Production scheduling of Aluminium Anodising plant

Carina Behr, 14010047

September 28, 2017

Project Report submitted in partial fulfilment of the requirements for the module BPJ 410 Project at the Department Industrial and Systems Engineering, University of Pretoria.

DECLARATION OF ORIGINALITY

I, Carina Behr, student number 14010047, 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.

Abstract

Wispeco Aluminium decided to reduce a double line aluminium anodising plant to a single line. This change forces the anodising plant to run at maximum capacity. To be able to run at maximum capacity, it is suggested that a production schedule be put in place to optimise the movement of the cranes and minimise the makespan of a day's work.

Production scheduling literature, including the popular α / β / γ notation was investigated to assist with identifying the characteristics of the relevant problem. An important characteristic that was identified is that the anodising problem is a flow shop problem. Once these characteristics were identified, peer-reviewed articles were investigated to identify similar problems. An article was identified which addresses the multiple cranes that cannot cross over one another, the flow shop characteristic and the parallel machines. This article was used to start formulating an algorithmic model as solution. This algorithm was coded in Python. To ensure that the results generated by the algorithm are relevant and correct, the results were compared to actual time-studies. The model tested the influence of three heuristic scheduling approaches on the makespan of the products. The heuristics tested were FIFO, Priority orders and Shortest Processing Times. The results compared over three production days indicated that the schedule does not have a large impact on the makespan of a day's work. This project suggests that the scheduling on the anodising plant is continued as it currently operates. The current method might not be the optimal solution for any given day, but produces an adequate makespan, which will not disrupt the normal processes that workers are used to.

Keywords: Production scheduling, Anodising, Flow shop

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Production scheduling is a significant part of optimising processes and facilities. Wispeco Aluminium (the company where this project is executed) has decided that their anodising plant needs to be improved by using production scheduling techniques. This chapter serves as an introduction to the company, the plant, and the problem, while focusing on the reasoning behind the project and how the outcomes will help Wispeco. In chapter 2, a literature study will describe production scheduling approaches and a few example articles will be revised. These articles will serve as a reference for the solution of the project, of which an overview will be given in chapter 4, which follows the as-is process described in chapter 3. Chapter 5 will conclude the report by giving a summary of what was discussed and what will be done to complete the project.

1.1 Introduction and Background

1.1.1 Wispeco Aluminium

Wispeco Aluminium is the leading aluminium extruder in Africa. The head office is located in Alberton, with smaller factories all over South Africa. Wispeco has three main units. The aluminium recycling and billet casting is where aluminium ore, scrap and other metals are melted together to create the ideal aluminium alloy for extrusion. The extrusion unit has its own die manufacturing section [1]. The two surfacing units are powder coating and anodising, which has South African bureau of Standards (SABS) certification.

Wispeco defines an extrusion as the long piece of aluminium that has been extruded into a specific form, before the piece of aluminium has been finished. These aluminium extrusions are used in transport, energy, agricultural, general engineering, automotive and other industries [15]. The company has over 3 000 different dies that are used to create extrusions. More or less 2 500 of these die forms are anodised in a financial year [11].

1.1.2 Anodising Plant

This project will focus on the finishing process called anodising. Anodising is an electrolytic reaction used to produce a layer of aluminium oxide on the aluminium alloy [1]. This layer functions as both decoration and protection of the aluminium. Currently, at the anodising plant of Wispeco, production is done on two lines: The manual anodising plant (MAP) and the automated anodising plant(AAP). Both lines have two teams, with 11 staff members per team for the AAP and 12 members per team at the MAP. The plant is currently producing jobs in the most convenient way for the workers and not the most efficient way. According to Xin-She Yang [21], an optimal solution can be identified by pursuing detailed studies and tested methods. The two lines share one support team of 13 members and a dispatch team with 16 members. The anodising plant has its own quality control, packaging and dispatch teams. The extrusions that enter or exit the anodising plant are referred to as profiles. Three anodic layer thickness options are produced by the plant (25μm, 15μm and 10μm). The thickness of this layer is determined by the duration of time that the profiles are submerged in the anodising tank. The thicker layers are for coastal customers, while inland customers do not require as thick a layer. Profiles are hung onto a structure called a beam, which is moved through the anodising process by a crane. All profiles on a beam, are as rule, orders with the same layer thickness. Currently, the plant produces five different colours of anodising or a natural finish. The colours vary from bronze to black, with three shades in between.

Figure 1.1 displays a few beams with profiles jigged (hung) onto the beam and ready for anodising.

The cranes that move the beams from tank to tank through the anodising process

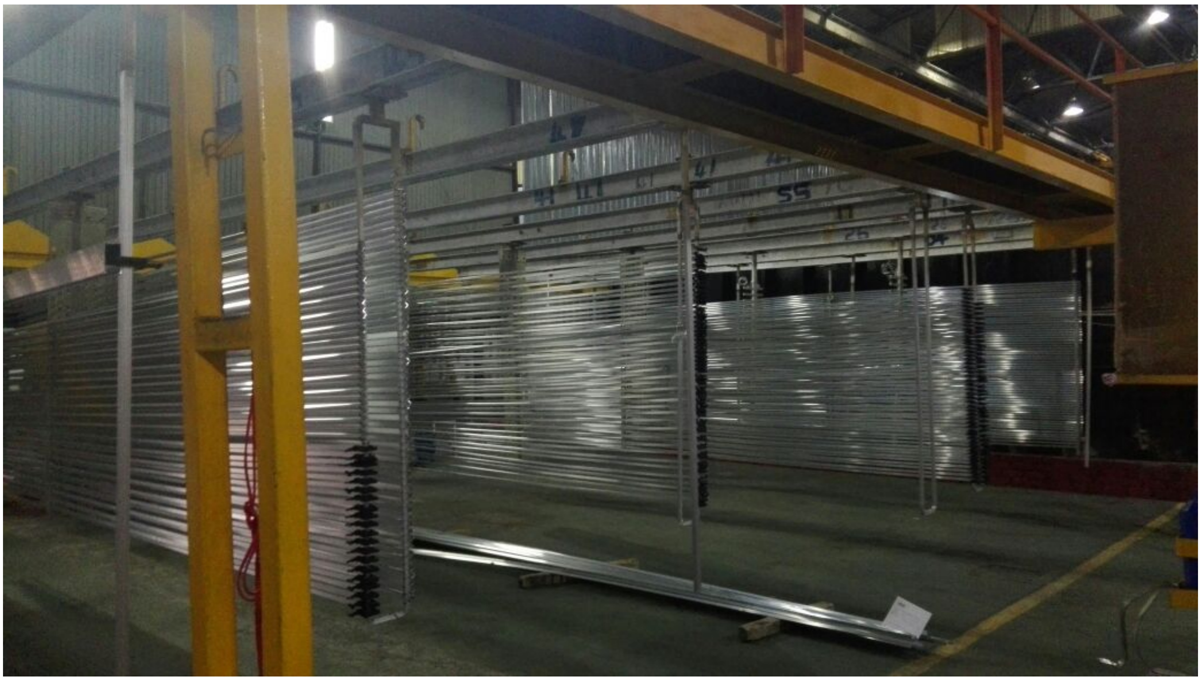


Figure 1.1: Beam holding profiles

are shown in figure 1.2. A more detailed study of the plant continues in chapter 2.

Currently, the two lines combined produce approximately 140 000m^2 of anodised aluminium per month. The market for anodising as a finishing process is declining. The main reason for this decline, according to Wispeco's management, is that Chinese manufacturers can produce the same product at a third of the cost. Powder coating is also a cheaper alternative in the current market. The AAP theoretically has the capacity to produce the total current throughput, thus management has decided to produce on a single line. If the AAP produces optimally, the single line should be able to meet demand. Anodising is not expected to grow significantly in the coming years, because of the cheaper alternative of powder coating. There is, however, still a place for anodising, as the finishing layer is more durable and protects the aluminium better than powder coated surfaces [14].

During a research project [10], the influence of variables on the thickness of the anodised layer was investigated. No new assignable causes were identified. During this project, the researcher noted that many of the profiles are over-anodised (over-processed).

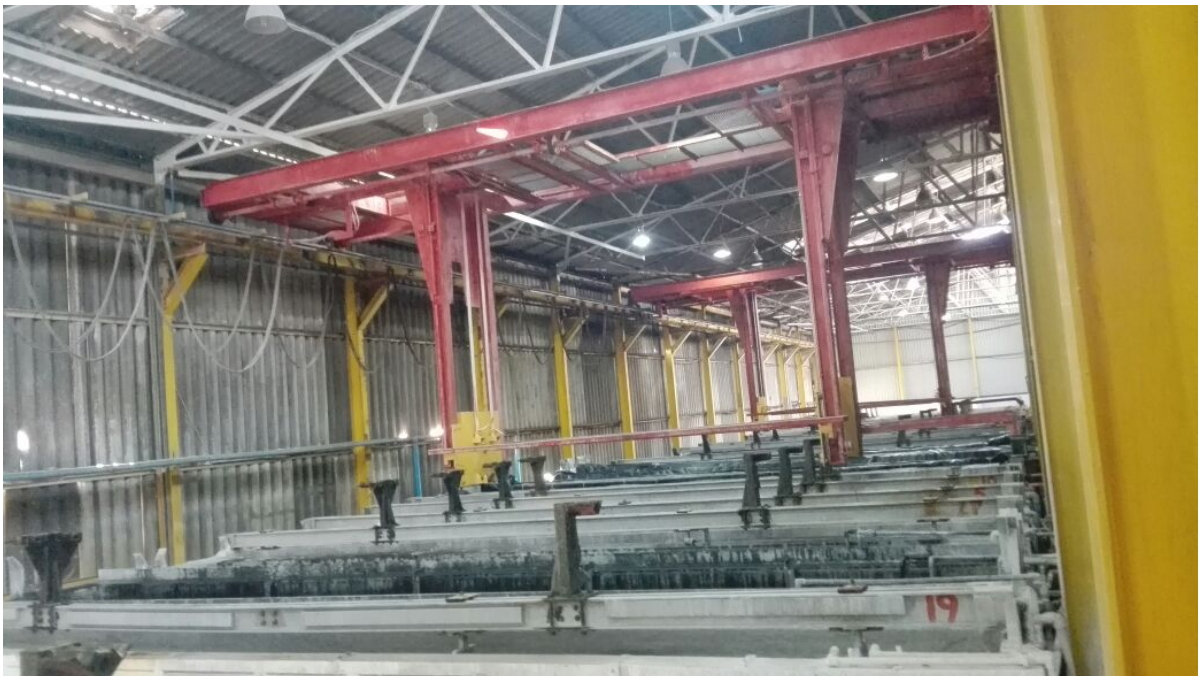


Figure 1.2: Cranes moving beams through anodising process

The setup times to change between layer thickness on the AAP is significant. To keep from losing production time, the operators anodise all 10μm profiles on a 15μm cycle. The 25μm are anodised longer, but as the change over from 25μm to 15μm takes place, the first two or three 15μm beams are over anodised.

Cranes are used to move the beams through the anodising process. There are currently two cranes on the AAP. The cranes are programmed to move profiles through a specific sequence. A new programmable logic controller (PLC) which can be programmed to move beams at specific times instead of at specific steps in a sequence is being investigated. Some staff members at the plant are of opinion that more cranes are needed to perform optimally however, purchasing new cranes will create added expense to the company.

1.2 Problem Statement

Currently, the change-over time to move between different micron layers takes too long. The different micron layers have predetermined times that the profiles must remain in the anodising tank for satisfactory quality (which is determined by measuring the thickness of the anodised layer in microns). The goal is to reduce the change-over times and minimise the total processing time of one day's production. When the time that the beam has to be in a certain station is up, the crane must react by immediately moving the beam to the next processing station or tank. Beams can be loaded to maximum capacity and the two cranes can move the beams through the process according to a time schedule, based on the anodising time or the time that causes the bottleneck in the process.

1.3 Project Aim and Rationale

A vacation work study was done on the anodising plant to conceptualise a "dream" plant that could compete with Chinese imports. The Wispeco anodising plant currently produces anodised aluminium at R30 per square meter, while the Chinese industry produces at about R10 per square meter. This study concluded that the best way forward is to run current production on a single anodising line. An estimation has been made that, on a single line, production cost will reduce from R30 per square meter of anodising to R21 per square meter [11]. This cost reduction results in an estimated R950 000 savings per month if all current orders are processed. The aim of this project following on the afore-mentioned study, is to make the new goal possible by optimising crane movement within the plant.

The management of the plant's main concern is to improve Overall Equipment Efficiency (OEE). A more detailed literature study will include the review of OEE.

$$OEE = PerformanceEfficiency + Quality + Availability \tag{1.1}$$

There will be a specific focus on the performance efficiency of OEE, which can be calculated with either of the following formulas [6]:

$$PerformanceEfficiency = \frac{ProcessedAmount \times TheoreticalCycleTime}{OperatingTime} \quad (1.2)$$

$$PerformanceEfficiency = NetOperatingRate \times OperatingSpeedRate \quad (1.3)$$

This term usually needs an engineering approach, where availability requires a management approach. The focus of this project will not be on the quality of throughput, however, the suggested solution cannot reduce the quality of the throughput.

Table 1.1 shows the distribution of production in terms of the anodising layer thickness in the financial year (July 2015 to June 2016). The production of each thickness is measured in square meters. Management of the anodising plant records performance measures and production in terms of square meters, as that is the unit in which the anodised finish of aluminium is sold to customers. During the afore-mentioned year, there were some 20μm orders, which can be ignored for future cases, as this order was specifically completed for a priority customer.

Table 1.1: Micron distribution in the anodising plant

Microns	Square meters produced	Percentage of production
10μm	287 244.27	20.80%
15μm	728 899.95	52.78%
20μm	17 494.01	1.27%
25μm	347 326.81	25.15%

To improve the OEE equation 1.1, the suggestion is to dynamically schedule the crane movement with an algorithm. Scheduling could be in the form of an algorithm, a simulation or a combination of these two as described by Herrmann [7]. This project uses an algorithm to solve the scheduling problem and sensitivity analysis to address uncertainty. Production scheduling is mainly focused on improving quality, production and delivery time whilst contributing to a continually improving manufacturing process [9]. The percentage of work allocated to each thickness layer as outlined in Table 1.1 must be taken into consideration when designing this scheduling system.

1.4 Activities, Tasks, Deliverables, Resources and Constraints

The desired deliverable is to build a algorithm which helps to schedule the movement of the cranes that move beams through the anodising process. The program should have scenarios with different numbers of cranes as a form of sensitivity analysis. These scenarios can be used to assist in determining the number of cranes required to work optimally and ensure that the number of cranes is not the bottleneck in the system. If the number of cranes needed (to ensure that the cranes are not the bottleneck) is identified, the new bottleneck must be identified to become a future project at the anodising plant. The theory behind bottlenecks and the use of DBR-scheduling (Drum, Buffer, Rope) [8] will be considered and be incorporated into the schedule if applicable.

Industrial engineering techniques should be used to complete this project. These include motion studies of current crane schedule, source problem identification, developing a dynamic scheduling model, eliminating the current bottleneck and identifying the new bottleneck. A possible problem with the implementation of this scheduling solution could be that the workers in the factory will override the proposed schedule and produce in such a way that suits them better. A possible managing solution could be walking the "gemba" [19], which refers to management who should be walking on the floor. This technique can be used in the pre phase of the project to make sure that the solution is practical and will contribute to the plant sufficiently. "Gemba" can be continued after the implementation of the scheduling system to make sure the algorithm works and to identify possible improvement opportunities. Other risks in the execution of the project include mismanagement of time, failure in support from management, creep in the scope of the project or industry could have expectations above the available skill set. A risk that could cause failure of the project is if the scheduling algorithm cannot integrate with the current system sufficiently or if the algorithm cannot be used by the AAP.

This project has a time constraint of finishing the development of the main deliverable by end of September. There are two cranes that can be observed in the current plant. The only way to predict the effectiveness of more cranes is by using simulation; More cranes cannot be purchased. The algorithm can be developed in any preferred language.

Python is the suggested method, as some experience has been gained in this language.

Chapter 2

Literature Review

To be able to identify possible algorithmic approaches for this scheduling problem, it is necessary to classify the current problem in a universal manner. The α / β / γ notation is useful to classify the problem and to identify key elements. This notation identifies the machines (resources), constraints and objective function respectively [2]. Heuristics need to be researched to identify similar solving models to use as a reference when solving a model. Three possible models that have been solved have been identified to use as possible references.

2.1 Classifying Scheduling Problems

2.1.1 α: Machine Environment

Alpha can either be a numerical value, indicating that the process has only one machine, or it can indicate the type of setup of machines in the relevant process [17]. α_1 can have the values of: {0,P,Q,R,PMPM,QMPM,G,X,O,J,F} [2], while α_2 is an integer indicating the number of machines in the process.

The following α_1 values have the stated meanings:

Table 2.1: Possible values

α-Value	Interpretation
0	All jobs are sent to a specific machine. α takes the value of $\alpha_2 = 1$.
P	All the parallel machines are exactly the same in purpose and machine speed.
Q	The parallel machines are similar (executes the same process, but the machine speed differs.
R	The parallel machines differ in terms of the operations executed.
PMPM	The identical speed machines can execute different operations (Multi-Purpose Machine).
QMPM	The multi-purpose machine performs different operations at different speeds.

The remaining options of {G,X,O,J,F} are for dedicated machines. G, is for general shop, which has only the general characteristics of dedicated machines and preceding relationships [2], where mixed shop (X), open shop (O), job shop (j) and flow shop (f) are all special types of general shops. A flow shop has a predetermined order of processing. All jobs will go through the same sequence in a flow shop. Job shop is where each job has a unique sequence to complete processing on specialized machines. Similarly, open shop can have unique sequences for jobs, but every job must be processed on every machine. Mixed shop is a combination of a job shop and a open shop [16].

2.1.2 β: Job Characteristics

The second descriptive variable is β. This variable describes certain characteristics of the job and the processes that it undergoes. There are six variables within β, which indicates the ins and outs of the process. All variables are equal to zero if the statement regarding the variable is false.

Table 2.2: β-variables

β-variable	Meaning
β_1	Job-splitting (also referred to as preemption). This variable indicates whether processing may be interrupted and continued at a later stage. β_1 = pmtn.
β_2	This variable indicated whether there are predetermined preceding relationships. A precedence tree can be drawn to indicate which operations need to take place before others. β_2 = prec.
β_3	$\beta_3 = r_i$, which indicates the release (starting date) of each job.
β_4	$\beta_4 = p_i$, where p_i is used to indicate the predetermined and controlled processing times of each job.
β_5	$\beta_5 = d_i$, which is the deadlines for each job. The jobs cannot be finished any later than this date.
β_6	This variable indicates whether the similar products can be batched when processing. There can be up to n jobs in a batch. β_5 = p-batch or s-batch. The sum of the processing times of all the jobs in a batch equals the processing time of the batch.

2.1.3 γ: Objective Function

The objective function is the criteria for the process to be optimised in a certain measure. The objective function must thus be defined by this measure. One of the most influential variables in terms of objective function is the finishing time of a job, which is defined by C_i for each job, J_i. Secondly, the change in cost that is caused by a schedule has an influence. This cost is denoted as $f_i(C_i)$, as it is a function of the finishing time per job. Popular objective functions are [2]:

Table 2.3: Objective functions

Measure	Equation to be minimised
Bottleneck	$f_{max}(C) := max\{f_i(C_i) \| i = 1, ..., n\}$
Sum objective	$\sum f_i(C) := \sum_{i=1}^{n} f_i(C_i)$
Lateness	$L_i := C_i - d_i$
Earliness	$E_i := max\{0, d_i - C_i\}$
Tardiness	$T_i := max\{0, C_i - d_i\}$
Absolute deviation	$D_i := \|C_i - d_i\|$
Squared deviation	$S_i := (C_i - d_i)^2$
Unit penalties	$U_i := \begin{cases} 0 & \text{if } C_i \leq d_i \\ 1 & \text{otherwise} \end{cases}$

Individual or combinations of the measures listed above can be used as objective functions, such as $\sum w_i D_i$ and $\sum w_i U_i$, where w_i denotes the weight assigned to each job. The objective function could also be to merely minimise the total processing time, or the maximum processing time per job i. The most important objective functions are C_{max} and L_{max}.

2.1.4 Summary

Based on the above definitions, the α / β / γ-notation can be used to describe nearly any discrete scheduling problem. Some examples of deterministic models are: $1|r_i,|L_{max}$ and $Q|pmtn, r_i|C_{max}$ [20].

The scheduling problem at Wispeco Aluminium is a flow shop problem with the cranes being the two machines available. A job is defined as a single beam, loaded with multiple profiles. The jobs can be batched, but not split. Each operation in the process has a predetermined time required for completion. Jobs cannot go over this time period. The minimal amount of work that the anodising plant is currently receiving that deadlines do not need to be assigned to the respective jobs. The makespan of the process must only be minimised. The following variables serves as the classification of the Wispeco anodising scheduling problem.

$\alpha_1 = F$

$\beta_2 =$ prec, with precedent relations according to the manufacturing proccess

$\beta_6 = p - batch$

$\gamma = max\{C_i|i = 1...n\}$

2.2 Solution Strategies and Applications

Heuristics and three applicable articles are investigated to be compared with this problem and to assist in finding possible solutions.

2.2.1 Heuristics

Metaheuristics are methods used in scheduling, to find a good solution which is not necessarily the optimal solution. These methods produce adequate solutions. However, they have the drawback of being difficult to apply to mulitple and varying problems. In addition, these methods produce only one solution whereas many problems have multiple working solutions. According to Jarboui et al [9], there are two method categories that metaheuristics can fall into. The first is where there is a random solution, which is improved using iterations to determine an improved solution like greedy randomized adaptive search procedure (GRASP) and variable neighbourhood search (VNS). The second method determines a family of solutions at every iteration. Metaheuristics can be applied on monocriterion scheduling or multicriteria scheduling.

Heuristics are metaheuristics, which have been applied to a specific problem. Pinedo [18] suggests the profile fitting heuristics for flow shop problems with limited or no storage in between machines. Other problems can be solved with heuristics used for travelling salesman problems.

2.2.2 Application: Flow Shop

In a flow shop application, Gupta et al. [5] a production schedule. The model is created for a flow shop environment with two machines and one transporter. The model uses processing times, setup times, transport times from machine one to machine two and transport times from machine two back to machine one. The Average High Ranking (AHR) of the processing and setup times are used in the model, as these actual times are “fuzzy” (which is defined as uncertain or imprecise data). The model must minimise the makespan by choosing a sequence with the shortest completion time. The model starts with two random sequences of jobs and determines the sequence with the shortest completion time. This process is then iterated to find a good solution.

The logic of this model can be used in the algorithm created for the Wispeco problem. Two models can be created one after the other, one for each crane. The cranes will serve as the transporters and each tank will serve as a machine.

2.2.3 Application: Single Hoist Scheduling of Electroplating PCB’s

Lim [13] scheduled a PCB electroplating line. The electroplating of PCB’s is a very similar process to that of anodising aluminium, as it is a smaller scale version of the same process. In this article, a genetic algorithm approach was followed instead of mathematical programming based approaches that are often followed. This example consists of a series of chemical tanks, moved through the system by means of a hoist, as shown in Figure 2.1.

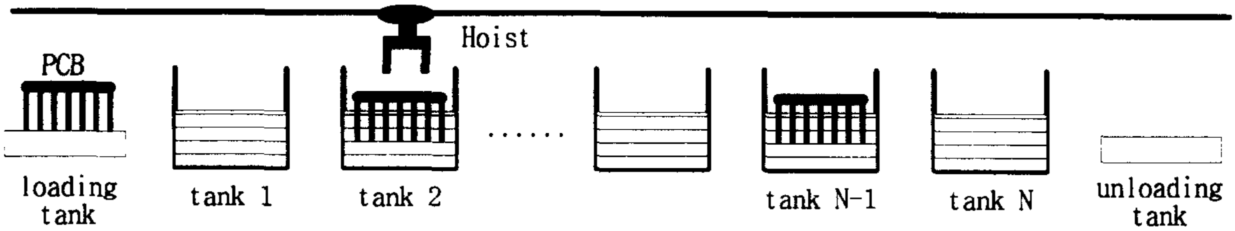


Figure 2.1: PCB electroplating line

In this model, S is the system, with $S = S_0, S_1 ... S_N, S_{N+1}$ and N = the number of tanks in the system. S_0 represents the loading station and S_{N+1} represents the unloading station. The time that the hoist takes to move from one station to the next, with or without a job, as well as the range of time that the job is required to be in each station are used as parameters. The hoist and tanks can only hold one job at a time. The objective of this model is to minimise the cycle time of one job coming into the system until it leaves the system.

This model is very similar to the anodising problem at Wispeco Aluminium. The hoist can be compared to the cranes in the anodising plant and there are also a series of chemical tanks creating the process. The only difference between this model and the anodising plant is that the anodising plant has multiple tanks at certain steps in the process. To be able to use this model as reference, it must be adapted to be able to skip a number of the tanks in the system.

2.2.4 Application: Multi-degree scheduling

Li et al. [12] describes a more relevant scheduling example which is similar to the problem at Wispeco. This example is a flow shop where there are n stages in the process and one of the stages have multiple machines while the other stages has single machines. Each machine can produce one unit at a time and each robot (moving the parts from machine to machine) can only move one part at a time, known as robot capacity constraints. A part pickup criteria called the time window is applicable to the example. This criteria is used when there is a lower and upper limit within which a part must be moved from one machine to the next. Other options for part pickup criteria is the no-wait criteria, where parts have to be picked up immediately after processing, or the free pickup, which implies that a part can be picked up at any time. This article deals with a process without any buffers between stages, but to reduce the bottleneck, the drum process usually has an extra machine to improve throughput [8]. The next bottleneck is often caused by the robots transporting the parts through the process, but instead of choosing the more expensive option of purchasing another robot, it is wise to first use scheduling techniques to improve sequences and throughput. In this example, it has been taken into consideration that the robots used for transporting cannot cross over each other.

Every robot thus has a few stages that it can move from and to. In this example, a cycle is defined as a period in which a certain number of parts (K) start and complete the process. One of the problems that the article identifies in the model is that the model will not be able to solve flow shop problems with more than thirty stages. The objective of the model is to maximise throughput, which is the same as minimising the cycle time in degree cycles. Mixed integer linear programming is used to solve the scheduling problem and a numerical example is used to show the implementation of the model. The model is illustrated in Figure 2.2, where S_n is the n^{th} stage (stage v refers to the multi-machine stage), M_n is the machine at stage n and H is the number of robots transporting the goods in the plant.

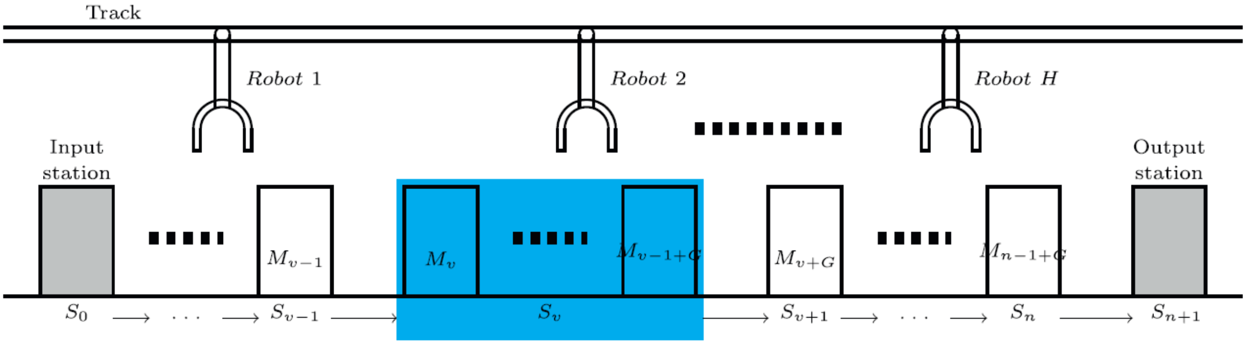


Figure 2.2: Multi-degree process with multiple robots

This example is similar to the problem at Wispeco and the model will be able to be adapted accordingly. The aluminium plant at Wispeco has thirteen stages with two cranes (robots) moving the parts, which cannot cross each other on the line. The model described has been implemented on a few examples, where the increase in throughput, and decrease in cycle time serves as proof of the success of this model. The latter seems to be the model with the most similarities with the anodising plant scheduling and will thus be used in the conceptual design of the scheduling algorithm.

Chapter 3

As-is Scheduling

This chapter aims to describe the environment and processes of scheduling as it currently operates at Wispeco Aluminium. The following was identified as the as-is process of anodising with specific focus on the scheduling for a 24 hour day's work.

3.1 Problem Investigation

The anodising plant at Wispeco Aluminium is moving toward a single line plant (where it is currently a double line plant). This decision forces the plant to have a larger throughput or smaller completion time of the same amount of aluminium profiles anodised. The following sections describes the plant.

3.1.1 Stages

The stations in the aluminium anodising plant serves as the stages in the desired model. The stations are presented in figure 3.1, where the brackets indicate all tanks that form one stage. These stages relate to stage v with G machines as described in chapter 2.

This process is controlled by two cranes which can be the robots in the related model. The cranes cannot cross each other on the line and must pick up the necessary materials within a short time frame.

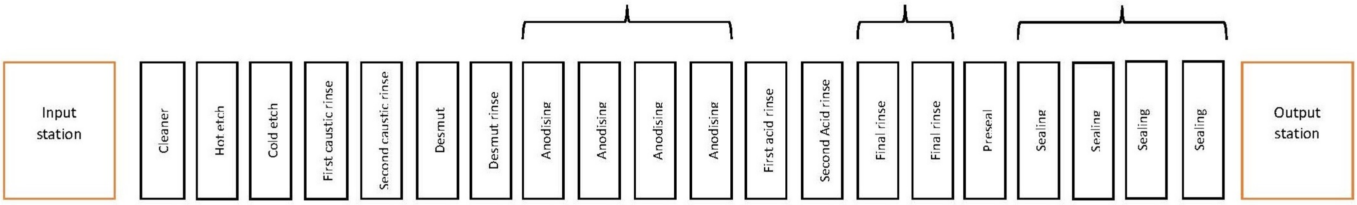


Figure 3.1: Anodising stations

3.1.2 Colouring

In addition to the process in Figure 3.1, the process includes a colouring station. The anodising plant produces six different colours, which contributed to production in the financial year of 2015 to 2016 as shown in Table 3.1. The natural colour does not require a colouring step in the process. All five other colours add one station to the process. The amount of time that the beam is submerged in the colouring tank determines the colour of the output. Based on Table 3.1 and the fact that all colours need an extra tank, management of the anodising plant has suggested to move away from producing colours other than natural. With the knowledge that all beams will then follow the exact same process, the scheduling problem can be classified as a flow shop problem [4].

Table 3.1: Anodising colours

Colour	Square meters produced	Percentage of production
N: Natural	1 307 559.61	94.68%
B1: Very light Bronze	2 708.84	0.20%
B3: Bronze	9 722.44	0.70%
B5: Dark Bronze	156.27	0.01%
B7: Darkest Bronze	502.71	0.04%
B9: Black	60 315.17	4.37%

The anodising plant is currently running the automated cranes on a predetermined sequence, which completes the necessary jobs between 2 hours, 23 minutes and 3 hours, 23 minutes without any downtime. This will be used as a basic metric to compare to the improved schedule. However, when the anodising plant changes to a single line plant, the throughput will increase and thus the cycle time must decrease to maintain customer satisfaction. Without sequencing to improve cycle time, the plant will not be able to keep up with the amount of work. This model will also be an indication of whether scheduling will improve the cycle time sufficiently and whether an additional crane is needed.

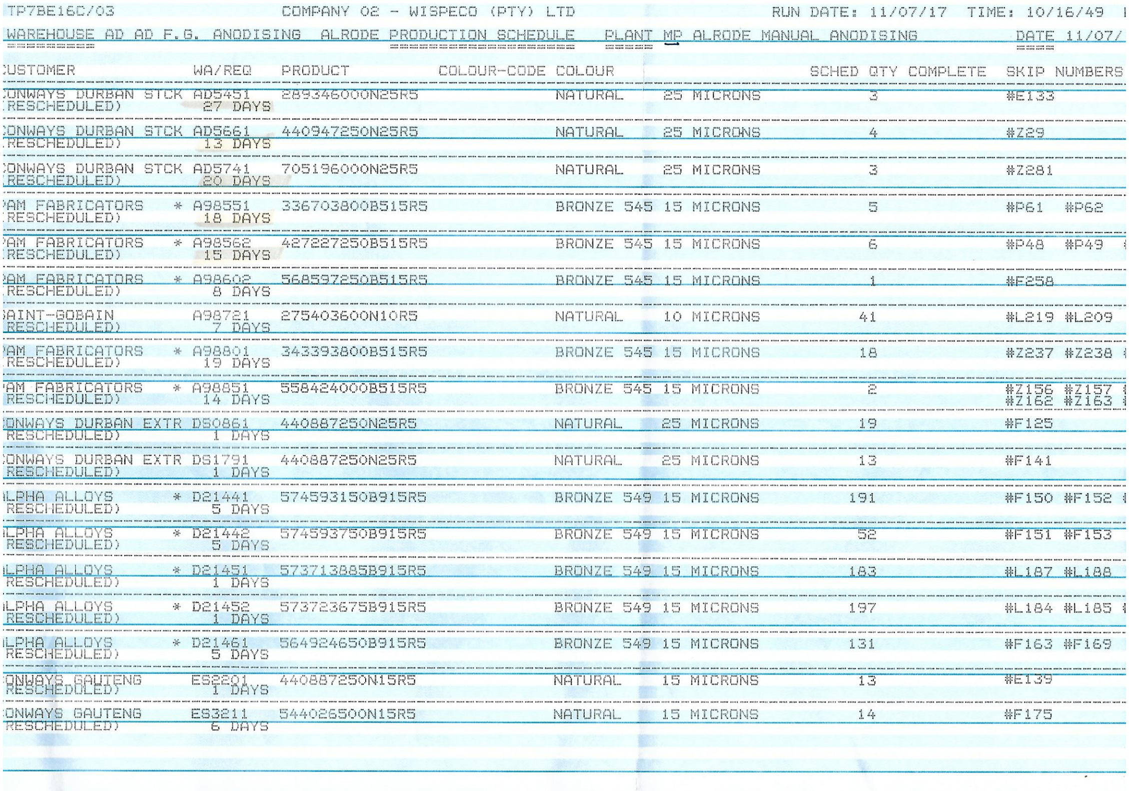
3.2 Scheduling

3.2.1 Order Scheduling

The person in charge of scheduling a day's work at the anodising plant is the planner. The planner sets up a list of all the orders received, where he specifies which orders are the most important. This is indicated on the shift handover form which is given to the FLM (First Line Manager) together with the list of jobs available to be processed. These important orders are usually reworks, orders where the customer calls to indicate the orders are urgent as well as orders from priority customers. Appendix B shows an example of the shift handover form, which indicates the priority of an order by means of dots. The more dots the form has, the more urgent the order is. Figure 3.2 shows the first page of five from the list of orders available to be processed.

From the list of ready orders, the FLM on the floor decides which orders to complete first. The beams are filled to physical capacity, which ranges between 35 and 120 square meters on a single beam, depending on the size of the profile to be jigged. Each profile has a specific die number which indicates its shape. The shape and the length of the profile determines the square meters that will be anodised on a single profile. This figure is used to determine the square meters that has been hung on one beam. The jiggers know how many profiles of a certain type should hang on a beam.

The rule of thumb used to determine the order of processing from the list is usually the most important orders first (according to the priority list received from the planner). Thereafter, the FLM processes work from the list of orders classified either as “nice” or “bad” work. “Nice” work commonly refers to profiles that are easy to jig and contribute the majority of square meters produced for the day (a measure monitored daily to ensure the plant is doing enough work). “Bad” work is the opposite. It refers to work that



```
TP7BE16C/03                COMPANY 02 - WISPECO (PTY) LTD                       RUN DATE: 11/07/17  TIME: 10/16/49
WAREHOUSE AD AD F.G. ANODISING  ALRODE PRODUCTION SCHEDULE   PLANT MP ALRODE MANUAL ANODISING            DATE 11/07/
==========                                          ====================   =====                                      ====

USTOMER                 WA/REQ   PRODUCT            COLOUR-CODE COLOUR                        SCHED QTY COMPLETE  SKIP NUMBERS
------------------------------------------------------------------------------------------------------------------------------
ONWAYS DURBAN STCK AD5451   289346000N25R5                  NATURAL     25 MICRONS            3                 #E133
RESCHEDULED)             27 DAYS
------------------------------------------------------------------------------------------------------------------------------
ONWAYS DURBAN STCK AD5661   440947250N25R5                  NATURAL     25 MICRONS            4                 #Z29
RESCHEDULED)             13 DAYS
------------------------------------------------------------------------------------------------------------------------------
ONWAYS DURBAN STCK AD5741   705196000N25R5                  NATURAL     25 MICRONS            3                 #Z281
RESCHEDULED)             20 DAYS
------------------------------------------------------------------------------------------------------------------------------
AM FABRICATORS    * A98551   336703800B515R5                 BRONZE 545 15 MICRONS            5                 #P61  #P62
RESCHEDULED)             18 DAYS
------------------------------------------------------------------------------------------------------------------------------
AM FABRICATORS    * A98562   427227250B515R5                 BRONZE 545 15 MICRONS            6                 #P48  #P49
RESCHEDULED)             15 DAYS
------------------------------------------------------------------------------------------------------------------------------
AM FABRICATORS    * A98602   568597250B515R5                 BRONZE 545 15 MICRONS            1                 #F258
RESCHEDULED)              8 DAYS
------------------------------------------------------------------------------------------------------------------------------
AINT-GOBAIN         A98721   275403600N10R5                  NATURAL     10 MICRONS           41                 #L219 #L209
RESCHEDULED)              7 DAYS
------------------------------------------------------------------------------------------------------------------------------
AM FABRICATORS    * A98801   343393800B515R5                 BRONZE 545 15 MICRONS           18                 #Z237 #Z238
RESCHEDULED)             19 DAYS
------------------------------------------------------------------------------------------------------------------------------
AM FABRICATORS    * A98851   558424000B515R5                 BRONZE 545 15 MICRONS            2                 #Z156 #Z157
RESCHEDULED)             14 DAYS                                                                                #Z162 #Z163
------------------------------------------------------------------------------------------------------------------------------
ONWAYS DURBAN EXTR DS0861   440887250N25R5                  NATURAL     25 MICRONS           19                 #F125
RESCHEDULED)              1 DAYS
------------------------------------------------------------------------------------------------------------------------------
ONWAYS DURBAN EXTR DS1791   440887250N25R5                  NATURAL     25 MICRONS           13                 #F141
RESCHEDULED)              1 DAYS
------------------------------------------------------------------------------------------------------------------------------
LPHA ALLOYS       * D21441   574593150B915R5                 BRONZE 549 15 MICRONS          191                 #F150 #F152
RESCHEDULED)              5 DAYS
------------------------------------------------------------------------------------------------------------------------------
LPHA ALLOYS       * D21442   574593750B915R5                 BRONZE 549 15 MICRONS           52                 #F151 #F153
RESCHEDULED)              5 DAYS
------------------------------------------------------------------------------------------------------------------------------
LPHA ALLOYS       * D21451   573713885B915R5                 BRONZE 549 15 MICRONS          183                 #L187 #L188
RESCHEDULED)              1 DAYS
------------------------------------------------------------------------------------------------------------------------------
LPHA ALLOYS       * D21452   573723675B915R5                 BRONZE 549 15 MICRONS          197                 #L184 #L185
RESCHEDULED)              1 DAYS
------------------------------------------------------------------------------------------------------------------------------
LPHA ALLOYS       * D21461   564924650B915R5                 BRONZE 549 15 MICRONS          131                 #F163 #F169
RESCHEDULED)              5 DAYS
------------------------------------------------------------------------------------------------------------------------------
ONWAYS GAUTENG      ES2201   440887250N15R5                  NATURAL     15 MICRONS           13                 #E139
RESCHEDULED)              1 DAYS
------------------------------------------------------------------------------------------------------------------------------
ONWAYS GAUTENG      ES3211   544026500N15R5                  NATURAL     15 MICRONS           14                 #F175
RESCHEDULED)              6 DAYS
```

Figure 3.2: List of orders

takes more time and effort to jig and has a significantly smaller contribution to the day's monitored square meters produced. The FLM usually prefers to do the "nice" work first.

As the day progresses, the FLM indicates on the planner's list what work has been completed. After the 12 hour night shift, the day shift continues with the same list and completes as many orders as possible. The work that has not been completed, are usually prioritised the following day. The planner sets up a new list every day, which is in line with the company goal: "One day delivery".

The schedule for the AAP was followed on 22 August 2017, when 34 beams and 2263 square meters were processed during the day shift, which started 2 hours late. The late start was due to the caustic tanks that were not at the desired temperature. On 23 August 2017, 42 beams and 2655 square meters were processed during the 12-hour day shift. On 7 August 2017, the AAP processed 36 beams and 2372 square meters during the day shift. This shift started 1 hour late, again due to incorrect temperatures in the caustic tanks.

3.2.2 Crane Scheduling

The two cranes on the AAP are controlled by a PLC (Programmable Logic Circuit), which has one operator. The operator must enter the square meters desired on each jigged beam into the PLC and switch the process to manual if there are any breakdowns. The PLC is currently programmed to move according to a predetermined sequence. The sequence does not change for any type of product, unless a breakdown forces the operator to control the line manually. The last column of Table 3.3 displays the average time it takes to complete each move. Based on these times, the algorithmic model assumes all crane movements (the combined move of moving a beam out of one tank and into another) takes one minute to complete.

Table 3.2: Crane 1 Sequence

Process	Into tank or Out of tank	Average Movement Time (Seconds)
Jigging	Out	46
Cleaning	In	14
Etch Rinse 1	Out	39
Etch Rinse 2	In	7
Desmut	Out	42
Desmut Rinse	In	6
Etch Rinse 2	Out	40
Desmut	In	7
Desmut Rinse	Out	59
Jigging	In	18
Stripping	Out	42
Etch Rinse 2	In	7
Desmut	Out	38
Desmut Rinse	In	15
Hot Etch	Out	43
Etch Rinse 1	In	8
Etch Rinse 2	Out	40
Desmut	In	15
Cleaning	Out	79
Hot etch	In	15
Desmut Rinse	Out	48
Anodising	In	12
Transfer station	Out	52
Stripping	In	18

Table 3.3: Crane 2 Sequence

Process	Into tank or Out of tank	Average Movement Time (Seconds)
Anodising	Out	
Acid Rinse 1	In	
Acid Rinse 1	Out	
Acid Rinse 2	In	
Sealing	Out	
Jigging	In	
Final Rinse	Out	
Sealing	In	
Off-jig	Out	
Transfer station	In	
Acid Rinse 2	Out	

Chapter 4

Algorithmic Approach and Solution

The multi-degree model in Section 2.2.4 was used as reference to create the model described in this chapter. In this model, there are two cranes and 12 processes with between one and 4 tanks for each process. There is a time frame within which the crane must pick up the relevant beam. In the application of an anodising plant, there are processes that have no time frame for stations like etching, the first caustic rinse, desmut, anodising and the first acid rinse. The processes which cannot be processed longer than the given processing times are referred to as critical processes.

4.1 Model Parameters

Notation usage to be noted when understanding the parameters and variables used in the model is as follows. A superscript 0, means that the set of relevant values include the loading station (station 0). A superscript +, means the unloading station is included. No super or subscripts, refers to the stations excluding the loading and unloading stations. A super- or subscript v, refers to all tanks, thus including each tank in stations a, v and f as described in Table 4.1 below.

The following parameters will be used (the naming of all variables and parameters are similar to those used by Lin et al. [12], for simplification):

Table 4.1: Model parameters

Parameter	Description and or value
n	Number of stages, 1-13. The loading station is 0 and the unloading station is number 14
a, v, f	The three stages with parallel machines, referring to anodising, final rinse and sealing respectively.
G, J, P	The number of parallel machines used in stage a, v and f respectively.
H	The number of cranes used to transport jobs = 2
Move(i)	The first transportation, in a cycle, of a part from stage i to i + 1, where $i \in N^0$
l_h	The crane assignment, where $h \in H$. The first crane is responsible for moves from station 0 to I_1 (0 to 7) and the second crane is responsible for moves I_1 to I_2, where I_2 = n+1 = 14
L_i	The minimum processing time a part requires in stage i, where $i \in N$
U_i	The maximum processing time a part may undergo in stage i, where $i \in N$
a_i	The time it takes to pick up a part from i, where $i \in N^0$. This time relates to move(i))
b_i	The time it takes to drop a part off at i + 1, where $i \in N^0$. This time relates to move(i+1)
d_i	The travel time of a loaded crane to move from i to i + 1, including the pick-up time (a_i) and the drop-off time (b_i), where $i \in N^0$
$e_{i,j}$	The travel time of an empty crane from i to j, where $i, j \in N^+$, $i \neq j$ and $e_{i,j} = e_{j,i}$
B	A small positive number
K	A large number

4.2 Model Variables

Table 4.2 introduces all variables to be used in the model.

Table 4.2: Model variables

Parameter	Description and or value
T	Cycle time
t_i	The starting time of move(i), where t_i0 and $i \in N^0$
t_{max1}	The starting time of the last move(i) within the cycle for crane 1
t_{min2}	The starting time of the first move(i) within the cycle for crane 2
t_{max2}	The starting time of the last move(i) within the cycle for crane 2
n_i^h	$n_i^h := \begin{cases} 1 & \text{if the move(i) is the last move for robot h} \\ 0 & \text{otherwise} \end{cases}$
u_j^h	$u_j^h := \begin{cases} 1 & \text{if the move(j) is the first move for robot h} \\ 0 & \text{otherwise} \end{cases}$
$w_{i,j}^h$	$w_{i,j}^h := \begin{cases} 1 & \text{if the move(j) is the first and move(i) is the last move for robot h} \\ 0 & \text{otherwise} \end{cases}$
$y_{i,j}$	$y_{i,j} := \begin{cases} 1 & \text{if } t_{r,i} < t_{u,j} \\ 0 & \text{otherwise} \end{cases}$ where $i < j$
$y_{i,i+1}$	$y_{i,i+1} := \begin{cases} 1 & \text{if } t_{r,i} + d_i - b_i < t_{u,i+1} + a_{i+1} \\ 0 & \text{otherwise} \end{cases}$ where $i \in \{I_1 ... I_{h-1}\}$

Objective function : $Z = min\{t_{max2}\}$

4.3 Model Constraints

What follows are constraints to be used as guidelines in the algorithm.

4.3.1 Constraints for Crane 1

$$t_{max1} + \sum_{i=1}^{I_1} (d_i + e_{i+1,0}) X_i <= T \tag{4.1}$$

$$t_{max1} >= t_i \qquad i \in \{I_1, I_2\} \tag{4.2}$$

$$t_{max1} <= t_i - (X_i - 1)K \qquad i \in \{I_1, I_2\} \tag{4.3}$$

$$\sum_{i=0}^{I_1} X_i = 1 \tag{4.4}$$

Constraint 4.1 ensures that the time that the last process starts,plus the crane movements to finish the cycle is not greater than the cycle time. Constraint 4.2 ensures that t_max1 is the largest possible value, which will be the starting time of the last process. Constraint 4.3 ensures that t_max1 only obtains a value when i is the last move for crane 1. Constraint 4.4 defines $X_i = 1$ with only one possible non-zero value for all possible i values.

4.3.2 Constraints Defining the Crane Capacity

$$t_j - t_i >= d_i + e_{i+1,j} - (1 - y_{i,j})K \qquad \forall i, j \qquad \in \{I_1, I_2\} \tag{4.5}$$

$$t_i - t_j >= d_j + e_{j+1,i} - y_{i,j}K \qquad \forall i, j \qquad \in \{I_1, I_2\} \tag{4.6}$$

Each crane can carry maximum one beam at a time.

4.3.3 Constraints Rejecting Crossover of Cranes

$$t_{i+1} - (t_i + d_i + L_i) >= (y_{i,i+1} - 1)K \qquad \forall i, j \qquad \in \{I_1, I_2\} \tag{4.7}$$

$$t_i + d_i - b_i - (t_{i+1} + a_{i+1}) >= -y_{i,i+1}K + \delta \qquad \forall i, j \qquad \in \{I_1, I_2\} \tag{4.8}$$

These constraints are valid where i+1 is the shared station between the two cranes.

The parameters, variables and constraints listed above is used as guidelines in the Python algorithmic model. These constraints are not a description of the entire model, but merely a starting point. Figure 4.1 indicates the logical approach of the algorithmic model and the complete algorithm is added as Appendix C.

The algorithm is used to test three scheduling heuristics. The best heuristic is the main deliverable of the project which can be implemented at Wispeco Aluminium. The heuristics that were tested to be used as guidance at Wispeco was: FIFO (First In, First Out), Priority (which completes priority orders first and then follows the schedule as Wispeco currently uses) and Shortest processing time (all 10μm orders, followed by 15μm and 25μm) [3].

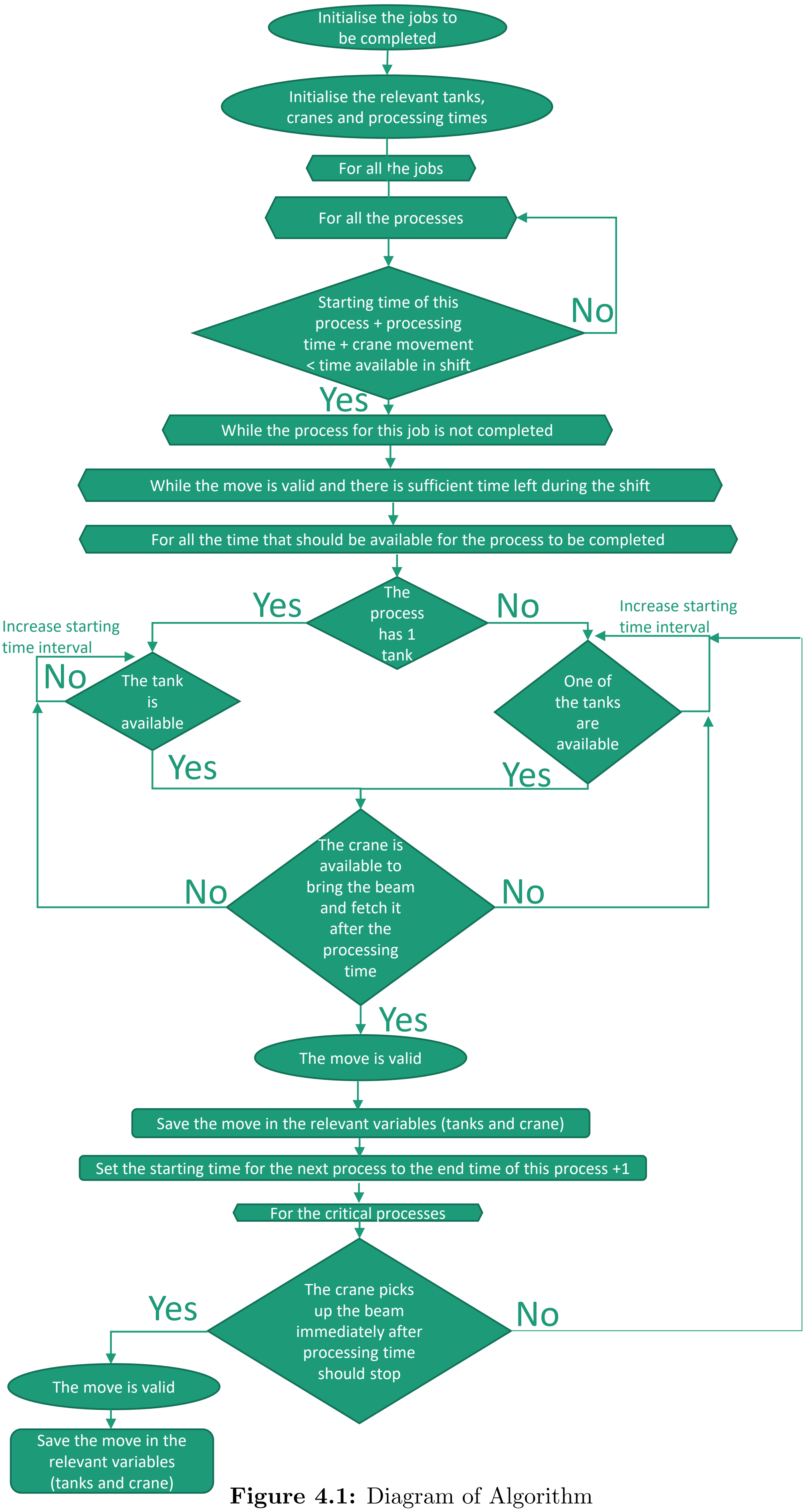

Initialise the jobs to be completed
Initialise the relevant tanks, cranes and processing times
For all the jobs
For all the processes
Starting time of this process + processing time + crane movement < time available in shift
No
Yes
While the process for this job is not completed
While the move is valid and there is sufficient time left during the shift
For all the time that should be available for the process to be completed
Yes
The process has 1 tank
No
Increase starting time interval
Increase starting time interval
No
The tank is available
One of the tanks are available
Yes
Yes
The crane is available to bring the beam and fetch it after the processing time
No
No
Yes
The move is valid
Save the move in the relevant variables (tanks and crane)
Set the starting time for the next process to the end time of this process +1
For the critical processes
Yes
The crane picks up the beam immediately after processing time should stop
No
The move is valid
Save the move in the relevant variables (tanks and crane)

Figure 4.1: Diagram of Algorithm

Chapter 5

Results

Three days (7 August, 22 August and 23 August 2017) of production at the anodising plant were followed to retrieve the results discussed in this chapter. The schedule for each day was entered into the algorithmic model, described in Chapter 3. The results retrieved from the algorithm is used to compare schedules for a shift's work according to three possible heuristics. The heuristics are FIFO, Priority and Shortest processing time.

The **current schedule** follows an inexact process. The beams are filled with jobs as the FLM sees fit, with special attention being paid to the priority orders that need to be completed before the end of two shifts.

The **FIFO** scheduling heuristic was created from the list of jobs as discussed in the "As-is" description. The same jobs that were processed in the current schedule of Wispeco are used. The **priority** schedule is the priority orders first (indicated in bold), followed by the rest of the orders from the schedule as it is currently set up at Wispeco. Most of the priority orders are completed during night shift. Day shift is thus completing the remaining priority orders. Schedules according to the **shortest processing time**, completes all 10μm orders, followed by all 15μm orders and then completing all 25μm orders.

The processing time according to the model starts at 06:00 and thus all anodising starts at 06:24 when the first beam is submerged in a anodising tank. The lists under each subsection indicate the microns required on every beam processed.

5.1 7 August 2017- Day shift

On this date, processing did not start at 06:00. The caustic rinse tanks were not at the desired temperature (either too hot or too cold), which caused processing to start after 08:00. The schedule was entered into the model, which does not take that fact into consideration. The processing times in this report will thus start at 06:00 and will end before the shift ends at 18:00.

Beams processed: 34

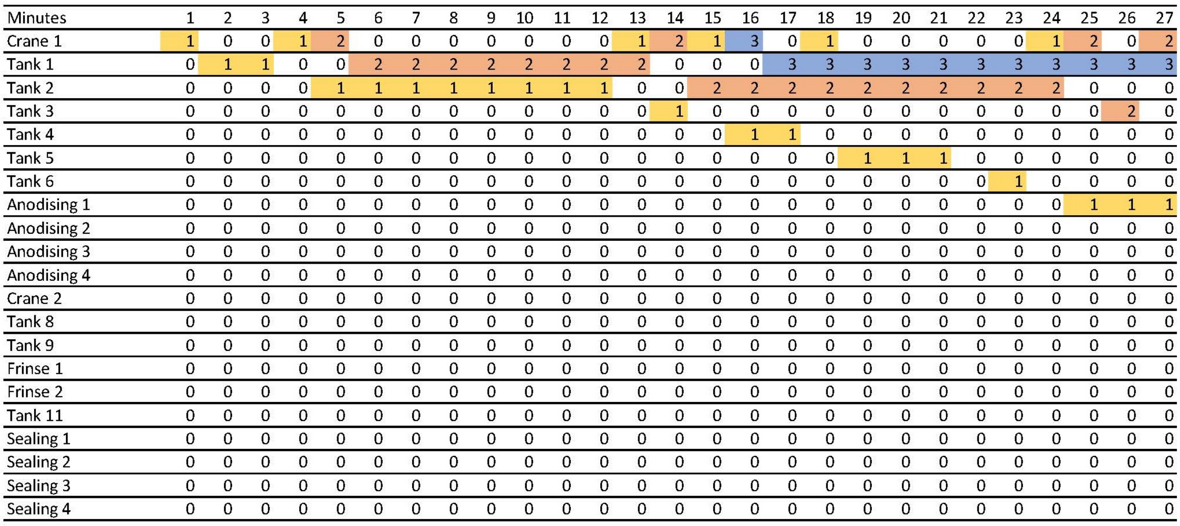
Square meters processed: 2 372

5.1.1 Current Schedule

[15, 15, 15, 15, 15, 15, 15, 10, 15, 25, 25, 25, 15, 15, 25, 25, 15, 25, 25, 25, 15, 15, 15, 15, 25, 15, 25, 25, 25, 25, 25, 25, 15, 25]

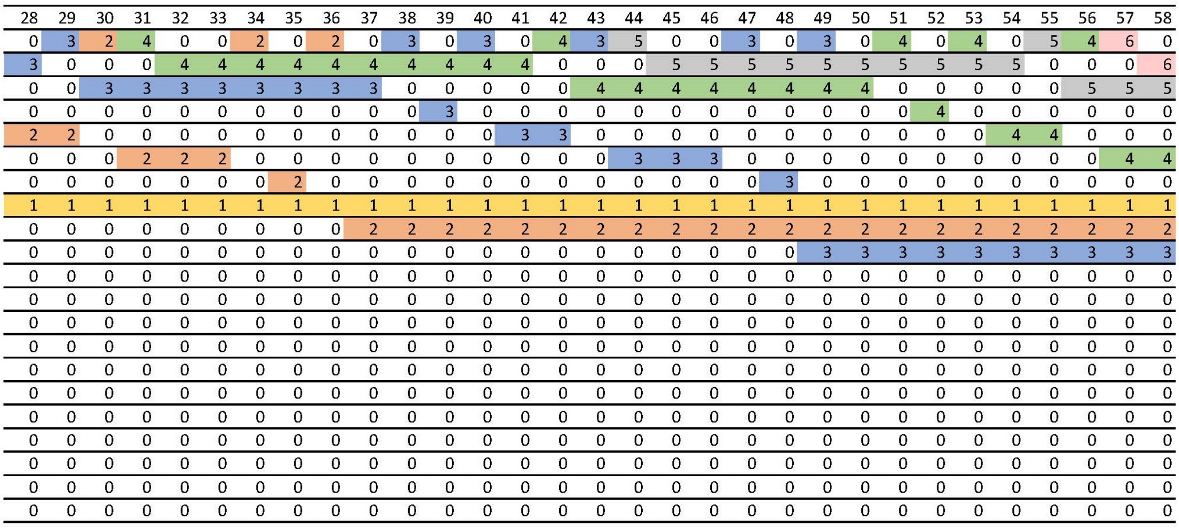
Last beam completed anodising: 15:01

Figures 5.1 to 5.11 displays a visual representation of the Gantt chart created by the algorithmic model for the current schedule on 7 August 2017. The first 15 beams are coloured to simplify reading of the Gantt chart. The minutes are indicated in the top row. Appendix D illustrates the remaining part of Gantt chart without colour.



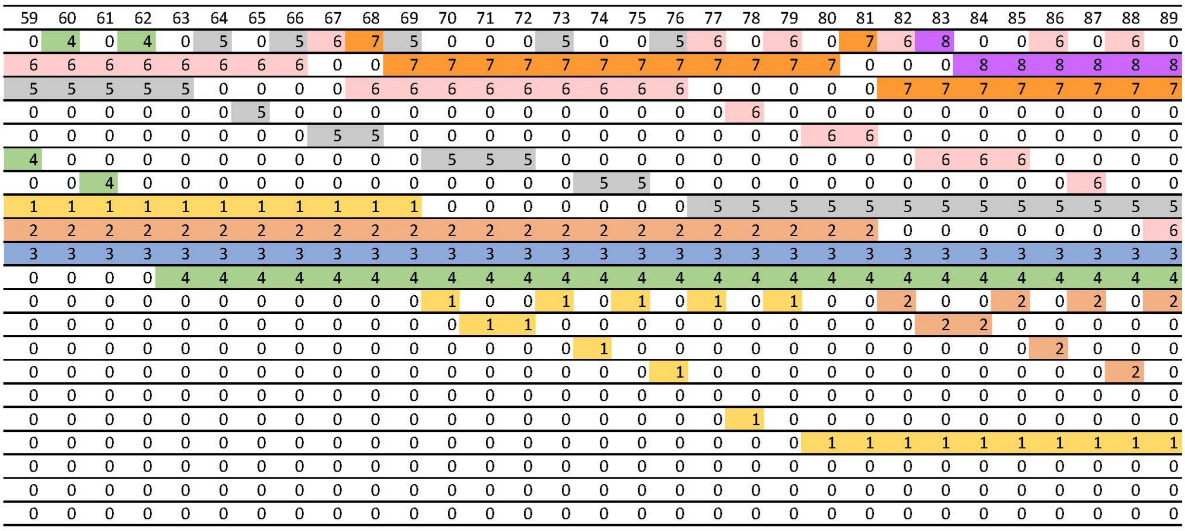
Minutes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Crane 1	1	0	0	1	2	0	0	0	0	0	0	0	1	2	1	3	0	1	0	0	0	0	0	1	2	0	2
Tank 1	0	1	1	0	0	2	2	2	2	2	2	2	2	0	0	0	3	3	3	3	3	3	3	3	3	3	3
Tank 2	0	0	0	0	1	1	1	1	1	1	1	1	0	0	2	2	2	2	2	2	2	2	2	2	0	0	0
Tank 3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0
Tank 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
Tank 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
Tank 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Anodising 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Anodising 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Anodising 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Anodising 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Crane 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tank 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tank 9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Frinse 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Frinse 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tank 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sealing 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sealing 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sealing 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sealing 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5.1: Gantt chart



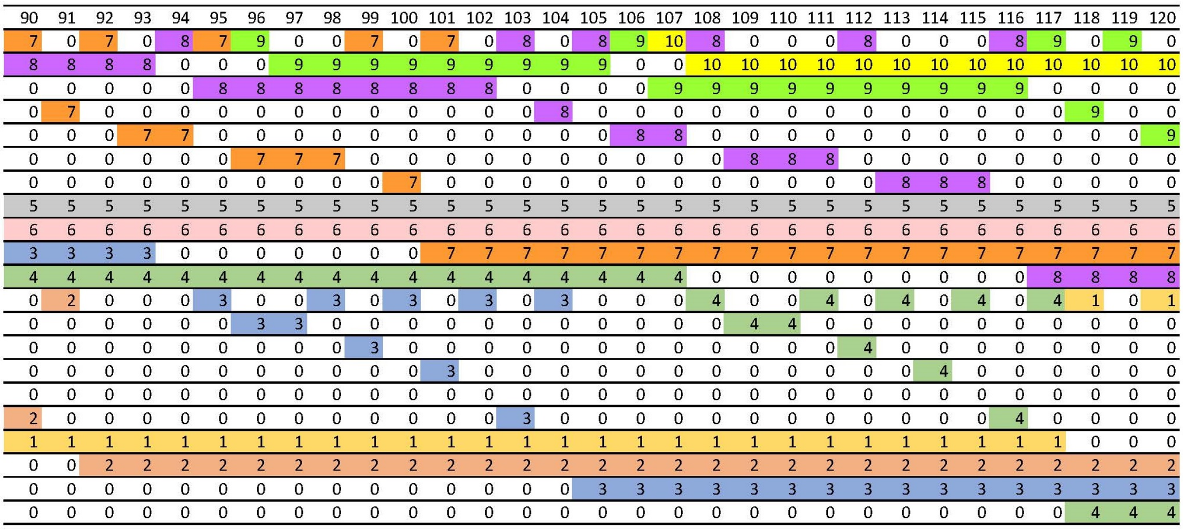
28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
0	3	2	4	0	0	2	0	2	0	3	0	3	0	4	3	5	0	0	3	0	3	0	4	0	4	0	5	4	6	0
3	0	0	0	4	4	4	4	4	4	4	4	4	4	0	0	0	5	5	5	5	5	5	5	5	5	5	0	0	0	6
0	0	3	3	3	3	3	3	3	3	0	0	0	0	0	4	4	4	4	4	4	4	4	0	0	0	0	0	5	5	5
0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0
0	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	3	3	3	0	0	0	0	0	0	0	0	0	0	4	4
0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3	3	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5.2: Gantt chart



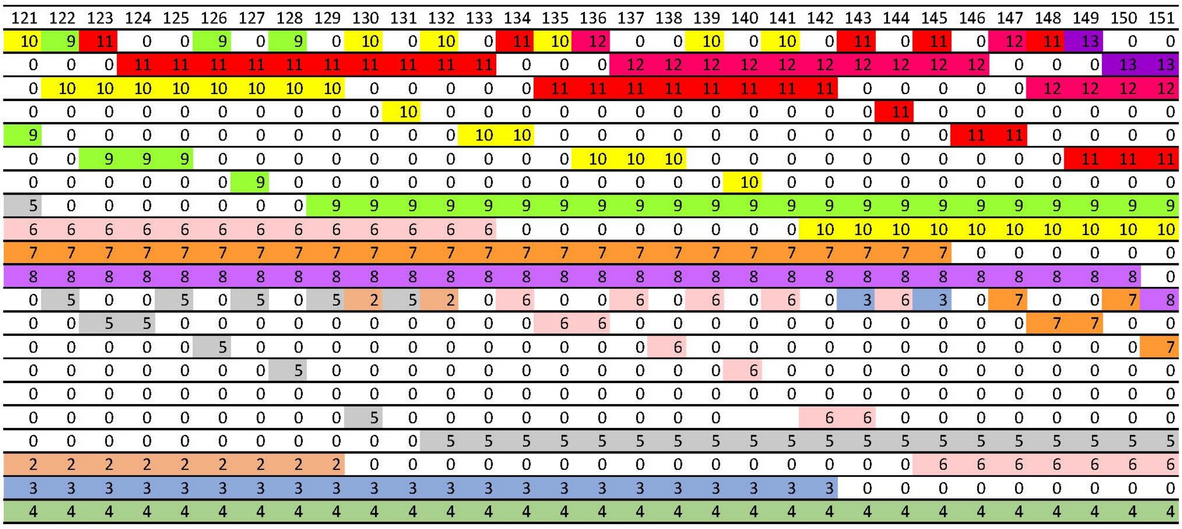
59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89
0	4	0	4	0	5	0	5	6	7	5	0	0	0	5	0	0	5	6	0	6	0	7	6	8	0	0	6	0	6	0
6	6	6	6	6	6	6	6	0	0	7	7	7	7	7	7	7	7	7	7	7	7	0	0	0	8	8	8	8	8	8
5	5	5	5	5	0	0	0	0	6	6	6	6	6	6	6	6	6	0	0	0	0	0	7	7	7	7	7	7	7	7
0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	5	5	0	0	0	0	0	0	0	0	0	0	0	6	6	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	5	5	5	0	0	0	0	0	0	0	0	0	0	6	6	6	0	0	0	0
0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	5	5	0	0	0	0	0	0	0	0	0	0	0	6	0	0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	5	5	5	5	5	5	5	5	5	5	5	5	5
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	6
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0	0	0	0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	2	0	0	2	0	2	0	2
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5.3: Gantt chart



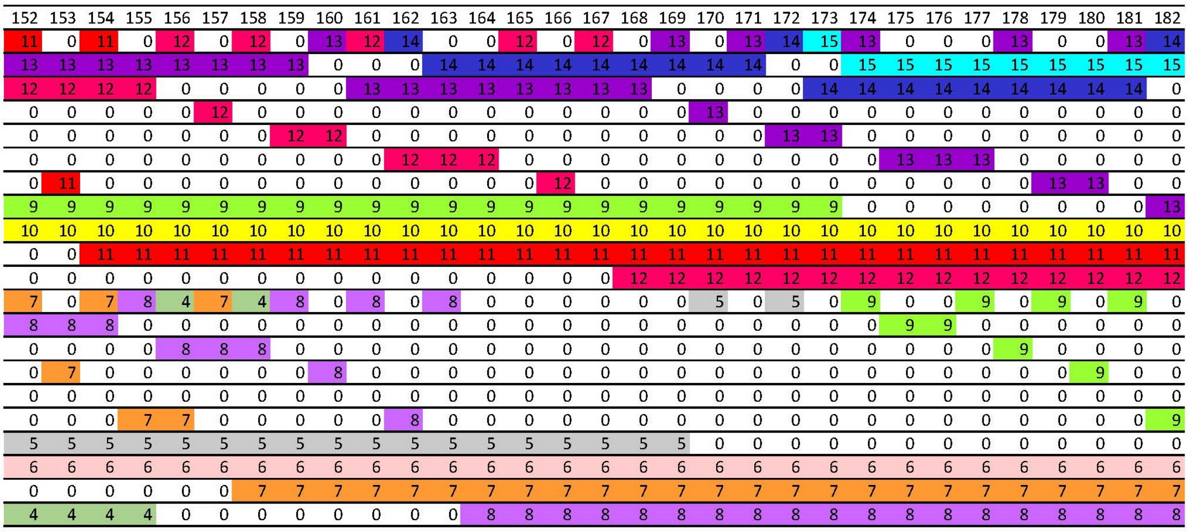
90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
7	0	7	0	8	7	9	0	0	7	0	7	0	8	0	8	9	10	8	0	0	0	8	0	0	0	8	9	0	9	0
8	8	8	8	0	0	0	9	9	9	9	9	9	9	9	9	0	0	10	10	10	10	10	10	10	10	10	10	10	10	10
0	0	0	0	0	8	8	8	8	8	8	8	8	0	0	0	0	9	9	9	9	9	9	9	9	9	9	0	0	0	0
0	7	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0
0	0	0	7	7	0	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	9
0	0	0	0	0	0	7	7	7	0	0	0	0	0	0	0	0	0	0	8	8	8	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	8	8	8	0	0	0	0	0
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3	3	3	3	0	0	0	0	0	0	0	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	0	0	0	0	0	0	0	0	0	8	8	8	8
0	2	0	0	0	3	0	0	3	0	3	0	3	0	3	0	0	0	4	0	0	4	0	4	0	4	0	4	1	0	1
0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4

Figure 5.4: Gantt chart



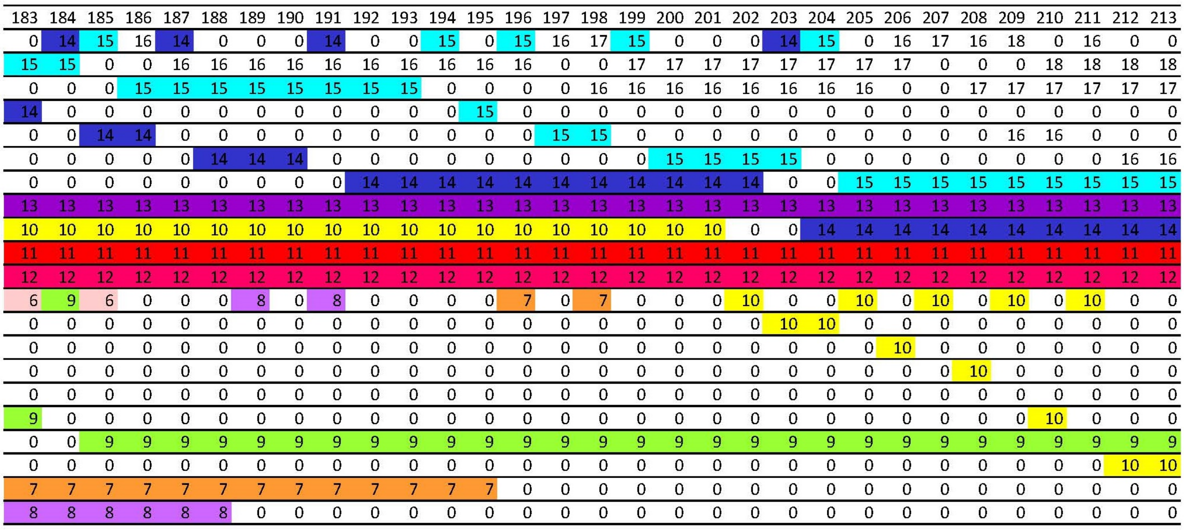
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151
10	9	11	0	0	9	0	9	0	10	0	10	0	11	10	12	0	0	10	0	10	0	11	0	11	0	12	11	13	0	0
0	0	0	11	11	11	11	11	11	11	11	11	11	0	0	0	12	12	12	12	12	12	12	12	12	12	0	0	0	13	13
0	10	10	10	10	10	10	10	10	0	0	0	0	0	11	11	11	11	11	11	11	11	0	0	0	0	0	12	12	12	12
0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	10	10	0	0	0	0	0	0	0	0	0	0	0	11	11	0	0	0	0
0	0	9	9	9	0	0	0	0	0	0	0	0	0	0	10	10	10	0	0	0	0	0	0	0	0	0	0	11	11	11
0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
6	6	6	6	6	6	6	6	6	6	6	6	6	0	0	0	0	0	0	0	0	10	10	10	10	10	10	10	10	10	10
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	0	0	0	0	0	0
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	0
0	5	0	0	5	0	5	0	5	2	5	2	0	6	0	0	6	0	6	0	6	0	3	6	3	0	7	0	0	7	8
0	0	5	5	0	0	0	0	0	0	0	0	0	0	6	6	0	0	0	0	0	0	0	0	0	0	0	7	7	0	0
0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	7
0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0			6	6	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6	6	6	6	6	6
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Figure 5.5: Gantt chart



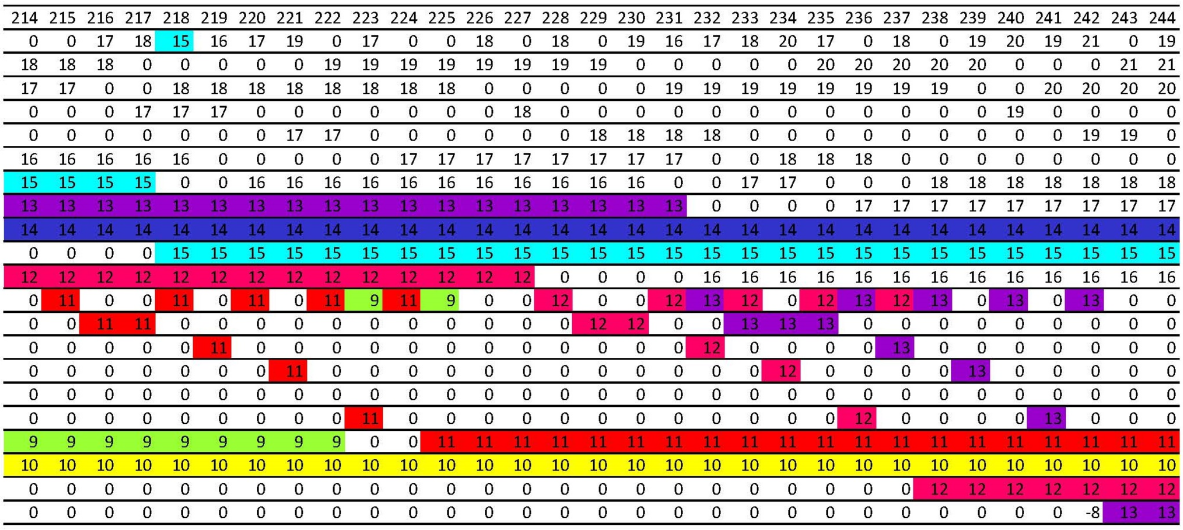
152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182
11	0	11	0	12	0	12	0	13	12	14	0	0	12	0	12	0	13	0	13	14	15	13	0	0	0	13	0	0	13	14
13	13	13	13	13	13	13	13	0	0	0	14	14	14	14	14	14	14	14	14	0	0	15	15	15	15	15	15	15	15	15
12	12	12	12	0	0	0	0	0	13	13	13	13	13	13	13	13	0	0	0	0	14	14	14	14	14	14	14	14	14	0
0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	12	12	0	0	0	0	0	0	0	0	0	0	0	13	13	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	12	12	12	0	0	0	0	0	0	0	0	0	0	13	13	13	0	0	0	0	0
0	11	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	13	13	0	0
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	0	0	0	0	0	0	0	0	13
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
0	0	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
7	0	7	8	4	7	4	8	0	8	0	8	0	0	0	0	0	0	5	0	5	0	9	0	0	9	0	9	0	9	0
8	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	9	0	0	0	0	0	0
0	0	0	0	8	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0
0	7	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	7	7	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
0	0	0	0	0	0	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	4	4	4	0	0	0	0	0	0	0	0	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8

Figure 5.6: Gantt chart



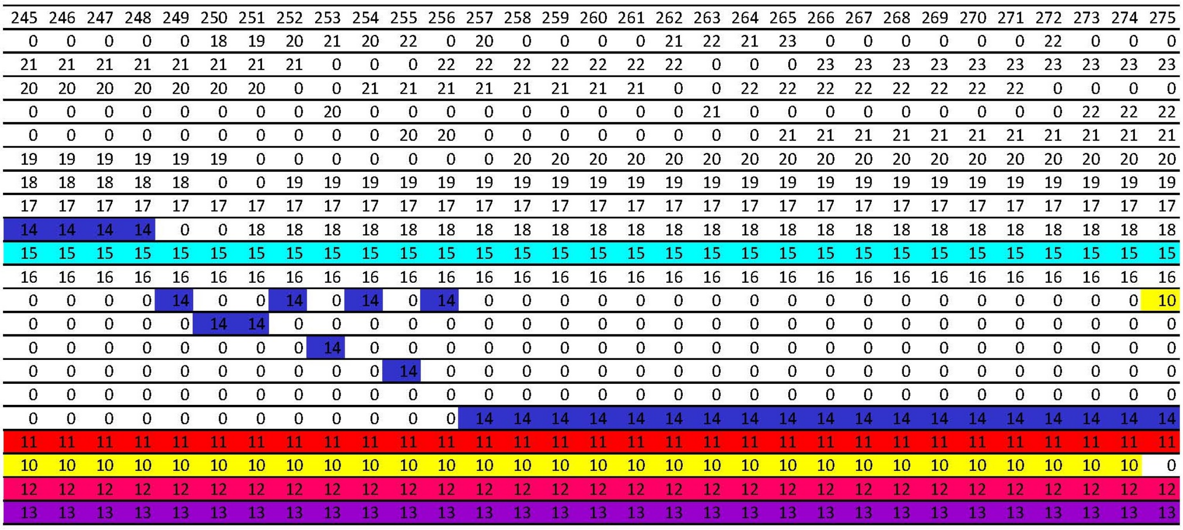
183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213
0	14	15	16	14	0	0	0	14	0	0	15	0	15	16	17	15	0	0	0	14	15	0	16	17	16	18	0	16	0	0
15	15	0	0	16	16	16	16	16	16	16	16	16	16	0	0	17	17	17	17	17	17	17	17	0	0	0	18	18	18	18
0	0	0	15	15	15	15	15	15	15	15	0	0	0	0	16	16	16	16	16	16	16	16	0	0	17	17	17	17	17	17
14	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	14	14	0	0	0	0	0	0	0	0	0	0	15	15	0	0	0	0	0	0	0	0	0	0	16	16	0	0	0
0	0	0	0	0	14	14	14	0	0	0	0	0	0	0	0	0	15	15	15	15	0	0	0	0	0	0	0	0	16	16
0	0	0	0	0	0	0	0	0	14	14	14	14	14	14	14	14	14	14	14	0	0	15	15	15	15	15	15	15	15	15
13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	0	0	14	14	14	14	14	14	14	14	14	14
11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
6	9	6	0	0	0	8	0	8	0	0	0	0	7	0	7	0	0	0	10	0	0	10	0	10	0	10	0	10	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0
0	0	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10
7	7	7	7	7	7	7	7	7	7	7	7	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	8	8	8	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5.7: Gantt chart



214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244
0	0	17	18	15	16	17	19	0	17	0	0	18	0	18	0	19	16	17	18	20	17	0	18	0	19	20	19	21	0	19
18	18	18	0	0	0	0	0	19	19	19	19	19	19	19	19	0	0	0	0	0	20	20	20	20	20	0	0	0	21	21
17	17	0	0	18	18	18	18	18	18	18	18	0	0	0	0	0	19	19	19	19	19	19	19	19	0	0	20	20	20	20
0	0	0	17	17	17	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0
0	0	0	0	0	0	0	17	17	0	0	0	0	0	0	18	18	18	18	0	0	0	0	0	0	0	0	0	19	19	0
16	16	16	16	16	0	0	0	0	0	17	17	17	17	17	17	17	17	0	0	18	18	18	0	0	0	0	0	0	0	0
15	15	15	15	0	0	16	16	16	16	16	16	16	16	16	16	16	0	0	17	17	0	0	0	18	18	18	18	18	18	18
13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	0	0	0	0	17	17	17	17	17	17	17	17	17
14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
0	0	0	0	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
12	12	12	12	12	12	12	12	12	12	12	12	12	12	0	0	0	0	16	16	16	16	16	16	16	16	16	16	16	16	16
0	11	0	0	11	0	11	0	11	9	11	9	0	0	12	0	0	12	13	12	0	12	13	12	13	0	13	0	13	0	0
0	0	11	11	0	0	0	0	0	0	0	0	0	0	0	12	12	0	0	13	13	13	0	0	0	0	0	0	0	0	0
0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	13	0	0	0	0	0	0	0
0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	13	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	13	0	0	0
9	9	9	9	9	9	9	9	9	0	0	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	12	12	12	12	12	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-8	13	13

Figure 5.8: Gantt chart



245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
0	0	0	0	0	18	19	20	21	20	22	0	20	0	0	0	0	21	22	21	23	0	0	0	0	0	0	22	0	0	0
21	21	21	21	21	21	21	21	0	0	0	22	22	22	22	22	22	22	0	0	0	23	23	23	23	23	23	23	23	23	23
20	20	20	20	20	20	20	0	0	21	21	21	21	21	21	21	21	0	0	22	22	22	22	22	22	22	22	0	0	0	0
0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	22	22	22
0	0	0	0	0	0	0	0	0	0	20	20	0	0	0	0	0	0	0	0	21	21	21	21	21	21	21	21	21	21	21
19	19	19	19	19	19	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
18	18	18	18	18	0	0	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17
14	14	14	14	0	0	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
0	0	0	0	14	0	0	14	0	14	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
0	0	0	0	0	14	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	0
12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13

Figure 5.9: Gantt chart

276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306
23	24	0	0	19	20	21	22	0	23	0	20	21	22	23	24	25	21	22	23	0	0	0	0	24	25	24	26	0	0	0
0	0	24	24	24	24	24	24	24	24	24	24	24	24	24	0	0	25	25	25	25	25	25	25	25	0	0	0	26	26	26
0	23	23	23	23	23	23	23	23	0	0	0	0	0	0	0	24	24	24	24	24	24	24	24	0	0	25	25	25	25	25
22	22	22	22	22	22	22	0	0	0	23	23	23	23	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0
21	21	21	21	21	21	0	0	22	22	22	22	22	0	0	23	23	23	23	0	0	0	0	0	0	0	0	24	24	24	24
20	20	20	20	20	0	0	21	21	21	21	21	0	0	22	22	22	22	0	0	23	23	23	23	23	23	23	23	23	23	23
19	19	19	19	0	0	20	20	20	20	20	0	0	21	21	21	21	0	0	22	22	22	22	22	22	22	22	22	22	22	22
17	17	17	17	17	17	17	17	17	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
15	15	0	0	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	0	0	21	21	21	21	21	21	21	21	21	21	21	21	21
14	10	0	15	0	13	15	13	15	17	15	0	11	15	11	17	16	17	0	16	17	16	17	16	0	12	16	12	0	0	0
0	0	0	0	15	15	0	0	0	0	17	17	17	17	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	17	17	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	15	15	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0
11	11	11	11	11	11	11	11	11	11	11	11	0	0	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
0	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	0	0	16	16	16	16
13	13	13	13	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	17	17	17	17	17	17	17

Figure 5.10: Gantt chart

307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337
0	0	0	25	26	27	0	0	22	23	24	25	0	26	27	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	26	26	26	0	0	27	27	27	27	27	27	27	27	0	0	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28
25	25	25	0	0	26	26	26	26	26	26	26	26	0	0	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27
0	0	0	0	25	25	25	25	25	25	25	0	0	0	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26
24	24	24	24	24	24	24	24	24	24	0	0	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25
23	23	23	23	23	23	23	23	23	0	0	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
22	22	22	22	22	22	22	22	0	0	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
18	18	18	18	0	0	0	0	0	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21
0	0	0	0	18	0	0	18	14	18	14	18	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17
0	0	0	0	0	18	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
14	14	14	14	14	14	14	14	0	0	0	0	0	0	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	0

Figure 5.11: Gantt chart

5.1.2 FIFO

[10, 15, 15, 25, 25, 25, 25, 15, 25, 25, 15, 25, 25, 25, 25, 25, 15, 25, 15, 25, 25, 25, 15, 25, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15]

Last beam completed anodising: 14:54

5.1.3 Priority

[**15, 15, 25, 25, 25, 25, 25, 25, 15, 15, 25**, 15, 15, 15, 15, 15, 10, 15, 25, 25, 15, 15, 15, 15, 15, 15, 25, 25, 25, 25, 25, 25, 15, 25]

Last beam completed anodising: 14:53

5.1.4 Shortest Processing Time

[10, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25]

Last beam completed anodising: 15:09

5.2 22 August 2017- Day shift

On the 22nd of August 2017, there was another problem with caustic rinse tank temperatures, which influenced the number of beams processed. Processing only started after 09:00.

Beams processed: 34

Square meters processed: 2 263

5.2.1 Current Schedule

[15, 15, 25, 15, 25, 25, 25, 25, 10, 25, 15, 10, 15, 25, 15, 15, 10, 10, 15, 15, 25, 15, 25, 15, 25, 10, 25, 25, 25, 25, 15, 15, 15, 15]

Last beam completed anodising: 14:29

5.2.2 FIFO

[25, 15, 15, 15, 15, 15, 15, 25, 15, 15, 15, 15, 15, 10, 10, 15, 10, 10, 25, 25, 15, 25, 25, 15, 25, 25, 25, 25, 10, 25, 25, 25, 15, 25]

Last beam completed anodising: 14:54

5.2.3 Priority

[**15, 15, 25, 25, 15**, 15, 15, 25, 15, 25, 25, 25, 10, 25, 15, 10, 15, 25, 15, 10, 10, 15, 25, 15, 15, 25, 10, 25, 25, 25, 25, 15, 15, 15]

Last beam completed anodising: 14:23

5.2.4 Shortest Processing Time

[10, 10, 10, 10, 10, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25]

Last beam completed anodising: 15:09

5.3 23 August 2017- Day shift

On 23 August 2017, processing did not start later than planned, but other breakdowns and human interferences could influence the actual processing time in comparison with the model output.

Beams processed: 42

Square meters processed: 2 655

5.3.1 Current schedule

[15, 15, 10, 15, 25, 15, 25, 15, 15, 25, 25, 25, 15, 15, 15, 15, 15, 15, 15, 15, 25, 15, 25, 25, 25, 25, 25, 25, 25, 15, 15, 25, 15, 25, 15, 25, 25, 25, 25, 25, 15, 15]

Last beam completed anodising: 16:48

5.3.2 FIFO

[15, 15, 15, 15, 15, 15, 15, 15, 10, 25, 25, 15, 25, 25, 25, 25, 25, 15, 15, 15, 15, 15, 15, 25, 25, 25, 15, 25, 15, 15, 15, 15, 15, 25, 15, 25, 25, 25, 25, 25, 25, 25]

Last beam completed anodising: 16:35

5.3.3 Priority

[**25, 25, 15, 25, 25, 25, 25, 25, 15, 15, 15, 15, 15, 15**, 15, 10, 25, 15, 15, 25, 15, 15, 15, 15, 15, 15, 15, 15, 25, 15, 25, 25, 25, 25, 25, 15, 15, 25, 25, 25, 25, 25]

Last beam completed anodising: 16:25

5.3.4 Shortest Processing Time

[10, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25]

Last beam completed anodising: 16:50

5.4 The best heuristic

The results that take the four possible scheduling methods into consideration show that the schedule does not make a large difference in the makespan of the process.

The priority method used produces the shortest makespan for the 22nd and 23rd of August, while the difference between the shortest makespan and that of the current method is only 5 to 33 minutes. This result indicates that a change in scheduling method is not significant. The current method provides the process with a relatively short makespan, with the team keeping the mindset that they do not need to make drastic changes to what they are used to. It would, however, be advised that all priority work is finished during the night shift to ensure they are all completed before the end of the day.

5.5 Validation

Although the algorithmic model cannot be compared to an existing model, certain characteristics can be compared to that of reality. In reality, there is human interference and breakdowns (which forces the operator of the PLC to set the AAP to manual). These contributing factors cause the makespan to be longer than the theoretical makespan (provided by the algorithmic model). All the days regulated indicate in Table 5.1, that the theoretical makespan is smaller than that of the actual makespan. The makespan indicated in this table is measured from the minute the first beam for the day enters the anodising tank, until the last beam is removed from the anodising tank.

Table 5.1: Theoretical vs Actual makespan

Date	Theoretical makespan (minutes)	Actual makespan (minutes)
7 Aug	541	610
22 Aug	489	496
23 Aug	648	720

5.6 Sensitivity Analysis

Sensitivity analysis is done on the number of tanks and a higher demand to determine the effect the change on the model and on the production line.

5.6.1 More Sealing or Anodising tanks

To determine the effect of more critical tanks on the process outcomes, the model was changed and the effect tabulated. In reality, the four anodising tanks that were used in the model consists of two tanks with two stations each. By adding another tank, the process gains two extra stations. Table 5.2 indicates the results for 23 August with the schedule as Wispeco currently uses. The effect of more anodising and sealing stations are investigated in terms of the makespan up to the anodising process and in terms of the makespan up to the sealing process.

Table 5.2: Effect of More Tanks in minutes

	4 stations each	6 anodising stations, 4 sealing stations	6 stations each
Makespan up to anodising	648	599	605
Makespan up to last tank	713	697	654

These results show that an extra anodising tank in the AAP can reduce the total production time by 16 minutes, where the AAP produces an average of 3,5m^2 per minute. An additional anodising tank can thus result in 56m^2 additional production. An additional anodising and a sealing tank results in a reduced production time of an additional 43 minutes (thus 59 minutes in total), which can result in 207m^2 produced more in a single shift. The sales for 2016 were on average R45/m^2 and the costs R30/m^2. The result of such a change could thus be R 3 100 more profit per shift or R124 000 per month of double shifts. The advantage of this decision will also be that there are no capital expenses for adding tanks, as the tanks for the current second production line could be used for the AAP. There will, however, be additional electricity costs which were not accounted for in this calculation.

5.6.2 Higher demand

The model can take any input, but will not successfully execute if there are too many orders to be processed during one 12-hour shift. A number of different combinations were tested and Table 5.3 indicates what the maximum orders are if all orders are of the same anodising thickness.

Table 5.3: Maximum orders

Anodising thickness	Maximum number of beams processed
10μm	58
15μm	52
25μm	35

These tests also indicated that there are only 3 sealing and anodising tanks necessary if all the orders are 10μm. The FLM can thus note that if there is a day where one of the tanks are out of order, it would be most beneficial to complete all the 10μm orders before underutilising the plant by waiting for a tank to open.

Chapter 6

Conclusion and Recommendations

6.1 Conclusion

The aluminium anodising plant at Wispeco Aluminium needs to implement a scheduling system to ensure minimum makespan of the process. To solve this problem, a mathematical model was formulated and solved by a heuristic method developed in Python. The model focuses on characteristics of reality like the two cranes which cannot cross over one another and the cranes both have a capacity of one. The model takes into consideration that the beams have a critical processing time where after the beams need to be moved to the next station (because over processing causes defects).

The algorithm is used to test three types of heuristics. All three heuristic scheduling approaches gave similar results. The suggestion to Wispeco is to keep the scheduling process as it currently is. The current scheduling approach produced the shortest makespan for two out of the three days that were tested. Keeping the scheduling system is better for the workforce, as they are comfortable with the known. The theoretical makespan of the model was compared with the actual makespan measured on three different days. All the theoretical makespans are shorter than the actual, which is realistic if factors like human error are considered.

There has not yet been any conclusion on number of cranes to be used on the AAP.

6.2 Recommendations

It is recommended that the scheduling procedure of the anodising plant at Wispeco aluminium is not changed. The current scheduling approach produces a schedule that has an adequate makespan. It is also recommended that members of management consider moving an anodising and a sealing tank from the current MAP to the AAP as soon as the plant is only operating on a single line. It could have financial benefits for the plant.

6.2.1 Future Projects

Different projects that can be pursued at the anodising plant could be an electronic administration system, Statistical Process Control and a more advanced scheduling algorithm.

The entire administration trail used in the anodising plant is paper based. The job cards for each beam is moved through the process with the beam, but sometimes the job cards fall in the chemicals and the information is lost. The operator of the PLC also receives a paper with the desired anodising thickness of each beam. All these paper based administration tasks can be done electronically. An electronic system will reduce the need to copy information by hand to ensure everyone has the desired information and it will also reduce the risk of losing information. A system could be implemented to interact with the current Enterprise Resource Planning (ERP) system.

Currently the anodising process does not always produce the desired thickness, even if the anodised material is in the anodising tank for the required time or even longer. Statistical Process Control can be applied to the anodising plant to determine the effect of parameters on the anodising thickness and to help eliminate assignable causes.

A more advanced scheduling algorithm will produce better results. This model only finds the best schedule between four options. A more advanced model could iterate all possible combinations to find the optimal scheduling solution.

Bibliography

[1] A.W. Barce and P.G. Sheasby. *The Technology of Anodizing Aluminium.* Technicopy Ltd., 2 edition, 1979.

[2] Peter Brucker. *Scheduling Algorithms.* Springer, 2004.

[3] DomagojJakobovi and Kristina Marasovi. Evolving priority scheduling heuristics with genetic programming. *Elsevier*, 2012.

[4] Jacomine Grobler. Particle swarm optimization and differential evolution for mulit-objective multiple machine scheduling. Master's thesis, University of Pretoria, 2008.

[5] Deepak Gupta, Sameer Sharma, and Shefali Aggarwal. Flow shop scheduling on 2-machines with setup time and single transport facility under fuzzy environment. *Springer Science & Business Media*, 2013.

[6] Robert C Hansen. *Overall equipment effectiveness: a powerful production/maintenance tool for increased profits.* Industrial Press, Inc., 2001.

[7] Jeffrey W. Herrmann, editor. *Handbook of Production Scheduling.* Springer, 2006.

[8] James F. Cox III and Jr. John G. Schleier. *Theory of Constraints Handbook.* McGraw-Hill Education, 2010.

[9] Bassem Jarboui, Patrick Siarry, and Jacques Teghem, editors. *Metaheuristics for Production Scheduling.* Wiley, 2013.

[10] Khethiwe Kunene. Variable data in an anodising plant. Data gathering project as Chemical Engineer of Wispeco Anodising plant., 2017.

[11] Khethiwe Kunene and Carina Behr. The dream anodising plant. January 2017 Project on a dream anodising plant at Wispeco, 01 2017.

[12] Xin Li, Felix T.S. Chan, and S.H. Chung. Optimal multi-degree scheduling of multiple robots without overlapping in robotic flowshops with parallel machines. *Elsevier*, 2015.

[13] Joon-Mook Lim. A genetic algorithm for a single hoist scheduling in the printed-circuitboard electroplating line. *Pergamon*, 1997.

[14] United Anodisers Ltd. Anodising vs powder coating. Website, 2016.

[15] Wispeco Pty Ltd. The largest aluminium extruder in africa. Webpage.

[16] Ying Ma, Chengbin Chu, and Chunrong Zuo. A survey of scheduling with deterministic machine availability constraints. *Elsevier*, 2010.

[17] Vladimir Marik, Luis M. Camarinha-Matos, and Hamideh Afsarmanesh, editors. *Knowledge and Technology Integration in Production and Services*. Springer, 2002.

[18] Michael L. Pinedo. *Scheduling Theory, Algorithms and Systems*. Springer, 2012.

[19] AR Rahani. Production flow analysis through value stream mapping: A lean manufacturing process case study. *Engineering Preocedia*, 2012.

[20] R.L.Graham, E.L.Lawler, J.K.Lenstra, and A.H.G.Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 1979.

[21] Xin-She Yang. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.

Appendix A

Signed 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 Aluminium**
Project Description:	**Production scheduling of aluminium profiles to be processed at an anodising plant.**
Student Name:	**Carina Marié Behr**
Student number:	**14010047**
Student Signature:	
Sponsor Name:	**Roland Röhrs**
Designation:	**Wispeco Chairman**
E-mail:	**roland@lantic.net**
Cell No:	**082 654 2246**
Sponsor Signature:	

Appendix B

Priority Indication Form

AAP FINISHING: SHIFT HANDOVER PROCEDURE

WISPECO Aluminium

DATE : 07 AUGUST 2017

FLM Night Shift: F. MONONYANA

CUSTOMERS	PRIORITIES	FEEDBACK ON PRIORITIES		AREA OF ID.
Customer	Product Code	Qty Scheduled	Qty Produced	Allocation
WISPECO IMPORTS	556846000N25R5	230	✓	IN SKIPS
ANSO BLOEM	558247250N15R5	20	✓	# F288
WISPECO GS	AS PER SCHEDULE	15 ORDERS	14 Done	SCHEDULED
ALUGLASS	AS PER SCHEDULE	3 ORDERS	✓	IN SKIPS
IDEAL ENGINEER	295586100N15R5	1093	Balance	#B137;142;146/7

TANK TEMPERATURES	DAY SHIFT	NIGHT SHIFT
Ano Tank 1 (78 & 79)	19	20
Ano Tank2 (Station 810 & 811)	19	19
Caustic	68	68
Stripping	50	50
Sealing Tank 1	96	96
Sealing Tank 2	96	96

No of Loads at Shift Change over	
Day Shift	Night Shift
5 Total Jigs (Target 5)	

Notes - Jigging Material (Clamps, wire, jigs, etc.)

No of Trolleys at Shift Change over	
Day Shift	Night Shift
1 Total Trolleys (Target 1.5)	

Housekeeping Notes

	Absentees	L.D.W's	Housekeeping
Day Shift	THARM	S	
Night Shift	NO	5	

Balances at the end of shift

Customer	Product Code	Qty Produced	Balance	Shift

Reworks left at the end of shift

Customer	Product Code	Qty left to be reworked	Shift

Messages / Notes

Day Shift: Signature: ______________ (FLM)

Night Shift: Signature: ______________ (FLM)

FLM`S PLEASE RECORD SCRAP ACCORDINGLY AT THE BACK, PLEASE GUYS !!!!!!!!!!!!

Appendix C

Python Algorithm

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jun 26 08:26:54 2017

@author: carina
"""

import sys
craneBus = [[0],[0]] #Crane busy/idle for a timeframe.  If busy, with which job
task =[15,15,15,15,15,15,15,15,15,15,10,10,10,10,25,25,25,25,25] #FIFO : One day's jobs
tank = [[],[],[],[],[],[],[],[],[],[],[],[],[]]  #timeframe in use/idle for a tank
anodisingtanks = [[],[],[],[]]
sealtanks = [[],[],[],[]]
frinsetanks = [[],[]]
done = [[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]] #Processes complete/incomplete (1/0) for a job
cranesplit = 6 #Index of Last tank that crane 1 moves to
cranemovetime = 1 #minutes for crane movment between processing times
processingtime =[2,8,1,2,3,1,30,2,1,1,1,30,1]
anodising = 6
anodisetime = [30,45,60]
seal = 11
sealtime = [25,38,63]
frinse = 9
daymin = 12*60
lastjob = [0,0,0,0,0,0,0,0,0,0,0,0,0]
jobtoano = []
jobtofrinse = []
jobtoseal = []
firsttry = 1
firstsealtry = 1
firstfrinsetry = 1
criticaltanks = [1,2,4,6]

for i in range(0,len(done)):
    jobtoano.append(0)
    jobtoseal.append(0)
    jobtofrinse.append(0)
    for u in range(0,len(tank)):
        done[i].append(0)

for i in range(0,daymin): #minutes in 12 hour shift
    craneBus[0].append(0)
    craneBus[1].append(0)
    for u in range(0,len(anodisingtanks)):
        anodisingtanks[u].append(0)
        sealtanks[u].append(0)
    for u in range(0,len(tank)):
        tank[u].append(0)
    for u in range(0,len(frinsetanks)):
        frinsetanks[u].append(0)
timedone = 0
varbreak = False

for job in range(0,len(task)):
    timeint = 0
    if task[job] == 10:
        processingtime[anodising] = anodisetime[0]
        processingtime[seal] = sealtime[0]
    elif task[job] == 15:
        processingtime[anodising] = anodisetime[1]
        processingtime[seal] = sealtime[1]
    elif task[job] == 25:
        processingtime[anodising] = anodisetime[2]
        processingtime[seal] = sealtime[2]
    for process in range(0,len(tank)):
        if (timeint + processingtime[process] + cranemovetime < daymin):
            toolittletime = 0
            while (toolittletime == 0) and (done[job][process] == 0):
                if process <= cranesplit:
                    cranenr = 0
                else:
                    cranenr = 1
                movevalid = True
                stopwhile = 0
                original = timeint
                while (movevalid) and (stopwhile == 0) and (toolittletime == 0):
                    varbreak = False
                    findanotank = 0
                    findsealtank = 0
                    findfrinsetank = 0
                    for i in range(timeint, processingtime[process] + timeint+1):
                        if varbreak == True:
                            break
                        if (timeint + processingtime[process] + cranemovetime > daymin):
                            toolittletime = 1
                            break
                        if process == 0:
                            if (craneBus[cranenr][timeint] == 0):
                                if (tank[process][timeint-1] == 0) and (tank[process][timeint] == 0) and (tank[process][i+1] == 0) and (tank[p
                                    movevalid = True
                                    if (movevalid) and (i == processingtime[process] + timeint):
                                        stopwhile = 1
                                else:
                                    timeint += 1
                                    if (timeint + processingtime[process] + cranemovetime > daymin):
                                        toolittletime = 1
                                    movevalid =False
                                    break
                            else:
                                timeint +=1
                                if timeint > i:
                                    break
                        elif process == cranesplit:
                            allanofull = 0
                            for u in range(0,len(anodisingtanks)):
                                if anodisingtanks[u][timeint-1] != 0:
                                    allanofull = 1
                                    if u == len(anodisingtanks)-1:
                                        timeint +=1
                                        varbreak = True
                                        break
                                else:
                                    if (craneBus[cranenr][timeint] == job + 1) or (craneBus[cranenr][timeint] == 0):
                                        if (craneBus[cranenr][timeint] == 0) and (firsttry == 1):
                                            firsttry = 0
                                            for r in range(timeint,0,-1):
                                                if craneBus[cranenr][r] == job + 1:
                                                    craneBus[cranenr][r] = 0
                                                    break
                                        if findanotank == 0:
                                            if (anodisingtanks[u][timeint-1] == 0) and (anodisingtanks[u][timeint] == 0) and (anodisingtanks[u
                                                movevalid = True
                                                jobtoano[job] = u
                                                findanotank = 1
                                                if (movevalid) and (i == processingtime[process] + timeint):
                                                    stopwhile = 1
                                                    varbreak = True
                                                break

                                            else:
                                                timeint += 1
                                                varbreak = True
                                                if (timeint + processingtime[process] + cranemovetime > daymin):
                                                    toolittletime = 1
                                                movevalid =False
                                                break
                                        else:
                                            if (anodisingtanks[u][timeint-1] == 0) and (anodisingtanks[u][timeint] == 0) and (anodisingtanks[u
                                                movevalid = True
                                                if (movevalid) and (i == processingtime[process] + timeint):
                                                    stopwhile = 1
                                                break
                                            else:
                                                timeint += 1
                                                varbreak = True
                                                if (timeint + processingtime[process] + cranemovetime > daymin):
                                                    toolittletime = 1
                                                movevalid =False
                                                if timeint > i:
                                                    break
                                    else:
                                        timeint += 1
                                        varbreak = True
                                        if (timeint + processingtime[process] + cranemovetime > daymin):
                                            toolittletime = 1
                                        varbreak = True
                                        break
                        elif process == seal:
                            allsealfull = 0
                            for s in range(0,len(sealtanks)):
                                if sealtanks[s][timeint-1] != 0:
                                    allsealfull = 1
                                    if s == len(sealtanks)-1:
                                        timeint +=1
                                        varbreak = True
                                        break
                                else:
                                    if (craneBus[cranenr][timeint] == job + 1) or (craneBus[cranenr][timeint] == 0):
                                        if (craneBus[cranenr][timeint] == 0) and (firstsealtry == 1):
                                            firstsealtry = 0
                                            for r in range(timeint,0,-1):
                                                if craneBus[cranenr][r] == job + 1:
                                                    craneBus[cranenr][r] = 0
                                                    break
                                        if findsealtank == 0:
                                            if (sealtanks[s][timeint-1] == 0) and (sealtanks[s][timeint] == 0) and (sealtanks[s][i+1] == 0):
                                                movevalid = True
                                                jobtoseal[job] = s
                                                findsealtank = 1
                                                if (movevalid) and (i == processingtime[process] + timeint):
                                                    stopwhile = 1
                                                    varbreak = True
                                                break

                                            else:
                                                timeint += 1
                                                varbreak = True
                                                if (timeint + processingtime[process] + cranemovetime > daymin):
                                                    toolittletime = 1
                                                movevalid =False
                                                break
                                        else:
                                            if (sealtanks[s][timeint-1] == 0) and (sealtanks[s][timeint] == 0) and (sealtanks[s][i+1] == 0):
                                                movevalid = True
                                                if (movevalid) and (i == processingtime[process] + timeint):
                                                    stopwhile = 1
                                                break
                                            else:
                                                timeint += 1
                                                varbreak = True
                                                if (timeint + processingtime[process] + cranemovetime > daymin):
                                                    toolittletime = 1
                                                movevalid =False
                                                if timeint > i:
                                                    break
                                    else:
                                        timeint += 1
                                        varbreak = True
                                        if (timeint + processingtime[process] + cranemovetime > daymin):
                                            toolittletime = 1
                                        varbreak = True
                                        break
                        elif process == frinse :
                            allfrinsefull = 0
                            for f in range(0,len(frinsetanks)):
                                if frinsetanks[f][timeint-1] != 0:
                                    allfrinsefull = 1
                                    if f == len(frinsetanks)-1:
                                        timeint +=1
                                        varbreak = True
                                        break
                                else:
                                    if (craneBus[cranenr][timeint] == job + 1) or (craneBus[cranenr][timeint] == 0):
                                        if (craneBus[cranenr][timeint] == 0) and (firstfrinsetry == 1):
                                            firstfrinsetry = 0
                                            for r in range(timeint,0,-1):
                                                if craneBus[cranenr][r] == job + 1:
                                                    craneBus[cranenr][r] = 0
                                                    break
                                        if findfrinsetank == 0:
                                            if (frinsetanks[f][timeint-1] == 0) and (frinsetanks[f][timeint] == 0) and (frinsetanks[f][i+1] ==
                                                movevalid = True
                                                jobtofrinse[job] = f
                                                findfrinsetank = 1
                                                if (movevalid) and (i == processingtime[process] + timeint):
                                                    stopwhile = 1
                                                    varbreak = True
                                                break

                                            else:
                                                timeint += 1
                                                varbreak = True
                                                if (timeint + processingtime[process] + cranemovetime > daymin):
                                                    toolittletime = 1
                                                movevalid =False
                                                break
                                        else:
                                            if (frinsetanks[f][timeint-1] == 0) and (frinsetanks[f][timeint] == 0) and (frinsetanks[f][i+1] ==
                                                movevalid = True
                                                if (movevalid) and (i == processingtime[process] + timeint):
                                                    stopwhile = 1
                                                break
                                            else:
                                                timeint += 1
                                                varbreak = True
                                                if (timeint + processingtime[process] + cranemovetime > daymin):
                                                    toolittletime = 1
                                                movevalid =False
                                                if timeint > i:
                                                    break
                                    else:
                                        timeint += 1
                                        varbreak = True
                                        if (timeint + processingtime[process] + cranemovetime > daymin):
                                            toolittletime = 1
                                        varbreak = True
                                        break
                        else:
                             if (craneBus[cranenr][timeint] == 0) or (craneBus[cranenr][timeint] == job + 1):
                                 if(tank[process][timeint-1] == 0) and (tank[process][timeint] == 0) and (tank[process][i+cranemovetime] == 0)
```

```
                     movevalid = True
                     if (movevalid) and (i == processingtime[process] + timeint):
                         stopwhile = 1
                         craneBus[cranenr][original] = 0
                         craneBus[cranenr][timeint] = job +1
                 else:
                     timeint += 1
                     varbreak = True
                     if (timeint + processingtime[process] + cranemovetime > daymin):
                         toolittletime = 1
                     movevalid =False
                     if timeint > i:
                         break
             else:
                 timeint += 1
                 varbreak = True
                 if (timeint + processingtime[process] + cranemovetime > daymin):
                     toolittletime = 1
                 movevalid =False
                 if timeint > i:
                     break
if movevalid:
    done[job][process] = 1
    if (process == 0):
        craneBus[cranenr][timeint] = job + 1 #before processing
        timedone = timeint + processingtime[process]
        timeint +=1
        for i in range(timeint, timedone+1):
            if i + cranemovetime < daymin:
                tank[process][i] = job + 1 #cranemovetime word benodig vir job0, proses0
    elif (process == cranesplit):
        if firsttry == 0:
            craneBus[cranenr][timeint] = job + 1 #before processing: if not directly into anodise
        timedone = timeint + processingtime[process]
        timeint +=1
        for i in range(timeint, timedone+1):
            if i + cranemovetime < daymin:
                anodisingtanks[u][i] = job + 1
    elif (process == seal):
        if firstsealtry == 0:
            craneBus[cranenr][timeint] = job + 1 #before processing: if not directly into anodise
        timedone = timeint + processingtime[process]
        timeint +=1
        for i in range(timeint, timedone+1):
            if i + cranemovetime < daymin:
```

```
                                sealtanks[s][i] = job + 1
                    elif (process == frinse):
                        if firstfrinsetry == 0:
                            craneBus[cranenr][timeint] = job + 1 #before processing: if not directly into anodise
                        timedone = timeint + processingtime[process]
                        timeint +=1
                        for i in range(timeint, timedone+1):
                            if i + cranemovetime < daymin:
                                frinsetanks[f][i] = job + 1
                    else:
                        timedone = timeint + processingtime[process]
                        for i in range(timeint, timedone):
                            if i + cranemovetime < daymin:
                                tank[process][i+1] = job + 1
                    timeint = timedone + cranemovetime
                    if timeint < daymin:
                        if process== cranesplit:
                            cranenr += 1
                        craneBus[cranenr][timeint] = job + 1 #timeint without cranemovetime,after processing
        else:
            print 'The day is too short to complete all this work.  We could only complete up to job nr ' + str(job+1)
            lastjob[process] = job
            break


#Keep jobs in tank until it is fetched by a crane
    for process in range(0,len(tank)):
        if job == lastjob[process]:
            break
        if process <= cranesplit:
            cranenr = 0
            print 'proses ' + str(process+1)
            print str(tank[process])
            print 'kraan ' + str(cranenr+1)
            print str(craneBus[cranenr])
        else:
            cranenr = 1
            print 'processs ' + str(process+1)
            print str(tank[process])
            print 'crane ' + str(cranenr+1)
            print str(craneBus[cranenr])
        if process == cranesplit:#anodising and crane split has been mixed
            lasttime = len(anodisingtanks[u]) - anodisingtanks[u][::-1].index(job+1) - 1
            if anodisingtanks[u][lasttime] != craneBus[cranenr+1][lasttime+1]:
                cranepickup = False
                while cranepickup == False:
                    for i in range(lasttime+1,len(craneBus[cranenr+1])-1):
                        if i > daymin:
                            cranepickup = True
                            break
                        if craneBus[cranenr+1][i+1] == job + 1:
                            anodisingtanks[u][i] = job + 1
                            timepickup = i
                            cranepickup = True;
                            for r in range(lasttime + 1,timepickup):
                                anodisingtanks[u][r] = job + 1
                            break
                        else:
                            continue
                    if (cranepickup == False) and (i == len(craneBus[cranenr+1])-lasttime):
                        break
        elif process == seal:
            lasttime = len(sealtanks[s]) - sealtanks[s][::-1].index(job+1) - 1
            if lasttime+1 > daymin:
                break
            if sealtanks[s][lasttime] != craneBus[cranenr][lasttime+1]:
                cranepickup = False
                while cranepickup == False:
                    for i in range(lasttime+1,len(craneBus[cranenr])-1):
                        if i > daymin:
                            cranepickup = True
                            break
                        if craneBus[cranenr][i+1] == job + 1:
                            sealtanks[s][i] = job + 1
                            timepickup = i
                            cranepickup = True;
                            for r in range(lasttime + 1,timepickup):
                                sealtanks[s][r] = job + 1
                            break
                        else:
                            continue
                    if (cranepickup == False) and (i == len(craneBus[cranenr])-lasttime):
                        break
        elif process == frinse:
            lasttime = len(frinsetanks[f]) - frinsetanks[f][::-1].index(job+1) - 1
            if lasttime+1 > daymin:
                break
            if frinsetanks[f][lasttime] != craneBus[cranenr][lasttime+1]:
                cranepickup = False
                while cranepickup == False:
```

Appendix D

Schedule Gantt Chart: 7 August 2017

338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368
0	0	0	0	23	24	0	25	26	24	0	25	26	25	27	28	26	27	29	0	27	0	0	0	28	29	28	26	27	28	30
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	0	0	0	0	29	29	29	29	29	29	0	0	0	0	0	0
27	27	27	27	27	27	27	27	27	27	27	27	27	27	0	0	28	28	28	28	28	28	28	28	0	0	29	29	29	29	29
26	26	26	26	26	26	26	26	0	0	0	0	0	0	0	27	27	0	0	0	0	0	0	0	0	28	0	0	0	0	0
25	25	25	25	25	25	25	0	0	26	26	26	0	0	0	0	0	0	27	27	0	0	0	0	0	0	0	28	28	0	0
24	24	24	24	24	0	0	0	25	25	25	0	0	26	26	26	0	0	0	0	0	27	27	27	27	27	27	27	0	0	28
23	23	23	23	0	0	24	24	24	0	0	0	25	0	0	0	0	26	26	26	26	26	26	26	26	26	26	0	0	27	27
20	20	20	20	20	20	20	20	20	20	20	0	0	0	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25
22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	0	0	0	0	26	26	26
19	19	0	0	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23
21	21	21	21	21	21	21	0	0	0	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
0	17	0	19	0	0	19	21	19	0	19	20	19	0	20	15	20	15	20	21	20	21	0	21	22	0	0	22	16	21	16
0	0	0	0	19	19	0	0	21	21	21	21	21	21	21	21	21	21	21	0	0	0	0	0	0	22	22	0	0	0	0
0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	20	0	0	0	0	21	0	0	0	0	0	0	0	22	22	22
0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	20	0	0	0	0	21	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	20	0	0	0	0	21	21	21	21	21	0	0	0	0
15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20
18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	0	0	21
0	0	0	0	0	0	0	0	0	0	0	0	0	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19

369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399
0	0	0	29	30	29	31	0	0	0	0	0	0	30	0	0	31	32	0	0	27	28	29	30	0	31	32	33	0	28	29
30	30	30	30	0	0	0	31	31	31	31	31	31	31	31	31	0	0	32	32	32	32	32	32	32	32	0	0	33	33	33
29	29	29	0	0	30	30	30	30	30	30	30	30	0	0	0	0	31	31	31	31	31	31	31	31	0	0	32	32	32	32
0	0	0	0	29	0	0	0	0	0	0	0	0	0	30	30	30	30	30	30	30	30	30	0	0	0	31	31	31	31	31
0	0	0	0	0	0	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	0	0	30	30	30	30	30	30	30
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	0	0	29	29	29	29	29	29	29	0
27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	0	0	28	28	28	28	28	28	28	0	0
25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25
26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26
23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	0	0	27	27	27	27	27	27	27	27	27	27	27
24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	0	0	0	0	0	28
22	0	22	0	0	0	0	0	0	0	0	0	0	0	0	18	22	18	0	23	0	0	23	0	23	24	23	0	24	0	24
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	23	0	0	0	0	24	24	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	24	0
0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	22	22	22	22	22	22	22	22	22	22	22	22	22	0	0	0	0	0	0	0	0	0	0	0	23	23	23	23	23	23
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	0	0	22	22	22	22	22	22	22	22	22	22	22	22	22	22
21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19

400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430
30	31	0	0	32	33	34	0	0	0	0	0	0	0	29	30	0	31	32	30	0	31	32	33	34	33	0	0	0	0	0
33	33	33	33	33	0	0	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	0	0	0	0	0	0	0
32	32	32	32	0	0	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	0	0	34	34	34	34	34	34
31	0	0	0	0	32	32	32	32	32	32	32	32	32	32	32	32	32	0	0	0	0	0	0	33	0	0	0	0	0	0
0	0	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	0	0	32	32	32	0	0	0	0	33	33	33	33	33
0	30	30	30	30	30	30	30	30	30	30	30	30	30	30	0	0	0	31	31	31	0	0	32	32	32	32	32	32	32	32
29	29	29	29	29	29	29	29	29	29	29	29	29	29	0	0	30	30	30	0	0	0	31	31	31	31	31	31	31	31	31
25	25	25	25	25	25	25	25	25	25	25	25	0	0	0	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29
26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	0	0	30	30	30	30	30	30	30	30	30	30	30
27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28
0	0	0	0	0	0	21	23	21	24	0	0	25	0	19	24	19	25	26	25	0	25	20	25	20	22	26	22	26	0	26
0	0	0	0	0	0	0	0	0	0	0	0	0	25	25	25	25	0	0	26	26	26	26	26	26	26	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	26	0	0	0
24	24	24	24	24	24	24	24	24	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	26	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	23	23	23	23	0	0	0	24	24	24	24	24	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	26	0
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	0	0	25	25	25	25	25	25	25
22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	0	0	0	0	0	0
21	21	21	21	21	21	0	0	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23
19	19	19	19	19	19	19	19	19	19	19	19	19	19	0	0	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24

431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461
0	0	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	32	33	34	0	0	0	0	0	0	32
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	0	0	0	0	0	0	0	0
33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	0	0	34	34	34	34	34	34	34
32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	0	0	33	33	33	33	33	33	33	33
31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	0	0	32	32	32	32	32	32	32	32	0
29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29
30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30
27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	0	0	31	31	31	31	31	31	31	31	31	31	31
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	0	0
0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	23	0	27	0	0	27	24	27	24	27	0	27	28	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	27	0	0	0	0	0	0	0	0	28
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0
25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25
0	0	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26
23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	27
24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	0	0	0	0	0	0	0	0

462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492
33	34	0	0	0	0	0	0	0	0	0	0	0	0	33	34	0	0	0	0	34	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	34	34	34	34	34	34	34	34	34	34	34	34	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	33	33	33	33	33	33	33	33	33	33	33	33	33	0	0	34	34	34	34	0	0	0	0	0	0	0	0	0	0	0
29	29	29	29	29	29	29	29	29	29	29	29	29	0	0	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	0	0	34	34	34	34	34	34	34	34	34	34
31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31
32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32
0	28	0	28	0	28	0	28	0	26	0	26	0	29	0	0	29	0	29	30	29	0	29	30	0	25	30	25	30	0	30
28	0	0	0	0	0	0	0	0	0	0	0	0	0	29	29	0	0	0	0	30	30	30	0	0	0	0	0	0	0	0
0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	0	0	0	0	0	0	30	30	0	0	0	0	0
0	0	0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	0	0	0	0	0	0	0	30	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	0	0	0	0	0	0	0	30	0	0	0
25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	0	0	0	0	0	0
26	26	26	26	26	26	26	26	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	29	29	29	29	29	29	29
27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27
0	0	0	0	0	0	0	0	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28

493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34
31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	0	0	0	0	0	0	0	0	0	0	0	0	0
32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0	0	31	0	31	0	31	0	0	32	27
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	31	0	0	0	0	0	0	0	0	32
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	31	31	31	0	0
30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30
29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29
27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	0
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28

524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	33	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	27	32	33	32	0	32	33	0	28	32	28	33	0	33	0	0	0	0	34	0	0	34	0	29	33	29	34	0	34	0
32	32	0	0	33	33	33	0	0	0	0	0	0	0	0	0	0	0	0	0	34	34	0	0	0	0	0	0	0	0	0
0	0	0	32	0	0	0	0	33	33	33	33	0	0	0	0	0	0	0	0	0	0	0	34	34	34	34	0	0	0	0
0	0	0	0	0	32	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	32	32	32	0	0	0	0	0	33	33	33	33	33	33	33	33	33	33	0	0	0	0	0	34	34	34
30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30
29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	0	0	33	33	33	33	33
0	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31
28	28	28	28	28	28	28	28	28	0	0	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32

555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	30	34	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34
33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31
32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32

586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	31	0	31	33	0	33	0	0	0	0	32	0	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34
33	33	33	33	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	32	32	32	32	32	32	32	32	32	32	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

617	618	619	620	621	622	623	624	625	626	627	628	629	630
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	34	0	34	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	34	34	34	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0