Single machine scheduling for powder coating aluminium profiles

Karien Krüger
14018986
2017/09/24

BPJ 420: Final Project Report

DECLARATION OF ORIGINALITY:

I, Karien Krüger, student number 14018986, hereby declare that this report is my own original work and that the references listed provide a comprehensive list of all sources cited or quoted in this report.

Single machine scheduling for powder coating aluminium profiles

by

Karien Krüger

Executive Summary

Scheduling is one of the most important issues in the planning and operation of manufacturing systems (Hoitomt, Luh, & Pattipati, 1993). It is becoming crucial for businesses to be able to schedule effectively. This is due to the effective management of time and money which is now more important to maintain a successful business.

This document describes the proposed final year project that will be performed at Wispeco Pty (LTD). Wispeco is a company extruding aluminium profiles. The project was executed at the powder coating department of the Alberton plant. The project aims to find an effective scheduling system, considering customer service and the cost of the different sequences that can be scheduled for the aluminium profiles that must be powder coated.

The document contains the preliminary concepts which will be used in the final deliverable. This document defines the problems at the powder coating department and motivates the need for this project. A literature study and a basic solution are also enclosed with an analysis done on these different solutions.

This project will cause the scheduling of the aluminium profiles which must be powder coated to be more cost-effective and customer-orientated. Providing Wispeco with this scheduling program will be a significant contribution to the powder coating department.

Table of Contents

List of Figures

List of Tables

Glossary

A-Colours: The four most common colours, namely black, white, charcoal, and bronze.

Big colours: See A-colours.

CUBE: A powder coating machine used at Wispeco.

Jobbing: Orders coming from private customers to be powder coated.

Milling: Orders coming from the extrusion department within Wispeco.

Operator: A person which performs the colour changes and changes the type of colour (more information in section 5.2).

Planner: Plans the sequence in which the orders must be powder coated, and produces a schedule that is provided for the FLM (First Line Manager).

Profile: Aluminium rods or dies.

Skip: An iron holder where the aluminium profiles are stored (about 0.75m wide and 0.75m high). This holder is different from a rubbish skip.

Small colours: All other colours which are not included in the A-colour category.

Spotter: A person who finds the next skip that must go into the CUBE.

Sqm: Square meter (m^2).

TOPP: Wispeco's ERP (Enterprise Resource Planning) System.

Acronyms

CSV: Comma Separated File

BFS: Breadth First Search

DFS: Depth First Search

EDD: Earliest Due Date

ERP: Enterprise Resource Planning

FIFO: First-in-first-out

FLM: First Line Manager

LPT: Largest Processing Time

PSO: Particle Swarm Observation

SMSP: Single Machine Scheduling Problem

SPT: Shortest Processing Time

TSP: Travelling Salesman Problem

WEDD: Weighted Earliest Due Date

WLPT: Weighted Largest Processing Time

WSPT: Weighted Shortest Processing Time

1. Introduction

The project execution took place at the Alberton plant of Wispeco. Wispeco is the biggest aluminium extrusion company in Africa (WISPECO PTY (LTD), 2017). Wispeco consists of different plants. The Alrode (Alberton) plant of Wispeco consists of different departments, including receiving, scrap, profiles, powder coating, and anodising. This project was performed at the receiving side of the powder coating department.

The powder coating department obtains profiles to powder coat from private customers and from the profile department of Wispeco, as shown in Figure 1. The profiles are powder coated in over 700 colours and then distributed to stockists of Wispeco or to private customers. The powder coating department works five days a week, divided into two shifts per day.

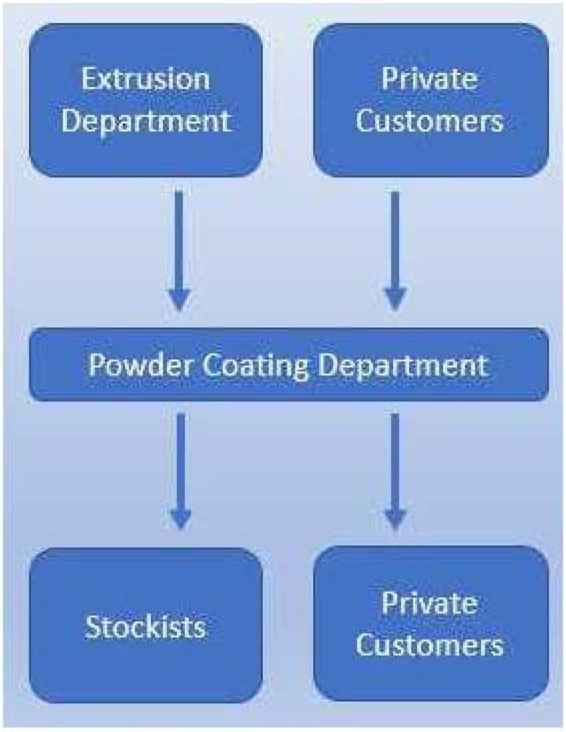

Figure 1: Different customers of the powder coating department

The department is struggling to plan and schedule the stock which must be powder coated. Scheduling is becoming more important as more jobs must be executed simultaneously with shorter manufacturing times (Lopez & Roubellat, 2008).

Section 2 contains background information on the company and the operations of the powder coating department. Then follows section 3 with the approach, scope, and deliverables of the project. Section 4 follows with a problem investigation of Wispeco, Section 5 contains a literature review, providing more information concerning what is known about this project, after which a development of supplementary mechanisms and different solutions is discussed in section 6 and section 7. Data analysis and validation of this solution follow in section 8 and section 9. This document concludes with section 10.

2. Project Aim / Rationale

Scheduling has been fascinating researchers since the 1950's (Jain & Meeran, 1999). Scheduling is a procedure regarding the sequence and time of necessary operations, and can also be described as the series of tasks or orders that must be performed (Dictionary, 2017).

For Wispeco scheduling includes:

- Arranging and controlling the orders that go into the CUBE at the powder coating department.
- Finding the most effective sequence in which the orders must be completed.

When orders come in from private customers (called Jobbing), the orders are powder coated before the extrusion orders (called Milling), in other words, the Jobbing orders are prioritised. Milling orders take on average 61% longer to be powder coated than Jobbing orders. This problem causes unhappy customers, as the policy for completing orders remain the same for all customers. By changing the schedule better customer service can be achieved. Each shift contains unnecessary colour changes. Each time an unnecessary colour change happen four to seven production minutes are wasted, depending on the type of colour change. With the proposed solution contained in this document, the powder coating department can save R118 636.71 per month (Discussed more in detail in section 4.3 and section 9.1).

The project's aim was to deliver a scheduling program for the powder coating department. The program contains a competitive scheduling method that receives a Comma Separated Values (CSV) file input from Wispeco's Enterprise Resource Planning (ERP) system. A CSV file is a file of data which can be used in a tabular form. This CSV input contains information for all orders currently in the receiving area. The program still considers customer service, the number of colour changes per day, and the number of sqm done each day.

Program objectives included the following:

- Providing the most effective planning sequence for the next two shifts (next 24-hours).
- Saving time and powder, by meeting the middle point between the number of change overs, production output, and the due date of the orders.
- Meeting customer requirements so that an order does not take too long to powder coat.

Further objectives that will be met with the final deliverable includes:

- Ensuring that the planned orders' mass does not exceed the limits (different rules apply).
- Recommending the amount and colour of powder needed for the next two shifts

3. Project Approach, Scope & Deliverables

3.1 Approach

Firstly, investigations of the project began. To 'walk the floor' was the primary method for investigations. It was decided to first chart the process to have a broader understanding of it.

The spotter, first-line manager, planner, Information Technology (IT) technician, the production staff, and the operations manager were interviewed. Problems were identified by questioning these people. These people's problems were observed and taken note of. This allowed for the whole process to be broadly understood so that all aspects could be considered and nothing overlooked. A whole analysis of the problems could then be done. The analysis was done with frequent input from the planner. The causes of the problems were found, and different problems were grouped together. The different constraints of the process were investigated (e.g. time, the number of production people, the number of forklifts and forklift drivers, rules of the CUBE).

The problems were now clear, and a literature study could be done. Firstly, definitions were sought where meanings were unclear, to gain a basic knowledge. Consequently, more in-depth connotations with the topic were found. This connotation was done by thoroughly reading journal articles and books to obtain a better idea of what the topic was about. All information was examined, and the information necessary selected. Subsequently, it was decided what would be important for the final deliverable, and the information was documented to provide the literature review. Different solutions were also combined with existing solutions.

From the problem analysis and the literature review, different scheduling heuristics were developed which satisfies all the constraints. These heuristics were tested against each other to find the optimal strategy for this problem. The optimal solution was also verified and validated. The verification and validation were done by performing a sensitivity analysis on the input parameters, and observing the effect on the output obtained. The solution was further validated by looking at the performance of the program against the given aims. The information and results will be used to refine the basic heuristic. This heuristic will form the basis for the final deliverable.

In the final deliverable an objective function, or a combination of functions, will be formulated. Each objective function will conform to the rules set out. By combining possible sequences that conform to the rules, the optimal sequence will be found (Malakooti, 2013). After further validation and evaluation of this written program (as with the basic heuristic), the final program will be delivered.

3.2 Scope

The project includes all stock, Jobbing – and Milling orders, in the receiving area of the powder coating department of the Alrode plant. There is assumed that the powder needed is always in stock.

For this document, a basic heuristic was developed to form the basis on which further programming can be done. The program takes into account the different types of colour changes, the date on which the order was received, and the production output per day of the powder coating department.

Internal decisions and policies of Wispeco are not examined or changed but only used to obtain an optimal solution within these boundaries.

3.3 Deliverables

The output will take the form of an Excel file containing an effective planning sequence for the next two shifts (next 24-hours). This schedule will provide an effective plan while still maintaining customer service. The schedule will also specify the amount and colour of powder needed for a specific shift.

4. Problem Investigation

It is very important to state the problems of a process when aiming to find a solution (T'kindt & Billaut, 2002). Different problems were identified at the powder coating department.

4.1 Problem Background

The receiving area of Wispeco consists of different bays, as illustrated in Figure 2.

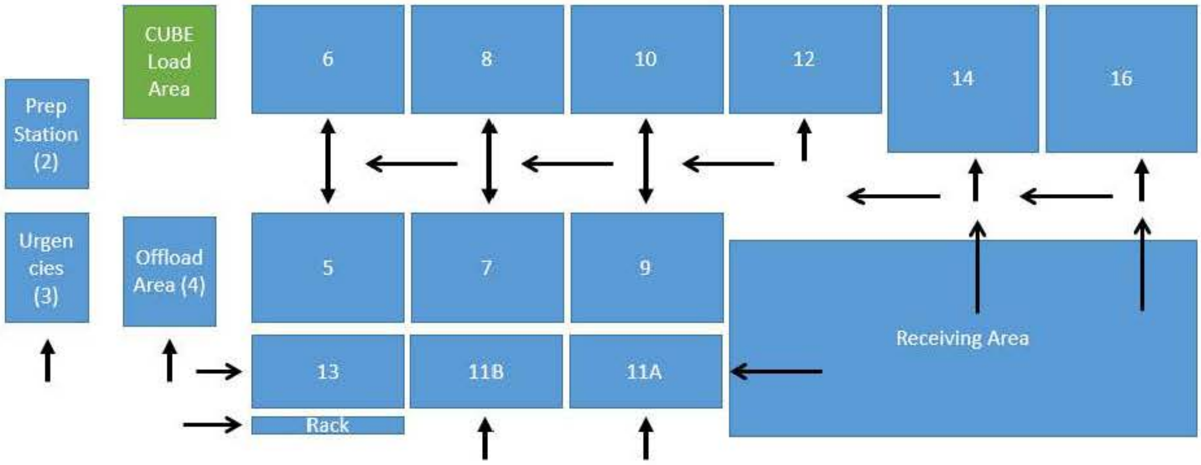

Figure 2: Receiving area

The profiles are received (at the receiving area) and sorted according to the colour it will be powder coated. Afterwards, the spotter communicates to the forklift driver to which bay the profiles must be taken. For example, if an order containing two skips (if an order is too big for one skip, two are used) must be sprayed charcoal, the spotter will transport both skips to a bay that must be sprayed charcoal, for example, bay 5. All other orders that must be sprayed charcoal will then also be transported to bay 5. If the next order must be sprayed bronze the profiles will then be transported, for example, to bay 6, and all other orders that must be sprayed bronze will also be transported to bay 6. If the number of bronze orders exceeds the limit of a bay, a second bronze bay will be created.

There are four "big" colours, called A-colours, namely: white, bronze, charcoal, and black. For the A-colours, each colour receives its own bay, as these colours amounts to about 80% of all stock in the receiving area, as seen in Figure 3. The rest of the colours are called "small" colours which are grouped together in a bay. Thus, each colour is not allocated its own bay. The rack area, as seen in Figure 2, is used for all small colour orders that are less than 20 profiles. The skips at bay 5-10, 12, 14, and 16 are for the A-colours, whereas the skips at bay 11A, 11B, 13, and the rack are for the small colours. The bays 5 to 10 and 12 are sized to hold skips that are sorted in 7 rows, stacked 7 skips high (as per safety regulations). Bay 14 and 16 are sized to take 10 rows with 7 skips high. The small colour bays are 4 rows stacked only 4 skips high (because it is easier to reach to the bottom orders if the profiles are stacked 4 rows high).

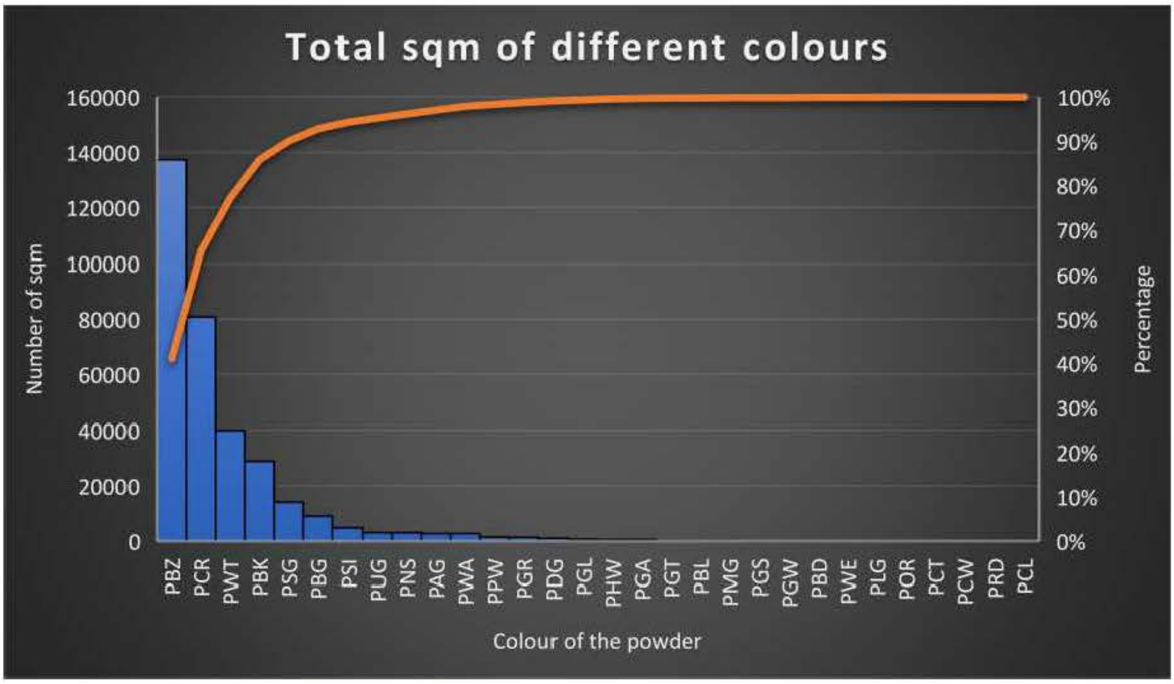

Figure 3: Total of sqm for different colours

4.2 Small Colours and Customer Service

At Wispeco there exist two types of colour changes, which differ in time and products that are used. Although there is not a considerable dependency on the setup time, as there are only two types of colour changes, this dependency would still play a role (discussed more broadly in section *5.2*).

The two types of colour changes at Wispeco are discussed here.

Reclaim: Reclaim is done after a production run that contained more than 40 sqm of profiles.
The time that a colour change takes: 7 minutes.
Powder wasted: 6% of all powder used.
Air pipes and pinch valves are used extra.

Spray-to-waste: Done after a production run that contained less than 40 sqm of profiles.
The time that a colour change takes: 4 minutes.
Powder wasted: 20% of all powder used.

The difference in time is due to the fact that when the powder is reclaimed an extra pipe is used that must also be cleaned.

As the amount of sqm done each day is very important for the plant, the FLM receives 10 points for each sqm that is powder coated. These points are then converted to a rand value. For each point, 40c is written up for the FLM. The FLM then receives a commissioned amount as bonus of that rand value. Countless times the FLM changes the planning schedule to achieve a higher production (output as sqm). The FLM exclude the small colours so that fewer colour changes take place, and then receives a higher bonus. Higher production can then take place, but unfortunately at the cost of customer service. Currently, Wispeco finishes a small colour within 6 days and an A-colour in 1.5 days on average. Thus, "small" colours take on average 75% longer to complete than the A-colours.

The due date for an order at Wispeco is 5 days. An A-colour thus takes 70% less than the specified due date, whereas small colours take 20% longer on average.

This late completion of orders creates a problem of poor customer service, as some orders take longer to be completed. When doing the planning this weight difference must be balanced as much as possible, without losing too much money.

Changing the policy of the due date would be an internal discussion within Wispeco, and would not be considered for this project.

4.3 Planning

Currently planning is performed according to the rule: Plan 2 A-colour bays, and then fill up the plan to 6500sqm per shift (as that is the average output per shift) with small colours. The sequence of the colours is planned according to the darkness of the colours (for example white, and then an off-white is planned, and not a black after a white, etc.). If there are two bays with the same colour, the bays are chosen at random. This type of planning is done because of the problem discussed in section 4.1.

This rule for planning is ineffective. The planning uses a rule-of-thumb method, which is not based on the most effective planning which can be performed. The rule causes small colours to not be powder coated, or if the order is powder coated, it is not powder coated in the most efficient way.

The possibility that unnecessary change overs are present is very likely, as there is no control over the number of colour changes. Each time an unnecessary colour change is done four to seven minutes is wasted, depending on the type of colour changes.

The different variable costs associated with a colour change is shown in Table 1.

Table 1: Variable costs associated with a colour change

Variable Expenses	Measure unit	A-Colours	Small colours
Powder	R/sqm	6.2595	7.8752

Chemicals	R/sqm	1.391	1.391
Losses	R/sqm	0.1284	0.1926
Plastic	R/sqm	1.1877	1.1877
Total Cost		**8.9666**	**10.6465**

As there is on average 0.1587sqm/s (calculations shown in section 6.2), and it takes four minutes in a reclaim colour change and seven minutes for a reclaim colour change (as discussed in section 4.2), the costs per colour change then totals up to the following amounts in Table 2:

Table 2: Cost for each type of colour change

	A-Colours	**Small colours**
Total cost for spray-to-waste colour change (Rand):	341.52	405.50
Total cost for reclaim colour change (Rand):	597.66	709.63

If looked at the number of times each colour change takes place, shown in Figure 4, and considering the fact that on average 18 change overs take place each shift. Wispeco will save R40 000 when minimising the number of change overs with 10%. This problem causes unnecessary production losses (discussed more in detail in section 5.2).

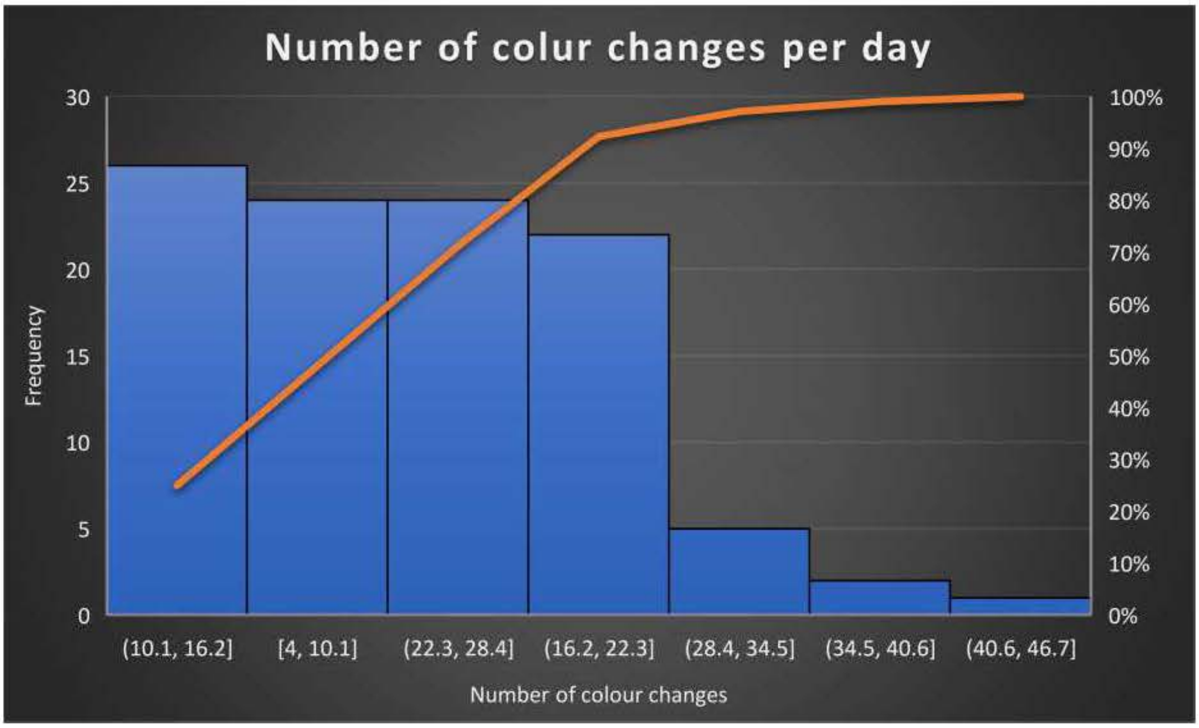

Figure 4: Number of colour changes that take place per shift

A further problem with the planning schedule is that it is not available for everyone to see. This problem causes further unhappy clients. A hard copy is provided to the FLM, who is then the only person who has the schedule. A client will call, for example, and ask if his bronze order is scheduled for that day. The planner will not be able to see exactly what is planned, as the schedule was only given through to the FLM. The planner will, nevertheless, remember that bronze was planned, and the information will be provided to the customer that his order is planned. Meanwhile, that specific bronze order was in another bronze bay which was not planned. This method of communicating without the correct information creates poor service.

4.4 CUBE

When the mass of the orders on the CUBE becomes heavy, because of the type of profiles currently hanging on the CUBE machine, the machine stops working (different rules that apply). This problem is an effect of the planning not executed effectively.

4.5 Powder Stolen

As there is no way to track the amount of powder needed per shift the possibility of powder being stolen is being investigated. When powder is needed the FLM collects powder from the warehouse store (the powder is notated in detail). The amount of powder taken is left to the discretion of the FLM how much powder is needed. A formula for providing the correct quantity of powder is already available, but not linked to the planning performed in that shift. To monitor the amount of powder being used is therefore almost impossible.

4.6 Jobbing Orders versus Milling Orders

Milling orders take up to 61% longer on average than Jobbing orders to be completed. This difference in lead times is shown in Figure 5. This effect implies that jobbing orders are prioritised over milling orders. This affects the type of service delivered to the customers, and as the policy for all orders are the same, namely that the order would be done in 5 days, many unhappy customers are created.

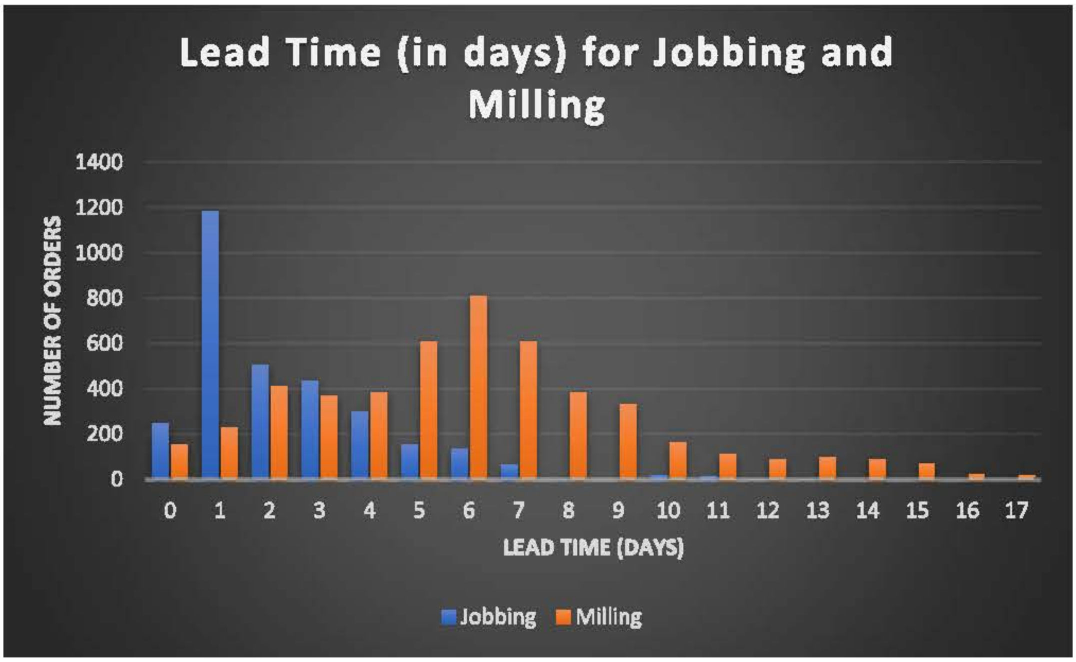

Figure 5: Lead time of jobbing orders and milling orders of the past three months

As seen above, different problems associated with process was established. In short, the problems entailed poor customer service, ineffective planning, the machine that stopped working, and the possibility that powder can be stolen.

5. Literature Review

Different kinds of problems can be solved with different types of algorithms. The kind of problem in the powder coating department was identified as a Single Machine Scheduling Problem (SMSP) which is discussed in section 5.1. Different topics that are linked to the problem in the powder coating department, as well as different solution algorithms for solving the problem, are also discussed in this section.

5.1 Single Machine Scheduling Problem (SMSP)

This problem is classified as an SMSP. Most models are classified based on the following:

- Characteristic steps that must be taken for a product.
- The number of operations for each product.
- Availability of resources to produce the product.

(Grobler, 2009).

A Single Machine Scheduling Problem occurs when there is only one machine available to work on. This problem is sometimes seen as the simplest type of scheduling (Pinedo, 2016), although the problem depends on the type of application. Many different algorithms have been applied successfully to single machine scheduling, amongst others the Particle Swarm Observation (PSO) algorithm (Anghinolfi & Paolucci, 2009) (Pan, Tasgetiren, & Liang, 2006). PSO is a stochastic population technique based on only one population or swarm, which was developed in 1995 (Hu, 2006).

An SMSP can be partially solved with two objectives, namely minimising the sum of flow times, and minimising the deviation from a common due date (Biskup, 1999). These two objectives will be discussed in detail in the rest of this section.

5.1.1 Minimal Deviation from a Common Due Date

Minimal deviation from a common due date means that all the orders must be completed as near as possible to the specified due date.

This objective includes earliness (completing an order early) and tardiness (completing an order late). The common due date is allocated an assigned cost for either earliness, tardiness, or both. This cost would depend on whether a job is finished before or after the due date (Nearchou, 2008). The goal is to mutually minimise the weighted earliness and penalty for tardiness. Meeting due dates is one of the most important deliverables in scheduling, as seen, for example, in Panwalker (1973).

As said, the sum of the earliness and tardiness costs must be minimised. It is, however, important to remember that both the earliness and tardiness's maximums should also be kept to a minimal, not just the sum of the two penalty costs (Biskup & Jahnke, 2001).

Minimal deviation from a common due date assumes that all jobs have the same due date. The due date can be internal, meaning that the company itself determines the due date to complete a job. The due date can also be external or settled beforehand with the customer (Biskup & Jahnke, 2001). Wispeco has an internal due date (as discussed in section 4.2. Process times cannot always be individually reducible, and therefore the whole process and its activities must be reduced. That is why the sequencing of orders is so important.

Scheduling with different due dates is very different from the strategies used with common due dates. Scheduling with different due dates are considerably more complex, and there are scarcely any such problems available (Lauff & Werner, 2004).

5.1.2 Minimal Flowtime

The flowtime is the period time it takes for a job to exit the production system (Webster & Baker, 1995). For each job, there is a weighted cost assigned to the job. The main objective of minimal flowtime is to minimise the sum of all the flowtimes and in turn the average flowtime. Minimising the flowtimes can be done by using the SPT rule (Isaacs, 1955). The sum of the flowtimes will also be equal to the sum of the completion time, as it is assumed that the release times are zero (Biskup, 1999).

5.2 Sequence Dependent Setup Times

Various times, the setup time depends on the existing current setup. (Dayama N.R. et al, 2015) developed, a scheduling model that includes algorithms that can be used where setup times are affected by the previous order. As the sequence influences the setup time, the production time changes depending on the sequence of orders. A certain sequence can decrease the production time, saving time and in effect money (as each minute of production costs money). On the other hand, a sequence can increase the production time, which will cost the company unnecessary money. The importance of scheduling because of different setup times is therefore crucial. At Wispeco, this problem can be seen with the different type of colour changes present, as discussed in section 4.3.

Different methods of solving these problems are available. Alternative solution strategies will now be discussed.

5.3 Heuristics

The origin of the word 'heuristic' comes from the Greek word 'heuriskin' meaning 'to discover' (Hall, 2008). Heuristics are used to help with decision making and problem solving (Cherry, 2016). Heuristics refers to a rule of thumb method where certain rules are followed to obtain a sequence that will be a near-optimal solution. This solution is not necessarily the 'best' method. Heuristics provide shortcuts to find a near-optimal solution when there is no way to find, with certainty, the best solution, or if the time to find the best solution will take unreasonably long. There are different types of heuristics that can be used. Heuristic methods are frequently applied to single machine scheduling.

It is necessary for a heuristic to also have a heuristic dominance rule (Chambers, Carraway, Lowe, & Morin, 1991). That is where some jobs must precede others to obtain an optimal solution.

Some dominance rules are:
1. Shortest Processing Time (SPT): Sequencing orders with increasing processing times (Hochbaum, 1999).
2. Weighted Shortest Processing Time (WSPT): Sort jobs in weighted order (Nieberg, 2010).
3. Earliest Due Date (EDD): Scheduled according to earliest due date.
4. Weighted Earliest Due Date (WEDD): Scheduled according to weighted earliest due date.
5. Largest Processing Time (LPT): Sort orders according to size.
6. Weighted Largest Processing Time (WLPT): Schedules according to weighted largest processing time (Arlow, 2016).

5.4 Meta-heuristics

The suffix 'meta' means 'in a higher level or beyond' (Vermeulen & Van den Akker, 2010). Meta-heuristics is thus algorithms that combine different types of basic heuristics for optimisation (Bianchi, Dorigo, Gambardella, & Gutjahr, 2009).

One of the most famous problems where heuristics are needed is the Travelling Salesman Problem (TSP). Basic heuristics for a single machine problem include:

5.4.1 Generate-and-Test

A very basic test, consisting of the following steps:
1. Generating a possible solution.
2. Test the solution
3. End the process, or repeat.

This method cannot be used for very complex problems (Robin, 2009).

5.4.2 Simulated Annealing

Simulated annealing is based on the annealing process used on metal. This method allows sequences of worse cases than the previous case, to allow a better solution to be found further on. Initially, this search jumps around in abundance within the state space, but as the searching progresses it follows a more firm and certain pattern (PEÑA, 2017).

5.4.3 Best-First Search

This method uses a combination of techniques from the Breadth First Search (BFS) and the Depth First Search (DFS). BFS does not fall into loops or stalemates, where DFS follows a single path. Best-first search explores the most promising method to date (Hollinger, 2017) using the advantages of the above-mentioned methods.

5.4.4 Tabu Search

This method's search is penalised by moves which were already taken in previous sequences (Glover, 1986). Therefore, this method can obtain a solution which is on a different optimal frontier to still be found, even if the solution lies far from the previously found solution and along the way worse solutions are found than the present solution.

The generate-and-test method will be used initially, to first obtain different scenarios and the effect it will have on the process. As this method was found to be impractical for complex problems (Robin, 2009) it will only be used as an interim method. Meta-heuristics will provide better solutions than a heuristic, but is more computationally complex.

The SMSP method can be solved by many different software programs. The best programs are discussed in the next section.

5.5 Different Software Programs

There are many programs available for scheduling. Many possible programs were inspected. These programs were narrowed down to three, namely:

- R
- Python
- Excel

These programs were chosen based on the availability of the programs, as well as training received in all three programs. Therefore, proficiency in the program is guaranteed.

R:

R is a statistical language. R can be used to perform statistical analysis on data (Theuwissen, 2017).

Python:

Python is best for application in an engineering and production environment (Theuwissen, 2017), although its uses can be extended for many other purposes.

Excel:

Excel is not a statistical language. Mistakes are prone to happen in Excel. The program may not function as intended due to missing data (Vaucher, 2017).

From above, it can be seen that Excel will not be the best program to use for this type of analysis. R is a statistical language and can be useful. Python, however, is more applicable than R. That is because Python's application is more suited for production environment, where R is used for more statistical approaches. Because Wispeco's powder coating department is a production environment, Python would, therefore, be the best-suited program to use.

6. Development of Supplementary Mechanisms

6.1 Chosen Application

The program used will depend on the problem and type of application needed. The application is done in an engineering environment that deals with production. As the program, Python, is already known, and the program works well with a production environment (as discussed in section 5.5), it was decided that the program will be written in Python.

6.2 Calculations and Parameters

There are two shifts for each 24 hours available. The first shift is from 06:00 till 18:00 and the second shift from 18:00 till 06:00 the next morning. Each shift consists of two tea breaks of 15 minutes each, and an hour lunch. Thus, the actual production time available is only 10.5 hours per shift or 21 hours per day. The number of hours acts as a constraint, together with the time of a colour change.

Different parameters exist:

The speed of the conveyor is 2 m/s.

$$S = 2\ [\frac{m}{s}]$$

The average conversion from kg to sqm is 3.2. That is, for every sqm, there is on average 3.2 kg of aluminium.

$$G = 2.9\ [\frac{kg}{sqm}]$$

There is an average of 12 000 sqm done per day. In each day, there are 21 working hours (24 hours – two 1-hour lunches, one for each shift – four 15-minute tea-times, two for each shift). The formula for calculating the average sqm/day is thus as follows:

$$A = \frac{12\,000\ [\frac{sqm}{day}]}{75\,600\ [\frac{s}{day}]} = 0.1587\ [\frac{sqm}{s}]$$

To calculate the average sqm per day the following was done:

$$B = \frac{A}{S} = \frac{0.1587\ [\frac{sqm}{s}]}{2\ [\frac{m}{s}]} = 0.07937\ [\frac{sqm}{m}]$$

The average kilogram per day was calculated by doing the following:

$$C = G \times B = 3.2\ \left[\frac{kg}{sqm}\right] \times 0.07937\ \left[\frac{sqm}{m}\right] = 0.2540\ [\frac{kg}{m}]$$

There are two types of colour changes, as discussed in section 4.2. The spray-to-waste colour change takes 4 minutes (Z_1) and the reclaim colour change 7 minutes (Z_2):

$$Z_1 = 4\ [min] * 60\ \left[\frac{s}{min}\right] = 240\ [s]$$

$$Z_2 = 7\ [min] * 60\ \left[\frac{s}{min}\right] = 420\ [s]$$

6.3 Basic Heuristic

The CSV file from TOPP is imported into Python. This CSV file was already processed to contain information in the correct format that it will be used.

6.3.1. Constraints

The production time was calculated by looking at the average speed of the conveyor, the average sqm that is done per second, and determining when colour changes will take place as well as what type of colour change. This information was used to constrain the number of orders that are being planned per shift. The number of hours available for production was used as discussed in section 6.2.

6.3.2. Objective Function

Three aspects were considered when developing the optimising function. Firstly, the cost of a colour change was considered. Secondly, the 'cost of goodwill' for finishing orders which have a long waiting time. Lastly, the fact that the FLM receives points for each sqm that is finished each day was considered.

Cost 1 - Cost of a colour change:

This cost was calculated using the information provided in section 4.3. The formula looks as follows with 4 used as a conversion factor:

$$C_1 = 4\ (\#\ colour\ changes\ that\ take\ place)(time\ required\ for\ the\ colour\ change)$$

Benefit 1 - Finishing orders with a long waiting time:

Benefit 1 was designed by using the 'policy of goodwill.' Wispeco has a policy of 'goodwill' which states that for each day the company delivers a finished product to the customer, the powder coating department will receive 40 points per order for each day the order was in the plant but not powder coated. Each point is equivalent to 10c of the colour change cost that the plant incurs. This link was used to establish the balance between having more colour changes and producing more.

$$B_1 = 40 \sum_{i=1}^{\#\ orders} waiting\ time\ [days]$$

Benefit 2 – Having a high productivity:

Calculating this benefit was done by using the fact that the FLM receives 10 points which are translated into R4 (as discussed in section 4.2) that is finished, as discussed in section 4.2. The conversion factor here is thus 10. The fact that the FLM receives, was also taken into account when creating the optimising function.

$$B_2 = \ 10 \sum_{i=1}^{\#\ orders} Sqm\ /\ order$$

The objective function is then:

$$F = C_1 - B_1 - B_2$$

The smaller an answer is the better is the selected solution, as the benefits are subtracted, and the cost is added.

6.4 Solution Output

The python program will give an output as a CSV file which can be opened with Excel. Such a solution will contain the order number, as well as the colour it must be powder coated and the time it would take to finish. An example of an actual planning sequence per shift can be seen in Table 3.

Table 3: Example of an actual solution

Order Name	Start Time (seconds)	End Time (Seconds)	Duration (Seconds)	Colour
'ALUM21'	0	33.62319	33.62319	'PAG'
'50888J'	33.62319	60.5041	26.88091	'PAG'
'J51309'	60.5041	71.69502	11.19093	'PAG'
'J51310'	71.69502	104.8141	33.11909	'PAG'
'J51311'	104.8141	151.317	46.50284	'PAG'
'J51312'	151.317	203.012	51.69502	'PAG'
'J51313'	203.012	229.477	26.46503	'PAG'
'J51315'	229.477	255.942	26.46503	'PBK'
'J51316'	255.942	332.426	76.48393	'PBK'
'J51314'	332.426	436.4713	104.0454	'PBK'
'WIPK18'	436.4713	527.1519	90.68053	'PBK'
'WIPK32'	527.1519	608.0025	80.85066	'PBK'
'391/1'	608.0025	624.1273	16.12476	'PBK'
'391/3'	624.1273	640.5293	16.40202	'PBK'
'391/7'	640.5293	660.9074	20.37807	'PBK'
'391/2'	660.9074	682.8355	21.92817	'PBK'
'391/6'	682.8355	706.1815	23.34594	'PBK'
'391/9'	706.1815	729.7732	23.59168	'PBK'
'391/4'	729.7732	755.5388	25.7656	'PBL'
'391/5'	755.5388	800.6301	45.09137	'PBL'
'J51307'	800.6301	809.5904	8.960302	'PBL'
'J51297'	809.5904	818.5885	8.99811	'PBL'
'J51287'	818.5885	834.0895	15.50095	'PBL'

'J51347'	834.0895	854.7322	20.64272	'PBL'
'WIPK14'	854.7322	884.1021	29.36988	'PBZ'
'R81736'	884.1021	918.1285	34.02647	'PBZ'
'R81735'	918.1285	1052.42	134.2911	'PBZ'
'R81739'	1052.42	1430.491	378.0718	'PBZ'
'R81734'	1430.491	2659.981	1229.49	'PBZ'
'J51314'	2659.981	2664.077	4.095778	'PBZ'
'J51313'	2664.077	2669.855	5.778198	'PBZ'
'J51315'	2669.855	2679.156	9.300567	'PBZ'
'J51316'	2679.156	2689.231	10.07561	'PBZ'
'J51314'	2689.231	2702.256	13.02457	'PBZ'
'J51317'	2702.256	2715.57	13.31443	'PBZ'
'J51319'	2715.57	2728.922	13.35224	'PBZ'
'J51318'	2728.922	2743.636	14.7133	'PBZ'
'J51327'	2743.636	2760.542	16.90611	'PBZ'
'J51324'	2760.542	2781.556	21.01449	'PBZ'
'J51334'	2781.556	2806.137	24.58097	'PBZ'
'J51344'	2806.137	2830.901	24.76371	'PBZ'
'J51364'	2830.901	2856.421	25.51985	'PBG'
'J51394'	2856.421	2882.647	26.22558	'PBG'
'J51384'	2882.647	2909.874	27.22747	'PBG'
'J51444'	2909.874	2962.804	52.93006	'PBG'
'J51514'	2962.804	3019.912	57.10775	'PBG'
'J56314'	3019.912	3084.417	64.50536	'PBG'
'AP5121'	3084.417	3088.198	3.780718	'PBG'
'AP5131'	3088.198	3100.674	12.47637	'PBG'
'AP5141'	3100.674	3123.359	22.68431	'PBG'
'AP5171'	3123.359	3188.765	65.40643	'PBG'
'AP5191'	3188.765	3283.283	94.51796	'PBG'
'S18561'	3283.283	3294.852	11.569	'PBG'
'R81757'	3294.852	3313.945	19.09263	'PBG'
'7010PT'	3313.945	3412.508	98.56333	'PBG'
'7011PT'	3412.508	3544.266	131.758	'PBG'
'7012PT'	3544.266	3679.238	134.9716	'PBG'
'7013PT'	3679.238	3885.665	206.4272	'PBG'
'WIPK29'	3885.665	3897.221	11.5564	'PBG'

The above solution with the orders over time can be seen in Figure 6.

Figure 6: Example of an actual solution per shift

7. Alternative Solutions

Three different solutions were investigated.

Solution 1: Sorting the orders according to EDD (discussed in section 5.3). In this problem, the orders can be handled the same as with FIFO, as all orders have the same due date. The orders having the greatest waiting time is thus prioritised.

Solution 2: Sorting the orders according to colour, therefore minimising the number of colour changes and the cost of colour changes. This solution can, however, cause bad customer service.

Solution 3: Firstly, grouping the orders according to their different colours, and then ordering these groupings according to those that have the orders with the greatest waiting time, thus combining solution 1 and 2 above.

8. Data Analysis and Best Solution

The three models were run for 4 different weeks, and the objective function associated with the different solutions can be seen in Figure 7. All data was for 2017 on the dates shown below.

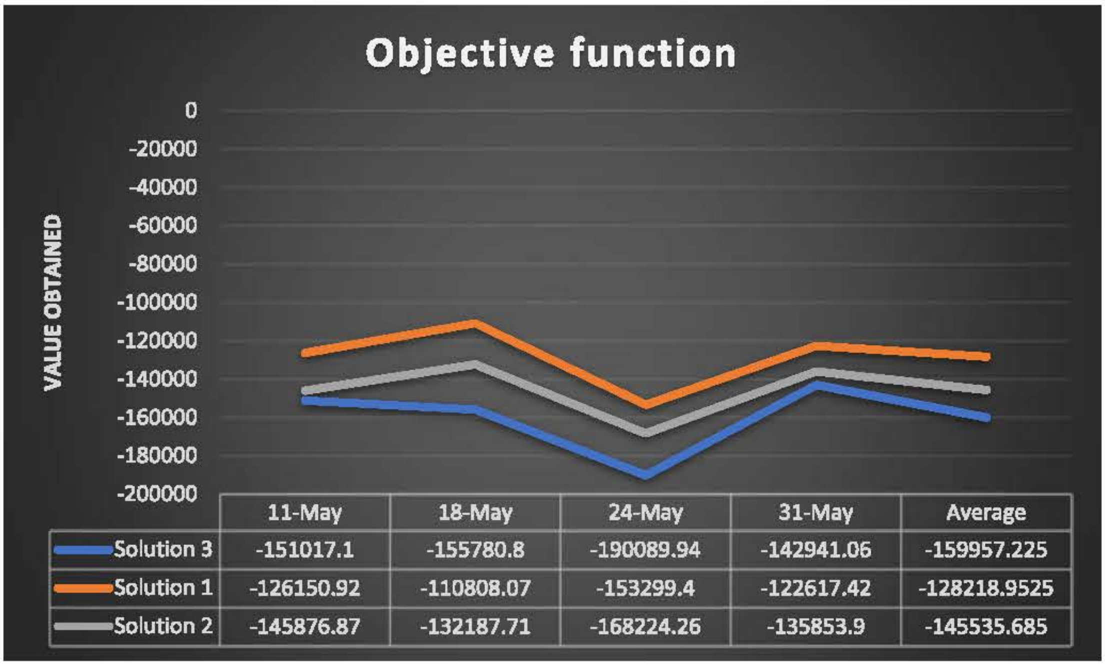

Figure 7: Values obtained with the objective function

Figure 7 shows that solution 3 was the best in all four instances. Solution 2 was the second-best in all instances, although sometimes close to solution three. Solution 1 can be seen to be least effective solution.

Because the objective function weighs the due dates (waiting time) and number of sqm done of all orders against the cost of colour changes needed to powder coat those orders in a specific sequence, it can be said that solution 3 would be the best solution considering all options. This solution gives the most effective value obtained from all three proposed solutions. Because this function is the best in all cases investigated it is safe to recommend solution 3 to the company, without being concerned that the wrong solution is used. This solution is further discussed in section 9.

9. Solution Validation and Verification

9.1 Project Aim Achievement

The selected solution method does not prioritise any orders based on colour, or how big an order is. The problem discussed in section 4.2 and section 4.6 would naturally thus not be a solution. The problems discussed in section 4.2 and section 4.6 is a problem because of human intervention which is not objective and does not take all factors into account. By following the proposed solution, it can be ensured that no decisions take place without the proper knowledge of the outcome of such decisions.

The problems discussed in section 4.3 mentions the fact that there takes on average 18 change overs place per shift. With Solution 3, an average of 12.75 change overs takes place, as shown in Figure 8. This solution provides a reducement of 29% in change overs. This reducement in colour changes creates an R118 636.71 in cost savings per month, which definitely makes it worthwhile.

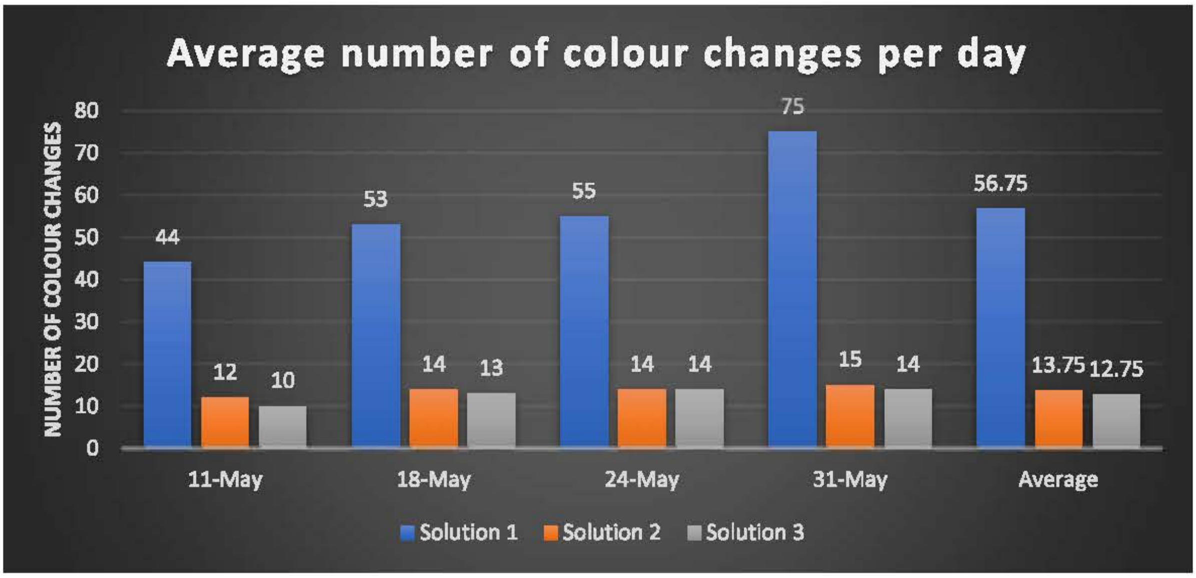

Figure 8: Number of colour changes of different solutions

One problem that might be noted, is that maybe the number of sqm done each day is not sufficient for the production needed by the powder coating department (although it was taken into account when developing the objective function). This point, however, is found to not be a problem, as seen in Figure 9. The average of the solution is just below 12 000 sqm (which is the average sqm per day at present).

Figure 9: Sqm per day of solution 3

With this solution, the number of colour changes is reduced, saving R118 636.71 per month, while maintaining lead time and average sqm per day. This solution is thus a very effective solution.

9.2 Viability to Company

The project findings were submitted to Wispeco Aluminium on 14 October 2017. They found it to be an efficient program and it will be used at the powder coating department.

9.3 Sensitivity Analysis

The three parameters that are changed is the conversion factor with which the colour change is multiplied, the policy of goodwill with which benefit 1 is multiplied, and the number of points that the FLM receives per sqm that is finished. The amount with which the parameters are multiplied, as well as the total cost in that regard, is shown Table 4. All values change as expected, and no surprises are seen.

Table 4: Sensitivity analysis

22-May						
20%	40%	60%	80%	100%	120%	140%
Changing benefit 2: The points the FLM receives for each sqm finished						
-50022	-70406.9	-90791.9	-111177	-131562	-151947	-172332
Changing benefit 1: The policy of goodwill is changed						
-100169	-108017	-115865	-123714	-131562	-139410	-147258
Cost 1: Changing the cost of colour changes						
-139245	-137324	-135403	-133483	-131562	-129641	-127720

Table 5: The percentage of change to the final cost

22-May	% Change				
40%	60%	80%	100%	120%	140%
Changing benefit 2: The points the FLM receives for each sqm finished					
0.550069	0.366712	0.183357	0	0.183356	0.366712
Changing benefit 1: The policy of goodwill is changed					
0.190315	0.126877	0.063438	0	0.063438	0.126877
Changing cost 1: The cost of colour changes is changed					
0.04317	0.02878	0.01439	0	0.01439	0.02878

From Table 5 it can be seen that the policy of goodwill has almost no effect on the final cost, with 6% being the most change at a 40% of the original policy. Changing benefit 1 has a slightly higher impact, with 19% being the highest change. Changing benefit two has a high impact, with a 37% change when changing the points to 40% more. The points that the FLM receives must be carefully considered when changing it. Making sure that the correct number of points are given now might also be an important point to make certain of, as this policy can highly affect the functioning of the program.

10. Conclusion and Recommendations

The written program which was chosen combines two other heuristics. The program first groups the orders together according to the different colours they must be powder coated and thereafter sort the orders according to EDD (as discussed in section 5.3). In other words, for each colour section, the orders within the specific section would be sorted according to the amount of waiting time that the orders are already in the receiving area, and the orders with the highest waiting time will be powder coated first.

The problems discussed in section 4.4 and section 4.5 are not that crucial to the powder coating department but can be considered when further enhancing the program. That is because the CUBE only stops working in extreme cases when the planner itself can foresee that it will be a problem (from section 4.4) and that the powder being stolen is not a problem yet, but may pose a problem in the future. These functions can be added to the preliminary solution provided in this document. It is further possible to solve the problem as a multi-objective optimization problem, as seen in section 5.

References

Anghinolfi, D., & Paolucci, M. (2009). A New Discrete Particle Swarm Optimization Approach for the Single-Machine Total Weighted Tardiness Scheduling Problem with Sequence-Dependent Setup Times. *European Journal of Operational Research*, 193(1), pp.73–85. Retrieved 2017, from http://www.sciencedirect.com.uplib.idm.oclc.org/science/article/pii/S0377221707010727

Arlow, R. (2016). *Single machine production scheduling with sequence dependent setup times at an Aluminium powder coating plant.* University of Pretoria, Dissertation: Final Year Industrial Engineering.

Bianchi, L., Dorigo, M., Gambardella, L., & Gutjahr, W. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing, vol. 8, no. 2*, 239-287. Retrieved from https://search-proquest-com.uplib.idm.oclc.org/docview/204868352?OpenUrlRefId=info:xri/sid:wcdiscovery&accountid=14717

Biskup, D. (1999). Single-Machine Scheduling with Learning Considerations. *European Journal of Operational Research*, 115(1), pp.173–178. Retrieved 2017, from http://www.sciencedirect.com.uplib.idm.oclc.org/science/article/pii/S037722179800246X

Biskup, D., & Jahnke, H. (2001). Common Due Date Assignment for Scheduling on a Single Machine with Jointly Reducible Processing Times. *International Journal of Production Economics*, 69(3), pp.317–322. Retrieved 2017, from http://www.sciencedirect.com.uplib.idm.oclc.org/science/article/pii/S0925527300000402

Chambers, R., Carraway, R., Lowe, T., & Morin, T. (1991). Dominance and Decomposition Heuristics for Single Machine Scheduling. *Operations research*, vol. 39, no. 4, pp. 639. Retrieved 2017, from http://search.proquest.com.uplib.idm.oclc.org/docview/219131805?OpenUrlRefId=info:xri/sid:wcdiscovery&accountid=14717

Cherry, K. (2016, 06 20). What Is a Heuristic and How Does It Work? *verywell*. Retrieved 2017, from https://www.verywell.com/what-is-a-heuristic-2795235

Dayama N.R. et al. (2015). History-Dependent Scheduling: Models and Algorithms for Scheduling with General Precedence and Sequence Dependence. *Computers and Operations Research*, 64, pp.245–261. Retrieved 2017, from http://www.sciencedirect.com.uplib.idm.oclc.org/science/article/pii/S0305054815001525

Dictionary. (2017). *Dictionary.com*. Retrieved from The definition of scheduling: http://www.dictionary.com/browse/scheduling

Glover, F. (1986). Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research*, 13(5), pp.533–549. Retrieved 2017, from http://www.sciencedirect.com.uplib.idm.oclc.org/science/article/pii/0305054886900481?_rdoc=1&_fmt=high&_origin=gateway&_docanchor=&md5=b8429449ccfc9c30159a5f9aeaa92ffb

Grobler, J. (2009). *Particle swarm optimization and differential evolution for multi-objective multiple machine scheduling.* University of Pretoria, Dissertation: MEng Industrial Engineering.

Hall, J. C. (2008). HEURISTICS. *ANZ Journal of Surgery*, 78: 1051. Retrieved 2017, from http://onlinelibrary.wiley.com.uplib.idm.oclc.org/doi/10.1111/j.1445-2197.2008.04748.x/full

Hochbaum, D. (1999). *The Scheduling Problem.* Retrieved from RIOT: http://riot.ieor.berkeley.edu/Applications/Scheduling/algorithms.html

Hoitomt, D., Luh, P., & Pattipati, K. (1993). A Practical Approach to Job-Shop Scheduling Problems. *IEEE Transactions on Robotics and Automation*, 9(1), pp.1–13. Retrieved 2017, from http://ieeexplore.ieee.org.uplib.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=210791

Hollinger, D. (2017). Heuristic Search. Retrieved 2017, from http://sequoia.ict.pwr.wroc.pl/~witold/ai/Dave_Hollinger_RPI/heuristicsearch.pdf

Hu, X. (2006). *Particle Swarm Optimization.* Retrieved 2017, from http://www.swarmintelligence.org/tutorials.php

Isaacs, R. (1955). Naval research logistics quarterly Volume 2. In *The Problem of Aiming and Evasion* (pp. 47-67). University of California. Retrieved 2017, from https://babel.hathitrust.org/cgi/pt?id=uc1.b3244129;view=1up;seq=7

Jain, A., & Meeran, S. (1999). Deterministic Job-Shop Scheduling: Past, Present and Future. *European Journal of Operational Research*, 113(2), pp.390–434. Retrieved 2017, from http://www.sciencedirect.com.uplib.idm.oclc.org/science/article/pii/S0377221798001131

Lauff, V., & Werner, F. (2004). Scheduling with Common Due Date, Earliness and Tardiness Penalties for Multimachine Problems: A Survey. *Mathematical and Computer Modelling*, 40(5), pp.637–655. Retrieved 2017, from http://www.sciencedirect.com/science/article/pii/S0895717704803285

Lopez, P., & Roubellat, F. (2008). *Lopez, P. & Roubellat, F.* John Wiley & Sons.

Malakooti, B. (2013). *Operations and Production Systems with Multiple Objectives.* John Wiley & Sons.

Nearchou, A. (2008). A Differential Evolution Approach for the Common Due Date Early/tardy Job Scheduling Problem. *Computers and Operations Research*, 35(4), pp.1329–1343. Retrieved 2017, from http://www.sciencedirect.com.uplib.idm.oclc.org/science/article/pii/S0305054806002024

Nieberg, T. (2010). *Single Machine Scheduling.* Retrieved 2017, from Research Institute for Discrete Mathematics: http://www.or.uni-bonn.de/lectures/ss10/scheduling_data/sched10_2.pdf

Pan, Q., Tasgetiren, M., & Liang, Y. (2006). Minimizing Total Earliness and Tardiness Penalties with a Common Due Date on a Single-Machine Using a Discrete Particle Swarm Optimization Algorithm. *Lecture notes in Computer Science pages* (pp. 460-467, Vol: 4150). Berlin, Heidelberg: Springer.

Panwalker, S., Dudek, R. A., & Smith, M. (1973). Sequencing research and the industrial scheduling problem. *Symposium on the Theory of Scheduling and its Applications* (pp. 29-38). Berling: Springer.

PEÑA, M. J. (2017). *Introduction and Simple Heuristics*. Master's degree in Modelling, Science and Engineering. Retrieved 2017, from Introduction and Simple: http://mat.uab.cat/~alseda/MasterOpt/IntroHO.pdf

Pinedo, M. (2016). *Basic Scheduling Algorithms for Single Machine Problems.* Retrieved 2017, from http://community.stern.nyu.edu/om/faculty/pinedo/scheduling/shakhlevich/handout02.pdf

Robin. (2009, December 14). Generate-And-Test Search. *World of Computing*. Retrieved 2017, from http://intelligence.worldofcomputing.net/ai-search/generate-and-test-search.html#.WQxtDeWGPIV

Theuwissen, M. (2017, 01 08). Kdnuggets.com. Retrieved from R vs Python for Data Science: The Winner is ...: http://www.kdnuggets.com/2015/05/r-vs-python-data-science.html

T'kindt, V., & Billaut, J. (2002). *Multicriteria scheduling : theory, models and algorithms.* Berlin: Springer.

Vaucher, P. (2017). Research Gate. Retrieved 01 10, 2017, from https://www.researchgate.net/post/Is_it_a_good_idea_to_use_Microsoft_Excel_as_a_statistical_software_package_for_analyzing_the_data_Is_it_acceptable_for_the_journals

Vermeulen, T., & Van den Akker, R. (2010). What meta means and does not mean. *Notes on Metamodernism*. Retrieved from http://www.metamodernism.com/2010/10/14/what-meta-means-and-does-not-mean/

Webster, S., & Baker, K. (1995). Scheduling groups of jobs on a single machine. *Operations research*, vol. 43, no. 4, pp. 692. Retrieved 2017, from http://search.proquest.com.uplib.idm.oclc.org/docview/219176594?OpenUrlRefId=info:xri/sid:wcdiscovery&accountid=14717

WISPECO PTY (LTD). (2017). *Wispeco*. Retrieved from http://www.wispeco.co.za/

Appendix A: Industry Sponsorship Form

Department of Industrial & Systems Engineering
Final Year Projects
Identification and Responsibility of Project Sponsors

All Final Year Projects are published by the University of Pretoria on *UPSpace* and thus freely available on the Internet. These publications portray the quality of education at the University and have the potential of exposing sensitive company information. It is important that both students and company representatives or sponsors are aware of such implications.

Key responsibilities of Project Sponsors:

A project sponsor is the key contact person within the company. This person should thus be able to provide the best guidance to the student on the project. The sponsor is also very likely to gain from the success of the project. The project sponsor has the following important responsibilities:

1. Confirm his/her role as project sponsor, duly authorised by the company. Multiple sponsors can be appointed, but this is not advised. The duly completed form will considered as acceptance of sponsor role.
2. Review and approve the Project Proposal, ensuring that it clearly defines the problem to be investigated by the student and that the project aim, scope, deliverables and approach is acceptable from the company's perspective.
3. Review the Final Project Report (delivered during the second semester), ensuring that information is accurate and that the solution addresses the problems and/or design requirements of the defined project.
4. Acknowledges the intended publication of the Project Report on UP Space.
5. Ensures that any sensitive, confidential information or intellectual property of the company is not disclosed in the Final Project Report.

Project Sponsor Details:

Company:	Wispeco Pty Ltd
Project Description:	
Student Name:	
Student number:	
Student Signature:	
Sponsor Name:	Esbé du Toit
Designation:	Organisational Improvement Manager B.Eng (Industrial)
E-mail:	esbe@wispeco.co.za
Tel No:	011 389 0034
Cell No:	084 584 1110
Fax No:	—
Sponsor Signature:	EduToit

Appendix B: Python Programming File

```python
# -*- coding: utf-8 -*-
"""
Created on Sat Sep 02 13:47:52 2017

@author: Karien
"""


import StringIO
import csv
from operator import itemgetter

from scipy.optimize import minimize
import Def_rosen




sqmlys=[]
cclys=[]
K=[]
m=[]
a=[]
lys=[0]
planning=[]
tel=1
'total time for colour change'
t=[]
c=0
Wano=[]
'wano, colour, enter date, mil/job = F, G, sqm, bye, row, height'
import numpy as np
order=np.genfromtxt('wispeco.csv', delimiter=',', skip_header=1,
                    dtype=('S30, S30, float, S30, float, float, float, float, float,
#print order
' Speed of conveyor [m/s]'
S=2.0
'Average [sqm/s] (average of 12 000 sqm/day and there is 75600s/day for 21 ho
A=0.1587
k=1
'Sqm converted to kg [kg/sqm]'
G=2.9
'Average sqm/m per day [sqm/m]'
B=A/S
'Average kg/m per day [kg/m]'
C=G*B
'Time per colour change [s]'
Z1=240.0
Z2=420.0

cost=1
P=1
tel=0
Tel=0
cost1=1
cost2=1
k=1
c=1
m=[]
a=[]
```

```
#x=order
x=sorted(order, key=itemgetter(1, 2))
#print x
#print x
#print x
'colour change tyd'
for i in range(len(order)-1):
    if x[i][1]!=x[i+1][1]:
#        print x[i][1]
#        print i
        lys.append(i+1)
#        print lys
#        while tel<20:
#            tel=tel+1
##            print tel
#
##            print (str('Gaar oor na kleur ')+  x[i+1][1]+ str(' na '),  i+1
#            lys.append(i+1)
##            print lys
lys.append(len(order))
#print lys

J=[]
cccost=[]
cclys=[]


#print cccost
#print len(cccost)
#print cclys
#print cost1

'Production tyd constraint'
while Tel<1:

    for j in range(len(x)):
        Tel=Tel+1
        m.append(x[j])
#        print j
#        print m
        a.append(m[j][4])
        Q=sum(a)
        P=Q/A
#        print P
        '21 ure is 75600 sekondes at 80% efficiency'
        if P<75600:
#            print P
            planning.append(x[j])
        'allows for orders to be carried over'
#print len(planning)
#print planning
#cost1=cccost[len(planning)-1]


#for z in range(len(lys)-1):
#    D=[]
#
#    for s in range(lys[z], lys[z+1]):
##        reeks = lys[z+1]-lys[z]
##        print reeks
```

```
#        D.append(x[s][4])
#        J=sum(D)
#
#    if J>=40:
#            k=k+343/0.8
#            c=c+Z2/0.8
##            print k
#    else:
#            c=c+Z1/0.8
#            k=k+710/0.8
##            print k
#    cclys.append(c)
#    cccost.append(k)
k=1
K=[]
kleur=0
for i in range(len(x)-1):
#    K=1
    if x[i][1]!=x[i+1][1]:
        kleur=kleur+1
        k=k+600
        K.append(k)
    else:
        k=k
        K.append(k)

G=[0.2,0.4,0.6,0.8,1.0,1.2,1.4,4.6,1.8]
Cost1=[]
for g in range(len(G)):

    cost1=4*K[len(planning)-1]*G[g]
    Cost1.append(cost1)
#print kleur/2.0
TC=[]
'choose goodwill cost as R40/day per order'
for u in range(len(x)):
    cost = cost + x[u][13]*40
    tccost=1.0/cost
#    print cost
    TC.append(cost)


cost2=TC[len(planning)-1]




Sm=1
#print planning
sqm=[]


for l in range (len(x)):
    Sm = Sm + x[l][4]*10
    sqm.append(Sm)
#print sqm
#print len(planning)


cost3=sqm[len(planning)-1]
```

```
#print Cost3
tt=[]
#print 'sqm is ' + str(cost3/10.0)
#print cost1
#print cost2
#print cost3
'Minus cost 2 omdat hoe groter die cost 2 is hoe beter, aangesien dit cost of
for h in range(len(G)):

    Totalcost=Cost1[h]-cost2-cost3
    E=round(Totalcost, 1)
    tt.append(E)
print tt
#print Totalcost
dag= (order[0][6])
#print dag
#print 'The total Rand value on ' + str(dag) + ' is ' + str(Totalcost)
```

```python
# -*- coding: utf-8 -*-
"""
Created on Sat Sep 02 13:41:45 2017

@author: Karien
"""


import StringIO
import csv
from operator import itemgetter

from scipy.optimize import minimize
import Def_rosen




sqmlys=[]
cclys=[]
K=[]
m=[]
a=[]
lys=[0]
planning=[]
tel=1
'total time for colour change'
t=[]
c=0
Wano=[]
'wano, colour, enter date, mil/job = F, G, sqm, bye, row, height'
import numpy as np
order=np.genfromtxt('wispeco.csv', delimiter=',', skip_header=1,
                    dtype=('S30, S30, float, S30, float, float, float, float,
#print order
' Speed of conveyor [m/s]'
S=2.0
'Average [sqm/s] (average of 12 000 sqm/day and there is 75600s/day for 21 ho
A=0.1587
k=1
'Sqm converted to kg [kg/sqm]'
G=2.9
'Average sqm/m per day [sqm/m]'
B=A/S
'Average kg/m per day [kg/m]'
C=G*B
'Time per colour change [s]'
Z1=240.0
Z2=420.0

cost=1
P=1
tel=0
Tel=0
cost1=1
cost2=1
k=1
c=1
m=[]
a=[]
```

```
#x=order
x=sorted(order, key=itemgetter(2))
#print x
#print x
#print x
'colour change tyd'
for i in range(len(order)-1):
    if x[i][1]!=x[i+1][1]:
#        print x[i][1]
#        print i
        lys.append(i+1)
#        print lys
#        while tel<20:
#            tel=tel+1
##            print tel
#
##            print (str('Gaar oor na kleur ')+  x[i+1][1]+ str(' na '),  i+1
#            lys.append(i+1)
##            print lys
lys.append(len(order))
#print lys

J=[]
cccost=[]
cclys=[]


#print cccost
#print len(cccost)
#print cclys
#print cost1

'Production tyd constraint'
while Tel<1:

    for j in range(len(x)):
        Tel=Tel+1
        m.append(x[j])
#        print j
#        print m
        a.append(m[j][4])
        Q=sum(a)
        P=Q/A
#        print P
        '21 ure is 75600 sekondes at 80% efficiency'
        if P<70200:
#            print P
            planning.append(x[j])
        'allows for orders to be carried over'
#print len(planning)
#print planning
#cost1=cccost[len(planning)-1]


#for z in range(len(lys)-1):
#    D=[]
#
#    for s in range(lys[z], lys[z+1]):
##        reeks = lys[z+1]-lys[z]
##        print reeks
```

```
#        D.append(x[s][4])
#        J=sum(D)
#
#    if J>=40:
#            k=k+343/0.8
#            c=c+Z2/0.8
##            print k
#    else:
#            c=c+Z1/0.8
#            k=k+710/0.8
##            print k
#    cclys.append(c)
#    cccost.append(k)
k=1
K=[]
kleur=0
for i in range(len(x)-1):
#    K=1
    if x[i][1]!=x[i+1][1]:
        kleur=kleur+1
        k=k+600
        K.append(k)
    else:
        k=k
        K.append(k)
cost1=4*K[len(planning)-1]
#print kleur/2.0
#print 75*600
TC=[]
'choose goodwill cost as R40/day per order'
for u in range(len(x)):
    cost = cost + x[u][13]*40
    tccost=1.0/cost
#    print cost
    TC.append(cost)
cost2=TC[len(planning)-1]



Sm=1
#print planning
sqm=[]
for l in range (len(x)):
    Sm = Sm + x[l][4]*10
    sqm.append(Sm)
#print sqm
#print len(planning)
cost3=sqm[len(planning)-1]
print cost1
print cost2
print cost3
'Minus cost 2 omdat hoe groter die cost 2 is hoe beter, aangesien dit cost of
Totalcost=cost1-cost2-cost3
#print Totalcost
dag= (order[0][6])
#print dag
print 'The total Rand value on ' + str(dag) + ' is ' + str(Totalcost)
```

```
# -*- coding: utf-8 -*-
"""
Created on Sat Sep 02 13:37:32 2017

@author: Karien
"""


import StringIO
import csv
from operator import itemgetter

from scipy.optimize import minimize
import Def_rosen




sqmlys=[]
cclys=[]
K=[]
m=[]
a=[]
lys=[0]
planning=[]
tel=1
'total time for colour change'
t=[]
c=0
Wano=[]
'wano, colour, enter date, mil/job = F, G, sqm, bye, row, height'
import numpy as np
order=np.genfromtxt('wispeco.csv', delimiter=',', skip_header=1,
                    dtype=('S30, S30, float, S30, float, float, float, float, float,
#print order
' Speed of conveyor [m/s]'
S=2.0
'Average [sqm/s] (average of 12 000 sqm/day and there is 75600s/day for 21 ho
A=0.1587
k=1
'Sqm converted to kg [kg/sqm]'
G=2.9
'Average sqm/m per day [sqm/m]'
B=A/S
'Average kg/m per day [kg/m]'
C=G*B
'Time per colour change [s]'
Z1=240.0
Z2=420.0

cost=1
P=1
tel=0
Tel=0
cost1=1
cost2=1
k=1
c=1
m=[]
a=[]
```

```
#x=order
x=sorted(order, key=itemgetter(1))
#print x
#print x
#print x
'colour change tyd'
for i in range(len(order)-1):
    if x[i][1]!=x[i+1][1]:
#        print x[i][1]
#        print i
        lys.append(i+1)
#        print lys
#        while tel<20:
#            tel=tel+1
##            print tel
#
##            print (str('Gaar oor na kleur ')+  x[i+1][1]+ str(' na '),  i+1
#            lys.append(i+1)
##            print lys
lys.append(len(order))
#print lys

J=[]
cccost=[]
cclys=[]


#print cccost
#print len(cccost)
#print cclys
#print cost1

'Production tyd constraint'
while Tel<1:

    for j in range(len(x)):
        Tel=Tel+1
        m.append(x[j])
#        print j
#        print m
        a.append(m[j][4])
        Q=sum(a)
        P=Q/A
#        print P
        '21 ure is 75600 sekondes at 80% efficiency'
        if P<75600:
#            print P
            planning.append(x[j])
        'allows for orders to be carried over'
#print len(planning)
#print planning
#cost1=cccost[len(planning)-1]


#for z in range(len(lys)-1):
#    D=[]
#
#    for s in range(lys[z], lys[z+1]):
##        reeks = lys[z+1]-lys[z]
##        print reeks
```

```
#        D.append(x[s][4])
#        J=sum(D)
#
#    if J>=40:
#            k=k+343/0.8
#            c=c+Z2/0.8
##            print k
#    else:
#            c=c+Z1/0.8
#            k=k+710/0.8
##            print k
#    cclys.append(c)
#    cccost.append(k)
k=1
K=[]
kleur=0
for i in range(len(x)-1):
#    K=1
    if x[i][1]!=x[i+1][1]:
        kleur=kleur+1
        k=k+600
        K.append(k)
    else:
        k=k
        K.append(k)
cost1=4*K[len(planning)-1]
print kleur/2.0


TC=[]
'choose goodwill cost as R40/day per order'
for u in range(len(x)):
    cost = cost + x[u][13]*40
    tccost=1.0/cost
#    print cost
    TC.append(cost)
cost2=TC[len(planning)-1]




Sm=1
#print planning
sqm=[]
for l in range (len(x)):
    Sm = Sm + x[l][4]*10
    sqm.append(Sm)
#print sqm
#print len(planning)
cost3=sqm[len(planning)-1]
print cost1
print cost2
print cost3
'Minus cost 2 omdat hoe groter die cost 2 is hoe beter, aangesien dit cost of
Totalcost=cost1-cost2-cost3
#print Totalcost
dag= (order[0][6])
#print dag
print 'The total Rand value on ' + str(dag) + ' is ' + str(Totalcost)
```

Appendix C: Excel Sheets

Variable Expenses	Measure unit		A-Colours	Small colours	
Powder	R/sqm		6.2595	7.8752	
Chemicals	R/sqm		1.391	1.391	
Losses	R/sqm		0.1284	0.1926	
Plastic	R/sqm		1.1877	1.1877	
Total Cost			8.9666	10.6465	
Average sqm/s =		0.1587			
Total Cost per Second			1.4229994	1.68959955	
			A-Colours	Small colours	
Total cost for spray-to-waste colour change:			341.51986	405.503892	
Total cost for reclaim colour change:			597.65976	709.631811	
1 uneccesary CC per month			13148.515	15611.89984	28760.41
			7513.4369	8921.085624	16434.52
TC assuming 1 colour change of each type each day (thus 2 types per shift)					45194.94
Lost Opportunity cost				-266508.0192	285163.6
Average of		18	colour changes / shift		
		792	colour changes / month		
		513.5788301	R/colour change		
Total cost on colour change		406754.4334			
If 10% of colour changes ar		40675.44334	118636.71		

18
12.75
0.291667