An optimization model to improve the production process: A case of a tag manufacturer in Sweden

Master's thesis in Supply Chain Management

Fredrik Dahl
Ramaswamy Maruthappan

DEPARTMENT OF TECHNOLOGY MANAGEMENT AND ECONOMICS
DIVISION OF SERVICE MANAGEMENT AND LOGISTICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se
REPORT No. E2022:137

REPORT NO. E2022:137

An optimization model to improve the production process: A case of a tag manufacturer in Sweden

Fredrik Dahl
Ramaswamy Maruthappan

Department of Technology Management and Economics
Division of Service Management and Logistics
Chalmers University of Technology
Gothenburg, Sweden 2022

An optimization model to improve the production process:
A case of a tag manufacturer in Sweden

Report No.E2022:137
Supervisor: Lokesh Kumar Kalahasthi, Researcher, Service Management and Logistics, Chalmers University of Technology and Management

Examiner: Ivan Sanchez-Diaz, Associate Professor, Service Management and Logistics, Chalmers University of Technology and Management

Industry Supervisor: Krister Svantesson, Head of Manufacturing and planning, Mark Bric AB

Master's Thesis 2022
Department of Technology Management and Economics
Division of Service Management and Logistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Gothenburg, Sweden 2022

Abstract

In recent years, the constant improvements in production planning has become increasingly important for manufacturing enterprises seeking to maximize their market value in a volatile industrial environment. This thesis is entitled towards providing an optimized production planning model. Mark Bric is a manufacturer of size marker hangers which are mainly used in the clothing industry. Size hangers are of different varieties which differ in size and color. When there are many varieties of products to be produced in manufacturing line it requires time for change-overs between each variant change. The aim of this thesis is to develop the resources needed for Mark Bric to become more efficient in their sequencing of these variation.

This thesis illustrates how an investment decision was analyzed by constructing an objective function, constraints, variables, and solving it using a mixed integer function created in Python. The authors of this thesis has researched the optimal number of machines needed to be procured to satisfy the future demand. Throughout this thesis 4 different numbers of machines (4-5-6-7 machines) are going to be analyzed over 4 different production demands, this will create 16 different scenarios of data which will be analyzed to make a the investment decision.

The comparison shows that using 6 machines allows for an advantage in terms of total machine run-time when compared to other scenarios. The majority of scenarios shows that there is a decline of around 50% compared to 4 machines and 68% compared to 5 machines. also, the overall change-over time during 6 machines is reduced. Therefore, it demonstrates that production alignment is well-organized while using 6 machines. Hence, purchasing 6 machines will be beneficial because it would allow Mark Bric to achieve a more efficient output.

Keywords: Production Planning, Mathematical Optimization, Operations Research, Mixed Integer Programming.

Acknowledgements

First of all, we would like to thank our supervisor Lokesh Kumar Kalahasthi at Chalmers University of Technology who guided us through sweat and tears during this thesis. His given inputs have played a major role in the final result.

We would also like to thank Krister Svantesson at Mark Bric who welcomed us an with warmth and openness to the direction of the scope of the thesis. Also, we would like to thank Karin Sundell and Kajo at Mark Bric who has has spent many hours providing, supporting and consulting this thesis.

We also like to thank Ivan Sanchez Diaz for his helpful ideas that helped us improve our thesis.

Finally, we would like to thank our friends and family for their continuous support and guidance, which has been a major role in guiding us in the right direction through out our life. Thank you all

Fredrik Dahl and Ramaswamy Maruthappan, Gothenburg, 2022

List of Acronyms

IJ	Injection moulding
MILP	Mixed-Integer programming
MTO	Make to order
ROP	Reorder point

Contents

List of Figures

List of Tables

1
Introduction

This chapter talks about the overview of the company's background, project background, aim and limitations of the project

1.1 Company background

Mark Bric AB (referred to as Mark Bric in the following) was founded by Gert Johansson in 1971, to create a more efficient shopping experience when locating sizes of different apparels in clothing stores. This invention, the marker is illustrated in figure 1.1 is also used for inventory tracking and making the process of identifying sizes easier. The marker is placed on the hangers in stores for customers to more easily identify just their variation of choice. In today's society, the marker is well-known all over the world due to its hassle-free buying experience and world wide presence in stores.

Today, Mark Bric has its operations in Sweden, USA, Germany and Hong Kong. The printing, packaging and the final delivery of plastic parts is carried out in Stenkullen, Sweden [1], which also is the headquarters of the company. The injection molding process of the several different variants is currently carried out in Borås, Sweden. After the moulding process the moulded parts are sent to the next process in process, printing and packaging and there after finally to a warehouse or directly to the customer. Mark Bric has an external warehouse in USA due to a large customer base in the united states and longer lead times.

1.2 Project background

Recently, the firm has decided to relocate the injection molding process from Borås to Stenkullen, bringing the entire production process to under the same roof. During this transition the company has made a decision to replace the 8 injection molding machines currently in usage with 5 new more efficient machines.

Mark Bric produces over 300 million markers per year over 139 colors and over 3 sizes. Due to the large number of variations, the production planning of the injection molding process is a complicated task.

Within the production process today there are two major restriction, color and tool change. These two restrictions requires the machine operator to preform a cleaning process for the color change and a fixture change for the changing of the tool. The different changes among products requires a different amount of time, so there is possible to plan the production in a optimal sequence to be more resource efficient. The reduction of machines will enhance the total tool and color changing time, this will be further explained in section 2. This reduction will put further put pressure on the company to be more resource efficient because a reduction in machines entails more downtime due to tool and color changes. According to the company a more structured production process is needed in order to meet the demand and obtain a high resource utilization.

Hence, together with the project owners the authors will be researching the possibilities of making the production planning process more efficient.

Figure (1.1) Product illustration

This thesis will be focused on the markers shown above in figure 1.1 and the production planning process of these markers. The markers represents the majority of the companies business and therefore the markers will be the focus of this thesis.

1.3 Aim and objective of the project

The primary focus of this thesis is to develop a production planning software based tool based on these restrictions to help Mark Bric become more resource efficient. Furthermore the aim of this thesis is to create structure within the production planning at the future injection molding plant.

The production process today has certain restrictions that can be leveraged in order to obtain the optimal production sequence to become more resource efficient. This will be further discussed in section 3.

1.4 Research questions

1. How can a more structured way of planning the production process be obtained at Mark Bric?
2. Is it suitable to use an optimized production planning model at Mark Bric?
3. What are the benefits by optimizing the production at Mark Bric?

1.5 Limitations

Due to the limits set by the company, the scope of this thesis had to be reduced, hence a few limitations in this thesis.

- The lack of historical data was an major disadvantage in improving the efficiency of the model. As the data provided does not consider any seasonal variation or peak demand period.

- The developed model process only a particular set of demand. It can be made more efficient, by improving the model to provide a plan for two or more set of demands at the same time.

2

Marc Bric's operations and Strategy Improvement Plan

In chapter 2.1 the overall operations within the production process of Mark Bric will be explained in order to get an understanding of the over all business. Later on in section 2.2 a brief explanation of the improvement plan of the future work is presented.

2.1 The operations of Marc Bric

In a nutshell, the larger majority of Mark Bric business is selling in-house molded plastic sizers and sale indicators called size markers, illustrated in image 1.1. Today the company produce over 300 million units per year and deliver to costumers all over the world. In the following section below the production process as well as the aim and overall processes of the company will be further elaborated

The production process of Mark Bric is currently located between two locations, stenkullen headquarters where the printing and packaging of the molded parts take place before the outbound delivery dispatch. The second location is Borås, where the injection molding process of the markers is carried out. At this molding plant, one machine operator is operating 8 machines producing the markers that will later be shipped to stenkullen for printing and packaging. This machine operator named Kajo receives the demand plan from the headquarters two times a week.

At Mark Bric, Karin is responsible for logistics and material planning works with the order intermediary. When a order is received given that there is a contract with that customer. After contract confirmation Karin checks the current stock if there is enough stock on hand to cover the re-order point (ROP), no parts is demanded from the molding plant. If the re-order point is reached then the order is sent to Stenkullen where the molding process is been done.

In the factory all the machines are connected to a individual flow of molted plastic of neutral color from the silo tank yet to be colorized, located outside of the molding facility. Beside each machine there is a connector that can connect a paste of chosen color which mixes the neutral plastic with the chosen color. This allows the machines to produce a chosen color separately. Each machine can therefore run an independent production of color. The individual molding plates that allow the machines to produce different sizes are currently 3 plats because of the 3 sizes. This information allows each machine to produce the sizes and colors of choice separately.

Before every color change the machinery have to be cleaned to get a good quality of coloring in each batch. The time it takes to clean the machine after each color change varies. In general, according to the machine operator Kajo, it is easier to clean the machine when going from a light to a darker shade due to that the darker shade fade out the lighter color and makes the light color less visible. The same methodology goes with the tool change in order to produce a different marker-size there is a molding tool changing time. The tool changing time is static.

During the interview with Kajo which will be further elaborated in section 4 these color changing times was mined and transformed into a color changing matrix. This was done in order to be able to optimize the production sequence.

The transportation truck with finished material go twice a week from Borås to Stenkullen in order to keep the Stenkullen printing process occupied. When the markers arrive at the Stenkullen plant they go through the printing process, where different prints are printed on the molded part. The print on each batch is selected by the customer. After the printing process the packaging process begins, involves packing the products in usually bags of 100 pieces each. These bags are then packaged in a cardboard box with 50 bags per box. Depending on the delivery service requested by the customer, it is then placed in pallets or individual boxes.

2.2 Improvement plan

This thesis was conducted in order to create more structure within the production planning in the molding process at Mark Bric. The structure will be created by designing and building an optimization tool that optimizes the production plan.

The main goal is to with the help of Python code, read the customer demand after Karin put in the revision demand, after the code will optimize the demand, plan the production sequence with the help of mathematical optimization. The finished planning process is outlined in section 2.1. Step 1 and 2 in the figure will be the same as the current process. Karin still have to check the availability for the demanded variants. The remaining steps in figure 2.1 will be done fully preformed in a python program with with integer programming solving for a optimal production plan. The Python application will take the demand list, optimize it, and then output the optimized list as a CSV file for the machine operators to retrieve the production plan from.

In order to solve this problem the authors have done a robust literature review further illustrated in section 4.1 about how to create more structure within an production process with optimization. This research allowed the authors to find a optimal way of solving the production planning problem at Mark Bric.

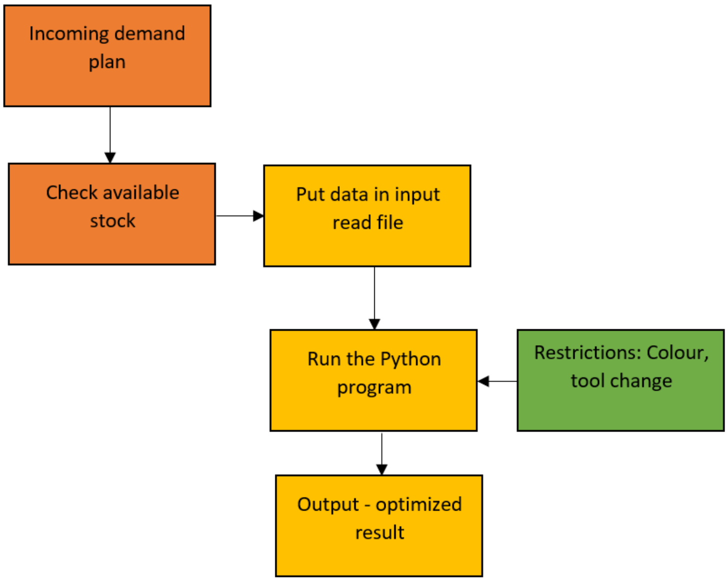


Figure (2.1) Flowchart of optimization of production planning

The previous section discussed how the production planning was done today at Mark Bric. Figure (2.1) illustrates how the authors of the thesis designed the improved flow of production process. The first two steps in the process: The incoming demand plan and the stock availability is similar in the current approach followed by Mark Bric and the one proposed in the thesis.

In the first step, the company tends to receive the demand from their customers, from then the stock availability of the recently received demand is checked by the material planner. If the stock availability is more than the re-order point, then product is neglected from being produced in the injection moulding process and it's being directly sent to printing where the appropriate print requested by the customer is been produced. If the product available in stock is less than the re-order point, then the quantity of material needs to replenish the re-order point and to meet the customer demand is summed up and the quantity to be manufactured is sent to the injection moulding operator to manufacture the requested quantity.

In this thesis technique, defining the input data file for the future algorithm is the third stage. Its main goal is to convert the demand file customers supplied into a Python-readable format for additional analysis. An optimization model with restrictions and an objective function will first be developed mathematically. By including an optimization solver, the mathematical model will be implemented in the Python programming language. which aids in producing the intended result of the production process, which will serve as a manual of instructions for the injection molding procedure.

The main purpose of the thesis is to optimize the production model in Python with the Pyomo extension which will be further explained in section 3.5 and 3.6.

3

Theory

This chapter is a theoretical introduction to the injection moulding process, basic understanding of production planning process and the complexities involved in planning the production process. In addition, this section discusses on how the production process can be enhanced through optimization using mathematical modelling in Python programming.

3.1 Injection molding process

Injection molding (IM) is a well-known manufacturing technology for producing various in large quantities and quickly. This method may yield a wide range of items with varying designs and applications [37]. The primary goal of the IM approach is to achieve a competitive advantage by manufacturing a high amount of items in a short period, hence increasing economies of scale. Due to the massive capacity to create a wide range of products, optimizing the injection molding process is critical to minimizing tool changing time and setting up a time during the transition from one product to another [37].

In order to optimize the production process, strategic planning is necessary to determine the best manufacturing sequencing and to create the product in a shorter amount of time. Previously, the machine operator made decisions based on his or her expertise [38]. Hence, it has become a drawback in terms of cost and time invested due to increased complexity when there is a high demand to be processed in the manufacturing [38].

However, due to the increased complexity caused by heavy fluctuation in demand, the firm must be dynamic to accommodate the various customer demands. It can be achieved by using optimization techniques, and most production industries have started to adopt these techniques to identify optimal planning [38].

3.2 Scheduling issues

In current technological breakthroughs, companies must be adaptive and dynamic to give consumers personalized services. To be dynamic, the company should have a sufficient amount of resources to survive in a fast-moving environment [36].

As the IM process can produce different variants of products, it is vital to optimizing the scheduling process. Since it involves set-up time and tool changing time, which often depend on sequencing of the products to be manufactured [36]. The more variation in the transition from one product to the next, the longer it generally takes to complete the set-up process for the following product in the sequence [36].

In general, a production process with variable change-over times creates the possibility of reducing the total production time through optimization. In order to attain the reduction, it is crucial to know the data regarding change over time to optimize accurately [36]. Once the customer places an order, each order should be analyzed to identify the best sequencing order depending on the type of product being produced in each machine. Through this ideology, the time taken to deliver the products to the customers and the total production time can be reduced [36].

3.3 Introduction to production planning

Production planning has grown in importance in recent years due to its critical ability to compete against rising competitors in the rapidly evolving technological world [21]. Production planning provides a precise strategy for deciding the quantity of the product and the appropriate time to manufacture a variety of products [22]. Through this methodology, the customers can procure better quality products with shorter delivery times [21].

The fundamental purpose of planning the production process is to respond quickly to a complex situation through customer demand [17]. Lean manufacturing is a planning process that focuses on lowering production costs by creating products in the right amount rather than making the products make-to-stock. However, focusing on maximizing the productivity of the resources and eliminating waste [23].

A hierarchical production planning method is primarily used to increase the efficacy of production planning choices by providing a framework for planning activities. Decisions taken at the top of the hierarchy function act as a restraint for persons at the bottom of the hierarchy throughout their decision-making process organizations [20]. The degree of planning may be separated into Strategic planning, tactical planning, and operational planning are the three functions in the hierarchical production planning system [20].

3.3.1 Operational planning

Elaborating the medium-term plan into a more specific form is known as short-term planning [17]. It is a method of streamlining the preliminary production schedule into a thorough planning strategy. The sequence in which the goods must be produced, the number of raw materials required, the delivery schedule, and the duration of the machine setup. So that a more accurate assessment may be made, hence, the planning is divided into short-term phases [17].

During short-term planning, the laborers are instructed on the number of orders that must be performed within the planning horizon. However, they are not provided the sequences in which it should be accomplished [17]. Hence, there should be a scheduling tool that can be used to plan the necessary sequence in which the production process must be followed [17]. As a result, schedule planning is strongly linked with the optimization model to fulfill the demand in an efficient production sequence [17].

3.4 Complexity in production planning

Proper forecasting for seasonal items, which have features of varying demand throughout the year, is one of the most significant issues encountered by manufacturing organizations that create a range of products. The need to keep things on hand until sold may drive up inventory expenses if sufficient planning is not done [28]. To avoid it, the business must decide between mass production's cost-effectiveness and customization's flexibility in response to customer demands. Flexibility may be attained by using mathematical algorithms to optimize the production process. As a result, inventory holding expenses are reduced, allowing for better planning for seasonal products. Additionally, this eliminates the complexity encountered during production planning and enables the business to adapt to consumer demand dynamically and flexibly [28].

3.5 Mathematical optimization

In recent years the importance of mathematical optimization has grown within the manufacturing industry due to its functionality in representing a real-life situation in a mathematical form. It provides a crucial tool for creating an optimization model for production and scheduling [29]. Furthermore, optimization has become an essential component for production managers, as it suggests the right timing to make decisions since it lowers total production costs by decreasing time lost in inefficient work processes and maximizes cost by leveraging existing resources. The significant difference between production planning and scheduling: Production is based on a tactical approach, and scheduling is based on an operational approach [29].

Statistics and predictive analysis have impressed the world with their unique capacity to influence decision-making. However, mathematics has also joined the race to guide decision making [30] since it can help in decision making by prioritizing the objective function and the constraints, which is a disadvantage through the application of statistics and predictive analysis [30].

The mathematical programming is set to define an optimized program from which the company gains an advantage by maximizing the profit or minimizing the production cost [31]. The following steps should be considered to achieve the perceived outcome: The first is to understand the bottlenecks in the current production process and select an issue where a satisfactory solution can be provided. The next is to create a mathematical model for the formulated problem, in which the different constraints involved in the issue and the decision variables are taken into consideration [31]. When the variables have been determined, the mathematical model identifies the most appropriate approach for organizing the manufacturing process [31].

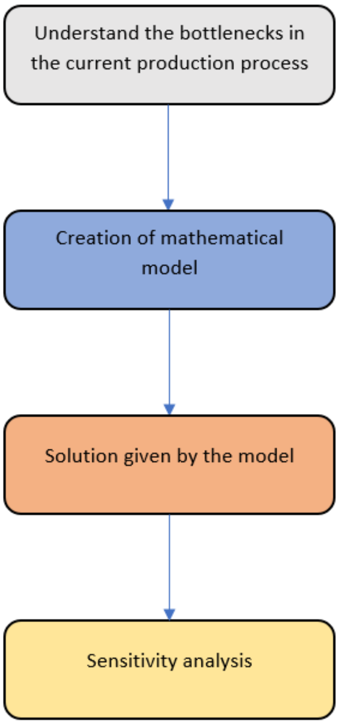


Figure (3.1) Methodology overview

Few organizations employ sophisticated and time-consuming worksheets to plan the production process due to the lack of various software-based optimization tools to support the production planning process. Since this method is not efficient, mathematical-based optimization can be employed to find an optimal plan for the production process [32].

3.5.1 Mixed integer programming

In a decision-making situation, model formulation is crucial because it communicates the core of the planning choice issue. The formulation is converting soft descriptions and numerical data into mathematical formulations. The vital link between decision variables, objectives, and resource limitations can be well-defined [33] through formulation.

The most commonly used approach to solve optimization or scheduling problems is mixed integer programming (MIP), which has been used in various fields such as sustainability, economics, and notably, in the area of manufacturing [33]. Hence, by utilizing the mathematical model in production planning, efficient usage of materials, labor, and equipment can be analyzed and achieved. So, planning is needed to run the production process more efficiently [33].

MIP is a mathematical model for solving many problems quite frequently in the area of production planning problems in the manufacturing industry[25]. Due to its ability to find an optimal solution to a highly complex and constrained program. A mixed integer is a combination of linear programming (LP) and integer programming (IP) [25].

3.6 Constraints, decision variable, objective

Essential elements forming mathematical modeling pillars are constraints, variables, and the objective function. These three will help convert a real-life application into a mathematical form [9].

3.6.1 Decision variables

Decisions variables are defined as the mathematical representation of the quantities set by the decision-makers, which they can control to alter the final result of the mathematical formulation. The decision variables can be presented as "Minimize f(x)," and X is the decision variable. If there are more decision variables, it becomes hard to solve the optimization problem [9].

$$f(x_1, x_2, x_n)$$

3.6.2 Objective function

The main functionality of an objective function is to either maximize the output or minimize resources used, and the decision on the objective function is usually decided by the stakeholders. [9].

The single objective function is the prime goal to be solved from the mathematical model, and it is the most frequent sort of objective function. Multi-objective objective functions are unusual since they comprise two or more objectives to be attained through the model, which makes the model difficult to execute. However, in today's world, most situations are in the form of a multi-objective function, making the model more complicated and harder to execute.

3.6.3 Constraints

Constraints are a set of functions that change the way the decision variables react. Inequalities and inequalities can represent constraints throughout the model formation process and can be physical or technical entities. The average demand that must be met is a typical example of setting a constraint. However, one of the essential things to remember when creating a restriction is that it should never accept a negative value and that it should be consistent [9].

3.7 Optimization software

3.7.1 Python

Python programming is widely utilized in today's technologically advanced environment. Python's capabilities of offering simple syntax, diverse libraries, and a sophisticated scripting language make programming applications simple to comprehend and maintain. Python has provided significant benefits to most industries, including automotive, data scientists, and software development, due to its simple syntax and user interface [14]. A Python library is a collection of related modules. It includes code bundles that may be utilized in many mobile/cellphone applications. For programmers, it simplifies and facilitates Python programming [27].

3.7.2 Pyomo

Pyomo, also called a Python optimization modeling object, is a software package with the Python programming language that can be used to solve various optimization problems. Python, an open-source programming language, is one of the key advantages of building mathematical models in this language. This allows users to benefit from open-source while simultaneously having more freedom to modify the program code [14].

Before the introduction of Pyomo in Python, algebraic modeling language (AML) was utilized to handle optimization issues in programming languages such as GAMS and AMPL [27], which was similar to the Python programming language in terms of features. However, one of Python's advantages was that it could be programmed in a low-level language and that it could optimize mathematical problems. As a result, the use of AMPL inspired the creation of Python.

4 Methodology

This thesis has been conducted with Marc Bric AB to improve the production planning processes and create a more structured production plan within the new production facility in Stenkullen, Sweden. As stated in the introduction, an investment in a new production line has been made, and along with this investment, the company demanded a need for a more structured working process.

The methodology section is divided into six parts: literature review, qualitative and quantitative data, Research workflow, data collection, interviews and illustration of the final model.

4.1 Literature review

The problem previously illustrated in several parts is a integer programming production planning optimization problem. A broad research study has been done within the production planning optimization scope. The research questions stated in section 1.3, if a more structured production planning can be obtained within Mark Bric.

In order to find relevant research Google scholar, Chalmers library and Science direct was used to explore previous research and relevant articles. The authors have found several interesting research papers regarding this case, these different cases will be evaluated below in order to find the best way to help Marc Bric create more structure within their production planning.

For decades many researchers and authors such as Stephen C.H. Leung (2006) [15] and others have raised concerns about production planning issues. Due to the increased complexity faced during high variation of customer demand. To provide a solution to cope with the fast-changing environment, these authors have developed various approaches such as stochastic optimization, mathematical-based optimization, and various more to plan the production process. These solutions have resulted in providing a dynamic approach to production planning to meet the customer demand.

The mathematical model of aggregate production planning (APP) optimization has been the subject of various prior studies. APP is the ultimate planning of available resources and labor to meet consumer demand. One of the research studies conducted by Sang-jin Nam and R. Logendran (1992) was based on developing an optimization programming to balance the forecasted sales demand and production capacity into manufacturing planning for the product demand, utilizing all of a company's human resources and equipment resources. The primary purpose of the production plan was to meet the forecasted demand for sales by incorporating the production, inventory, and workforce. APP is one of the most used methodologies by various researchers in designing production planning. Even though the APP model has been widely developed since 1950, it is not greatly accepted by the industries. Since it can only take into account one product family, one benefit of employing this strategy is that it aggregates or substitutes all of the product families.

The most common techniques used to solve production planning applications are (linear) mixed integer programming(MIP) and satisfiable modulo theory [13]. The production planning was used to assist Robert Bosch GmbH in determining the appropriate sequence of manufacturing line assignments. The production planning results give information about what sequence is the most optimal to be produced in due to certain restrictions [13]. In order to reduce the computational issue and to benefit from performance and flexibility, solvers were developed. In the Robert Bosch GmbH case, SAT solvers were used in deciding which set of constraints should be satisfied. However, SAT solver is not powerful enough to encode computational problems [13].

Through the broad literature research, the authors found that the Pyomo solver, which is used to solve mathematical problems in Python, is another popular solver used as SAT solver. These solvers are used in production planning applications. The main characteristic of pyomo solver is that the modeling objects are integrated into a full-featured high-level programming language with a numerious collection of supporting libraries [13]. This is frequently used to express models for complex real-world issues, including hundreds of constraints and variables, including linear, mixed-integer, non-linear, and non-linear mixed-integer models. Python consists of classes that define sets, parameters, and variables. This helps formulate algebraic expression for defining objectives and constraints [13].

Pyomo can describe mathematical concepts reasonably intuitively and concisely due to the advantage of Python's uncomplicated syntax [9]. Pyomo can also be utilized within an interactive Python shell, allowing users to interact with pyomo-based models directly. pyomo thus inherits the majority of the benefits of both Algebraic Modeling Languages (AML) interfaces, and modeling packages [9]. Hence, the proposed integer programming in this thesis is a version of the mathematical problem developed by Henri Lotze (2015) [13] to solve Mark Bric's production planning challenges. As a result, the model can assign the incoming orders automatically to each machine depending upon the sequence in which each constraint is to be met [9]. As an outcome, the total cleaning time in-between color changes can be reduced, minimizing the total production time. Furthermore, this method can create structure in the production planning by implementing the mathematical model in the Python interface.

As discussed by William E. Hart (2018) [14], using a pyomo solver would be beneficial since it improves the production planning of Mark Bric as it includes packages that can read the demand list and translate it to the production sequence. Hence, it will be a time-efficient model for the company. Also, it has the necessary components to formulate the optimization problems: Variables, objectives, and constraints. Pyomo can display a structured sequence in which the products should be produced in the injection molding process. The pyomo solver would be effective in the Mark Bric application since it possesses the required function and the ability to address the research objectives presented in section 1.3. Hence the proposed model in this thesis is the extended version of the pyomo solver developed by William E. Hart.

4.2 Qualitative and quantitative data

When designing an optimization model, qualitative data is crucial; mathematics is the language of a model of such sort. As stated above, qualitative data has been collected via interviews, factory visits, and meetings with the decision-makers. The primary understanding of the operations has been meditated through the meeting with the stakeholders. In the early stages of this project, the authors collected easily mined data like the number of machines, pace per machine, and the number of working hours for the staff. Later in this project, the authors and several employees worked together to collect the correct data to fulfill the goal of making the model via interviews and observations.

Several meetings with people in several positions at the company were conducted to get a deeper understanding of how the production planning process was outlined at Mark Bric. The two different operational locations and the connection between these two create difficulties for planning have been further researched to understand the production planning process and how to design the software to suit the company. Both quantitative and qualitative data have been meditated to the authors via two semi-conducted interviews and several more casual meetings with the top staff at the company.

Several conversations with employees occurred to make the back-end of the Python software statistics similar to the actual operations; quantitative data was created via an estimated model made with the machine operator Kajo. In order to create the tool and color change matrix, the backbone of the optimization program, the authors and machine operator had to work together to create a model that resembled an accurate operation.

4.3 Strategy used for data collection

Together with the company, the authors have evaluated different ways of creating a more efficient solution to plan the production at Mark Bric.

The authors have done a full literature review of previous research within the field of production planning optimization to consult the project owners on the most optimal way of solving the problem. A literature study on this subject matter has been done to understand how much the research question could be solved.

4.3.1 Research workflow

The research topic was not yet determined at the beginning of this project. The research topic was conducted with the company to improve the supply chain process.

The company's supply chain was examined to review possible improvements. Certain areas were found where improvements could be made; later on, the production planning area was chosen to create a more structural production planning process.

Hence a mutual desire for the project to succeed, the company has consulted the authors continuously during the project and provided information about the process with employee assistance, data access production visits, and progress consultation. Direct input on the final product has been crucial for project success. Continuous improvement and pivoting with users and owners of the final product have been conducted.

4.3.2 Data collection

The authors recognize continuous problem exploration in deep connection with employees and the manager of the project which has led to a continuous mining of qualitative and quantitative data. The majority of the data was collected during two interviews with employees, factory visits and introductions of the supply chain by the management and reviews of the supply chain.

A time study has been made to approximate the tool change and color change within the machinery to build to the back-end of the Python code. The time study approximation was done in order to get the constraints necessary for creation the optimization model. The data mining process was done together with a machine operator with over 20 years of operating the production line. This will be further discussed in the analysis section.

This process of working together with the employees of Mark Bric has allowed for continuous data validation. Hence, the employees and top management possesses a great deal of expertise in the manufacturing process

4.3.3 Interviews

According to Nigel Mathers (2000) [10] an interview is a data gathering technique that requires verbal exchange between two parts, often between the subject and the researcher. Bryman and Bell (2011) [11] explain 3 different types of interviews, structured, unstructured and semi-structured. The two interviews conducted during this thesis have been conducted in a semi-structured matter according the Bryman and bell(2011) [11]. The authors in this case had predetermined questions, but a certain openness to elaborate in order for the interview to lead into an open discussion which according to Roen, K. (2007) [12] can lead to more qualitative data and different perspectives.

As stated in the previous part, throughout this project two key interviews have been conducted. Several meetings with management have taken place but in a more civilised matter of information exchange. The majority of information conducted to curve this thesis into its final form was conducted during during two interviews. The first with the production planner Karin, the second one with the machine operator Kajo. The second meeting was a face to face semi-structured interview took place during a facility visit at the injection molding plant in Borås.

Another semi-structure interview was preformed with the machine operator with predetermined questions but with the main goal of understanding the operators views and learn the thought process of production planning. The interview did not have a structured path, Kajo led the interview into new areas of improvement. Kajo ended up contributing to the outline of the Python program. This contribution was crucial to the design of the mathematics behind the construction of the constraints in the model that will later be explained in section 4.5 in the results part of thesis.

Data	Company	Position	Duration(minutes)
2022-02-23	Mark Bric	Head of logistics	60
2022-03-08	Mark Bric	Machine operator/production planner	180

Table (4.1) Interviews

Product		Light	Light	Light	Dark	Dark	Dark
	Size	1	3	5	1	3	5
Light	1	25	145	145	15	135	135
Light	3	145	25	145	135	15	135
Light	5	145	145	15	135	135	15
Dark	1	35	155	155	25	145	145
Dark	3	155	35	155	145	25	145
Dark	5	155	155	35	145	145	25

Table (4.2) Change over time in minutes

Figure (4.1) above represent the color and tool chaining matrix. The numbers represent the change over time in minutes. The markers come in 3 different sizes and 4 different shade change combinations these are represent in the change over matrix above, changes from and to each shade and size to each shade and size. The possible changes are: light to dark, dark to light, light to light and dark to dark. That represents 36 different possible combinations of changes with respect to the 3 different sizes. The matrix above represent each of these individual changing times in minutes.

In order to create the tool and color change matrix as shown in the figure 4.1, several interviews and email conversations with the company occurred. With over 20 years of experience operating the machines Kajo has an excellent knowledge about the different color changing times. Furthermore, this knowledge has been put down on paper with was used to develop this matrix. The tool and color change matrix is the representation of the qualitative data collected. The numbers in the matrix have been approximated and evaluated by Kajo the machine operator.

4.4 Final model

The final model includes an objective function and eight restrictions for optimizing production planning. In which the (i) reflects the number of accessible machines (1,2,3,4,5). N represents the number of jobs (1,2,3,...417), 417 represents all possible variation combinations, all colours times all sizes. The dummy jobs, denoted as N_0, was added to the index N. The N_0 is defined as the number of jobs in the specific demand plus the dummy jobs. The model also includes parameters, Pr, which represent the production rate of each injection molding machine, and decision variables, X_{ijk}, Y_{ik}, and C_j.

4.4.1 Model assumptions

- **The production rate:** The production rate of each injection molding machine is assumed to be 250 pcs per minute irrespective of each size. This value is defined through the interview conducted with the machine operator (Kajo), as explained in the interview section.

- **The changeover time:** As the total possibilities of job available ranges from 1, 2, 3, ..., 139, which are of dark and light colors. Hence, the change over time from one job to another job is determined by the type of size and color of the successor and predecessor job.

- **The dummy job:** During the formulation of the mathematical model and execution of Python model, a dummy variable is introduced for testing and operational purpose of the program. Adding to it, as dataset may contain numerous types of values, including categorical values. So, in order to conveniently employ those category values in programming, we build dummy variables.

- **Machine starting time:** The program is designed to assign each job to each machine available in the sequence. Which results in neglecting any ideal time of machines.

4.4.2 Model implementation

This section describes how the final model (4.7) is used on Mark Bric production and the process of implementing in an optimization solution.

4.4.3 Description of parameters, sets and variable

Notation	Description
Pr	The number of products produced per minutes, which is 250 pcs per minute.
C_k	The total completion time for job k.
S_{ijk}	The change over time required to change from job j to job k.
P_{ik}	The processing time for the job k in machine i.
V	Biggest processing integer.
i	Sets of machine from (1...5).
N	Number of set of job available. The type of color (1,2,3...139).
N_+	It denotes the set of jobs m which will the start of the production.
N_-	It denotes the set of jobs m that which will be the end of the production.
N_{t+}	It is the set of n+m jobs that include our n jobs as well as m dummy jobs that start production.
N_{t-}	It is the set of n+m jobs that include our n jobs as well as m dummy jobs that end production.
N_t	It is the set of n+2m jobs that include our n jobs as well as 2m dummy jobs.

Table (4.3) Different parameters, sets and variables used in the model

4.4.4 Description of variables and indices

Notation	Type	Description
j	Indices	j denotes the index of the current job.
k	Indices	k denotes the index of the next set of jobs.
X_{ijk}	Binary variable	The value is 1, if the job j is scheduled before job k.
Y_{ik}	Binary variable	The value is 1, if the job k is scheduled in machine i.

Table (4.4) Different variables and indices used in the model

4.4.5 Objective function:

$$\min = C \tag{4.1}$$

Equation (4.1) - The objective function minimizes the end time of the last job.

4.4.6 Constraints:

$$\sum_{i \in M} Y_{ik} = 1, \quad k \in N_t \tag{4.2}$$

Equation (4.2) ensure that a job is assigned exactly to one machine. It can be interpreted as the summation of all the machines i, in which a the incoming job K must be assigned to each machine i. Hence Y_{ik} becomes 1, once the job K is assigned to machines i.

$$\sum_{i \in M} \sum_{j \in N_{t+j} \neq k} x_{ijk} = 1, \quad k \in N_{t-} \tag{4.3}$$

Equation (4.3) establish that every job has exactly one predecessor. This can be interpreted as the total summation of machines i and the job j with respect to the types of job N_0 and the current job j will not be equal to the next job $j \neq k$. Hence, x_{ijk} is 1 if the job j is scheduled before job k, otherwise 0

$$\sum_{i \in M} \sum_{k \in N_{t-,j \neq k}} x_{ijk} = 1, \quad j \in N_{t+} \tag{4.4}$$

Equation (4.4) establish that every job has exactly one successor. This constraint can be defined as summation of machines i and the next set of job K with respect to N_{t+}. Hence, x_{ijk} equals 1 if there is always a successor job j defined before the job K in every machines.

$$\sum_{j \in N_{t+}, j \neq k} x_{ijk} = y_{ik}, \quad k \in N_{t-}, i \in M \tag{4.5}$$

Equation (4.5) establish that every job has exactly one predecessor and both are assigned to the same machine. The constraint (4.5) can be described as the summation of j with respect to N_{t-} and $j \neq k$. Hence, $x_{ijk} = Y_{ik}$, when the previous job and the next job are aligned to the same machine.

$$\sum_{k \in N_{t-}, j \neq k} x_{ijk} = y_{ij}, \qquad j \in N_{t+}, i \in M \tag{4.6}$$

Equation (4.6) guarantee that every job has exactly one successor and both are assigned to the same machine. This constraint is same as the previous one (4.5), this is to make sure that the next job and the previously assigned job are aligned to the same machine.

$$Y_{ik} = 1, \quad k, i \in (N_{+}, M) \tag{4.7}$$

Equation (4.7) is defined to in-order to assign a assign a dummy job N_{+}in each machine during the start of the production sequence.

$$Y_{ik} = 1, \quad k, i \in (N_{-}, M) \tag{4.8}$$

Equation (4.8) has the similar function of the previous constraint, but this is defined to assign the dummy jobN_{-} in-order to end the production sequence.

$$C_k - C_j + V\,(1 - x_{ijk}) \geq s_{ijk} + p_{ik}, \quad j \in N_0, k \in N, j \neq k, i \in M \tag{4.9}$$

Equation (4.9) provide a right processing order, avoiding loops. Basically, they establish that if $x_{ijk} = 1$, then $C_k \geq C_j + s_{ijk} + p_{ik}$. If $x_{ijk} = 0$, the constraint becomes redundant. Basically, it forces that completion time for job k will be after completion time of j plus changeover time from j to k and processing time of job k.

$$C_k = 0, \quad k \in N_{+} \tag{4.10}$$

Equation (4.10), is formulated for the purpose of setting the time of the dummy job to be 0.

$$\geq C_k, \quad k \in N_t \tag{4.11}$$

Equation (4.11), is introduced to linearize the objective function and get max completion time from all jobs.

According to the set of constraints formulated in this thesis, every machine should have a predecessor and successor. As a result, the dummy variable is represented as a predecessor at the start of the production chain and as a successor at the conclusion. These functions are introduced to run the model effectively in the start and end. The time change from and to a dummy job is always zero. The limitations (4.7) and (4.8) set the time necessary to perform the fake task to zero. As a result, it is designed to ensure that the software flows efficiently.

4.5 Data

The data provided in thesis is of three parts - demand scenario, the change over time needed to change from one color and between sizes. Finally the production rate and the number of machines available. These data were defined by the Mark bric executives through the course of interviewing process.

- The change overtime: This data file includes the time needed during the transmission of molding process from one job of a color and a size to an another job of different or similar color and size.

- The production rate: The production rate of 250 pcs per minute were determined through the interview session with the logistics manager. Where the total products being manufactured per hour was known. Hence, the total number of produced per minute were scaled down to per minute, in-order to be convenient while using it in a Python program.

- Demand: The demand of a particular order was determined through the historical data of the company. This was specifically chosen since, it had a divers range of colors and sizes to be produced.

4.6 Software

The optimization model in this thesis was created using the pyomo solver within the Python interface. Through diverse research on what tool should be used, pyomo was suitable due to it's efficiency in creating the model which included variables, constraints, and the objective function. In addition, pyomo provides an interface to CBC optimization solvers.

Various libraries are used during the pre-processing and post-processing of data in Python. The essential libraries utilized in this thesis are pandas, NumPy, and plotly. Pandas assist in importing data from an external sheet or storing the model's output on an external sheet. Since the mathematical formulation is a crucial part of this thesis, Numpy is utilized, as it provides a wide range of mathematical operations. Finally, Plotly package has been utilized in Python to transform the results summary to a visual representation.

4.7 Model output

The model output is determined by the input of the excel file which provides the necessary data for an optimized production planning sequence. Python interface represents the output visually, in which the necessary information about the production planning will be provided. The type of information is of:

- **Machine number:** Once the customer's orders are processed by the Python program, it gives the relevant information on the sort of task to be started on which machine.

- **Production sequence of color and size:** The output, provides the best suited sequence of product to be produced in each machine.

- **Machining time:** Adding to it, output provides the machining start time and end-time of each job. Through which the time required for the set-up and the cleaning process of the upcoming job can be determined.

5 Results and Discussion

The thesis describes a method for optimizing the production planning process using mixed-integer programming. The model was established to improve the efficiency of the production planning by making the sequencing more efficient within production process. This was made with a designed solution using constraints and a objective function as mentioned in section (4.4.2).

Initially a strategic plan was formulated and illustrated in Figure (2.1). The groundwork for the development was planned out using this framework describing the roadmap. At first the authors studied the current process of the company. Through this knowledge the process could be improved. Through a broad literature research, the proposed optimization model was determined to be implemented in Python due to the perk of a large variety of supporting libraries as panda and pyomo.

The upcoming sections discuss the final result achieved through the Python model which will be evaluated through 16 different scenarios over 4 different demands, in which each demand is been evaluated with 4 scenarios: scenario 1 (4 machines), scenario 2 (5 machines), scenario 3 (6 machines), scenario 4 (7 machines). Adding to it, an individual production rate scenario will be illustrated which was build to validate the model further and for and it also illustrates the benefits of using the model. Also, it covers the key-finding which is distinguished in the analysis. The 4th machine scenario in each demand is going to be called scenario 1, 5th machine scenario in each demand will be called scenario 2, 6th machine scenario in each demand will be called scenario 3, 7th machine scenario in each demand will be called scenario 4

5.1 Scenarios overview

In order to validate the model, the upcoming subsections analyze and discuss the validation of the different demands regarding the developed model.

Demand 1 and 2	Demand 1 and 2 have the same tool color, size, quantity but inverse shade combination. Demand 1 has 75% light respectively 25% dark jobs and demand 2 has the inverse of demand 1. These two demands were chosen as it had characteristics that is apt for comparison among each other. The hypothesis of this comparison was that demand 2 should finish later than demand 1 due to the matrics in the change-over matrix due to the shade ratio. The original demand was a historic demand received by the company.
Demand 3 and 4	These two demands were chosen together with Mark Bric based on previous demand records. The orders have average characteristics in form of color, quantity, shade and size ratio. These demands were chosen to analyze the functionality of the developed model and be given to the machine operators if the final results are optimal in their view.
Individual production rates	This section was developed to validate the model further. In this model the production rate in each machine can be adjusted. This will help Mark Bric continue their planning if the production rate would be variable. The demand in this scenario is based on demand 2

Figure (5.1) Scenarios overview

In order to validate the model the result section will be analyzed upon 16 scenarios scenarios. Each demand scenarios is going to run in Python 4 times over a different number of machines, i.e each demand one time for 4 machines, 5 machines, 6 machines and 7 machines. These scenarios will later be compared and reviewed in the discussion. Figure 5.2 is illustrating the incentives of each demand. All of the demands in this section except for demand 3 has been produced at Mark Bric. Demand 3 is developed with the help of the stakeholders and it is the inverse of demand 1 in terms of shade ration, in order to compare the results with equal basics

To attain a better solution, the Python solver should be solving the program for a longer period of time. The program run-time for the 16 scenarios has been 10 minutes each. For further improvements, the results of the model can be improved via a longer run-time.

5.1.1 Demand 1

The table (5.1) describes the initial demand scenario, which have the attributes of 75% bright color jobs and 25% dark color jobs. One of the key reasons for selecting demand 1 was to assess the following production sequence from the model, to see if the outcomes had shorter production time and total change-over than demand 3 (5.1.13).

Color	Size	Quantity	Time	UoM	Shade
99	1	18500	74	Minutes	Light
45	3	75000	300	Minutes	Light
62	1	75000	300	Minutes	Light
71	5	85000	340	Minutes	Light
3	1	54500	218	Minutes	Dark
36	5	15000	60	Minutes	Light
7	3	55000	220	Minutes	Dark
117	1	75000	300	Minutes	Light
21	3	65000	260	Minutes	Light
23	5	17500	70	Minutes	Dark
27	1	15000	60	Minutes	Light
6	3	85000	340	Minutes	Light
120	1	65000	260	Minutes	Dark
100	1	18000	72	Minutes	Light
121	1	10000	40	Minutes	Dark
28	3	85500	342	Minutes	Light
63	5	62500	250	Minutes	Light
44	1	57000	228	Minutes	Light
90	1	37000	148	Minutes	Light
6	5	30000	120	Minutes	Light

Table (5.1) Demand data 1

The resulting production sequence shown in the figure (5.2) from the demand data validation implies when there is a change in tool size between the jobs the change-over time increases. For example, In scenarios 1, where 4 machines, the change-over between the job (117_1) and job (28_3) is longer than the other change-over time between the jobs. When machines are producing from producing light color jobs to dark color jobs and also a tool change involved, the time taken for the changeover process is 135 minutes (finish time of job (117_1): 11:45 am and the start of job (28_3): 14:10 pm, total 2 hours and 25 minutes). Hence, lot of time is being consumed between the two particular job. Which could potentially be reduced if there is a similar job with respective to tool size is aligned to the machine.

Adding to it, when the number of machines utilized increases, the overall production time of the demand decreases. In-comparison with the above 4 scenarios, the reduction of production time between the scenario 3 (6 machines) to scenario 4 (7

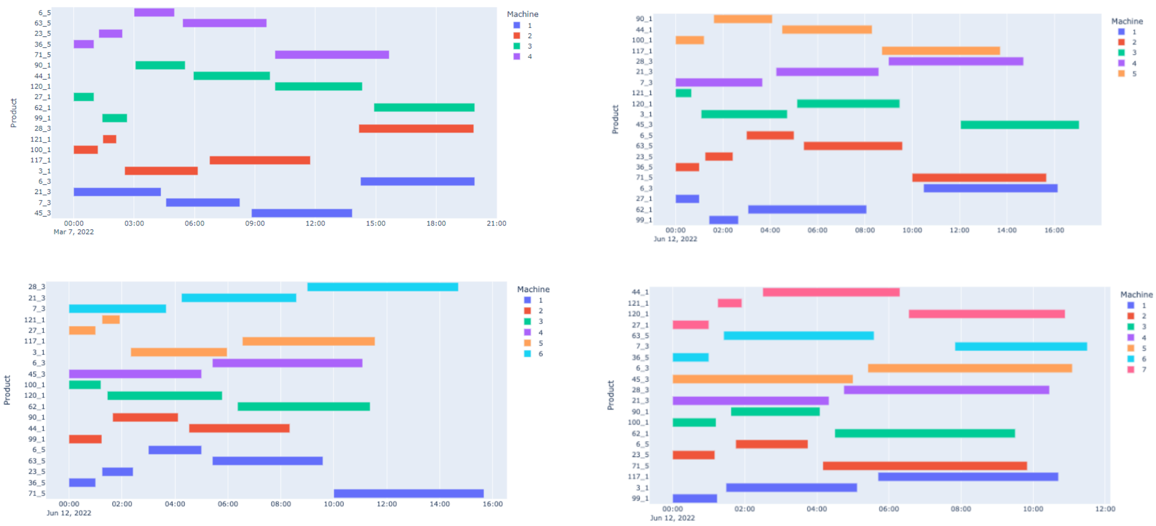


Figure (5.2) Production plan for demand 1

machines) has a greater effect in the production time. The end-production time of scenario 3 is 15:45 and scenario 4 is 11:35, however there is total reduction of 4 hours and 10 minutes. Compared to other the reduction between scenario 1 (4 machines) and 2 (5 machines) is 2 hours and 55 minutes and between scenario 2 and 3 (6 machines)is close to 1 hour and 15 minutes. Hence scenario 4 (7 machines) overall has an effective production plan.

Demand 1				
No.	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Machine 1	1120	774	840	592
Machine 2	1042	840	450	530
Machine 3	1000	818	632	520
Machine 4	840	822	640	602
Machine 5		822	618	640
Machine 6			822	530
Machine 7				588

Table (5.2) Total run time in minutes of demand 1

Total run time of machines in all the scenarios is show is illustrated in the table (5.2), from the analysis, it is seen that almost all the machines in scenario 1 is has equal machining time except machine 1. Hence, it experiences an inefficient time till the end production. During scenario 2, the job are well assigned in such a way that the production time is equally maintained throughout all machines. However, in scenario 3, there is a single machines out of 6 which has a less production time than the others. Hence, it is not quite efficient throughout the process. Finally, scenario 4 is similar to scenario 4 in a way that the job are assigned in an optimized way that provides an almost equal amount of production time to all the machines available.

Machine 4	Machine 5	Machine 6	Machine 7
15	25	35	15
35	25	25	35
25	25	15	15
25	35	25	25
25	25	35	135
25	25	25	25
25	25	15	25
15	155	35	25
35	15	25	25
15	35	25	35
25	25	15	25
35	25	35	15
145	25	25	35
15	25	25	
35	145		
25			
520	635	340	435

Table (5.3) Change-over time in minutes of demand 1

Table (5.3), explains the total change-over time of each scenario analyzed in demand 1. As the number of machines used in each scenario increases there is an decrease in the change-over time in all of the scenarios. However, during scenario 3, where 6 machines are utilized there is a major difference in the reduction of change-over time. However, in scenario 4, the change-over time is raised by 95 minutes; however, the decrease in production time is rather considerable in comparison to the preceding one. As a result, the increased change-over time during the use of seven machines will not result in inefficiency in the production sequence.

Demand 1 - Scenario 1	
No. Machine	End-production time of each machine
1 machine	19hr and 55 mins
2 machine	19hr and 52 mins
3 machine	19hr and 55 mins
4 machine	15hr and 40 mins
Total idle time of scenario 1	**4hr and 18 mins**

Demand 1 - Scenario 2	
No. Machine	End-production time of each machine
1 machine	16hr and 5 mins
2 machine	13hr and 45 mins
3 machine	17hr
4 machine	14hr and 45 mins
5 machines	13hr and 45 mins
Total idle time of scenario 2	**7hr and 40 mins**

Demand 1 - Scenario 3	
No. Machine	End-production time of each machine
1 machine	15hr and 45 mins
2 machine	8hr and 25 mins
3 machine	11hr and 35 mins
4 machine	11hr and 25 mins
5 machine	11hr and 45 mins
6 machine	14hr and 45 mins
Total idle time of scenario 3	**20hr and 50 mins**

Demand 1 - Scenario 4	
No. Machine	End-production time of each machine
1 machine	10hr and 35 mins
2 machine	9hr and 45 mins
3 machine	9hr and 35 mins
4 machine	10hr and 25 mins
5 machine	11hr
6 machine	11hr and 35 mins
7 machine	10hr and 45 mins
Total idle time of scenario 4	**6hr and 25 mins**

Figure (5.3) Idle time in minutes of demand 1

Figure (5.3) illustrates machine idle time for the demand 1 scenario analysis. The idle time between each scenario, according to the research, varies dramatically. The graph illustrates that scenario 2 (5 machines) has an increase of 202 minute over scenario 1 (4 machines). While scenario 3 (6 machines) has a significant increase in idle compared to scenario 2. However, in scenario 4 (7 machines) the idle time is reduced to 385 minutes. As a result, using six machines has an disadvantage in-regards to the idle time of few machines.

5.1.2 Demand 2

The demand 3 was developed with the guidance of the company stakeholders in-order to compare and analyze the production sequence of both the demands. Demand 2 consist of 75% of dark colors jobs and 25% of light colors jobs. Since, the majority of jobs are in dark colors the production time compared to demand 1 should be longer. Hence, the the selection of the demand 2 is important in the result analysis.

Color	Size	Quantity	Time	UoM	Shade
99	1	18500	74	Minutes	Dark
45	3	75000	300	Minutes	Dark
62	1	75000	300	Minutes	Dark
71	5	85000	340	Minutes	Dark
3	1	54500	218	Minutes	Light
36	5	15000	60	Minutes	Dark
7	3	55000	220	Minutes	Light
117	1	75000	300	Minutes	Dark
21	3	65000	260	Minutes	Dark
23	5	17500	70	Minutes	Light
27	1	15000	60	Minutes	Dark
6	3	85000	340	Minutes	Dark
120	1	65000	260	Minutes	Light
100	1	18000	72	Minutes	Dark
121	1	10000	40	Minutes	Light
28	3	85500	342	Minutes	Dark
63	5	62500	250	Minutes	Dark
44	1	57000	228	Minutes	Dark
90	1	37000	148	Minutes	Dark
6	5	30000	120	Minutes	Dark

Table (5.4) Demand sheet 2

The figures (5.4) illustrate that when the machine quantity grows, the change-over process time decreases dramatically. At the same time as the number of machines is increasing, the production time is decreasing as planned. Due to the fact that, demand 2 consist of more dark colors and less number of light jobs. Hence, changing from Dark to light in the production sequence could be a potential possibility for the

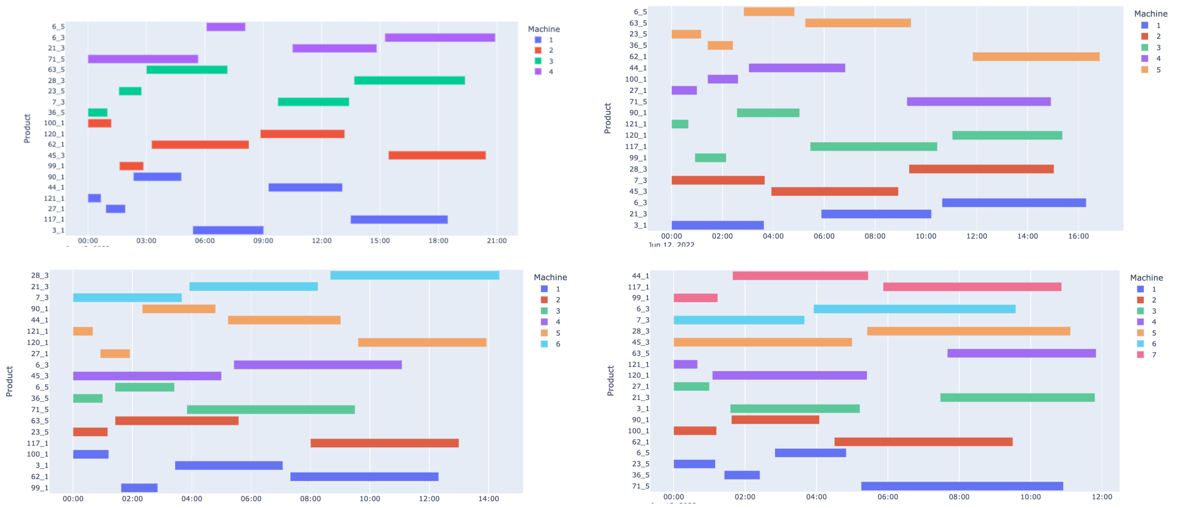


Figure (5.4) Production plan for demand 2

increased in change-over time compared to other scenarios which lead to an increase in the production time.

Machine 4	Machine 5	Machine 6	Machine 7
25	15	15	25
145	25	25	25
25	25	15	15
35	25	25	25
15	25	25	25
155	25	35	135
15	145	25	35
25	15	25	135
25	25	25	25
35	25	15	25
135	35	145	15
15	15	25	25
25	25	35	25
35	135	15	
15	25		
25			
750	585	450	535

Table (5.5) Change-over time in minutes for demand 2

According to Table (5.5), increase of utilizing 5 to 6 machines reduces change-over time much more than increasing from 4 to 5 machines or 6 to 7 machines. Hence, utilizing 6 machines have an greater advantage in sequence of the production demand.

Demand 2				
No.	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Machine 1	1125	800	822	602
Machine 2	1228	700	736	560
Machine 3	1185	822	640	642
Machine 4	1255	862	520	550
Machine 5		818	620	538
Machine 6			664	520
Machine 7				590

Table (5.6) Total run time in minutes for Demand 2

Table (5.6), explains the run-time of machines in all four scenarios. The run-time of demand 2 and demand 1 is similar since both have the equal quantity and tool size. As the number of machines increases the average run-time of machines decreases since jobs are evenly assigned to the resources available which reduced the work load evenly. Hence, the average run-time of each scenario is always less than the previous scenario.

Figure (5.5) explains the end-production time of demand 2 scenarios. From the analysis, it can be seen that the reduction in end-time is almost similar in all four

Demand 2 - Scenerios 1	
No. Machine	End-production time of each machine
1 machine	18hr and 35 mins
2 machine	20hr and 25 mins
3 machine	19hr and 45 mins
4 machine	20hr and 55 mins
Total idle time of scenerios 1	**4hr**

Demand 2 - Scenerios 2	
No. Machine	End-production time of each machine
1 machine	16hr and 25 mins
2 machine	15hr
3 machine	15hr and 35 mins
4 machine	14hr and 50mins
5 machines	17hr
Total idle time of scenerios 2	**6hr and 10 mins**

Demand 2 - Scenerios 3	
No. Machine	End-production time of each machine
1 machine	12hr and 25 mins
2 machine	13hr
3 machine	9hr and 35 mins
4 machine	11hr
5 machine	13hr and 50 mins
6 machine	14hr and 25 mins
Total idle time of scenerios 3	**12hr and 15 mins**

Demand 2 - Scenerios 4	
No. Machine	End-production time of each machine
1 machine	10hr and 45 mins
2 machine	9hr and 35 mins
3 machine	11hr and 45 mins
4 machine	11hr and 45 mins
5 machine	11hr and 45 mins
6 machine	9hr and 35 mins
7 machine	10hr and 45 mins
Total idle time of scenerios 4	**7hr and 5 mins**

Figure (5.5) Idle time in minutes of demand 2

scenarios. However, by comparing four scenarios, reduction from utilizing 4 machines to 5 machines has longer reduction in end-production time. The statement is also supported by the total change-over time, as the utilizing 5 machines instead of 4 has a greater reduction in the total change-over time as mentioned in table (5.11).

5.1.3 Demand 3

The table (5.7) shows the demand 3, this was chosen as it comprised of jobs with a total production time more than 100 minutes. As the overall goal of the thesis is to lower the overall production time of all machines, it will be appropriate to analyze the resulting outcome of the demand.

Color	Size	Quantity	Time	UoM	Shade
100	1	98500	394	Minutes	Dark
45	3	75000	300	Minutes	Dark
65	1	75000	300	Minutes	Dark
77	5	85000	340	Minutes	Light
3	1	54500	218	Minutes	Dark
36	5	55000	220	Minutes	Light
9	3	55000	220	Minutes	Dark
111	1	75000	300	Minutes	Light
21	3	65000	260	Minutes	Dark
23	5	67500	270	Minutes	Dark
27	1	75000	300	Minutes	Light
6	3	85000	340	Minutes	Light
120	1	65000	260	Minutes	Dark
101	1	88000	352	Minutes	Light
121	1	80000	320	Minutes	Dark
28	3	85500	342	Minutes	Dark
69	5	62500	250	Minutes	Dark
44	1	57000	228	Minutes	Light
98	1	37000	148	Minutes	Light
6	5	30000	120	Minutes	Light

Table (5.7) Demand 3

Figure (5.6) Production plan for demand 3

The obtained results figure (5.6) show that as the number of machines increases, the total production time declines, but the total change-over time is less in machine 4 when compared to the rest of the machines. Though the production time is decreased scenario 2 has lead to an increase in the time consumed during the change-over process. Which leads to an inefficiency during the reduction of production time.

Demand 3				
No.	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Machine 1	1370	1200	882	820
Machine 2	1200	952	966	830
Machine 3	1450	1160	942	642
Machine 4	1462	1120	930	838
Machine 5		1050	972	840
Machine 6			790	670
Machine 7				842

Table (5.8) Total run time in minutes of Demand 3

Table (5.8) explains the run-time of all the machines for the above demand scenario, from the values from different scenario it is clear that scenario 3(6 machines) and scenario 4(7 machines) is quite efficient compared to the other 2 scenarios. Since, during 3 and 4, the production time of all the machines is well aligned. However, during the other 2 scenarios the idle time of couple of machines are high which makes them inefficient during the process.

Machine 4	Machine 5	Machine 6	Machine 7
25	15	25	35
15	25	15	15
25	35	35	35
25	15	15	25
15	35	35	35
25	15	15	15
35	25	25	15
25	25	15	35
25	25	35	15
15	25	15	145
25	35	35	25
25	15	25	15
25	35	145	
35	135		
15			
340	460	435	410

Table (5.9) Change-over time in minutes of demand 3

The table (5.9) describes the change-over time obtained between each job in all 4 four scenarios of demand 3. According to scenario 1(4 machines) has lesser change-change over time than the other scenarios. Which lead to an less time consumed during the production process. One of the main reason to increase in scenario 2(5 machines) is that there is a tool change occurring in the machine 5. Hence, leads to a increase of 100 minutes in the change-over process.

Demand 3 - Scenario 1	
No. Machine	End-production time of each machine
1 machine	14hr
2 machine	14hr
3 machine	21hr and 45 mins
4 machine	12hr and 35 mins
Total idle time of scenario 1	**5hr and 40 mins**

Demand 3 - Scenario 2	
No. Machine	End-production time of each machine
1 machine	20hr and 45 mins
2 machine	20hr and 35 mins
3 machine	20hr and 42 mins
4 machine	17hr and 25 mins
5 machines	21hr and 35 mins
Total idle time of scenario 2	**6hr and 53 mins**

Demand 3 - Scenerio 3	
No. Machine	End-production time of each machine
1 machine	16hr
2 machine	15hr
3 machine	14hr and 55 mins
4 machine	14hr and 52 mins
5 machine	17hr and 5 mins
6 machine	15hr and 25 mins
Total idle time of scenario 3	**9hr and 13 mins**

Demand 3 - Scenario 4	
No. Machine	End-production time of each machine
1 machine	14hr and 45 mins
2 machine	13hr and 55 mins
3 machine	15hr
4 machine	15hr
5 machine	11hr
6 machine	14hr and 45 mins
7 machine	14hr and 45 mins
Total idle time of scenario 4	**5hr and 50 mins**

Figure (5.7) Idle time in minutes of demand 3

The figure (5.7), shows an overview of the idle time of each machines experiences until the end-production time. From the 4 scenarios tested for demand 3, there is been a huge variations of idle time through the different scenarios. From scenario 1 to 3, there is been gradual increase of idle time of machine. However, during scenario 4, where 7 machines are used, there is steep decrease in the idle time as it decreases to total of 350 minutes. Hence, it can said that using scenario 4 is quite efficient in regard to equal amount of utilization of all the resources available. Since, all the machines are occupied till the end-process.

5.1.4 Demand 4

One of the distinguishing features of demand 4 was that it included an equal number of tasks with production times less than 100 minutes and the other half with production time higher than 100 minutes. Also, demand 4 has a characteristics of demand 2, hence this would be helpful in comparison of the result between the two demands.

Color	Size	Quantity	Time	Shade
98	1	12500	50	Light
44	3	75000	300	Dark
63	1	75000	300	Light
89	5	85000	340	Light
2	1	14500	58	Dark
33	5	15000	60	Light
5	3	55000	220	Dark
119	1	75000	300	Light
49	3	65000	260	Dark
21	5	17500	70	Dark
23	1	15000	60	Light
12	3	85000	340	Light
123	1	65000	260	Dark
99	1	18000	72	Light
111	1	10000	42	Dark
26	3	15500	62	Dark
140	5	12500	50	Light
47	1	57000	228	Dark
91	1	16000	64	Light
67	5	30000	120	Light

Table (5.10) Demand sheet 4

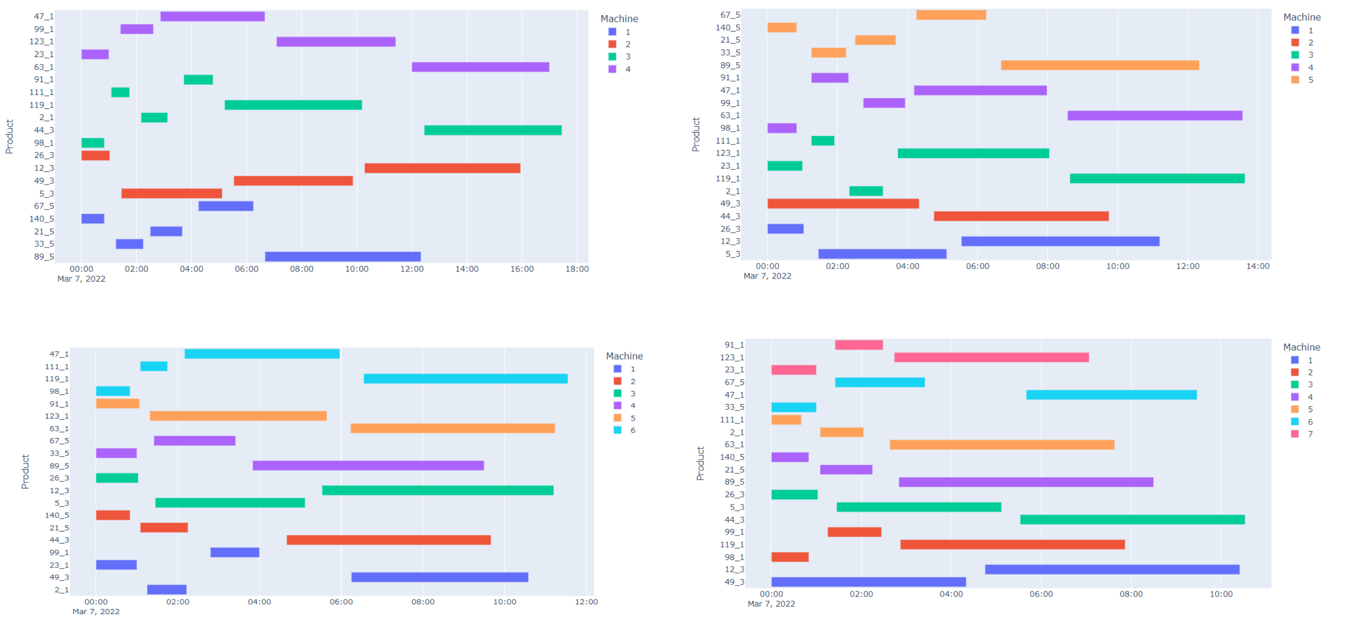


Figure (5.8) Production plan for demand 4

Figure (5.8) illustrates the production sequence of the different scenarios. From the figure as seen in the previous demands there is an decrease in the production time as there is an increase in the number of machines used. But, scenario 2 has a significant amount of reduction in the production time compared to scenario 1, i.e., End-time of scenario 1 - 17:35 and scenario 2 - 13:35, there is an decrease of 4 hours in the production time. The reduction of time in comparison with scenario 2 and 3, 3 and 4 on a average equals to 2 hours. Hence, according to the figure (5.8), utilizing 5 machines is quite efficient to this demand profile.

Machine 4	Machine 5	Machine 6	Machine 7
25	25	15	25
15	15	25	15
25	35	35	25
35	25	15	135
15	25	35	25
25	25	25	35
35	15	25	15
25	35	25	35
135	15	25	25
25	25	15	25
25	25	145	25
25	35	15	25
25	25	35	25
15	25	135	
35	25		
25			
485	375	570	435

Table (5.11) Change-over time in minutes of demand 4

The overall change-over time for all machines while changing from one job to another is described in table (5.11). The table illustrates that in general the number of machines used increases, so does the entire change-over time process. However, when 6 machines are used in this demand, there is a substantial increase in time. The increase is due to the procedure involves two tool changes that makes the overall change-over time increase. As previously discussed in the demand analysis, machine 5 is a better suited option for this demand.

Demand 4				
No.	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Machine 1	920	640	618	384
Machine 2	812	714	624	408
Machine 3	882	720	520	398
Machine 4	640	560	622	460
Machine 5		620	420	580
Machine 6			450	424
Machine 7				600

Table (5.12) Total run time in minutes of Demand 4

As discussed in the previous 3 demand scenarios, table (5.12), explains the run-time of machines in all the four scenarios. However, from the analysis, utilizing 5 machines has a positive impact on the average run-time of machines. As seen from the table average run-time of 4 machine in total is 813.5 mins and machine 5 is 650.2, hence a total reduction of approximately 200 minutes. But by comparing other scenarios there is only a reduction of 100 minutes. Which conveys that in the scenarios 3 and 4, few machines are underutilized. Hence, it is not profitable to invest in buying more machines.

Demand 4 - Scenario 1	
No. Machine	End-production time of each machine
1 machine	12hr and 25 mins
2 machine	15hr and 55 mins
3 machine	17hr and 25 mins
4 machine	17hr and 25 mins
Total idle time of scenario 1	**6hr and 55 mins**

Demand 4 - Scenario 2	
No. Machine	End-production time of each machine
1 machine	11hr and 25 mins
2 machine	9hr and 45 mins
3 machine	13hr and 45 mins
4 machine	13hr and 40 mins
5 machines	12hr and 25 mins
Total idle time of scenario 2	**7hr and 45 mins**

Demand 4 - Scenario 3	
No. Machine	End-production time of each machine
1 machine	10hr and 35 mins
2 machine	9hr and 45 mins
3 machine	11hr and 25 mins
4 machine	9hr and 35 mins
5 machine	11hr and 30 mins
6 machine	11hr and 45 mins
Total idle time of scenario 3	**5hr and 55 mins**

Demand 4 - Scenario 4	
No. Machine	End-production time of each machine
1 machine	10hr and 25 mins
2 machine	7hr and 50 mins
3 machine	10hr and 30 mins
4 machine	8hr and 25 mins
5 machine	7hr and 45 mins
6 machine	9hr and 45 mins
7 machine	7hr and 5 mins
Total idle time of scenario 4	**11hr and 40 mins**

Figure (5.9) Idle time in minutes of demand 4

The figure (5.9), explains the total idle time of each machine in all the 4 scenarios. During demand 4, there is lot of variations in the idle time of machines, since there is an sudden steep in the idle time in the scenario 3 and there is large increase in the idle time during scenario 4. As scenario 4, has a total idle time of 700 minutes. Hence, the resources are underutilized during the process. From the analysis, scenario 3 is efficient in comparison as all the machines in the scenario have more or less the same production end-time.

5.1.5 Comparison

In this section a comparison between the 4 different demands is being evaluated.

The first set of comparison can be made between demand 1 and demand 2 due to the characteristics of the job available in the demand profile. As discussed in the interview section, demand 1 has major quantity of light color jobs, hence it has an advantage over demand 2 in-terms of less total production time. Since, the change-over between the jobs in demand 1 is lesser, since switching from light to dark will not consume much time during the change-over process.

The next comparison can be analyzed between demand 3 and 4. In demand 4, since most of the job had less production time it is being influenced by frequent change-over process which tends to increase the total production time. However, in the case of demand 3 change-over time is comparatively less. It is due to cause that the two factors in the demand 3, shade and quantity were well interconnected in the demand, through which the production sequence had a uniform flow throughout the process.

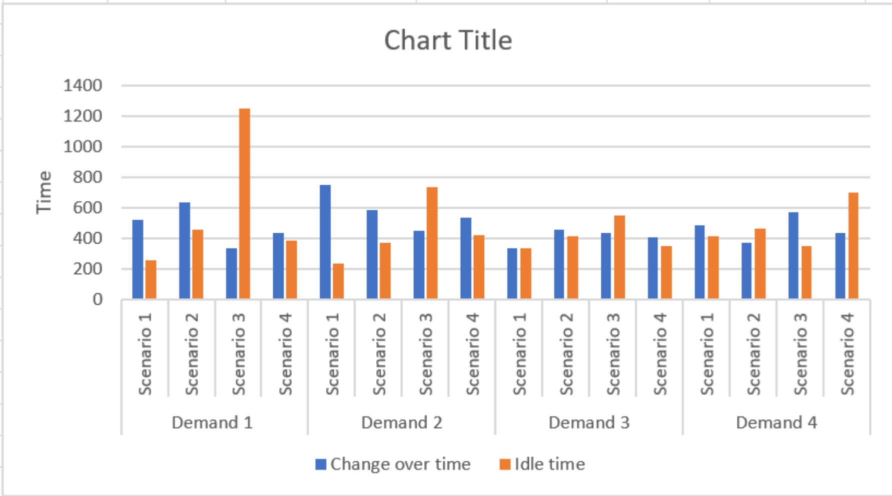


Figure (5.10) Result summary

Figure (5.10) compares change over time and idle time of the 16 different scenarios. These are compared and analyzed to determine the number of machines to be implemented to achieve production efficiency. From the comparison of idle time and change-over-time, it is seen that using 6 machines tends to decrease the total change-over time in all 4 demand profiles. This indicates that all the jobs are well aligned to the available 6 machines. However, as the change-over time decreases, the idle time of the machines tends to increase, as using 6 machines, few machines have their end-time much earlier than the other machines; this is seen in all the 4 demand profiles. Hence, from the figure, we can conclude that a higher production rate can be achieved by using 6 machines in the production process.

5.1.6 Individual production rates scenario

The initial step in modifying the model towards individual production rate is to define the different production rate of each machine in the demand file as show in the table(5.13), which is being inserted into the python algorithm. Mathematical equation (5.1), explains that by dividing the production quantity of each job (j) by the production rate of available machine (i), the processing time of the respective job can be determined. In the python algorithm, the processing time is added in the form as parameter: t_f_new(), the defined parameter is being included in constraint 7 as shown in the equation (5.2), through which the processing time of each job is determined to the each machine. A parameter for idle time ($idle_{i,j,k}$) is also included in constraint 7, which adds value by making it possible for the machines to idle and not strictly have an assigned job ready to be produced after color nor tool change. This makes it possible for the program to allocate better suited jobs more freely.

$$P_{ik} = j_q/pr_i \tag{5.1}$$

$$C_k >= C_j + idle_{i,j,k} \geq s_{ijk} + p_{ik} - V\left(1 - x_{ijk}\right), \quad j \in N_0, k \in N, j \neq k, i \in M \tag{5.2}$$

The variable production rate for each machine has been included as an extra feature to the present model. As all of the machines were recently bought, it is determined that all of the machines will have the same production time. However, fixed production rate was considered as the main focus of this thesis. But, when the machines used through a period of time, it is exposed to wear and tear on the machinery. This might slow down the manufacturing pace of the machinery. To cope with the dynamic environment the developed model and to monitor the performance of the production process continuously, improvement towards variable production rate was added to the developed algorithm.

The main step in varying the manufacturing process is to enter the production rate of each machine into the demand sheet. The python code has been modified to read the variable production rate of machine from the input excel file. Hence, time taken to produce each job varies according to the production rate of the machine it's been assigned to. Parameter for the variable production rate is introduced to the program through which is the key feature in determining the production time of each job.

Variable production rate for each machine defined in the pattern as shown in the table (5.13), through inserting the excel file into the program assigns production rate to the available resources according to the values defined in the sheet. Through introducing the variable production rate, the time specified in each demand profile is neutralized by allocating the variable production rate. Since, the time taken to produce the necessary quantity is defined at the rate in which each machine process

Demand 3	
No.	Scenario 4
Machine 1	250
Machine 2	250
Machine 3	500
Machine 4	500

Table (5.13) Variable production rate for demand 3

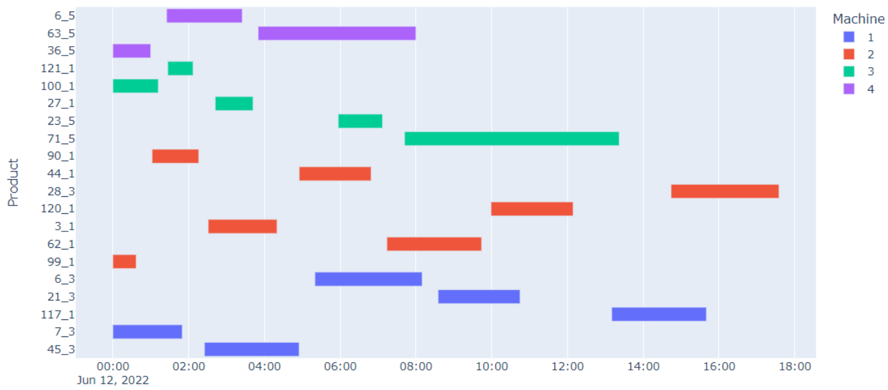


Figure (5.11) Change over time

per minute.

Figure (5.11) is an representation of demand 2; scenarios 1(4 machines) in which different production rate for the machines is defined as shown in the table (5.13). The production rate of machine 1 and 2 is set to 250 products per minute and machine 3 and 4 is set to 500. Hence, from the figure(5.10) it can be seen that the machine machine 3 and 4, is quite faster in processing the assigned task compared to 1 and 2 machines. However, if model is based on standard production rate, the time taken to finish job 99 and size 1 is 74 minutes. As the production rate is increased to 500 products per second the total time taken to finish the job is 37 minutes. Hence, as the machines are used farther down the line, the production rate should be regularly updated in order to synchronize the product delivery schedule to the consumers. Mark brick will be able to sustain using the optimization model due to variable manufacturing rate. Since it has the ability to adapt to current and future situations.

5.2 Key findings of the achieved result

The organization will receive a well-developed method for planning the production. The new planning process is more time and resource efficient than the present production process in several ways.

As outcome of the achieved result, the production process can now be planned through the visualization graph generated by Python code which illustrates an optimal sequence that shows in which order the products can efficiently be produced. The software is designed in such a manner that it tries to minimize the total production time.

When comparing idle time for the comparable demands 1 and 2, it's important to take end time into consideration. The end time of each scenario is what the total idle time of each scenario is calculated against. So the end time sets the gravity for the comparison. If we compare the comparable demands, 1 and 2 we see that for scenario 1 in demand 1 finishes an hour earlier that demand 2. On top of the earlier finish time, the idle time is also larger, which means that the mean end time is longer, which strengthens the hypothesis about that demand 1 finishes before demand 2. The same goes for each scenario in demand 3 and 4. This information can be validated by the table (5.9) and (5.11) where the total change over time is explained. These two tables support the analysis by showing that the total change over time in demand 1 is less that demand 2.

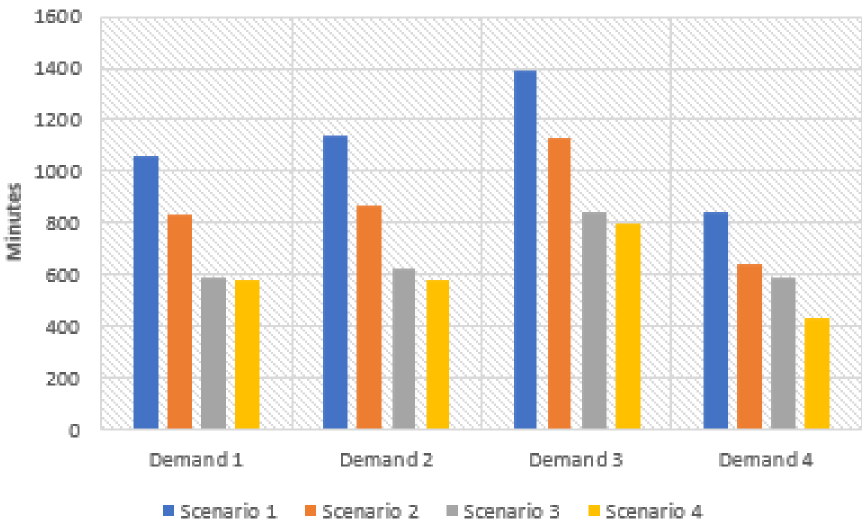


Figure (5.12) Average run-time of each machine in scenarios

Figure (5.12) illustrates the average machining time of each scenario, since the idle time can vary demand to demand due to it randomness characteristics it is neglected in finding the average run-time of machines in all the scenarios. However, following the analysis, utilizing 6 machines during demand 1 has a significant advantage in machining time. Since the average machining time is comparatively reduced in comparison to scenario 2 (5 machines) and the average run-time of scenario 4 is the same as scenario 3. Hence, utilizing 7 machines is quite insufficient, Since, the using 7 machines does not bring any advantage to the production process.

According to demand 2 and demand 3 in figure (6.1), it is the same pattern as demand 1, using 6 machines have a greater advantage in comparison to the other scenarios.

However, in demand 4, there is a drastic change in the reduction of run-time, while moving towards utilizing 5 machines instead of 4 machines. But, using 6 machines leads to inefficiency since it has the same output has the previous scenarios.

5.3 Future work

The fundamental purpose of the thesis was to optimize the production planning. This was achieved through the use of mathematical programming and the Python code. In the present suggested technique, the production planner is in charge of inserting the demand sheet into the Python program, which then returns an optimal plan as an output. Eventually, the developed process can be modified and improved through the use of machine learning algorithm. With a machine learning solution jobs can be added to the production plan in real time so there is a continuous flow of automation and not a sequence of a new production optimization once every week.

However, by developing a machine learning solution, a dedicated end-to-end flow from the initial step of receiving the demand to the final step of receiving the desired output which is finally sent to the injection moulding operator can be defined as a complete flow can potentially be developed in the future.

The vast amount of data accessible about customer preferences is a major technical advancement in today's environment. Which allows for the interpretation of consumer behavior and the forecasting of future production, which is only feasible owing to the preservation of client data.

5.4 Effect of limitation

- Since, the lack of historical data the model lacked in providing optimal solutions if there is an significant increase in the demand.

- The effect of less number of interview with the stakeholders had an effect on the data perception. The clear understanding of the data can have lead to an increased efficiency of the model.

6

Conclusions and Recommendation

6.1 Answering the research questions

1. **How can a more structured way of planning the production process be obtained at Mark Bric?**

It is feasible to accomplish a more ordered production planning process with the aid of Python programming and the techniques described in this thesis. A more organized procedure can be achieved by using Python, an optimization model framework built with combining constraints and an objective function, and illustrating output frameworks.

The optimized output is being strategically presented in a graph of the optimal production plan by integrating pyomo optimization modeling software with the Python application. An effective method for the optimization of production planning is clearly defined in the graph. The stakeholder can utilize this illustrative graph to improve production process.

2. **Is it suitable to use a optimized production planning model at Mark Bric?**

As Mark Bric has a lot of variations of markers in their portfolio, it is critical for the company to have a well-organized manufacturing process. The operations at mark bric today require a large portion of manual work. Through this thesis it has been shown that the majority of the manual work can be automated and made more efficient. As mark bric uses the change-over matrix described in figure (4.1), increase in the efficiency of the production process can be obtained by integrating the optimized build model and the change-over matrix. Hence, optimization is well suited method to be followed by mark bric in-order to improve the efficiency of the resource available.

3. **What are the benefits by optimizing the production at Mark Bric?**

Mark Bric can benefit from optimizing the use of available resources. For example, if the production sequence is not structured effectively and is planned based on the flexibility of the machine operator, it may result in more change-overs, increasing the total production time of a specific set of demand. This may cause delays in the subsequent stages of the production process, such as printing, packing, and ultimate delivery to clients.

Therefore by implementing optimization model build the company will benefit a more efficient production flow, this perk enhances certain other metrics as economical enhancement and employee satisfaction in the form of increased production in an efficient way.

6.2 Recommendations for Mark Bric

The recommendation for the company is to implement the outlined improved production process outlined in this thesis. This improvement will be a strategic move to make business more efficient.

- It's important to create standards around how often a new production plan should be sent to the injection molding operators in order to maximize output and minimize idle time.

- The future investment mark bric makes in boosting its resources to achieve quicker production is one of the important factors in developing an efficient manufacturing process. It is clear from the key findings that mark bric can meet its target when it uses 6 machines to produce size hangers. Given that scenario 3's average machining time—which involves six machines—has a bigger benefit in terms of overall production time.

- As the model is enhanced to support the variable production rate, it is important that the production rate of each machine is up to date in-order to achieve the desired production performance.

Bibliography

[1] Sizers.com. (2018, November 5). Hanger size markers, labels & amp; rack dividers. Mark Bric Sizers.

[2] Master thesis. Chalmers. (n.d.). Retrieved March 11, 2022, from https://www.chalmers.se/en/departments/tme/education/masters-thesis

[3] Lindhard, S. (2014). Understanding the effect of variation in a production system. Journal of Construction Engineering and Management, 140(11), 04014051. https://doi.org/10.1061/(asce)co.1943-7862.0000887

[4] G. B. Dantzig and M. N. Thapa, Linear Programming. New York: Springer, 2010.

[5] "What is manufacturing optimization (and why should you care)?,"

[6] S.-jin Nam and R. Logendran, "Aggregate production planning — a survey of models and methodologies," European Journal of Operational Research, vol. 61, no. 3, pp. 255–272, 1992.

[7] Y.I.-F.E.N.G.HUNG and G.I.A.J.I.N. CHENG, "Hybrid capacity modeling for alternative machine types in linear programming production planning," IIE Transactions, vol. 34, no. 2, pp. 157–165, 2002.

[8] R.O.B.E.R.T.C.LEACHMAN and T.A.L.I.F.CARMON, "On capacity modeling for production planning with alternative machine types," IIE Transactions, vol. 24, no. 4, pp. 62–72, 1992.

[9] Pyomo - optimization modeling in Python. William E. Hart, 2018.

[10] Mathers, N. J., Fox, N. J., & Hunn, A. (1998). Using interviews in a research project. NHS Executive, Trent.

[11] A. Bryman, "Mission accomplished?: Research methods in the first five years of leadership," Leadership, vol. 7, no. 1, pp. 73–83, 2011.

[12] Roen, K. (2007). Review: U. flick (2006).an introduction to qualitative research. Qualitative Research in Psychology, 4(3), 259–260. https://doi.org/10.1080/14780880701473623

[13] Lotze, H. (2018). Automated Optimization in Production PlanningHenri Lotze (thesis).

[14] Hart, W. E., Watson, J.-P., & Woodruff, D. L. (2011). Pyomo: Modeling and solving mathematical programs in Python. Mathematical Programming Computation, 3(3), 219–260. $https://doi.org/10.1007/s12532 - 011 - 0026 - 8$

[15] Leung, S. C., Wu, Y., & amp; Lai, K. K. (2006). A stochastic programming approach for multi-site aggregate production planning. Journal of the Operational Research Society, 57(2), 123–132. https://doi.org/10.1057/palgrave.jors.2601988

[16] ALVFORS, O. S. K. A. R., & BJÖRELIND, F. R. E. D. R. I. K. (2015). Optimization of Production Scheduling in the Dairy Industry (thesis).

[17] Paschos, V. T. (2010). Applications of combinatorial optimization. ISTE.

[18] Silver, E. A., Pyke, D. F., & Peterson, R. (2006). Inventory management and Production Planning and Scheduling. John Wiley & Sons.

[19] Barbarosoğlu, G. (1995). Hierarchical Production Planning. In: Burton, R.M., Obel, B. (eds) Design Models for Hierarchical Organizations. Springer, Boston, MA. $https://doi.org/10.1007/978-1-4615-2285-0_8$

[20] Hax, A. C., & Meal, H. C. (2012). Hierarchical integration of production planning and scheduling. The Roots of Logistics, 245–267. $https://doi.org/10.1007/978-3-642-27922-5_19$

[21] Zavvar Sabegh, M. H., Mirzazadeh, A., Maass, E. C., Ozturkoglu, Y., Mohammadi, M., & Moslemi, S. (2016). A mathematical model and optimization of total production cost and quality for a deteriorating production process. Cogent Mathematics, 3(1), 1264175. https://doi.org/10.1080/23311835.2016.1264175

[22] Bueno, A., Godinho Filho, M., & Frank, A. G. (2020). Smart production planning and control in the industry 4.0 context: A systematic literature review. Computers & Industrial Engineering, 149, 106774. $https://doi.org/10.1016/j.cie.2020.106774$

[23] Gupta, S., & Jain, S. K. (2013). A literature review of Lean Manufacturing. International Journal of Management Science and Engineering Management, 8(4), 241–249. $https://doi.org/10.1080/17509653.2013.825074$

[24] Y. V. Via, I. Y. Purbasari and R. W. Arianti, "Optimization of Production Profits Using The Firefly Algorithm," 2020 6th Information Technology International Seminar (ITIS), 2020, pp. 291-296, doi: 10.1109/ITIS50118.2020.9321073.

[25] Nehring, M., Topal, E., & Knights, P. (2010). Dynamic short term production scheduling and machine allocation in underground mining using mathematical programming. Mining Technology, 119(4), 212–220. $https://doi.org/10.1179/1743286310y.0000000001$

[26] Vanderbei, R. J. (2014). Linear Programming. International Series in Operations Research & Management Science. $https://doi.org/10.1007/978-1-4614-7630-6$

[27] Bynum, M. L., Hackebeil, G. A., Hart, W. E., Laird, C. D., Nicholson, B. L., Siirola, J. D., Watson, J.-P., & Woodruff, D. L. (2021). Pyomo – Optimization Modeling in Python. Springer

[28] Pochet, Yves & Wolsey, Laurence. (2006). Production Planning by Mixed Integer Programming. $10.1007/0-387-33477-7$.

[29] Ferschl, F. (1961). operations research. assembled by the Operations Research Center, M.I.T. Unternehmensforschung, 234–235. $https://doi.org/10.1007/978-3-662-39426-7_37$

[30] Prior, R. C., Slavens, R. L., Trimarco, J., Akgun, V., Feitzinger, E. G., & Hong, C.-F. (2004). Menlo Worldwide Forwarding optimizes its network routing. Interfaces, 34(1), 26–38. $https://doi.org/10.1287/inte.1030.0056$

[31] Szopa R, Marczyk B. Optimization of production problems using mathematical programming. Polish Journal of Management Studies. 2011;4:231-8.

[32] Shapiro, & F, Jeremy & Center, Massachusetts. (2022). Mathematical programming models and methods for production planning and scheduling.

[33] Shapiro, J. F. (1993). Chapter 8 mathematical programming models and methods for production planning and scheduling. Logistics of Production and Inventory, 371–443. $https://doi.org/10.1016/s0927-0507(05)80188-4$

[34] Moghadam, S. H., Mina, H., Iranmanesh, S. H., & Keyvandarian, A. (2015). A new mathematical model for single machine scheduling with learning effect: Continuous approach. International Journal of Mathematics in Operational Research, 7(3), 348. $https://doi.org/10.1504/ijmor.2015.069153$

[35] Ma, G.-feng, Yang, J.-hong, & Zhang, P.-chuan. (2012). Achieving optimal solution of linear programming based on Mobile Agent Technology. Physics Procedia, 24, 1364–1368. https://doi.org/10.1016/j.phpro.2012.02.203

[36] Klement, N., Abdeljaouad, M. A., Porto, L., & Silva, C. (2021). Lot-sizing and scheduling for the plastic injection molding industry—a hybrid optimization approach. Applied Sciences, 11(3), 1202. $https://doi.org/10.3390/app11031202$

[37] Kale, P. D., Darade, P. D., & Sahu, A. R. (2021). A literature review on injection moulding process based on runner system and process variables. IOP Conference Series: Materials Science and Engineering, 1017, 012031. $https://doi.org/10.1088/1757-899x/1017/1/012031$

[38] Bharti, Prem & Khan, Minhaj & Harbinder, Singh. (2010). Recent methods for optimization of plastic injection molding process—a retrospective and literature review. International Journal of Engineering Science and Technology.

A

Questionnaire

Interview With Karin
1. How does it work when you receive an order?
2. How is the information about the demand handed to Kajo the operator?
3. Who decides the production plan?
4. How is the production planning done today?
5. Do you have any stock? if so, how do you determine stock levels? re-order point?
6.Do you have any stock in house? .
7. How does the old plant differ from the new plant when it comes to production?
8. What will be the major benefit of having the production line and office building under the same roof?

Interview with Kajo
1. How do you plan the production today?
2. How long time does it take to make a color change?
4. Do you ever choose the wrong sequence? if you do how do you make up for the mistake?
5. Do you think the production planning will be done the same way at the new plant?
6. Do you think you run the production at the best pace today?
7. Would it benefit the production with a visual production plan of the most optimal sequence?
8. In what ways would a visual production plan benefit you as machine operator?

B

Appendix I - Python code

B.0.1 Production sequence optimization

!pip install pyomo

In [12]:
```
from pyomo import environ as pym
import pandas as pd
import numpy as np
import os
```

In [13]:
```
from pyomo.environ import *
from pyomo.core.base import Constraint, Var, NonNegativeReals, Objective, ConcreteModel, Param, \
    minimize, Binary, Set
```

In [14]:
```
filename = "Demand_1.xlsx"
```

In [15]:
```
demand = pd.read_excel(filename,sheet_name='demand',index_col=(0,1))
demand.head()
```

Out[15]:

color	size	quantity	time	deadline	shade
99	1	18500	74	AnyDate	Light
45	3	75000	300	AnyDate	Light
62	1	75000	300	AnyDate	Light
71	5	85000	340	AnyDate	Light
3	1	54500	218	AnyDate	Dark

In [16]:
```
demand.index
```

Out[16]:
```
MultiIndex([( 99, 1),
            ( 45, 3),
            ( 62, 1),
            ( 71, 5),
            (  3, 1),
            ( 36, 5),
            (  7, 3),
            (117, 1),
            ( 21, 3),
            ( 23, 5),
            ( 27, 1),
            (  6, 3),
            (120, 1),
            (100, 1),
            (121, 1),
            ( 28, 3),
            ( 63, 5),
            ( 44, 1),
            ( 90, 1),
            (  6, 5)],
           names=['color', 'size'])
```

In [17]:
```
parameter = pd.read_excel(filename,sheet_name='static',index_col=[0])
```

In [18]:
```
parameter
```

Out[18]:

parameter	value
production_rate	250
machines	4

In [19]:
```
change_over = pd.read_excel(filename, sheet_name='change_over',header=[0,1],
                            index_col=[0,1],)
change_over.head()
```

Out[19]:

	Size	Light			Dark		
		1	3	5	1	3	5
Light	1	25	145	145	15	135	135
	3	145	25	145	135	15	135
	5	145	145	25	135	135	15
Dark	1	35	155	155	25	145	145
	3	155	35	155	145	25	145

In [20]:
```
def get_shade(idx):
    return demand.loc[idx,'shade'] if idx in demand.index else 0
```

In [21]:
```
get_shade((99,1))
```

Out[21]: 'Light'

In [22]:
```

M = ConcreteModel()


def demand_f(M,color,size):
    return demand.loc[(color,size),'quantity']

def t_f(M,color,size):
    return demand.loc[(color,size),'time']


def change_over_f(M,color_1,size_1,color_2,size_2):

    # return 0 if size of color 0
    if color_1==0 or color_2==0 or size_1==0 or size_2==0:
        return 0
    shade_from = get_shade((color_1,size_1))
    shade_to = get_shade((color_2,size_2))
    return change_over.loc[(shade_from,size_1),(shade_to,size_2)]


m = m = parameter.loc['machines','value']

indices_with_0 = list(demand.index)
indices_with_0.insert(0,(0,0))

M.M = Set(initialize=RangeSet(m))
M.N = Set(initialize = list(demand.index))
M.N_0 = Set(initialize = indices_with_0 )

M.S = Set(initialize= list(set([idx[1] for idx in demand.index])))
```

```
M.demand = Param(M.N, initialize=demand_f)
M.s = Param(M.N_0, M.N_0, initialize=change_over_f)
M.t = Param(M.N, initialize=t_f)


M.x = Var(M.M, M.N_0, M.N_0, within=Binary)
M.y = Var(M.M, M.N, within=Binary)
M.z = Var(within=NonNegativeReals)
M.C = Var(M.N_0, within=NonNegativeReals)
```

In [23]:

```
def obj_expression(M):
    #return quicksum(M.C[i] for i in M.N_0)
    M.p = quicksum(M.C[i] for i in M.N_0)
    return M.p + M.z

M.obj = Objective(rule=obj_expression, sense=minimize)
```

In [24]:

```
M.one_prod = ConstraintList()



for j in M.N:
    M.one_prod.add(quicksum(M.y[i,j]  for i in M.M) ==1)


for k in M.N:
    M.one_prod.add(quicksum(M.x[i,j,k] for i in M.M for j in M.N_0 if j!=k)==1)


for j in M.N:
    M.one_prod.add(quicksum(M.x[i,j,k] for i in M.M for k in M.N_0 if j!=k)==1)

M.product_seq = ConstraintList()

for i in M.M:
    for k in M.N:
            M.product_seq.add(quicksum(M.x[i,j,k] for j in M.N_0 if j!=k) <= M.y[i,k])


for i in M.M:
    for j in M.N:
            M.product_seq.add(quicksum(M.x[i,j,k] for k in M.N_0 if j!=k) <= M.y[i,j])


for i in M.M:
    M.product_seq.add(quicksum(M.x[i,0,0,k] for k in M.N) == 1)
```

In [25]:

```
M.timing = ConstraintList()
for j in M.N_0:
    for k in M.N:
        for i in M.M:
            if j!=k:
                M.timing.add(M.C[k] >= M.C[j] + M.s[j,k] + M.t[k] -  2200*(1-M.x[i,j,k]))
```

```
M.demand = Param(M.N, initialize=demand_f)
M.s = Param(M.N_0, M.N_0, initialize=change_over_f)
M.t = Param(M.N, initialize=t_f)


M.x = Var(M.M, M.N_0, M.N_0, within=Binary)
M.y = Var(M.M, M.N, within=Binary)
M.z = Var(within=NonNegativeReals)
M.C = Var(M.N_0, within=NonNegativeReals)
```

In [23]:

```
def obj_expression(M):
    #return quicksum(M.C[i] for i in M.N_0)
    M.p = quicksum(M.C[i] for i in M.N_0)
    return M.p + M.z

M.obj = Objective(rule=obj_expression, sense=minimize)
```

In [24]:

```
M.one_prod = ConstraintList()



for j in M.N:
    M.one_prod.add(quicksum(M.y[i,j]  for i in M.M) ==1)


for k in M.N:
    M.one_prod.add(quicksum(M.x[i,j,k] for i in M.M for j in M.N_0 if j!=k)==1)


for j in M.N:
    M.one_prod.add(quicksum(M.x[i,j,k] for i in M.M for k in M.N_0 if j!=k)==1)

M.product_seq = ConstraintList()

for i in M.M:
    for k in M.N:
            M.product_seq.add(quicksum(M.x[i,j,k] for j in M.N_0 if j!=k) <= M.y[i,k])


for i in M.M:
    for j in M.N:
            M.product_seq.add(quicksum(M.x[i,j,k] for k in M.N_0 if j!=k) <= M.y[i,j])


for i in M.M:
    M.product_seq.add(quicksum(M.x[i,0,0,k] for k in M.N) == 1)
```

In [25]:

```
M.timing = ConstraintList()
for j in M.N_0:
    for k in M.N:
        for i in M.M:
            if j!=k:
                M.timing.add(M.C[k] >= M.C[j] + M.s[j,k] + M.t[k] -  2200*(1-M.x[i,j,k]))
```

In [26]:
```
M.span = ConstraintList()
for i in M.N:
    M.span.add(M.C[i] <=M.z)
```

In [27]:
```
M.span.add(M.C[(0,0)]==0)
for j in M.N:
    M.span.add(M.C[j] >= 0)
```

In [28]:
```
M.write("production_plan.lp")
solvername = 'cplex'
opt = SolverFactory(solvername)
opt.options['time'] = 30


os.environ['NEOS_EMAIL'] = 'fwfw@live.se'

model = M
solver_manager = pym.SolverManagerFactory('neos')
results=solver_manager.solve(model,opt=opt)

print(results)
```

```
WARNING: Loading a SolverResults object with a warning status into
    model.name="unknown";
      - termination condition: maxIterations
      - message from solver: CPLEX 20.1.0.0\x3a time limit with integer
        solution; objective 13191; 664496 MIP simplex iterations; 46100
        branch-and-bound nodes; absmipgap = 8638, relmipgap = 0.65484

Problem:
- Lower bound: -inf
  Upper bound: inf
  Number of objectives: 1
  Number of constraints: 1865
  Number of variables: 1782
  Sense: unknown
Solver:
- Status: warning
  Message: CPLEX 20.1.0.0\x3a time limit with integer solution; objective 13191; 664496 MIP simplex iterations; 46100
branch-and-bound nodes; absmipgap = 8638, relmipgap = 0.65484
  Termination condition: maxIterations
  Id: 422
Solution:
- number of solutions: 0
  number of solutions displayed: 0
```

In [29]:
```
result_data=[]
for i in M.M:
    for j in M.N:
        if M.y[i,j].value==1:
            result_data.append({"Machine":i, "Product":j,"End_time":round(M.C[j].value,1),"Production_Time":M.t[j],
result_summary = pd.DataFrame(result_data)
```

In [30]:
```
M.z.pprint()
```

```
z : Size=1, Index=None
    Key  : Lower : Value  : Upper : Fixed : Stale : Domain
    None :     0 : 1322.0 :  None : False : False : NonNegativeReals
```

In [31]:
```
result_summary
```

Out[31]:

	Machine	Product	End_time	Production_Time	Start_Time
0	1	(36, 5)	480.0	60	420.0
1	1	(28, 3)	967.0	342	625.0
2	1	(63, 5)	250.0	250	0.0
3	1	(6, 5)	395.0	120	275.0
4	2	(71, 5)	1168.0	340	828.0
5	2	(23, 5)	793.0	70	723.0
6	2	(120, 1)	578.0	260	318.0
7	2	(121, 1)	40.0	40	0.0
8	2	(44, 1)	303.0	228	75.0
9	3	(99, 1)	74.0	74	0.0
10	3	(62, 1)	997.0	300	697.0
11	3	(3, 1)	392.0	218	174.0
12	3	(117, 1)	1322.0	300	1022.0
13	3	(27, 1)	159.0	60	99.0
14	3	(100, 1)	499.0	72	427.0
15	3	(90, 1)	672.0	148	524.0
16	4	(45, 3)	830.0	300	530.0
17	4	(7, 3)	495.0	220	275.0
18	4	(21, 3)	260.0	260	0.0
19	4	(6, 3)	1195.0	340	855.0

In [32]:
```
result_summary.to_csv('result.csv')
```

In [33]:
```
result_summary.groupby('Machine').agg(list)
```

Out[33]:

	Product	End_time	Production_Time	Start_Time
Machine				
1	[(36, 5), (28, 3), (63, 5), (6, 5)]	[480.0, 967.0, 250.0, 395.0]	[60, 342, 250, 120]	[420.0, 625.0, 0.0, 275.0]
2	[(71, 5), (23, 5), (120, 1), (121, 1), (44, 1)]	[1168.0, 793.0, 578.0, 40.0, 303.0]	[340, 70, 260, 40, 228]	[828.0, 723.0, 318.0, 0.0, 75.0]
3	[(99, 1), (62, 1), (3, 1), (117, 1), (27, 1), ...	[74.0, 997.0, 392.0, 1322.0, 159.0, 499.0, 672.0]	[74, 300, 218, 300, 60, 72, 148]	[0.0, 697.0, 174.0, 1022.0, 99.0, 427.0, 524.0]
4	[(45, 3), (7, 3), (21, 3), (6, 3)]	[830.0, 495.0, 260.0, 1195.0]	[300, 220, 260, 340]	[530.0, 275.0, 0.0, 855.0]

In [34]:
```
df = result_summary.copy()
```

In [35]:
```
df
```

Out[35]:

	Machine	Product	End_time	Production_Time	Start_Time
0	1	(36, 5)	480.0	60	420.0
1	1	(28, 3)	967.0	342	625.0
2	1	(63, 5)	250.0	250	0.0
3	1	(6, 5)	395.0	120	275.0
4	2	(71, 5)	1168.0	340	828.0
5	2	(23, 5)	793.0	70	723.0
6	2	(120, 1)	578.0	260	318.0
7	2	(121, 1)	40.0	40	0.0
8	2	(44, 1)	303.0	228	75.0
9	3	(99, 1)	74.0	74	0.0
10	3	(62, 1)	997.0	300	697.0
11	3	(3, 1)	392.0	218	174.0
12	3	(117, 1)	1322.0	300	1022.0
13	3	(27, 1)	159.0	60	99.0
14	3	(100, 1)	499.0	72	427.0
15	3	(90, 1)	672.0	148	524.0
16	4	(45, 3)	830.0	300	530.0
17	4	(7, 3)	495.0	220	275.0
18	4	(21, 3)	260.0	260	0.0
19	4	(6, 3)	1195.0	340	855.0

click to expand output; double click to hide output

In [36]:
```
import datetime as dt
```

In [37]:
```
df.columns = ["Machine","Product","Finish","Production_Time","Start"]
df['Start']=dt.datetime(2022,6,12) + pd.TimedeltaIndex(df['Start'], unit='m')
df['Finish']=dt.datetime(2022,6,12) + pd.TimedeltaIndex(df['Finish'], unit='m')
df['Machine'] = df['Machine'].astype(str)

df['Product'] = df['Product'].apply(lambda o: str(o[0])+'_'+str(o[1]))
```

In [38]:
```
df
```

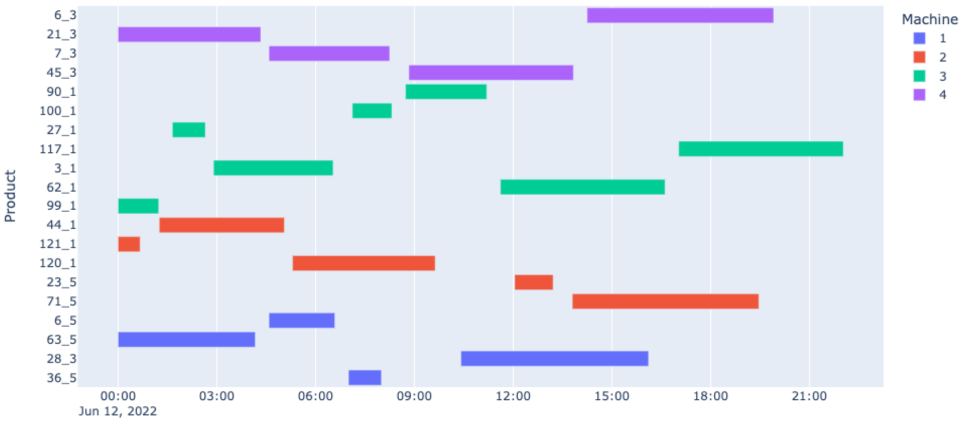
Out[38]:

	Machine	Product	Finish	Production_Time	Start
0	1	36_5	2022-06-12 08:00:00	60	2022-06-12 07:00:00
1	1	28_3	2022-06-12 16:07:00	342	2022-06-12 10:25:00
2	1	63_5	2022-06-12 04:10:00	250	2022-06-12 00:00:00
3	1	6_5	2022-06-12 06:35:00	120	2022-06-12 04:35:00
4	2	71_5	2022-06-12 19:28:00	340	2022-06-12 13:48:00
5	2	23_5	2022-06-12 13:13:00	70	2022-06-12 12:03:00
6	2	120_1	2022-06-12 09:38:00	260	2022-06-12 05:18:00
7	2	121_1	2022-06-12 00:40:00	40	2022-06-12 00:00:00
8	2	44_1	2022-06-12 05:03:00	228	2022-06-12 01:15:00
9	3	99_1	2022-06-12 01:14:00	74	2022-06-12 00:00:00
10	3	62_1	2022-06-12 16:37:00	300	2022-06-12 11:37:00
11	3	3_1	2022-06-12 06:32:00	218	2022-06-12 02:54:00
12	3	117_1	2022-06-12 22:02:00	300	2022-06-12 17:02:00
13	3	27_1	2022-06-12 02:39:00	60	2022-06-12 01:39:00
14	3	100_1	2022-06-12 08:19:00	72	2022-06-12 07:07:00
15	3	90_1	2022-06-12 11:12:00	148	2022-06-12 08:44:00
16	4	45_3	2022-06-12 13:50:00	300	2022-06-12 08:50:00
17	4	7_3	2022-06-12 08:15:00	220	2022-06-12 04:35:00
18	4	21_3	2022-06-12 04:20:00	260	2022-06-12 00:00:00
19	4	6_3	2022-06-12 19:55:00	340	2022-06-12 14:15:00

In [39]:

```
import plotly.express as px
import pandas as pd

fig = px.timeline(df, x_start="Start", x_end="Finish", y="Product", color="Machine")
fig.show()
```



B.0.2 Production sequence optimization - variable production rate

!pip install pyomo

In [1]:
```
from pyomo import environ as pym #Modelling framework
import pandas as pd
import numpy as np
import os
```

In [2]:
```
from pyomo.environ import *
from pyomo.core.base import Constraint, Var, NonNegativeReals, Objective, ConcreteModel, Param, \
    minimize, Binary, Set
```

In [3]:
```
filename = "Demand_2.xlsx"
```

In [4]:
```
demand = pd.read_excel(filename,sheet_name='demand',index_col=(0,1))
demand.head()
```

Out[4]:

color	size	quantity	time	deadline	shade
99	1	18500	74	AnyDate	Light
45	3	75000	300	AnyDate	Light
62	1	75000	300	AnyDate	Light
71	5	85000	340	AnyDate	Light
3	1	54500	218	AnyDate	Dark

In [5]:
```
demand
```

Out[5]:

color	size	quantity	time	deadline	shade
99	1	18500	74	AnyDate	Light
45	3	75000	300	AnyDate	Light
62	1	75000	300	AnyDate	Light
71	5	85000	340	AnyDate	Light
3	1	54500	218	AnyDate	Dark
36	5	15000	60	AnyDate	Light
7	3	55000	220	AnyDate	Dark
117	1	75000	300	AnyDate	Light
21	3	65000	260	AnyDate	Light
23	5	17500	70	AnyDate	Dark
27	1	15000	60	AnyDate	Light
6	3	85000	340	AnyDate	Light
120	1	65000	260	AnyDate	Dark
100	1	18000	72	AnyDate	Light
121	1	10000	40	AnyDate	Dark
28	3	85500	342	AnyDate	Light
63	5	62500	250	AnyDate	Light
44	1	57000	228	AnyDate	Light
90	1	37000	148	AnyDate	Light
6	5	30000	120	AnyDate	Light

In [6]:
```
list(demand.index)
```
Out[6]:
```
[(99, 1),
 (45, 3),
 (62, 1),
 (71, 5),
 (3, 1),
 (36, 5),
 (7, 3),
 (117, 1),
 (21, 3),
 (23, 5),
 (27, 1),
 (6, 3),
 (120, 1),
 (100, 1),
 (121, 1),
 (28, 3),
 (63, 5),
 (44, 1),
 (90, 1),
 (6, 5)]
```
In [7]:
```
#parameter = pd.read_excel(filename,sheet_name='static',index_col=[0])
```
In [8]:
```

# Show parameters
"""Keep in mind that the the number of machines defined in static sheet
must be the same as the number of machines defined in PROD_RATES sheet"""
#parameter
```
Out[8]:
```
'Keep in mind that the the number of machines defined in static sheet\nmust be the same as the number of machines defined in PR
OD_RATES sheet'
```
In [9]:
```
prod_rate_df=pd.read_excel(filename,sheet_name='PROD_RATES')
```
In [10]:
```
prod_rate_df
```
Out[10]:

	machine	production_rate
0	1	250
1	2	230
2	3	500
3	4	300
4	5	70
5	6	1000

In [11]:
```
len(prod_rate_df)
```
Out[11]:
```
6
```
In [12]:
```
prod_rate_dict=prod_rate_df.set_index(['machine'])['production_rate'].to_dict()
```
In [13]:
```
prod_rate_dict
```
Out[13]:
```
{1: 250, 2: 230, 3: 500, 4: 300, 5: 70, 6: 1000}
```
In [14]:
```
demand_df = pd.read_excel(filename,sheet_name='demand')
prod_quantity = demand_df.set_index(['color','size'])['quantity'].to_dict()
```
In [15]:
```
prod_quantity
```
Out[15]:
```
{(99, 1): 18500,
 (45, 3): 75000,
 (62, 1): 75000,
 (71, 5): 85000,
 (3, 1): 54500,
 (36, 5): 15000,
 (7, 3): 55000,
 (117, 1): 75000,
 (21, 3): 65000,
 (23, 5): 17500,
 (27, 1): 15000,
 (6, 3): 85000,
 (120, 1): 65000,
 (100, 1): 18000,
 (121, 1): 10000,
 (28, 3): 85500,
 (63, 5): 62500,
 (44, 1): 57000,
 (90, 1): 37000,
 (6, 5): 30000}
```
In [16]:
```
Process_time_dict={}
for job in prod_quantity:
    for m in prod_rate_dict:
        process_time= round(prod_quantity[job]/ prod_rate_dict[m],2)
        key=(m,)+ job
        Process_time_dict[key]=process_time
```

```
In [17]: Process_time_dict

 (3, 45, 3): 150.0,
 (4, 45, 3): 250.0,
 (5, 45, 3): 1071.43,
 (6, 45, 3): 75.0,
 (1, 62, 1): 300.0,
 (2, 62, 1): 326.09,
 (3, 62, 1): 150.0,
 (4, 62, 1): 250.0,
 (5, 62, 1): 1071.43,
 (6, 62, 1): 75.0,
 (1, 71, 5): 340.0,
 (2, 71, 5): 369.57,
 (3, 71, 5): 170.0,
 (4, 71, 5): 283.33,
 (5, 71, 5): 1214.29,
 (6, 71, 5): 85.0,
 (1, 3, 1): 218.0,
 (2, 3, 1): 236.96,
 (3, 3, 1): 109.0,
 (4, 3, 1): 181.67,
```

```
In [18]: Process_time_df = pd.Series(Process_time_dict).reset_index()
         Process_time_df.columns = ['machine', 'color', 'size','time']
         Process_time_df.set_index([ 'machine','color', 'size'], inplace=True)
```

```
In [19]: Process_time_df
```

Out[19]:

machine	color	size	time
1	99	1	74.00
2	99	1	80.43
3	99	1	37.00
4	99	1	61.67
5	99	1	264.29
...	...	...	...
2	6	5	130.43
3	6	5	60.00
4	6	5	100.00
5	6	5	428.57
6	6	5	30.00

120 rows × 1 columns

```
In [20]: change_over = pd.read_excel(filename, sheet_name='change_over',header=[0,1],
                                     index_col=[0,1],)
         change_over.head()
```

Out[20]:

	Size	Light			Dark		
		1	3	5	1	3	5
Light	1	25	145	145	15	135	135
	3	145	25	145	135	15	135
	5	145	145	25	135	135	15
	1	35	155	155	25	145	145

```
In [22]: def get_shade(idx):
             """
             This method takes a tuple (color,size) and returns the relevant shade
             from the "demand" table.
             If it does not find the index (color,size) in the list of indices,
             it returns 0
             """
             return demand.loc[idx,'shade'] if idx in demand.index else 0
```

```
In [23]: get_shade((99,1))
Out[23]: 'Light'
```

```
In [24]:
         M = ConcreteModel()
```

```
In [25]: def demand_f(M,color,size):
             """
             The function takes in color and size, and returns the "quantity"
             from the demand table
             """
             return demand.loc[(color,size),'quantity']
```

```
In [26]: def t_f_new(M,machine,color,size):

             return Process_time_df.loc[(machine,color,size),'time']
```

In [27]:
```
def change_over_f(M,color_1,size_1,color_2,size_2):

    # return 0 if size of color 0
    if color_1==0 or color_2==0 or size_1==0 or size_2==0:
        return 0
    shade_from = get_shade((color_1,size_1))
    shade_to = get_shade((color_2,size_2))
    return change_over.loc[(shade_from,size_1),(shade_to,size_2)]
```

In [28]:
```

m = len(prod_rate_df)

indices_with_0 = list(demand.index)
indices_with_0.insert(0,(0,0))
```

In [29]:
```

M.M = Set(initialize=RangeSet(m))
M.N = Set(initialize = list(demand.index))
M.N_0 = Set(initialize = indices_with_0 )



M.S = Set(initialize= list(set([idx[1] for idx in demand.index])))
```

In [30]:
```
# parameters
M.demand = Param(M.N, initialize=demand_f)
#M.pr = Param(M.N, initialize=pr)
M.s = Param(M.N_0, M.N_0, initialize=change_over_f)

M.t_new= Param( M.M, M.N ,initialize=t_f_new)
```

In [31]:
```
M.x = Var(M.M, M.N_0, M.N_0, within=Binary)
M.y = Var(M.M, M.N, within=Binary)
M.z = Var(within=NonNegativeReals)
M.C = Var(M.N_0, within=NonNegativeReals)
```

In [32]:
```
M.Idl  = Var(M.M, M.N_0, M.N_0, within=NonNegativeReals)


M.MIdl = Var(M.M , within=NonNegativeReals)
```

In [33]:
```



def obj_expression(M):

    M.p = quicksum(M.C[i] for i in M.N_0)
    return M.p + M.z

M.obj = Objective(rule=obj_expression, sense=minimize)
```

In [34]:
```
M.one_prod = ConstraintList()


for j in M.N:
    M.one_prod.add(quicksum(M.y[i,j]  for i in M.M) ==1)


for k in M.N:
    M.one_prod.add(quicksum(M.x[i,j,k] for i in M.M for j in M.N_0 if j!=k)==1)


for j in M.N:
    M.one_prod.add(quicksum(M.x[i,j,k] for i in M.M for k in M.N_0 if j!=k)==1)

M.product_seq = ConstraintList()

for i in M.M:
    for k in M.N:
            M.product_seq.add(quicksum(M.x[i,j,k] for j in M.N_0 if j!=k) <= M.y[i,k])


for i in M.M:
    for j in M.N:
            M.product_seq.add(quicksum(M.x[i,j,k] for k in M.N_0 if j!=k) <= M.y[i,j])


for i in M.M:
    M.product_seq.add(quicksum(M.x[i,0,0,k] for k in M.N) == 1)
```

In [35]:
```
M.timing = ConstraintList()


for j in M.N_0:
    for k in M.N:
        for i in M.M:
            if j!=k:
                M.timing.add(M.Idl[i,j,k]  <= 2200* M.x[i,j,k])


for k in M.N_0:
    M.timing.add(quicksum(M.Idl[i,0,0,k] for i in M.M) == 0) #

for j in M.N_0:
    M.timing.add(quicksum(M.Idl[i,j,0,0] for i in M.M) == 0) #


for i in M.M:
    M.timing.add(quicksum(M.Idl[i,j,k] for j in M.N for k in M.N if j!=k) == M.MIdl[i] )



for j in M.N_0:
    for k in M.N:
        for i in M.M:
            if j!=k:
                M.timing.add(M.C[k] >= M.C[j] + M.Idl[i,j,k]  + M.s[j,k] + M.t_new[i,k] -  2200*(1-M.x[i,j,k]))
```

In [36]:
```
M.span = ConstraintList()
for i in M.N:
    M.span.add(M.C[i] <=M.z)
```

In [37]:
```
M.span.add(M.C[(0,0)]==0)
for j in M.N:
    M.span.add(M.C[j] >= 0)
```

In [ ]:
```
M.write("production_plan.lp")
solvername = 'cplex'

opt = SolverFactory(solvername)
t= opt.options['time'] =1*60


os.environ['NEOS_EMAIL'] = 'fwfw@live.se'


model = M
solver_manager = pym.SolverManagerFactory('neos')
results=solver_manager.solve(model,opt=opt)

print(results)
```

In [ ]:
```
result_data=[]
for i in M.M:
    for j in M.N:
        for k in M.N_0:
            if j!=k and M.x[i,j,k].value==1:

                result_data.append({"Machine":i, "Product":j,"Start_Time":round(M.C[j].value-M.t_new[i,j],1),
                                    "Production_Time":M.t_new[i,j],"End_time":round(M.C[j].value,1),
                                    "Idle_Time": M.Idl[i,j,k].value,"Change_Over":M.s[j,k] })
result_summary = pd.DataFrame(result_data)
```

```
In [ ]: """
        # Collect results from the solver output, for us that is the M.y variable. Only collect it if it is 1, means the job was
        # set in that machine in that time


        #the code here is modfied from the old M.t[j] --> M.t_new[i,j]
        result_data=[]
        for i in M.M:
            for j in M.N:
                    if M.y[i,j].value==1:
                        result_data.append({"Machine":i, "Product":j,"End_time":round(M.C[j].value,1),
                                            "Production_Time":M.t_new[i,j],"Start_Time":round(M.C[j].value-M.t_new[i,j],1)})
        result_summary = pd.DataFrame(result_data)
        """
```

```
In [ ]: M.z.pprint()
```

```
In [ ]: result_summary
```

```
In [ ]: result_summary.to_csv('result_Oct8.csv')
```

```
In [ ]: result_summary.groupby('Machine').agg(list)
```

```
In [ ]: df = result_summary.copy()
```

```
In [ ]: df
```

```
In [ ]: import datetime as dt
```

```
In [ ]: df.columns = ["Machine","Product","Start","Production_Time","Finish","Idle_Time","Change_Over"]
        df['Start']=dt.datetime(2022,6,12) + pd.TimedeltaIndex(df['Start'], unit='m')
        df['Finish']=dt.datetime(2022,6,12) + pd.TimedeltaIndex(df['Finish'], unit='m')
        df['Machine'] = df['Machine'].astype(str)
        df['Product'] = df['Product'].apply(lambda o: str(o[0])+'_'+str(o[1]))
```

```
In [ ]: df
```

```
In [ ]: import plotly.express as px
        from matplotlib import pyplot as plt
        import pandas as pd


        fig = px.timeline(df, x_start="Start", x_end="Finish", y="Product", color="Machine")
        fig.show()
```

```
In [ ]: for i in M.M:
            print(i, M.MIdl[i].value)
```

```
In [ ]: for j in M.N_0:
            for k in M.N_0:
                for i in M.M:
                     print((i,j,k), M.Idl[i,j,k].value)
```

DEPARTMENT OF TECHNOLOGY MANAGEMENT AND ECONOMICS
DIVISION OF SERVICE MANAGEMENT AND LOGISTICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY