the measurement of the moisture levels inside the mixers. Each mixer has a sensor for the moisture level, however, this sensor is not calibrated and therefore also not used in the system. The sensor allows for automatically controlling the moisture level. We therefore recommend to calibrate the moisture sensor and start using it for automatically controlling the moisture level.

3. *Flexibly assign mixing jobs of the bottom-layer to Mixer 4 and Mixer 5*

   At the moment mixing jobs of the bottom layer are assigned to either Mixer 4 or 5 throughout the entire length of the day. In order to improve the system further, mixes should be dynamically assigned to Mixer 4 or 5. However, to make this possible the moisture level of the delivered mixes should be stable. When a mix is taken from Mixer 4 it should have the same contents as an order from Mixer 5. We recommend flexibly assigning mixing orders to Mixer 4 and 5 as soon as the consistency of the moisture level is improved.

4. *Add a buffer for plateaus between curing chamber and press*

   Tiles are transported on plateaus to the curing chamber. When the plateaus return from the curing chamber the tiles are taken of the plateaus and the plateau is transported to the press to be filled with tiles again. However, when a process at the retrieval of the tile from the plateau or the packaging of the tile breaks down, the press also comes to a halt as there is no buffer for plateaus. The press needs to wait until an empty plateau becomes available again. This causes a lot of waiting time for plateaus that can easily be solved by adding a buffer of plateaus between the retrieval of tiles from the plateau and the press.

7.3 Further Research

Our research is one step in minimising the waiting time for raw materials at the presses at the plant of SVI in Dordrecht.

1. *Generalisation of the model*

   SVI has several more plants and the holding of SVI, CRH plc, has many plants that may also benefit our improved planning model. Also, clients of TMS may also benefit our improved planning model. Our model is most suitable for flexible system that makes use of multiple resources and that host multiple transporters in the same system. Plants that have similar conditions may also benefit this model. However, the model would require adaption to the specific needs of those plants. Nevertheless, further research could be done in order to investigate which plants may also benefit this model.

2. *Simulation of Model with Breakdowns and Robustness*

   For this research, we did not take breakdowns into account. A simulation study is required in order to test the effect of breakdowns by the presses, transporters, or mixers. The simulation study can also be used for determining the parameters of the robustness. The robustness of schedules can only be tested when breakdowns or other delays are taken into account.

References

Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management science, 34*(3), 391-401.

Armentano, V., & Scrich, C. (2000). Tabu search for minimizing total tardiness in a job shop. *International Journal of Production Economics, 63*(2), 131-140.

Baker, K., & Scudder, G. (1990). Sequencing with earliness and tardiness penalties: a review. *Operations research, 38*(1), 22-36.

Baptiste, P., Flamini, M., & Sourd, F. (2008). Lagrangian bounds for just-in-time job-shop scheduling. *Computers & Operations Research, 35*(3), 906-915.

Beck, J., & Refalo, P. (2003). A hybrid approach to scheduling with earliness and tardiness costs. *Annals of Operations Research, 118*(1-4), 49-71.

Chen, H., & Luh, P. (2003). An alternative framework to Lagrangian relaxation approach for job shop scheduling. *European Journal of Operational Research, 149*(3), 499-512.

Cordeau, J., Gendreau, M., Laporte, G., Potvin, J., & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 512-522.

Danna, E., & Perron, L. (2003, January). Structured vs. unstructured large neighborhood search: A case study on job-shop scheduling problems with earliness and tardiness costs. *Principles and Practice of Constraint Programming-CP 2003*, 817-821.

De Bontridder, K. (2005). Minimizing total weighted tardiness in a generalized job shop. *Journal of Scheduling, 8*(6), 479-496.

dos Santos, A., raujo, R., & Arroyo, J. (2010). A combination of evolutionary algorithm, mathematical programming, and a new local search procedure for the just-in-time job-shop scheduling problem. *Learning and Intelligent Optimization*, 10-24.

Glover, F. (1989). Tabu search-part I. *ORSA Journal on computing, 1*(3), 190-206.

Gurufocus. (2016, May 16). Retrieved from Gurufocus: http://www.gurufocus.com/term/wacc/CRH/Weighted%2BAverage%2BCost%2BOf%2BCapital%2B%2528WACC%2529/CRH%2BPLC

Haupt, R. (1989). A Survey of Priority Rule-Based Scheduling. *Operations-Research-Spektrum, 11*(1), 3-16.

He, Z., Yang, T., & Tiger, A. (1996). An exchange heuristic imbedded with simulated annealing for due-dates job-shop scheduling. *European Journal of Operational Researsch, 91*(1), 99-117.

Holthaus, O., & Rajendran, C. (1997). Efficient dispatching rules for scheduling in a job shop. *International Journal of Production Economics, 48*(1), 87-105.

Huang, R., & Yang, C. (2008). Ant colony system for job shop scheduling with time windows. *The International Journal of Advanced Manufacturing Technology, 39*(1-2), 151-157.

Jain, A., & Meeran, S. (1999). Deterministic job-shop scheduling: Past, present and future. *European journal of operational research, 113*(2), 390-434.

Kanet, J., & Sridharan, V. (2000). Scheduling with inserted idle time: problem taxonomy and literature review. *Operations Research, 48*(1), 99-110.

Kelbel, J., & Hanzálek, Z. (n.d.). Constraint programming search procedure for earliness/tardiness job shop scheduling problem. *Procedings of the 26th Workshop of the UK Planning and Scheduling Special Interest Group*, 67-70.

Kutanoglu, E. (1999). An analysis of heuristics in a dynamic job shop with weighted tardiness objectives. *International Journal of Production Research, 37*(1), 165-187.

Land, A., & Doig, A. (1960). An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, 497-520.

Law, A. (2007). *Simulation Modeling and Analysis* (4th Edition ed.). McGraw-Hill.

Mazzini, R., & Armentano, V. (2001). A heuristic for single machine scheduling with early and tardy costs. *European Journal of Operational Research, 128*(1), 129-146.

Monette, J., Deville, Y., & Van Hentenryck, P. (2009, September). Just-In-Time Scheduling with Constraint Programming. *ICAPS*.

Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling, 12*(4), 417-431.

Pinedo, M. (2012). *Scheduling: theory, algorithms, and systems.* Springer.

Pinedo, M., & Singer, M. (1999). A shifting bottleneck heuristick for minimizing the total weighted tardiness in a job shop. *Naval Research Logistics, 46*(1), 1-17.

Rajendran, C., & Holthaus, O. (1999). A comparative study of dispatching rules in dynamic flowshops and jobshops. *European journal of operational research, 116*(1), 156-170.

Ramasesh, R. (1990). Dynamic job shop scheduling: a survey of simulation research. *Omega, 18*(1), 43-57.

Roy, S., & Sussman, B. (1964). Les problèmes d'ordonnancement avec contraintes disjonctives. *SEMA, Note D.S. No. 9 bis*.

Thiagrajan, S., & Rajendran, C. (2005). Scheduling in dynamic assembly job-shops to minimize the sum of weighted earliness, weighted tardiness and weighted flowtime of jobs. *Computers & Industrial Engineering, 49*(4), 463-503.

Valencia, L., & Rabadi, G. (n.d.). A multiagents approach for the job shop scheduling problem with earliness and tardiness. *Systems, Man and Cybernetics, 2003. IEEE International Conference On*, 1217-1222.

Vieira, G., Herrmann, J., & Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies and methods. *Journal of scheduling, 6*(1), 39-62.

Wan, G., & Yen, B. (2002). Tabu search for single machine scheduling with distinct due windows and weighted earliness/tardiness penalties. *European Journal of Operational Research, 142*(2), 271-281.

Weng, M., & Ren, H. (2006). An efficient priority rule for scheduling job shops to minimize mean tardiness. *IEE transactions, 38*(9), 789-795.

Yang, H., Sun, Q., Saygin, C., & Sun, S. (2012). Job shop scheduling based on earliness and tardiness penalties with due dates and deadlines: an enhanced genetic algorithm. *The International Journal of Advanced Manufacturing Technology, 61*(5-8), 657-666.

Zhu, Z., & Heady, R. (2000). Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach. *Computers & Industrial Engineering, 38*(2), 297-305.

Appendices

Appendix A: Processing Times of Job Operations

This appendix contains the processing times of all operations for each possible job. Table 0.1 shows the list of possible jobs and the processing time of Operations 1 to 8 for each job. The total processing time is also displayed. The job is identified with first the departing mixer and the destination silo. For example, a job starting at Mixer 2 and delivery at the top-layer silo of Press 2 is listed as 'M2-2D'.

Table 0.1: Processing Times of Operations for Each Job and Total Processing Time of the Job in Seconds

Job	**Op. 1**	**Op. 2**	**Op. 3**	**Op. 4**	**Op. 5**	**Op. 6**	**Op. 7**	**Op. 8**	**Total (sec.)**
dummy	0	0	0	0	0	0	0	0	406
M1-1D	210	60	0	22	56	0	46	12	390
M1-2D	210	60	0	22	48	0	38	12	358
M1-3D	210	60	0	14	40	0	22	12	380
M1-5D	210	60	0	25	40	0	33	12	396
M2-1D	210	60	0	17	56	0	43	10	380
M2-2D	210	60	0	17	48	0	35	10	364
M2-3D	210	60	0	17	40	0	27	10	370
M2-5D	210	60	0	20	40	0	30	10	384
M3-1D	210	60	0	11	56	0	38	9	368
M3-2D	210	60	0	11	48	0	30	9	368
M3-3D	210	60	0	19	40	0	30	9	362
M3-5D	210	60	0	16	40	0	27	9	366
M4-1K	160	60	0	21	62	0	49	14	336
M4-2K	160	60	0	21	47	0	34	14	328
M4-3K	160	60	0	24	40	0	30	14	308
M4-5K	160	60	0	16	38	0	20	14	376
M5-1K	160	60	0	26	62	0	52	16	346
M5-2K	160	60	0	26	47	0	37	16	338
M5-3K	160	60	0	29	40	0	33	16	322
M5-5K	160	60	0	23	38	0	25	16	406

Appendix B: Experiments for Day Setups in Year 2015

Table 0.2: Experiments for Different Setups with Waiting Time in Hours for Current Situation and Planning Cycle

Experiment	Press 1	Press 2	Press 3	Press 5	Frequency	Current Situation	Planning Cycle
Experiment 1	Low	High	Off	Low	28	20.31	0.00
Experiment 2	Low	High	Off	Off	25	5.41	0.00
Experiment 3	High	Low	Off	Low	17	7.91	0.00
Experiment 4	Low	Low	Off	Low	16	7.97	0.00
Experiment 5	Low	High	Off	High	16	9.75	0.00
Experiment 6	High	High	Off	Low	16	9.91	0.00
Experiment 7	High	Low	Off	High	12	9.65	0.00
Experiment 8	Low	Low	Off	Off	12	4.52	0.00
Experiment 9	High	High	Off	Off	12	5.49	0.00
Experiment 10	Low	Low	Low	Low	11	8.74	0.00
Experiment 11	High	Low	Off	Off	10	3.38	0.00
Experiment 12	Low	Low	Off	High	10	4.42	0.00
Experiment 13	High	High	Off	High	9	4.93	0.00
Experiment 14	High	High	Low	Low	8	6.00	0.00
Experiment 15	Low	High	Low	Low	7	5.50	0.00
Experiment 16	Low	High	Low	High	7	3.18	0.12
Experiment 17	High	High	Low	High	4	2.62	0.10
Experiment 18	High	Low	Low	Low	4	1.29	0.00
Experiment 19	Low	High	Low	Off	3	1.08	0.00
Experiment 20	Low	Off	Off	Off	3	0.38	0.00
Experiment 21	Low	Low	Low	High	3	1.35	0.05
Experiment 22	High	Low	Low	High	2	2.13	0.03
Experiment 23	Off	High	Off	Low	1	0.49	0.00
Experiment 24	High	High	Low	Off	1	0.49	0.00
Experiment 25	High	Off	Off	Off	1	0.15	0.00
Experiment 26	Off	Off	Off	Low	1	0.59	0.00
Experiment 27	Off	Off	Off	High	1	0.73	0.00
Experiment 28	Off	High	Off	Off	1	0.47	0.00
Total					**241**	**128.82**	**0.30**

Table 0.3: Calculation of Cost Savings for Each Press

	Cost of 1 Hour Waiting Time	Total Savings in Hours	Total Cost Savings
Press 1	€ xxx,-	35.10	€ xx,xxx.xx
Press 2	€ xxx,-	43.56	€ xx,xxx.xx
Press 3	€ xxx,-	N/A	N/A
Press 5	€ xxx,-	49.99	€ xx,xxx.xx
		Total	**€ xx,xxx.xx**

Appendix C: Experiments for Effect of Mixer Assignment and Tile Height

Table 0.4 contains a list of combinations of presses assigned to a mixer ranked on the score given the tile height produced on each mixer. The lower the score the more favourable the assignment of a press to a mixer is. The tile height settings for this scenario is 50 mm, 80 mm, 50 mm, and 80 mm for Press 1, 2, 3, and 5 respectively. Thus the most favourable combination for the top-layer is assigning Press 1 and 2 to the same mixer. Assigning Press 3 to a separate mixer and assigning Press 5 to a separate mixer. The most favourable combination for the bottom-layer is assigning Press 1 and 5 to the same mixer and Press 2 and 3 to the same mixer. The list also shows combining Press 2 and 5 to the same bottom-layer mixer is unfavourable. This can be explained as they both produce tiles of 80 mm which require a lot of input material for the bottom-layer.

Table 0.4: Example of a List of Mixer Assignments Ranked on Favourability Given the Tile Height on Each Press

	Top-layer mixer assignment				*Bottom-layer mixer assignment*				
	Press 1	**Press 2**	**Press 3**	**Press 5**	**Press 1**	**Press 2**	**Press 3**	**Press 5**	**Score**
1	M1	M1	M2	M3	M4	M5	M5	M4	0
2	M1	M2	M3	M1	M4	M4	M5	M5	26,6
3	M1	M2	M3	M2	M4	M4	M5	M5	26,6
4	M1	M2	M3	M3	M4	M4	M5	M5	26,6
5	M1	M1	M2	M3	M4	M4	M5	M5	26,6
6	M1	M2	M1	M3	M4	M4	M5	M5	26,6
7	M1	M2	M2	M3	M4	M4	M5	M5	26,6
8	M1	M2	M3	M1	M4	M5	M5	M4	28
9	M1	M2	M3	M2	M4	M5	M5	M4	28
10	M1	M2	M3	M3	M4	M5	M5	M4	28
11	M1	M2	M1	M3	M4	M5	M5	M4	28
12	M1	M2	M2	M3	M4	M5	M5	M4	28
13	M1	M1	M2	M3	M4	M5	M4	M5	37514,5
14	M1	M2	M3	M2	M4	M5	M4	M5	106928,4
15	M1	M2	M1	M3	M4	M5	M4	M5	150952,7
16	M1	M2	M3	M1	M4	M5	M4	M5	171960,7
17	M1	M2	M3	M3	M4	M5	M4	M5	178077,9
18	M1	M2	M2	M3	M4	M5	M4	M5	194014,6

Appendix D: Significance Results of Planning Cycle Parameters from Minitab

Step 1

Use all factors, quadratic effects, and interaction effects.

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	9	61211197	6801244	20,05	0,006
Linear	3	42608860	14202953	41,88	0,002
Interval	1	2529990	2529990	7,46	0,052
Commitment zone	1	1973330	1973330	5,82	0,073
Horizon	1	40663512	40663512	119,90	0,000
Square	3	16356680	5452227	16,08	0,011
Interval*Interval	**1**	**241933**	**241933**	**0,71**	**0,446**
Commitment zone*Commitment zone	**1**	**72675**	**72675**	**0,21**	**0,667**
Horizon*Horizon	1	9733623	9733623	28,70	0,006
2-Way Interaction	3	2215455	738485	2,18	0,233
Interval*Commitment zone	**1**	**4010**	**4010**	**0,01**	**0,919**
Interval*Horizon	1	1853901	1853901	5,47	0,080
Commitment zone*Horizon	**1**	**386036**	**386036**	**1,14**	**0,346**
Error	4	1356558	339139		
Lack-of-Fit	3	658587	219529	0,31	0,827
Pure Error	1	697971	697971		
Total	13	62567755			

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
582,357	97,83%	92,95%	61,97%

Step 2

Leave out Interval versus Fixed Zone and Fixed zone versus Horizon. These are *not* significant with a p-value of 0.919 and 0.346 respectively.

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	7	60778885	8682698	29,12	0,000
Linear	3	50804948	16934983	56,80	0,000
Interval	1	2502169	2502169	8,39	0,027
Commitment zone	1	1882982	1882982	6,32	0,046
Horizon	1	50776874	50776874	170,31	0,000
Square	3	17618286	5872762	19,70	0,002
Interval*Interval	**1**	**241933**	**241933**	**0,81**	**0,402**
Commitment zone*Commitment zone	**1**	**72675**	**72675**	**0,24**	**0,639**
Horizon*Horizon	1	9817620	9817620	32,93	0,001
2-Way Interaction	1	1783143	1783143	5,98	0,050
Interval*Horizon	1	1783143	1783143	5,98	0,050
Error	6	1788870	298145		
Lack-of-Fit	5	1090899	218180	0,31	0,866
Pure Error	1	697971	697971		
Total	13	62567755			

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
546,027	97,14%	93,81%	81,78%7

Step 3

Leave out quadratic effects of Interval and Fixed Zone. These are *not* significant with a p-value of 0.402 and 0.639 respectively.

```
Analysis of Variance

Source                DF    Adj SS    Adj MS  F-Value  P-Value
Model                  5  60479683  12095937    46,34    0,000
  Linear               3  53247570  17749190    68,00    0,000
    Interval           1   2502169   2502169     9,59    0,015
    Commitment zone    1   1882982   1882982     7,21    0,028
    Horizon            1  53201147  53201147   203,83    0,000
  Square               1  17319085  17319085    66,35    0,000
    Horizon*Horizon    1  17319085  17319085    66,35    0,000
  2-Way Interaction    1   1783143   1783143     6,83    0,031
    Interval*Horizon   1   1783143   1783143     6,83    0,031
Error                  8   2088072    261009
  Lack-of-Fit          7   1390101    198586     0,28    0,897
  Pure Error           1    697971    697971
Total                 13  62567755


Model Summary

      S    R-sq  R-sq(adj)  R-sq(pred)
510,890  96,66%     94,58%      83,68%
```

The remaining factors are all significant.