Ana Lúcia Fernandes da Silva Pinto

Modeling and Optimization of Production by Extrusion

Departamento de Matemática
Faculdade de Ciências da Universidade do Porto
November 2015

Ana Lúcia Fernandes da Silvia Pinto

Modeling and Optimization of Production by Extrusion

A thesis submitted to the Faculty of Sciences, University of Porto in partial fulfillment of the requirements for the degree of Master in Mathematical Engineering, supervised by Prof. Dr. João Pedro Pedroso and by Prof. Dr. João Nuno Tavares.

Departamento de Matemática
Faculdade de Ciências da Universidade do Porto
November 2015

Acknowledgments

First and foremost, I would like to thank Fersil - Freitas & Silva S.A. for giving me this opportunity in an industrial context, contributing to my personal and professional growth. A special thank to Eng. Jorge Leite for all the support and motivation over the project.
I am deeply grateful to Prof. Dr. João Pedro Pedroso and Prof. Dr. João Nuno Tavares, of Faculty of Sciences of University of Porto, for the constant guidance, dedication and inspiration.
Finally, thank to my family for the unconditional presence.

Abstract

The main goal of any company is to reach the maximum efficiency for keeping its competitiveness in the market. A good planning system has an important role to achieve this target.
Production planning consists of a detailed analysis of the availability of the resources needed to meet a set of requirements in a given time horizon, building the best plan for that horizon. The plan establishes the optimal production quantities, or *lot-sizes*, and timing. With the industrial growth, this task has become a more complex challenge, thus, the development of tools which help building of optimal production plans has been growing.
This dissertation is the result of an internship in a company producing pipes and accessories in plastic materials, Fersil - Freitas & Silva S.A.. The goal of this project is the development of a planning support tool. Since the production and sales processes of this company are slightly different from what has been studied in the literature, it was necessary to do a detailed analysis of all the constraints of this problem. From standard lot-sizing models, we designed some extensions of them, to reach a good adaptation to the proposed problem.
After modeling these processes, we proceeded with implementation and test of different instances, in order to better understand the solutions of the developed models and their adaptation to the company's problem.
Even though this being one of the most interesting problems of the industrial world, it is also very complex. More powerful software has been developed, but for high complexity situations, that characterize this problem, finding a feasible solution in a reasonable computational time can be a very hard task.

Resumo

O objetivo de qualquer empresa passa por atingir a máxima eficiência, mantendo a competitividade no mercado. O sistema de planeamento desempenha um importante papel para atingir esta meta, pelo que a sua melhoria é cada vez mais uma grande aposta.
O planeamento de produção consiste numa análise cuidada da disponibilidade dos recursos necessários para satisfazer um conjunto de exigências num certo horizonte temporal, estabelecendo o melhor plano de produção para esse período de tempo. O plano de produção estabelece as quantidades ótimas, ou *lot-sizes*, e frequência de produção. Com o crescimento industrial, essa tarefa tem vindo a tornar-se cada vez mais num desafio complexo, pelo que o desenvolvimento de ferramentas de apoio à construção de planos de produção ótimos tem vindo a crescer.
Esta dissertação é resultado de um projeto de estágio numa empresa de tubos e acessórios em materiais plásticos, Fersil - Freitas & Silva S.A.. O objetivo do projeto consiste no desenvolvimento de uma ferramenta de apoio ao planeamento. Uma vez que os processos de produção e vendas desta empresa seguem linhas um pouco diferentes das que têm vindo a ser estudadas na literatura, foi necessária uma análise cuidada e detalhada de todas as restrições deste problema. Partindo de modelos standard de lot-sizing, construímos extensões destes, de modo a atingir uma boa adaptação ao problema proposto. Após a modelação, prosseguimos com a implementação e teste de diferentes instâncias, de modo a perceber a solução dos modelos desenvolvidos e a sua adaptação à situação real da empresa.
Sendo dos mais interessantes no universo industrial, os modelos de lot-sizing são também bastante complexos. Embora o software que tem vindo a ser desenvolvido seja cada vez mais eficaz no tratamento destes problemas, em situações de elevada complexidade, que caracterizam este problema, encontrar uma solução viável em tempo computacional útil transforma-se numa tarefa difícil.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Production planning represents one of the most important tasks for a company. It consists of an analysis of the resources needed to produce a set of items, in such a way as to satisfy the customers in a profitable manner for the company.
Over the years the business world has been growing, what became the resources management a complex task. The growth of a company typically implies an increasing number of items to produce. At the same time, it is important not to neglect the best possible customer service, due to the competitive market. In this way, the need to develop computerized decision support tools emerged. The introduction of *materials requirement planning* (MRP) systems, with computational planning production systems responsible for the requirement analysis and subsequent production decisions, was a big step in this field. However, these systems were not sufficient to tackle many industrial complex problems. Therefore, it emerged the need to develop more sophisticated planning systems, which extended the production planning to decisions on different levels. These are called *advanced planning systems* (APS), and they include decisions in three time ranges: *strategic problems* involve strategic decisions on a long-term horizon; *tactical problems* focus on decisions in a medium-term planning horizon; *operational problems* involve the production planning in short-term horizons, usually of a few weeks. This project focuses on the short-term problem class, more specifically, on a *lot-sizing* problem. A *lot-size* is the quantity of an item to produce in one go, also called *batch size*. Based on an analysis of the requirements and the resources' availability, the goal of these models is to compute lot-sizes, respecting a set of constraints imposed by the problem and optimizing an objective function, usually a cost function.
The first lot-sizing models were linear programs, where the aim was to find optimal production amounts. The extension to mixed integer programs (MIP) allowed not only the modeling of the batch sizes but also the production timing, with the introduction of a binary variable called *setup variable*. A setup operation indicates a production occurrence, since it represents the preparation of a machine every time it is going to produce some item. It has a time and a cost associated. This cost is fixed per batch.
Ford Whitman Harris was a pioneer in this field, with his publication in 1913, entitled "How Many Parts to Make at Once" ([Harris, 1913]). The goal of the model presented in this paper is to find the economical production quantity, in such a way as to meet a static demand in an infinite time horizon and continuous time scale, balancing the average setup costs and the inventory costs. This publication originates an important model of inventory management, "economic order quantity" (EOQ).
The generalization of this model to a dynamic demand in a finite time horizon was realized by [Wagner and Whitin, 1958]. Since then, several developments and adaptations of this model have been done over the years; among them, the starting point model of this project: the capacitated lot-sizing problem (CLSP). Let us consider a set of items which can be produced in a machine. Analyzing the dynamic demand of these items in a given time horizon, as well as the machine availability in that horizon, the goal of this model is to compute the batch size and time of production for each product, minimizing a cost function and meeting a set of imposed constraints. Costs involved in this process

are inventory costs, proportional to the stock amounts, and setup costs.
Since the first approach to the CLSP, several other models have been developed, to keep up with companies' growth and the consequent complexity increase of their production planning. New concepts have been introduced, as production scheduling, parallel machines, multi-level production, backlogging and setup carryover; all of them will be described in detail along this dissertation.
Being mixed integer programs, these models are rather complex. The CLSP briefly presented above belongs to the class of NP-hard problems, hence all adaptations of it also belongs to this class. The difficulty in practice of a model depends on its features, as the planning horizon, number of levels, number of items and machines. With the growth of the companies, instances to solve are also higher, which increases even more the problems' difficulty. Computing an optimal solution for an industrial problem can be a challenge.
The usage of decision support tools has become an important factor in the companies' performance, in the sense that a good planning system can represent a significant improvement in their costs, as well as in their customer's satisfaction.

1.2 Aims of Investigation

Fersil - Freitas & Silva S.A. is a Portuguese company producing pipes and accessories in plastic materials. The planning system is done under the responsibility of the Planning and Logistic Department, and has become a complex process with the growth of the company. In this way, it emerged the need to develop a decision support mechanism, able to build an optimal production plan which establishes a trade-off between the firm's interests and the customers' satisfaction.
Fersil has a wide range of items, for which production occurs in a single- stage in parallel machines. The challenge of this project is to model their particular production and sales processes, which will be described over this dissertation.
Machine setup in this company is a complex process, which requires skilled labor. This is the reason why the so-called *setup team* is a scarce resource. This fact makes the machine's setup a very important task for the company. The number of setups per machine is limited, as well as the total time spent in this process each period.
The sales process is the most challenging part in Fersil's production planning. In contrast to what has been studied in the literature, this company works mostly for keeping appropriate inventory levels, under a *make-to-stock* production strategy. Customers' orders represent a small percentage of the sales. Moreover, when the company is not able to meet all the demand, some customers usually allow delays in the deliveries, the so-called *backlogging*. However, there are customers do not allow delays and thus a percentage of this unmet demand is totally lost. This means that, contrarily to what happens in problems described in the literature, the backlogged sales are not completely recovered in the next periods. Even though backlogging is allowed, customers' fixed orders do not allow it.
This is a brief summary of the features of the proposed problem. All the details will be presented along this dissertation. The goal of this project consists in developing a mathematical model of the Fersil's problem, building a decision support tool.

1.3 Dissertation Outline

This dissertation consists of six chapters, including this brief introduction.
Chapter 2 consists of a literature review, presenting some important lot-sizing models with discrete and finite time horizon and dynamic demand, which is the class problem under study. We present detailed features of some models, showing the evolution and adaptations done through the years. Over this chapter, we introduce some important notation and concepts in lot-sizing.
Chapter 3 concerns a description of the company, Fersil, where the internship that originated this work took place. We describe the main features of the company, paying attention to production and sales processes.
In Chapter 4, we propose two different models to Fersil's problem. After introducing notation, each

model is detailed described in, and the evolution between them is also explained.
In Chapter 5, we present the data and some computational results. As the only information provided by the company about the demand consists of a sales history, it was necessary to develop a demand generator, which is presented in the first section of this chapter. The second section consists of the results in terms of computational runtime and solution quality for four classes of different sizes instances. The third section is split into three parts. In the first one, we present two detailed examples to better understand the developed models. In the second part, we present a variant of the final model developed, which constitutes a suggestion for the company. We make a comparison between this variant and the original model. In the last part, we present a real Fersil's instance, and an analysis of the application of the final model to this dataset, to understand the performance of the model in the real world.
Finally, Chapter 6 consists of some final considerations and ideas for possible future work following this project.

Chapter 2

Literature Review

The planning department assumes an important role in all companies. It is responsible for the management of all resources involved in the production process. Based on an analysis of the demand, the inventory levels and the availability of the resources, the goal of this department is the computation of the needed quantities to produce of each item, when to produce and what resources to use in a given time horizon. This is called *production plan*, and it consists in a set of decisions that considers the company's constraints and, at the same time, the customer's satisfaction.

With the growth of companies and their production capacity, building an optimal production plan has become a complex task. It has been object of study over the years and it led to the development of new decision support tools to simplify this work. Thus appeared the *lot-sizing* models. Based on a dataset of items and resources and on all constraints of the problem, the goal of these models is computing a production plan, optimizing an objective function, usually, a cost function. These constraints and the objective function are linear expressions, and the variables involved in the model are continuous and binary, as we will see later in this dissertation. This kind of models is called *mixed integer programs* (MIP) and they can be solved by branch-and-bound and branch-and-cut algorithms, presented in many MIP solvers. Small instances can be easily solved by these mechanisms, but for bigger instances, the computational complexity increases. In these cases, it is necessary the development of more effective optimization algorithms.

Now, we present an introduction to the lot-sizing problems, with some important models in the literature and some terminology.

The goal of the lot-sizing models consists of determining production quantities, or *batch size*, and its timing, for a given *planning horizon*. The planning horizon can be *finite* or *infinite*. Infinite planning horizon is usually related to static demand, and finite to dynamic demand. Moreover, we can work in a *continuous* or *discrete* time scale. If the demand is known in advance, the model is called *deterministic*. Otherwise, it is *stochastic*. Since Fersil's model is deterministic, with discrete and dynamic demand and finite planning horizon, only this class of problems will be studied in this dissertation.

A lot-sizing model includes an objective function to optimize and a set of constraints. These rules define the production process, and they depend on the problem to be modeled. This is the reason why it is very important to have a thorough knowledge of the problem. For a modeling to be good, it has to achieve the best results for the company as well as for the customer.

The first lot-sizing models were classified as *single-item*, *single-level*. *Single-item* models deal only with one item. With the progress of companies, developed studies led to *multi-item* models where planning is thought for a set of products, in contrast with previous problems. In a *single-level* model, production occurs in only one level. That is, between the raw material and the final product, there is only one production stage. However, there are production processes where an item goes through several levels until the final product. This led to *multi-level* models.

Now, it is important to introduce two new concepts: *big-bucket* and *small-bucket* models. In small-bucket models, only one item can be produced each period. In contrast, in big-bucket models, several items can be produced in each period, with a machine's preparation between them, called *setup*. In the small-bucket class, several models were developed over the years. The *discrete lot-sizing and scheduling problem* (DLSP) is one example, where the "all-or-nothing" production policy is present. In each period, at most one item can be produced and, if it occurs, the produced amount has to be enough to occupy the total machine's capacity. [Jordan and Drexl, 1996] developed a model for one machine and sequence dependent setups. Later, [Bruggeman and Jahnke, 2000] developed a two-phase simulated annealing heuristic to solve the DLSP with batch availability.

The generalization of this model to another where this assumption does not exist is given by the *continuous setup lot-sizing problem* (CSLP). [Karmarkar et al., 1987] was one of the researchers on this problem, developing two algorithms to solve the uncapacitated and the capacitated cases.

Since CSLP does not use the "all-or-nothing" policy, sometimes there is unused capacity in the machine. The goal of the *proportional lot-sizing and scheduling problem* (PLSP), is to schedule a second item in a period where the machine's capacity is not totally full. [Drexl and Haase, 1995] proposed a solution method for this problem, followed by several other authors.

The problem under study belongs to the big-bucket class, thus, we will not go into details about small-bucket problems.

One of the simplest big-bucket lot-sizing model is the *single-item single-level uncapacitated lot-sizing problem*. Given the demand of the item for the planning horizon, the goal of this model is the determination of the batch sizes and production timing, to ensure the demand satisfaction and, at the same time, to minimize the total costs for the company. Machine's capacity is not considered, as well as the setup times.

However, the machine has a limited capacity in each period. The time spent with the whole production task each period cannot exceed the machine's capacity in that period. In addition, in many problems there is a set of items to produce, and not only one. In other words, usually we deal with multi-item problems. Hence appeared the *multi-item single-level capacitated lot-sizing problem* (CLSP), [Drexl, 1997]. This is an extension of the previous model, including the adaption for a set of items and a constraint to define the machine's capacity limit. The first CSLP did not consider setup times. In the mathematical formulation presented below, these times are considered.

Multi-item Single-level Capacitated Lot-sizing Problem

We start to define some notation, followed by the mathematical formulation of the problem. Let $\mathcal{T} = \{1, ..., T\}$ represents the planning horizon and $\mathcal{N}$ represents the set of items.

Indices

t - time period identifier;

p - item identifier;

Parameters

C_t - available time for production in period t;

D_{pt} - demand of item p in period t;

H_p - unit inventory cost for item p;

M - big number;

P_p - setup time for item p;

S_p - setup cost for item p;

U_p - unit production time for item p.

Decision Variables

x_{pt} - non-negative real variable that indicates the batch size of p in period t;

i_{pt} - non-negative real variable that indicates the amount of p in inventory at the end of period t;

y_{pt} - binary variable which takes value 1 if item p is produced in period t, and 0 otherwise.

$$\text{minimize} \quad \sum_{p\in\mathcal{N}}\sum_{t\in\mathcal{T}}(\mathrm{S}_p y_{pt} + \mathrm{H}_p i_{pt}) \tag{2.1a}$$

subject to:

$$i_{p,t-1} + x_{pt} = \mathrm{D}_{pt} + i_{pt}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T} \tag{2.1b}$$

$$x_{pt} \leq \mathrm{M} y_{pt}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T} \tag{2.1c}$$

$$\sum_{p\in\mathcal{N}} \mathrm{U}_p x_{pt} + \sum_{p\in\mathcal{N}} \mathrm{P}_p y_{pt} \leq \mathrm{C}_t, \qquad \forall t \in \mathcal{T} \tag{2.1d}$$

$$y_{pt} \in \{0, 1\}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T} \tag{2.1e}$$

$$x_{pt}, i_{pt} \geq 0, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T} \tag{2.1f}$$

$$i_{p0} = 0, \qquad \forall p \in \mathcal{N} \tag{2.1g}$$

The objective function, given by (2.1a), is a cost function, therefore the objective is to minimize it. Costs associated with the whole production are unit inventory costs, dependents of the quantities in stock, and the setup costs, fixed per batch. A machine setup is its preparation to produce an item in a given time period. This initial stage may consist in the cleaning of the machine, the change of some tools, among others. Every time an item is going to be produced, a setup for this must occur. The setup cost is fixed, that is, it does not depend on the batch size. It occurs every time there is production.
Constraint (2.1b) is called *flow balance*. This rule represents the obligation to satisfy the demand, establishing a flow conservation of inventory, demand and production between periods. This constraint can be better understood with the scheme 2.1.

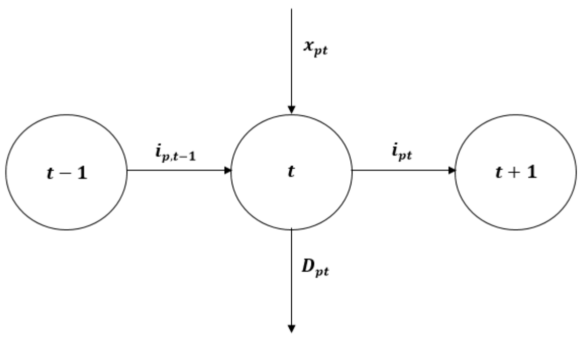


Figure 2.1: Flow balance of the multi-item single-level capacitated lot-sizing problem (CLSP)

Constraint (2.1c) enforces the setup variable to be 1 in period t for item p if there is production ($x_{pt} > 0$), and zero otherwise. It is important to note that M should be as small as possible to the model be tighter. Usually, it is used $\mathrm{M} = \frac{\mathrm{C}_t}{\mathrm{P}_p}$, which represents the maximum possible batch size of p in period t.
Constraint (2.1d) ensures that the total time spent with the production task in period t is less than or equal to the available time for production in that period. Constraints (2.1f) and (2.1e) define the decision variables' domain. It was shown by [Bitran and Yanasse, 1982] that the single-item CLSP is

NP-hard. All its extensions belong to this complexity problem's class.
The CLSP can be seen as an extension of the [Wagner and Whitin, 1958] formulation. Over the years, several extensions of this model have been developed, in such a way as to answer to different problems. It is important to note that, until now, all the problems work only with one machine, and it is responsible for the production of all items. However, in the majority of the companies and industries, there is a set of parallel machines, thus, it is possible to produce different items at the same time or to produce the same item in two machines simultaneously. The previous model was easily adaptable to the parallel machines' problem ([Quadt, 2004]).
In multi-level problems, the production process is more complex. While single-level problem has only one stage of production, in multi-level, there can be several stages between the raw material and the final product. In other words, an item can be an output of a stage and an input of another. This dependency between items is represented by a product structure, called *Bill of Materials* (BOM) ([Pochet and Wolsey, 2006]). There are different types of structures, but in all of them each intermediate item has at least one *predecessor* (input) and one *successor* (output). In these problems, the demand of an item is split into two values: *external demand*, which corresponds to customers' orders, and *internal demand*, which are the set of predecessors needed for the production. In the figure above, we can see three different product structures.

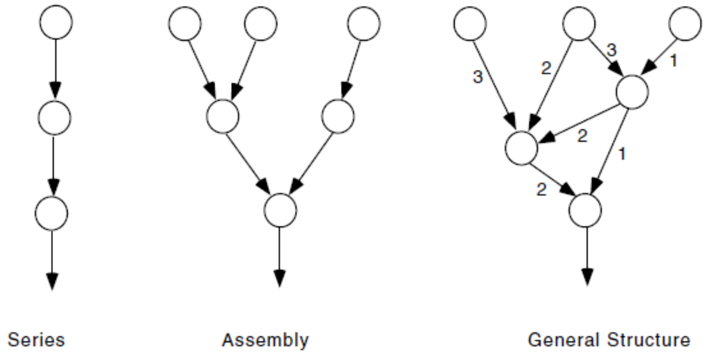


Figure 2.2: Different product structures

In all the presented product structures, each item, except the final product, goes through a set of production levels. In the first, *Series*, each product has exactly one predecessor and one successor. In the second case represented in the figure above, each intermediate item has exactly one successor, but can have more than one predecessor. In the last example, *General Structure*, each item can have more than one predecessor and successor.
A machine's setup has a cost and a time associated, as mentioned before. Problems analyzed until now are *sequence independent*, that is, setup costs and times do not depend on items sequence of production. However, there are cases where this sequence is important, because these costs and times may vary. In other words, if a company has two items, A and B, to be produced in a machine, and the setup from A to be B is different in terms of time or cost from B to A, then the sequence of production is important and the model must tackle this case. This kind of problems origins the extension of the previous one to cover this situation. Thus, the goal of a model to this problem is not only to optimize the batch size and production timing, but also to find the optimal production sequence. Parameters and variables associated with the setup process are indexed not only for the item that is going to be produced, but for the two products involved in the process. These models are called *sequence dependent*, and they are more difficult to solve than sequence independent problems. In Fersil's problem, changing the production sequence, the difference in setup costs and times are not significant, thus, it belongs to the sequence independent problem class.
Still on setup topic, models seen until now assume that, at the beginning of each time period, a machine's setup for the item that is going to be produced is always necessary, even if this was the last one in the previous period. This methodology causes unnecessary costs, since the state of the machine can be carried between periods. In other words, if an item is the last one to be produced in a given

time period and the first one in the next period in the same machine, it is necessary only one setup in the first period, and it is carried to the second. This process is called *setup carryover*. There were some extensions of the CLSP over the years to include the setup carryover, as the CLSP with linked lot sizes of [Haase, 1994] and later with the [Sox and Gao, 1999] and the [Suerie and Stadler, 2003] approaches.
Even though this problem belongs to the sequence independent class, it includes a partial scheduling. When an item is going to be produced in two consecutive periods, the model can choose to do a setup carryover. If it occurs, we know that this item is the last one to be produced in the previous period and the first one in the next. This is the only situation where we can see a scheduling in this model. This is the reason why we call it "partial scheduling".
Until now, it is possible to see that all the demand has to be satisfied in each period. Nonetheless, there are cases where available time in a certain time period is lower than the necessary to produce the demand in that period. In this situation, it is not possible to find a solution to the problem. To cover this situation, a new model has been presented, where if in a given period the available capacity is not enough, then the exceeded quantity can be produced in next periods, with an associated cost for unit delayed. This is called *backlogging*
Even though there are several approaches of the CLSP on literature, problems with backlogging and setup carryover are scarcer. In this dissertation, due to the particular production and sales processes of the company, some formulations of the literature will be extended for this problem. In Chapter 4, it will be possible to see these adaptations.
Now, we will present a mathematical formulation for *multi-item single-level capacitated lot-sizing problem with backlogging, setup carryover and parallel machines*, based on [Caserta and Voß, 2013].

Multi-item Single-level CLSP with Backlogging, Setup Carryover and Parallel Machines

We start to define some notation, followed by the mathematical formulation of the problem. Let $\mathcal{T} = \{1, ..., T\}$ represents the planning horizon, $\mathcal{N}$ represents the set of items and $\mathcal{M}$ represents the set of machines.

Indices

t - time period identifier;

p - item identifier;

m - machine identifier;

Parameters

B_p - unit backlog cost for item p;

C_{mt} - available time for production of machine m in period t;

D_{pt} - demand of item p in period t;

H_p - unit inventory cost for item p;

M - big number;

P_{pm} - setup time for item p in machine m;

S_{pm} - setup cost for item p in machine m;

U_{pm} - unit production time for item p in machine m.

Decision Variables

x_{pmt} - non-negative real variable that indicates the batch size of p in period t in machine m;

i_{pt} - non-negative real variable that indicates the amount of p in inventory at the end of period t;

b_{pt} - non-negative real variable that indicates the unmet amount of p in period t;

y_{pmt} - binary variable which takes value 1 if machine m produces item p in period t, and 0 otherwise;

z_{pmt} - binary variable which takes value 1 if machine m is prepared to produce p at the beginning of period t, and 0 otherwise.

$$\text{minimize} \quad \sum_{p\in\mathcal{N}}\sum_{m\in\mathcal{M}}\sum_{t\in\mathcal{T}} \mathrm{S}_{pm}(y_{pmt} - z_{pmt}) + \sum_{p\in\mathcal{N}}\sum_{t\in\mathcal{T}}(\mathrm{H}_p i_{pt} + \mathrm{B}_p b_{pt}) \tag{2.2a}$$

subject to:

$$i_{p,t-1} + \sum_{m\in\mathcal{M}} x_{pmt} + b_{pt} = \mathrm{D}_{pt} + i_{pt} + b_{p,t-1}, \quad \forall p \in \mathcal{N}, t \in \mathcal{T} \tag{2.2b}$$

$$x_{pmt} \leq \mathrm{M} y_{pmt}, \quad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T} \tag{2.2c}$$

$$\sum_{p\in\mathcal{N}} \mathrm{U}_{pm} x_{pmt} + \sum_{p\in\mathcal{N}} \mathrm{P}_{pm}(y_{pmt} - z_{pmt}) \leq \mathrm{C}_{mt}, \quad \forall m \in \mathcal{M}, t \in \mathcal{T} \tag{2.2d}$$

$$\sum_{p\in\mathcal{N}} z_{pmt} \leq 1, \quad \forall m \in \mathcal{M}, t \in \{2, ..., T\} \tag{2.2e}$$

$$z_{pmt} \leq y_{pmt}, \quad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \{2, ..., T\} \tag{2.2f}$$

$$z_{pmt} \leq y_{pm,t-1}, \quad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \{2, ..., T\} \tag{2.2g}$$

$$\mathrm{M}(2 - z_{pmt} - z_{pm,t+1}) + 1 \geq \sum_{p'\in\mathcal{N}} y_{p'mt}, \quad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \{2, ..., T\} \tag{2.2h}$$

$$z_{pm1} = 0, \quad \forall p \in \mathcal{N}, m \in \mathcal{M} \tag{2.2i}$$

$$y_{pmt}, z_{pmt} \in \{0, 1\}, \quad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T} \tag{2.2j}$$

$$i_{pt}, b_{pt} \geq 0, \quad \forall p \in \mathcal{N}, t \in \mathcal{T} \tag{2.2k}$$

$$x_{pmt} \geq 0, \quad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T} \tag{2.2l}$$

$$i_{p0} = b_{p0} = z_{p,T+1} = 0, \quad \forall p \in \mathcal{N} \tag{2.2m}$$

Is important to note that, in this model, all the items can be produced in all machines. The unit production time of an item varies with the machine, as well as setup time and cost. Constraint (2.2b) is the adaptation of the flow balance to backlogging and parallel machines problem. It can be seen in the following scheme.

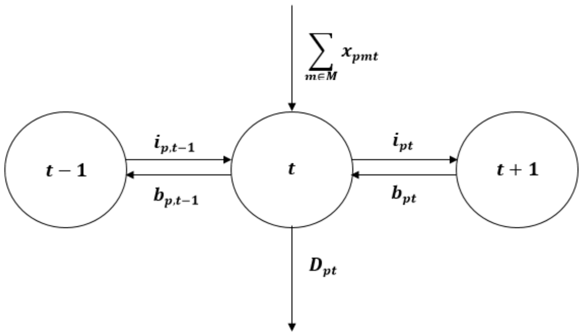


Figure 2.3: Flow balance of the multi-item single-level capacitated lot-sizing problem with backlogging, setup carryover and parallel machines

Equation (2.2c) is similar to (2.1c). Limitation of the used capacity is given by equation (2.2d). This rule indicates that, for each machine and for each time period, the total time spent with the production task has to be less than or equal to the available capacity of the machine in the period. It is important to note that where $y_{pmt} = 1$, it means that there is production of item p in machine m in period t. To make this happen, there can be a setup, if $z_{pmt} = 0$, or a setup carryover, if $z_{pmt} = 1$. Constraint (2.2e) ensures that, at the beginning of a time period, each machine is prepared at most for one item. Constrains (2.2f), (2.2g) and (2.2h) establish the relation between setup and setup carryover. At the beginning of the planning horizon, any machine can be prepared to an item. This is ensured by equation (2.2i). Finally, (2.2j), to (2.2l) define the variable's domain.
Lot-sizing problems has an important role in the industrial world. However, they are also very hard to solve. This difficulty is increased not only by the size of the instance to solve, but also by some features of the problem. The number of levels and the capacity constraints are two of these features, as well as the introduction of the sequence depend setups structure, backlogging and setup carryover.

Chapter 3

The Company

3.1 Company Presentation: Grupo Fersil

Fersil is a Portuguese company producing pipes and accessories in plastic materials. Built in 1978, in Cesar, Oliveira de Azeméis, began that year the production of PVC pipes, producing at the moment PVC, PP and PEAD pipes. Since then, Grupo Fersil has been characterized by an increasing level of expansion and, at the moment, it is in the African market, with FIL - Tubos Angola, and Fersil Moçambique.
As a national leader in production of pipes and accessories in plastic materials, Fersil has a permanent focus on quality and innovation, investing in new production and quality control technologies, responding to today's demanding market.

3.2 Product Classification

Italian sociologist and economist Vilfredo Pareto, born in 1848, observed that 80% of the wealth of his country belonged to 20% of the total population, which generated one of the most important decision support tools: Pareto's Law. This rule allows companies to make a selection of their priorities, paying more attention to the products that generate more profit. In general, these items represent a small portion of the total.
The Pareto's Law generates the ABC Classification, used to split items into classes, according to their importance level of profit generation. Thus, the class A holds the items that generate more profit, which are those with higher total annual demand. These are the most important products, and correspond to 20% of the total. In contrast, class C holds about 50% of the total, and it concentrates the items that contribute the less to the total profit. Finally, B is an intermediate class, and it holds the last 30% of the total.
Fersil produces a wide range of products for Construction (internal pipeline system of buildings), Public Works (external pipeline system of buildings), Agricultural Applications, Technological Market and Industry. These items are classified according Pareto's Law, in three classes: A, B and C.

3.3 Production Process

The production planning assumes an important role in all companies. Its first priority is to satisfy customers and, at the same time, to minimize related costs, always looking at production constraints. Fersil has a wide variety of items, such as pipes and important accessories for pipeline systems. In all pipe production, extrusion and co-extrusion machines are used. Accessories are produced by injection sectors, rotomoulded and manufactured product units. This work focus on determining batch quantities and production timing of extrusion items.
The production process belongs to the single-level class of problems, that is, between the raw material and the final product there is only one production stage. Each item can be produced by a specific

subset of machines with different setup times, setup costs and unit production times. Beyond a set of machines, each item has a set of tools also responsible for its production. Thus, it is possible to see items that can be produced by only one machine, others that can be produced by different machines, but have only one set of tools and finally others with two sets of tools. The extrusion items have only one set of extrusion tools.
Based on an analysis of inventory and demand, the needed quantities of each item are computed and production is initialized. This process can be split into three stages: machine's setup, production and machine's stoppage.
Machines' setup is the most important one due to its high cost and complexity. Here, the machine is prepared for the production of an item. This first stage can be split into four steps: change of tools, begin of production and dimensional adjustments, detailed optimization and maintenance and storage of tools. This process can be seen schematically below.

Figure 3.1: Machine's setup process: **Blue** - change of tools; **Green** - begin of production and dimensional adjustments; **Yellow** - detailed optimization; **Brown** - maintenance and storage of tools.

In the first step, the tools which allowed the previous production are removed, and there are introduced the necessary tools for the next item. After this update, the production begins, with the pipe diameter adjustments. Then, on the third stage, the final optimization is done, according to the required pipe features. It should be emphasized that through these two last phases, there is production. Although, the product is not prepared to be used, because it is result of machine adjustments. When the machine is totally prepared to produce, the maintenance and storage of tools removed on the first setup step are done and, simultaneously, the second stage of the global process starts: production. It is important to note that the production starts immediately after the detailed optimization, during the last phase of the setup process.
Once produced the needed amount, the machine's stoppage is initiated. In contrast to injection, the stoppage in extrusion machines is not instantaneous. From the moment that this process begins, the machine starts to decelerate, until really stop. There is production along this process, but the resultant product is also lost as well as in the second and third steps of setup.
First stage, machine's setup, needs skilled labor, has a cost associated to waste produced regulating equipment and a profit loss due to the setup operation. These factors turn this process the most important for the company. The number of setups per machine is limited, as well as the total time spent with setups each period.
Fersil allows the setup carryover described in the previous chapter, which constitutes a good measure to reduce the setup occurrence.
Given all features described above, one of the goals of this project consists not only in computing batch quantities in order to satisfy customer's demand, but also in finding the optimal production timing and machine for each product on each time period, taken into account the companies' restrictions.

3.4 Sales Process

Fersil usually does not work directly to satisfy customers' orders. Its production is mostly done to inventory replenishment, under a make-to-stock policy. In this way, the task of the Planning and Logistic Department consists of ensuring adequate inventory levels to provide the best customer's service.
However, with this sales policy, sometimes the amount of an item available in a specific time period is lower than the requested by the customer, leading to the so-called *backlogging*. In this case, the customer buys the available amount of product and, with respect to shortage quantity, there are two situations that can happen. In one of them, the customer awaits the production on next periods. However, there is a cost to the company associated with the fraction of demand unavailable when

was required, so-called *performance loss factor*. In the second case, the customer does not wait for next periods, and he just buys the amount available in the current period. Thus, there is a fraction of demand that is lost. In this situation, there is a higher cost to the company. In addition to performance loss, there is also a profit loss.
This proportion of non-recoverable backlogged sales depends on the product type. Items with high turnover, like these in class A, have a higher percentage of client's withdrawal, because they are easily found in other markets. Products in classes B and C have a lower percentage of loss because they are more difficult to find, what leads customers to await its production in next time periods.
For the most part, Fersil's production is done to inventory replenishment. Even though this is the predominant production policy, sometimes this company has some customers' orders of products not usually in inventory. These orders do not allow backlogging. While routine customers' requests of items in inventory are managed by the warehouse, allowing delays in the deliveries, customers' orders of products not usually kept in inventory have to be planned by the Planning and Logistic Department to be produced in such a way as to be supplied in time. To include this feature, we split the demand into two values: demand for inventory replenishment and customer's direct demand.
In conclusion, in contrast to the most studied problems of literature, Fersil backlogging is not totally recovered in next periods. There is a percentage, depending on the item, that is lost. A goal of this work consists in reducing this loss of sales, providing a higher customer's satisfaction.

Chapter 4

Modeling of Problem

The goal of this project is to develop a mathematical model for the production planning of the Fersil's extrusion items. As a starting point, we used the [Caserta and Voß, 2013] approach, presented in Chapter 2, and we adapted it to the problem under study.
As has been discussed until now, the Fersil's backlogging process differs from what has been studied in the literature. This is one of the challenges of this project and is what differentiates this problem from others seen before.
When the company receives a request for an item and cannot fully satisfy it, the client buys the available amount in the current period and, with respect to the shortage quantity, there can be two situations: he waits for the production of this portion in next periods or he does not wait. As explained before, both cases have an associated cost. The total is the backlog cost.
There is a profit loss only in the second case. If the client awaits and buys the shortage amount in next periods, the company sells this quantity. Even though it has a performance loss cost associated, proportional to this portion, it has also the profit from the sale. However, if the client does not wait, there is a profit loss because the company only sells the available amount at the moment and all the rest is lost. Thus, the goal of the Planning and Logistic Department is to keep good inventory levels to ensure the maximum customer's satisfaction, avoiding this profit loss.
In the models studied in the literature, all the backlogged sales are restored in next periods. Since in Fersil's problem this not happen, and a profit concept is present when the backlogging process is defined, we considered more natural to develop a profit maximization model rather than a cost minimization model. Hence appeared the *Base Model*.
Observing the solution of this model, we concluded that, in some situations, we obtained unmet demand of some items and available time for production in their machines at the same time. From the point of view of the Planning and Logistic Department, customers' must be served if possible. To tackle this situation, we developed a second model: the *Penalization of Backlog with Available Capacity*. This model is an extension of the previous one, and it includes an additional penalization on the backlog cost if machines have available time to produce when demand is not totally satisfied.
This chapter is organized in three sections. In the first one, we describe all the indices, parameters and variables used to model the problem. In the remaining sections, we present in detail each model aforementioned.

4.1 Indices, Parameters and Variables

In this first section, we present all the indices, parameters and variables that will be used in the following mathematical formulations. Let $\mathcal{T} = \{1, ..., T\}$ represents the planning horizon, $\mathcal{N}$ represents the set of items and $\mathcal{M}$ represents the set of machines.

Indices

t - time period identifier;

p - item identifier;

m - machine identifier.

Parameters

C_{mt} - available time for production of machine m in period t *(hours)*;

D_{pt} - total demand of item p in period t *(meters)*;

E_{mt} - maximum number of setups for machine m in period t;

F_{p} - performance loss factor for item p *(euros/meter)*;

G - gross margin excluding fixed costs *(percentage)*;

H_{p} - unit inventory cost for item p *(euros/meter)*;

I_{pt} - demand for inventory replenishment of item p in period t *(meters)*;

L_{t} - time limit to setups in period t *(hours)*;

M - big number;

O_{pt} - customer's direct demand of item p in period t *(meters)*;

P_{pm} - setup time for item p in machine m *(hours)*;

R_{p} - unit price of item p *(euros/meter)*;

S_{pm} - setup cost, excluding fixed costs, for item p in machine m *(euros)*;

U_{pm} - unit production time for item p in machine m *(hours/meter)*;

Z_{pm} - indicative of the subset of machines where item p can be produced;

ρ_{p} - percentage of demand of item p that is lost when this item is not available in time *(percentage).*

The indices were described in Chapter 2, in the *multi-item single-level capacitated lot-sizing problem with backlogging, setup carryover and parallel machines*.
Because a machine setup is a complex process, it needs skilled labor. To realize this task, Fersil has a setup team, which has limited availability. In this way, the number of setups in each machine in each period is limited to E_{mt}. Furthermore, the total number of hours spent with setups in each period is also limited to L_{t}.
Fersil allows backlogging, as explained before in Chapter 3. It has an associated cost, proportional to the shortage amount, called *performance loss factor* and represented by F_{p}.
The backlogging model in this company is different from what has been studied in the literature. While in most problems the shortage quantities are totally recovered in next periods, in this problem there is a percentage of this quantity that is lost, represented by ρ_{p}. This value dues to the customers'

withdrawal when the total required amount of an item is not available in time. While some customers wait for the production in next periods, others do not. Hence, a percentage of the total quantity that the company could sell is completely lost if not supplied in time.
The total demand, D_{pt}, is split into two values: I_{pt} is the demand for inventory replenishment; O_{pt} is the customers' orders. Naturally, the total demand is given by $\mathrm{D}_{pt} = \mathrm{I}_{pt} + \mathrm{O}_{pt}$.
The total setup costs include skilled labor, the cost associated to waste produced regulating equipment and the profit loss due to the setup operation. In this model, we only considered the variable costs, that is, the costs which the company incurs each time a machine's setup is done. In this way, the labor cost is excluded, since they are fixed. The setup cost excluding the fixed costs is represented by S_{pm}. The gross margin, represented by G corresponds to an approximation of the profit percentage earned with each sale, also excluding the fixed costs. R_p represents the sale price of each meter.
In contrast with most models with parallel machines seen until now, in this case, each item has a specific subset of machines where it can be produced. This information is represented by the coefficients Z_{pm}, which take value 1 if the item p can be produced in machine m, and 0 otherwise.

Decision Variables

x_{pmt} - batch size of item p in period t in machine m *(meters)*;

i_{pt} - amount of item p in inventory at the end of period t *(meters)*;

b_{pt} - unmet amount of item p in period t *(meters)*;

y_{pmt} - binary variable which takes value 1 if machine m produces item p in period t, and 0 otherwise;

z_{pmt} - binary variable which takes value 1 if machine m is prepared to produce item p at the beginning of period t, and 0 otherwise.

All the decision variables are the same used in the model presented in Chapter 2, thus they are known. In the next section, we present the first model studied in this project: the *Base Model*.

4.2 The Base Model

The first model studied in this project is an extension of the *multi-item single-level capacitated lot-sizing problem with backlogging, setup carryover and parallel machines* presented in Chapter 2, based on [Caserta and Voß, 2013]. We call it the *Base Model* and, being and extension of the Caserta's model, belongs to the class of NP-hard problems. In this section, we present the mathematical formulation and all its details.

Objective Function

The objective is to maximize a profit function, given by the next expression.

$$\text{maximize} \qquad \sum_{p\in\mathcal{N}}\sum_{t\in\mathcal{T}}(\theta_{pt} - \beta_{pt} - \delta_{pt} - \gamma_{pt})$$

The sales revenue is given by θ_{pt}. When, in a given period, the demand of an item is not totally satisfied, the client buys the amount available at the moment and, with respect to the remaining quantity, two situations can occur: the customer waits for its production in next periods or he does not. If the client waits, he will buy all the amount in which he was interested from the beginning. But if he does not wait, there is a portion of demand that is never sold, leading to a profit loss. Each item has associated a percentage ρ_p, which represents the customers' withdraw. Therefore, a goal of the company is to keep the best inventory levels, to be able to avoid this profit loss. Considering this backlogging model, the sales revenue is given by the expression below:

$$\theta_{pt} = (\mathrm{D}_{pt} - \rho_p b_{pt})\mathrm{R}_p\mathrm{G}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

The setup cost is represented by β_{pt} and includes the cost associated to waste produced regulating equipment and the profit loss due to the setup operation. It is important to note that a setup carryover has no associated costs, since in these situations the setup team does not need to execute any task. The setup cost is represented by the expression below:

$$\beta_{pt} = \sum_{m \in \mathcal{M}} \mathrm{S}_{pm}(y_{pmt} - z_{pmt}), \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

The inventory cost includes capital costs, average damage costs (burned and broken items), and logistic operational costs related to each product unit keep in inventory, and it is represented by the following expression:

$$\delta_{pt} = \mathrm{H}_p i_{pt}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

Backlogging cost is the performance loss that occurs every time the company cannot satisfy all the demand in time, and it is proportional to the shortage quantity. It is represented by γ_{pt}:

$$\gamma_{pt} = \mathrm{F}_p b_{pt}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

Constraints

The first constraint is the adaptation of the flow balance equation studied in the literature to the proposed problem. One of the main features of the Fersil's problem is the backlogging process. In contrast to what happens in literature, in this case, the backlogged sales are not totally recovered in next periods. This situation is represented by the following expression and can be seen more clearly in the scheme presented below:

$$i_{p,t-1} - (1 - \rho_p)b_{p,t-1} + \sum_{m \in \mathcal{M}} x_{pmt} - \mathrm{D}_{pt} - i_{pt} + b_{pt} = 0, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

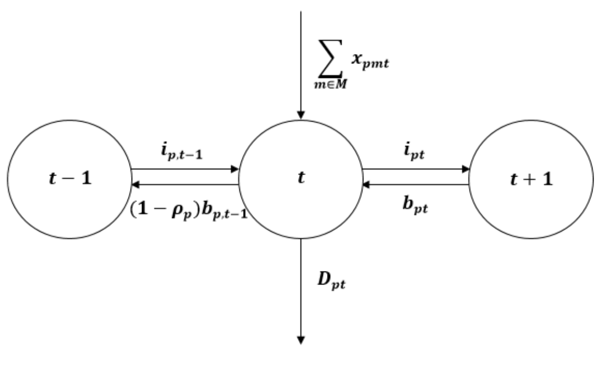


Figure 4.1: Flow balance of the Base Model

Until now, we saw that if, in a certain period, the demand was not totally satisfied, the shortage quantity was entirely recovered in next periods. In Fersil's model, this occurs in a different way, and it is the challenge of this problem. Because this business works mostly for inventory replenishment, sometimes clients do not wait for the production of items that are not available when they request them. In some cases, they only buy the available amount at the moment and the backlogs are completely

lost. Looking for the equation and for the scheme, if in period t the demand of an item p is not totally satisfied, only $(1-\rho_p)$ of the unmet amount in the current period is recovered later.
The second constraint of this problem is:

$$x_{pmt} \leq \frac{\mathrm{C}_{mt}}{\mathrm{U}_{pm}} y_{pmt} \mathrm{Z}_{pm}, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T}$$

It ensures that y_{pmt} takes value 1 if there is production of item p in machine m in period t ($x_{pmt} > 0$), and 0 otherwise. In the model studied in Chapter 2, all items could be produced on all machines. At Fersil, this does not happen. Each item has a specific subset of machines, represented by Z_{pm}, where it can be produced. If this parameter takes value 1 for a product p and a machine m, it means that this machine can produce this item and x_{pmt} can be positive. Otherwise, there cannot occur production. The big number M used before is now replaced by the maximum possible batch size of p on machine m, given by $\frac{\mathrm{C}_{mt}}{\mathrm{U}_{pm}}$.
The equation below represents the machine's capacity limitation, seen in the *multi-item single-level capacitated lot-sizing problem with backlogging, setup carryover and parallel machines* presented before. It is important to note that setup carryovers do not reduce production time:

$$\sum_{p \in \mathcal{N}} \mathrm{U}_{pm} x_{pmt} + \sum_{p \in \mathcal{N}} \mathrm{P}_{pm}(y_{pmt} - z_{pmt}) \leq \mathrm{C}_{mt}, \qquad \forall m \in \mathcal{M}, t \in \mathcal{T}$$

The most important production stage of Fersil is the machine's setup. Since it requires skilled labor and is a complex process, the setup team has limited availability. Because of that, the company has to impose limits in its occurrence. In each period, there can be at most E_{mt} per machine:

$$\sum_{p \in \mathcal{N}} (y_{pmt} - z_{pmt}) \leq \mathrm{E}_{mt}, \qquad \forall m \in \mathcal{M}, t \in \mathcal{T}$$

Moreover, in each period, the total time spent with machines' setup has to be less than or equal to an established limit given by the parameter L_t. This limitation is represented by the following expression:

$$\sum_{p \in \mathcal{N}} \sum_{m \in \mathcal{M}} \mathrm{P}_{pm}(y_{pmt} - z_{pmt}) \leq \mathrm{L}_t, \qquad \forall t \in \mathcal{T}$$

There is a reason to include two different setup's limits in the model. Currently, the company is able to do at most one setup in each working day on each machine, from what the limit of setups per period and per machine is limited. Setup times may vary by machine and items, as seen before. Some of them are relatively brief, but others can be very long. When planning the production for a given time horizon, the company must ensure that the setup team will be able to prepare all the planned machines. If we are dealing with big setup processes, and the model used to build the production planning only limits the number of setups, it can happen that the number of work hours of the setup team is lower than the needed time to do all the setups planned. This is the reason why it is important to limit not only the number of setups, but also the total time spent with this process in each period.
The following equation defines backlogging policy of customers' orders:

$$b_{pt} \leq \mathrm{I}_{pt}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

Even though Fersil works mostly for stock replenishment, sometimes there are special orders of items not in inventory. Typically, they are small quantities, representing about $3-4\%$ of the total sales. When it happens, the company has to ensure that these quantities are available in time, that is, backlogging is not allowed in this case. Routine customers' requests of items in inventory are managed by the warehouse, allowing delays if the ordered quantities are not available. These are hence treated

differently of customers' orders of products not usually kept in inventory, which have to be planned by the company's managers to be produced in such a way as to be supplied in time.
In under-capacity situations, in which Fersil mostly works, the model presented until now decides to produce items with higher rotation. It makes sense and is one of the model's objectives, since the best production plan is the one that gives priority to the most profitable items. However, items with low demand, as these in class C, are seldom produced, because their sales revenue does not compensate their high production costs. In fact, in the model described until now, these items may not be produced even if there is capacity available, as no-production may be cheaper than production.
As usually customer's orders of items not in inventory are small quantities, it was necessary to extend the present model to ensure that these amounts are available in time. To adapt our model, we considered a split of the demand into two values: O_{pt} and I_{pt}. O_{pt} represents the customers' orders and I_{pt} corresponds to the demand for the stock replenishment. Naturally, $\mathrm{D}_{pt} = \mathrm{O}_{pt} + \mathrm{I}_{pt}$. The previous equation imposes that customers' orders are always produced in time.
The next constraint ensures that each machine at the beginning of each period can be prepared at most for one item, that is, at most one z_{pmt} is non-zero:

$$\sum_{p \in \mathcal{N}} z_{pmt} \leq 1, \qquad \forall m \in \mathcal{M}, t \in \{2, ..., T\}$$

As seen before, y_{pmt} takes value 1 if there is production of p in machine m in period t, and 0 otherwise. If machine m is not prepared for item p at the beginning of period t, a setup must occur. In this case, we have $y_{pmt} = 1$ and $z_{pmt} = 0$. On the other hand, if this item was the last one to be produced in this machine in the previous period, no setup is needed. Hence we have $y_{pmt} = 1$ and $\alpha_{pmt} = 1$. In other words, z_{pmt} can only take value 1 if y_{pmt} is also non-zero:

$$z_{pmt} \leq y_{pmt}, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \{2, ..., T\}$$

If a machine is prepared for some item at the beginning of a time period, either a setup has been done in the previous period, or the machine was already prepared for this item:

$$z_{pmt} \leq y_{pm,t-1}, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \{2, ..., T\}$$

Next constraint ensures that if there are consecutive setup carryovers for the same item, no other item can be produced in that machine:

$$\mathrm{M}(2 - z_{pmt} - z_{pm,t+1}) + 1 \geq \sum_{p' \in \mathcal{N}} y_{p'mt}, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \{2, ..., T\}$$

We assume that at the beginning of the planning horizon, machines are not prepared to produce any item. In case of production in the first period, a setup must occur:

$$z_{pm1} = 0, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M}$$

As seen before, to produce an item it is necessary to have both a machine and a set of extrusion tools available. Each extrusion item has a subset of machines where it can be produced, but its set of tools is unique. Hence an item cannot be produced on two machines simultaneously:

$$\sum_{m \in \mathcal{M}} y_{pmt} \leq 1, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

These constraints define the variables' domain.

$$
\begin{aligned}
&y_{pmt}, z_{pmt} \in \{0,1\}, && \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T} \\
&i_{pt}, b_{pt} \geq 0, && \forall p \in \mathcal{N}, t \in \mathcal{T} \\
&x_{pmt} \geq 0, && \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T} \\
&i_{p0} = b_{p0} = z_{p,T+1} = 0, && \forall p \in \mathcal{N}
\end{aligned}
$$

The complete mathematical formulation is presented below.

$$
\text{maximize} \qquad \sum_{p\in\mathcal{N}}\sum_{t\in\mathcal{T}}(\theta_{pt} - \beta_{pt} - \delta_{pt} - \gamma_{pt}) \tag{4.1a}
$$

subject to:

$$
i_{p,t-1} - (1-\rho_p)b_{p,t-1} + \sum_{m\in\mathcal{M}} x_{pmt} - \mathrm{D}_{pt} - i_{pt} + b_{pt} = 0, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T} \tag{4.1b}
$$

$$
x_{pmt} \leq \frac{\mathrm{C}_{mt}}{\mathrm{U}_{pm}} y_{pmt} \mathrm{Z}_{pm}, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T} \tag{4.1c}
$$

$$
\sum_{p\in\mathcal{N}} \mathrm{U}_{pm} x_{pmt} + \sum_{p\in\mathcal{N}} \mathrm{P}_{pm}(y_{pmt} - z_{pmt}) \leq \mathrm{C}_{mt}, \qquad \forall m \in \mathcal{M}, t \in \mathcal{T} \tag{4.1d}
$$

$$
\sum_{p\in\mathcal{N}} (y_{pmt} - z_{pmt}) \leq \mathrm{E}_{mt}, \qquad \forall m \in \mathcal{M}, t \in \mathcal{T} \tag{4.1e}
$$

$$
\sum_{p\in\mathcal{N}}\sum_{m\in\mathcal{M}} \mathrm{P}_{pm}(y_{pmt} - z_{pmt}) \leq \mathrm{L}_t, \qquad \forall t \in \mathcal{T} \tag{4.1f}
$$

$$
b_{pt} \leq \mathrm{I}_{pt}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T} \tag{4.1g}
$$

$$
\sum_{p\in\mathcal{N}} z_{pmt} \leq 1, \qquad \forall m \in \mathcal{M}, t \in \{2,...,T\} \tag{4.1h}
$$

$$
z_{pmt} \leq y_{pmt}, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \{2,...,T\} \tag{4.1i}
$$

$$
z_{pmt} \leq y_{pm,t-1}, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \{2,...,T\} \tag{4.1j}
$$

$$
\mathrm{M}(2 - z_{pmt} - z_{pm,t+1}) + 1 \geq \sum_{p'\in\mathcal{N}} y_{p'mt}, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \{2,...,T\} \tag{4.1k}
$$

$$
z_{pm1} = 0, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M} \tag{4.1l}
$$

$$
\sum_{m\in\mathcal{M}} z_{pmt} \leq 1, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T} \tag{4.1m}
$$

$$
y_{pmt}, z_{pmt} \in \{0,1\}, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T} \tag{4.1n}
$$

$$
i_{pt}, b_{pt} \geq 0, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T} \tag{4.1o}
$$

$$
x_{pmt} \geq 0, \qquad \forall p \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T} \tag{4.1p}
$$

$$
i_{p0} = b_{p0} = z_{p,T+1} = 0, \qquad \forall p \in \mathcal{N} \tag{4.1q}
$$

4.3 Penalization of Backlog with Available Capacity

When we analyzed the solution of the previous model, we found an interesting aspect. In certain periods, even though the demand of some products is not totally satisfied, the machines where these items can be produced still have available time to produce. In other words, in some situations, the costs associated with the production process are larger than the no-production costs of certain products.
However, by decision of the company's managers, customers must be served if possible. In particular, backlogging should be more penalized if there is available capacity to produce.
Based on these facts, the second model of this problem emerged: the *Penalization of Backlog with Available Capacity*. It is an extension of the previous model, thus, it also belongs to the class of NP-hard problems. In each period, if the demand of a certain item is not totally satisfied, and the subset of machines able to produce it has available capacity, taking into consideration the setup time for this item, then the backlog cost of this product suffers a penalization proportional to the delayed amount. The goal of this model is to promote a better customer service, even with higher costs for the company.
To extend the *Base Model* to the *Penalization of Backlog with Available Capacity*, it was necessary to introduce some parameters and decision variables. These are presented below, followed by the mathematical formulation of the final model.

Parameters

K - increasing factor in the backlog cost.

Decision Variables

a_{pt} - non-negative real variable that indicates the total available time of the machines subset of p in period t, taking into consideration the setup time for p in each machine *(hours)*;

c_{mt} - total used time of machine m in period t *(hours)*;

e_{pt} - increasing on backlog cost *(euros)*;

v_{pt} - binary variable which takes value 1 if the subset of machines of item p has available capacity, taking into consideration the setup time for this item in each machine, and 0 otherwise;

w_{pt} - binary variable which takes value 1 if penalization on backlog cost of item p in period t must occur;

α_{pt} - binary variable which takes value 1 if there is unmet demand of item p in period t, and 0 otherwise.

Since this is an extension of the previous model, the objective is to maximize a profit function.

Objective Function

$$\text{maximize} \quad \sum_{p \in N} \sum_{t \in T} (\theta_{pt} - \beta_{pt} - \delta_{pt} - \gamma_{pt})$$

Sales revenue, setup cost and inventory cost are the same used in the previous model:

$$\theta_{pt} = (\mathrm{D}_{pt} - \rho_p b_{pt})\mathrm{R}_p\mathrm{G}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

$$\beta_{pt} = \sum_{m \in \mathcal{M}} \mathrm{S}_{pm}(y_{pmt} - z_{pmt}), \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

$$\delta_{pt} = \mathrm{H}_p i_{pt}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

The backlog cost is defined by a new expression, which includes the penalization mentioned before:

$$\gamma_{pt} = \mathrm{F}_p b_{pt} + \mathrm{K} e_{pt}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

Constraints

All constraints of the previous model remain the same in the present, except (4.1g). It is replaced by the following expression to also ensure that the binary variable α_{pt} takes value 1 if there are backlogged sales of p in period t, and 0 otherwise:

$$b_{pt} \leq \mathrm{I}_{pt}\alpha_{pt}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

The total available time to produce of the machines' subset of p in period t, taken into consideration the setup time for p in each machine, is represented by the expression below:

$$a_{pt} \geq \sum_{m \in \mathcal{M}} (\mathrm{C}_{mt} - c_{mt} - \mathrm{P}_{pm})\mathrm{Z}_{pm}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

The following constraint ensures that the binary variable v_{pt} takes value 1 if a_{pt} is positive, that is, if there is at least one machine with available time to produce p in period t, taken into consideration the setup time for p in each machine:

$$a_{pt} \leq \sum_{m \in \mathcal{M}} \mathrm{C}_{mt}\mathrm{Z}_{pm}v_{pt}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

Penalization only occurs if the demand of p is not fully satisfied in period t and there is available time to produce this item in at least one machine of its machines' subset of p, taken into consideration the setup time for p. In other words, penalization only occurs if both α_{pt} and v_{pt} take value 1. To concentrate both cases on a single variable, we introduce w_{pt}, that takes value 1 when a penalization must occur, and 0 otherwise. The following constraint defines the values of this variable:

$$w_{pt} \geq \alpha_{pt} + v_{pt} - 1, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

The last constraint presented establishes the penalization in the backlog cost. As mentioned before, penalization should occur only when w_{pt} takes value 1, which is ensured by this expression. This penalization is proportional to the delayed amount of p in period t, b_{pt}. If there are backlogged sales of an item in a certain period, but the machines' subset of this product is totally occupied, then $w_{pt} = 0$ and no extra backlog cost is incurred.
The penalization is increased by the factor K, presented in the backlog cost formulation. If $\mathrm{K} = 0$, there is no increase in the backlog cost and the model reverts to *Base Model* presented in the previous section. If $\mathrm{K} = 1$, the backlog cost doubles, if $\mathrm{K} = 2$ it triplicates, and so on. This factor allows the user to obtain different solutions, which corresponds to different production plans, for the same instance, allowing the managers to do a more informed decision.

$$e_{pt} \geq \mathrm{F}_p b_{pt} - \mathrm{M}(1 - w_{pt}), \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

It follows the complete mathematical formulation of the *Penalization of Backlog with Available Capacity*.

$$\text{maximize} \quad \sum_{p\in\mathcal{N}}\sum_{t\in\mathcal{T}}(\theta_{pt}-\beta_{pt}-\delta_{pt}-\gamma_{pt}) \tag{4.2a}$$

subject to:

$$i_{p,t-1}-(1-\rho_p)b_{p,t-1}+\sum_{m\in\mathcal{M}}x_{pmt}-\mathrm{D}_{pt}-i_{pt}+b_{pt}=0, \quad \forall p\in\mathcal{N},t\in\mathcal{T} \tag{4.2b}$$

$$x_{pmt}\leq\frac{\mathrm{C}_{mt}}{\mathrm{U}_{pm}}y_{pmt}\mathrm{Z}_{pm}, \quad \forall p\in\mathcal{N},m\in\mathcal{M},t\in\mathcal{T} \tag{4.2c}$$

$$\sum_{p\in\mathcal{N}}\mathrm{U}_{pm}x_{pmt}+\sum_{p\in\mathcal{N}}\mathrm{P}_{pm}(y_{pmt}-z_{pmt})\leq\mathrm{C}_{mt}, \quad \forall m\in\mathcal{M},t\in\mathcal{T} \tag{4.2d}$$

$$\sum_{p\in\mathcal{N}}(y_{pmt}-z_{pmt})\leq\mathrm{E}_{mt}, \quad \forall m\in\mathcal{M},t\in\mathcal{T} \tag{4.2e}$$

$$\sum_{p\in\mathcal{N}}\sum_{m\in\mathcal{M}}\mathrm{P}_{pm}(y_{pmt}-z_{pmt})\leq\mathrm{L}_t, \quad \forall t\in\mathcal{T} \tag{4.2f}$$

$$b_{pt}\leq\mathrm{I}_{pt}\alpha_{pt}, \quad \forall p\in\mathcal{N},t\in\mathcal{T} \tag{4.2g}$$

$$\sum_{p\in\mathcal{N}}z_{pmt}\leq 1, \quad \forall m\in\mathcal{M},t\in\{2,...,T\} \tag{4.2h}$$

$$z_{pmt}\leq y_{pmt}, \quad \forall p\in\mathcal{N},m\in\mathcal{M},t\in\{2,...,T\} \tag{4.2i}$$

$$z_{pmt}\leq y_{pm,t-1}, \quad \forall p\in\mathcal{N},m\in\mathcal{M},t\in\{2,...,T\} \tag{4.2j}$$

$$\mathrm{M}(2-z_{pmt}-z_{pm,t+1})+1\geq\sum_{p'\in\mathcal{N}}y_{p'mt}, \quad \forall p\in\mathcal{N},m\in\mathcal{M},t\in\{2,...,T\} \tag{4.2k}$$

$$z_{pm1}=0, \quad \forall p\in\mathcal{N},m\in\mathcal{M} \tag{4.2l}$$

$$\sum_{m\in\mathcal{M}}z_{pmt}\leq 1, \quad \forall p\in\mathcal{N},t\in\mathcal{T} \tag{4.2m}$$

$$a_{pt}\geq\sum_{m\in\mathcal{M}}(\mathrm{C}_{mt}-c_{mt}-\mathrm{P}_{pm})\mathrm{Z}_{pm}, \quad \forall p\in\mathcal{N},t\in\mathcal{T} \tag{4.2n}$$

$$a_{pt}\leq\sum_{m\in\mathcal{M}}\mathrm{C}_{mt}\mathrm{Z}_{pm}v_{pt}, \quad \forall p\in\mathcal{N},t\in\mathcal{T} \tag{4.2o}$$

$$w_{pt}\geq\alpha_{pt}+v_{pt}-1, \quad \forall p\in\mathcal{N},t\in\mathcal{T} \tag{4.2p}$$

$$y_{pmt},z_{pmt},\alpha_{pt},v_{pt},w_{pt}\in\{0,1\}, \quad \forall p\in\mathcal{N},m\in\mathcal{M},t\in\mathcal{T} \tag{4.2q}$$

$$i_{pt},b_{pt},e_{pt},a_{pt}\geq 0, \quad \forall p\in\mathcal{N},t\in\mathcal{T} \tag{4.2r}$$

$$x_{pmt}\geq 0, \quad \forall p\in\mathcal{N},m\in\mathcal{M},t\in\mathcal{T} \tag{4.2s}$$

$$c_{mt}\geq 0, \quad \forall m\in\mathcal{M},\forall t\in\mathcal{T} \tag{4.2t}$$

$$i_{p0}=b_{p0}=z_{p,T+1}=0, \quad \forall p\in\mathcal{N} \tag{4.2u}$$

Chapter 5

Data and Computational Results

In this chapter, we start with a brief data description followed by a presentation and analysis of some computational results.

5.1 Data Description

All the necessary data was provided by the company, except the demand matrix. As the company produces mostly to inventory rather than to directly satisfy customers' orders, they provided us a sales history of the last periods. To test the developed models, we use this history to generate the two demand matrices that are needed.

Analyzing the sales history matrix, we observed that there are periods for which some items have no demand. Thus, for each product, each period is assumed to have a probability of having sales. Based on this value, computed from the sales history, we randomly select the periods of the planning horizon for which demand values will be generated. The next step consists of generating these values. The details of this demand matrices generator can be consulted in Appendix A.

As seen before, this model includes a parameter, K, which establishes the penalization on the backlog cost when there is available time to produce in the machines. Varying this parameter, we build two versions of the model:

- **Version 1**: $\mathrm{K} = 0$
- **Version 2**: $\mathrm{K} = 1$

To better understand the difference between both versions of the model, let us remind the expressions of the backlog cost and of the penalization:

$$\gamma_{pt} = \mathrm{F}_p b_{pt} + \mathrm{K} e_{pt}, \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$
$$e_{pt} \geq \mathrm{F}_p b_{pt} - \mathrm{M}(1 - w_{pt}), \qquad \forall p \in \mathcal{N}, t \in \mathcal{T}$$

Observing these equations, it is easy to see that in Version 1, there is no extra penalization on the backlog cost, that is, this model reverts to the *Base Model* presented in Chapter 4. In Version 2, if there is available time to produce, the backlog cost doubles.

5.2 Computational Results

The computational results were obtained using a Intel Celeron CPU running 1000M 1.80 Ghz, with 4Gb of RAM using an Oracle VM Virtual Box Version 4.3.28 Edition running Linux. We used the Gurobi Python Interface, with Python version 2.7.9 and Gurobi version 5.0, to model the problem and solve it. All the results obtained were stored in an SQL database, using the sqlite3 module in Python. The demand matrices generator was also codded in Python.
Both versions of this model were tested and compared on 40 instances of different sizes, grouped in four classes, as reported in table 5.1, according to the number of items and machines.

Table 5.1: Instances grouped by number of items and number of machines

Group ID	Number of Items	Number of Machines
A	3	4
B	6	4
C	15	4
D	30	4

For each group, we generate ten instances and both versions of the model were tested for each of them, in a planning horizon of six periods. We use the following stopping criteria:

- MIPgap = 0.0001 %
- Time Limit = 3600 seconds

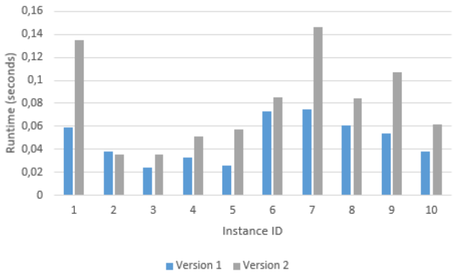
We measured the average runtime and the average GAP of each class. The GAP is a measure of the quality of the solution; for minimizing problems, it is $\frac{U-L}{U} \times 100\%$, where U is the objective value for the best feasible solution found, and L is the best lower bound. The closer this value is to zero, the better is the solution. For a better understanding of this measure, see Appendix B.
In the table 5.2 we present the average runtime and the average GAP for both versions.

Table 5.2: Runtime average and GAP average of each instance class

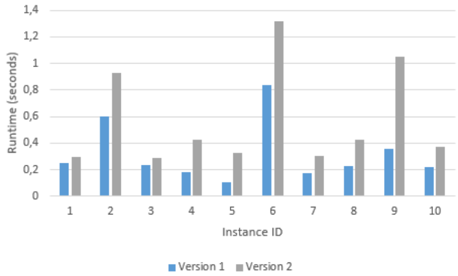
Group ID	Runtime Average (seconds)		GAP Average (%)	
	Version 1	**Version 2**	**Version 1**	**Version 2**
A	0.0481	0.0798	<0.01	<0.01
B	0.318	0.573	<0.01	<0.01
C	30.6	174	<0.01	<0.01
D	1673	3600	0.0409	0.347

Comparing the results of the two versions in the tables above, we can conclude that the average runtime and GAP are higher in Version 2 than in Version 1. These values show that in fact the model where the backlog cost is doubled when there are shortage sales and available capacity of the machines in the same period is more difficult in practice than the model without any additional penalization.

Each graphic presented below represents the results for each instance's class. In each of them, we present the runtime of each instance for both versions.

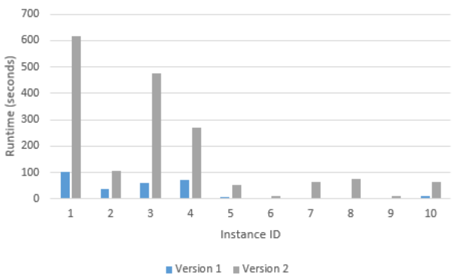


(a) Runtime of instances class A

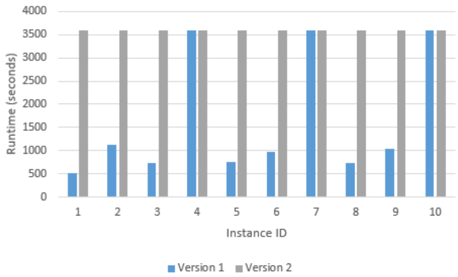


(b) Runtime of instances class B

Figure 5.1: Runtime of instances classes A and B



(a) Runtime of instances class C



(b) Runtime of instances class D

Figure 5.2: Runtime of instances classes C and D

Looking at the graphics, we can see again that there is a significant difference between the runtime between the two versions. We can further add that with the growth of the instances, this difference increases.

It is important to note that the MIP solver [Gurobi Optimization, 2012] used to test the *Penalization of Backlog with Available Capacity* was able to solve Version 1 of this model within the time limit established as stopping criteria for almost all instances with an average GAP very close to zero. This did not happen with Version 2, where for instances of class D, the time limit was always crossed, obtaining results with a higher average GAP, of 0.347%. However, this is a small value for the GAP, which allow us to assert that the MIP solver found a good solution. The detailed results of all instances can be consulted on Appendix C.

5.3 Results Analysis

In this section, we present some details of the computational results obtained with the two versions of the final model, *Penalization of Backlog with Available Capacity*. For all instances used in the following subsections, the GAP obtained by the MIP solver, [Gurobi Optimization, 2012], was less than 0.01%.

5.3.1 Two Small Instances

It is important to understand the penalization effect in the *Penalization of Backlog with Available Capacity*. In this section, we analyze in detail two different examples of the application of the two versions of this model. In the first case, we will see a decreasing on the backlogging between the Version 1 and the Version 2, as were our goal. In the second case, even with the penalization, the backlogging values will not change.

The First Example

Let us consider a set of 3 items and 3 machines, with the following relation item-machine, given by the parameter Z_{pm}:

Table 5.3: Relation item-machine given by parameter Z_{pm}

Item/Machine	1	2	3
A	0	1	1
B	1	0	0
C	1	0	0

All the data of the parameters involved in the model can be consulted in Appendix D. Next, we present some results to better understand the behavior of the two versions.

Table 5.4: Setup costs, inventory costs, backlogging costs, sales revenue and total profit

Version 1

	Period						Profit = 4202
	1	**2**	**3**	**4**	**5**	**6**	**Total**
Setup	301	196	0	0	0	0	497
Inventory	0	50	34	17	0	0	101
Backlogging	21	0	0	0	0	3	24
Revenue	993	897	869	923	262	880	4824

Version 2

	Period						Profit = 4182
	1	**2**	**3**	**4**	**5**	**6**	**Total**
Setup	497	0	0	0	0	0	497
Inventory	66	50	34	17	0	0	167
Backlogging	0	0	0	0	0	6	6
Revenue	1021	897	869	923	262	880	4852

Observing the results presented in the table above, the first aspect to note is the fact that the total profit decreased between versions. This was expected, since we added an extra cost to the objective function. After that, we can see that the backlog costs decreased and the sales revenue increased between the two versions, which allow us to conclude that the penalization worked as intended. Let us analyze a comparison between the values of the demand, inventory and backlogging per period to better understand the model's solution.

Table 5.5: Total demand, customers' orders, demand for inventory replenishment, inventory and backlogging

Version 1							
Period							
Variable	**1**	**2**	**3**	**4**	**5**	**6**	**Total**
D	2237	1636	1576	1678	413	1609	9149
O	0	0	0	0	0	0	0
I	2237	1636	1576	1678	413	1609	9149
i	0	1257	839	413	0	0	2509
b	1241	0	0	0	0	411	1652

Version 2							
Period							
Variable	**1**	**2**	**3**	**4**	**5**	**6**	**Total**
D	2237	1636	1576	1678	413	1609	9149
O	0	0	0	0	0	0	0
I	2237	1636	1576	1678	413	1609	9149
i	1652	1257	839	413	0	0	4161
b	0	0	0	0	0	411	411

Observing the previous table, we can verify that, for the same demand, the backlogging values significantly decreased. However, the inventory levels also increased, what at first sight apparently has no explanation. Analyzing the production details, which can be consulted in Appendix D, we can justify these results.

In Version 1 of the model, there are two periods of the planning horizon with backlogging: the first one and the last one. The demand not satisfied in the first period belongs to the item B and of the last period belongs to item C, both produced in the machine 1. Analyzing the total occupation of this machine in these periods, which can also be consulted in Appendix D, we can verify that its limit was not reached in any of them, from what we are in a situation where the penalization must be applied. In this way, looking for the results of the Version 2, we can find two different cases. The first one is about item B. While in Version 1, it was not produced in the first period of the planning horizon, in Version 2 it was, thus, all its demand was satisfied. However, the amount produced in the first period of item C increases in Version 2. To not suffer the penalization on the backlog cost, the model chose to produce item B in the first period, contrary to what happened in Version 1. As in the period 2 this item was also produced, the machine 1 did a setup carryover. Since each machine can do only a setup carryover each period, the amount of C that was produced in periods 1 and 2 in the Version 1, was totally produced in the first period in Version 2, remaining in inventory.

The other situation corresponds to item C. Its backlogging values did not change between the two versions. That is, the amount of unmet demand in Version 1 remained the same in Version 2, suffering the penalization on the objective function, once there was available time to produce in this period in machine 1. To produce the demand of the last period, it would be necessary a new setup, what would be less profitable. The other option would be to produce a higher quantity in the first period, taking advantage of the setup already done. However, this amount would be in inventory until the last period, what would incur a higher cost than the penalization. Taken into account all these cases, the model chose to suffer the penalization on the backlog cost. This kind of situation may occur when the delay quantity is small, and its sales revenue does not compensate a new setup or the inventory costs, as in this case.

It is important to note that the item A can be produced on two machines, as we can see in table 5.3. It is a role of the model to choose the best machine, in term of costs. In fact, the model selected machine 2 to produce this item, which is where the setup cost is lower.

The Second Example

Let us consider a set of 3 items and 3 machines, with the next relation item-machine:

Table 5.6: Relation item-machine given by parameter Z_{pm}

Item/Machine	**1**	**2**	**3**
D	1	0	0
E	0	1	1
F	0	1	1

All the data of the parameters involved in the model can also be consulted in Appendix D. Next we present some results and its analyzes.

Table 5.7: Setup costs, inventory costs, backlogging costs, sales revenue and total profit

Version 1

	Period						**Profit = 10637**
	1	**2**	**3**	**4**	**5**	**6**	**Total**
Setup	755	0	0	0	0	0	755
Inventory	0	0	0	0	0	0	0
Backlogging	636	822	881	929	956	945	5169
Revenue	2879	2770	2749	2749	2722	2692	16561

Version 2

	Period						**Profit = 10387**
	t1	**t2**	**t3**	**t4**	**t5**	**t6**	**Total**
Setup	755	0	0	0	0	0	755
Inventory	7	22	35	48	62	76	250
Backlogging	636	822	881	929	956	945	5169
Revenue	2879	2770	2749	2749	2722	2692	16561

The backlog cost penalization of an item occurs if there are backlogged sales and available time to produce in the machines where this item can be produced at the same time. In this situation, the model tries to totally occupy the machines in the most profitable way. What is expected to happen is the production of the backlogged items, decreasing the backlogging and increasing the machines occupation, avoiding the penalization in the objective function. However, in some situations, this does not happen in this way.

Observing the tables, it is clearly that the backlog costs are the same in the two versions, what enables us to conclude that the backlogs were not reduced. Table 5.8 proves this conclusion, showing that for the same values of demand, the two versions obtained the same values of backlogging in each time period.

Table 5.8: Total demand, customers' orders, demand for inventory replenishment, inventory and backlogging

Version 1							
	Period						
Variable	**1**	**2**	**3**	**4**	**5**	**6**	**Total**
D	17211	17194	17209	17313	17266	17062	103255
O	0	0	0	0	0	0	0
I	17211	17194	17209	17313	17266	17062	103255
i	0	0	0	0	0	0	0
b	3250	4198	4502	4743	4883	4827	26403

Version 2							
	Period						
Variable	**1**	**2**	**3**	**4**	**5**	**t6**	**Total**
D	17211	17194	17209	17313	17266	17062	103255
O	0	0	0	0	0	0	0
I	17211	17194	17209	17313	17266	17062	103255
i	804	2388	3814	5232	6704	8252	27194
b	3250	4198	4502	4743	4883	4827	26403

The setup costs and the sales are also the same. The only values which changed between the two versions were the inventory levels and, as a consequence, the inventory costs.
Analyzing the production details, we verified that the item F was the only one with unmet demand, in the Version 1. The model chose to produce this item on machine 2, where the setup cost is lower. A setup in the first period was done and this machine state was carried over all the planning horizon (these results can be consulted in Appendix D). With the penalization on backlog costs, we verified that the backlogging of this item did not decrease. In this way, we decided to analyze the occupation of machine 2 in all periods in Version 1. These values are also presented in Appendix D. Taken into account that the total available time of all machines was 15 hours per period, we concluded that machine 2 was fully occupied all periods. Even though item F could be produced also in machine 3, and this had available time to produce, this model does not allow simultaneous production, as discussed before. Therefore, there was no possibility to satisfy all the demand of this item. However, when we applied the Version 2 to this instance, the backlog cost suffered a penalization, because machine 3 had available time and it is able to produce the item with backlogged sales. The solution of the model to avoid this extra cost was to produce higher quantities of item E, which had been produced in machine 3, occupying all its time capacity. These quantities were stored in inventory. This is the reason why, in this particular case, only the inventory costs and levels changed between the two versions. It is easy to conclude that even increasing the penalization factor on backlog cost, the results will not change, since the machine is totally occupied.
It is important to note that for items E and F, both produced in machines 2 and 3, machine 2 is where the setup cost is lower. However, the model chose to produce item F in machine 2 and item E in machine 3. The reason behind this is that the setup cost difference between both machines is higher for item F than for item E.
To conclude this results analysis, it is important to note that, to generate the optimal production plan, the model establishes a trade-off between the setup and inventory costs. On one hand, it is important to use the maximum machine's capacity each time a setup is done, to offset its costs. On the other hand, producing high quantities, it obtain high inventory costs. In this way, the balance between both costs is very important.

5.3.2 A Variant of The Penalization of Backlog with Available Capacity

As discussed before, Fersil does not allow simultaneous production. In other words, two items cannot be produced in the same period on two different machines. The explanation behind this rule is the fact that, even though each extrusion product has a set of machines where its production can occur, its set of tools is unique.

In this section, we build a variant of the final model, where the simultaneous production is possible. The main goal of this model is to verify if an investment in new sets of tools would be or not profitable for the company.

Next we present a comparison between the original model and its variant without the constraint which does not allow the simultaneous production. To do this, we used the instance of the second example of the subsection 5.3.1 and we applied to this set the Version 1 of the final model, that is, the version without penalization of the backlog cost where there are backlogging and available time to produce in the machines at the same time. This comparison is done in terms of costs and profit, and these values can be analyzed in the tables below.

Table 5.9: Setup costs, inventory costs, backlogging costs, sales revenue and total profit

The Penalization of Backlog with Available Capacity							
	Period						**Profit = 10637**
	1	**2**	**3**	**4**	**5**	**6**	**Total**
Setup	755	0	0	0	0	0	755
Inventory	0	0	0	0	0	0	0
Backlogging	636	822	881	929	956	945	5169
Revenue	2879	2770	2749	2749	2722	2692	16561
Variant of The Penalization of Backlog with Available Capacity							
	Period						**Profit = 15529**
	1	**2**	**3**	**4**	**5**	**6**	**Total**
Setup	842	290	0	377	290	0	1799
Inventory	34	28	0	0	28	0	90
Backlogging	417	327	0	72	342	0	1158
Revenue	3024	3037	3166	3162	3047	3140	18576

Observing the values presented above, we can conclude that it would be more profitable to invest in new sets of tools, to allow the simultaneous production of an item on two or more machines at the same time. Even though the setup and inventory costs are higher in the variant of the model, the backlog costs are smaller, what enables us to conclude that with more tools we would have a better satisfaction of the demand. In fact, the sales revenue is higher in the second case, as well as the total profit.

Analyzing these results, we can say that new sets of tools could be good for the company, depending on the investment value. However, since this update leads to an increasing in the setups occurrence, this change in the production process could lead to a second investment, which consists in the growth of the setup team.

5.3.3 A Real Fersil's Instance

In the section 5.3.1, we tested the model in two small instances, with 3 items and 3 machines. However, we consider important understanding the performance of the model in a real instance, with more items per machine and in an under-capacity situation. In this way, in this section we present a Fersil's instance, with 15 items and 4 machines. The set of products is composed of 13 items of class A and 2 of class C. It is important to note that all the class A production is done to inventory replenishment, and all the class C corresponds to customer's orders. Thus, this last class does not allow backlogging. We started with the application of the Version 1 of the final model, that is, no extra penalization on the backlog cost is done. All the data related to this instance can be consulted on the Appendix E, as well as the production details.

Along this section, we present some summary results, where we can analyze the model performance and some aspects not yet discussed with the other instances studied.

We start to present the total costs and sales revenue per period, the total profit and the costs' distribution. The costs' distribution is the percentage of each cost in the revenue: $\frac{Cost}{Revenue}$.

Table 5.10: Setup costs, inventory costs, backlogging costs, revenue and total profit

	Period						**Profit= 343221**
	1	**2**	**3**	**4**	**5**	**6**	**Total**
Setup	3190	2055	737	2152	1399	680	10213
Inventory	642	666	385	556	525	0	2774
Backlogging	211	0	0	0	0	103	314
Revenue	58343	60424	56929	60227	60411	60188	356522

Table 5.11: Costs' distribution (%)

	Period					
	1	**2**	**3**	**4**	**5**	**6**
Setup	5.47	3.40	1.29	3.57	2.32	1.13
Inventory	1.10	1.10	0.68	0.92	0.87	0
Backlogging	0.36	0	0	0	0	0.17

Observing the results above, we can verify that the setup costs represent the highest percentage of the sales revenue. This enables us to conclude that, in fact, this is the most expensive process for the company, and deserves special focus.

At certain times of the year, Fersil works in under-capacity situations, reason why in these periods it is not able to produce all the demand. In this way, the company's objective consists of making a selection of the items to produce, optimal for the company and for the customers.

As seen before, the most profitable items are usually those which has a higher rotation. In addition, these are the items with higher withdrawal's percentage in the case of backlogging. With these two features, it is easy to conclude that the model's goal must be giving the production priority to these items.

In the following table, we present the percentage of the unmet demand of each item in each period of the planning horizon.

Table 5.12: Percentage of unmet demand (%)

	Period					
Item	**1**	**2**	**3**	**4**	**5**	**6**
A	0	0	0	0	0	0
B	0	0	0	0	0	0
C	0	0	0	0	0	0
D	0	0	0	0	0	0
E	0	0	0	0	0	0
F	0	0	0	0	0	0
G	0	0	0	0	0	0
H	0	0	0	0	0	0
I	0	0	0	0	0	0
J	100	0	0	0	0	0
K	0	0	0	0	0	0
L	0	0	0	0	0	0
M	0	0	0	0	0	0
N	0	0	0	0	0	0
O	100	0	0	0	0	100

First and foremost, it is important to note that items A and M belong to class C. Observing the results above, we can see that only two items, J and O, have demand not satisfied, whose quantity corresponds to the total amount requested in that period. What we are going to analyze is why the model selected these items to not produce.

Table 5.13: Total demand for each item, sorted in descending order, in Version 1

Item	**TotalDemand**
D	234669
G	148890
E	147732
H	104652
F	76932
C	61530
I	57728
N	30038
K	28726
L	22544
B	21772
J	20406
O	19758
A	6881
M	296

The table presented above contains the total demand of each item, sorted in descending order. In fact, items J and O are those with lower total demand, after the items of class C, what allow us to conclude that the model made the right products' selection.
The next step in this analysis consists of verifying the total occupation of the machines each period, to understand if there is available time to produce the shortage quantities.

Table 5.14: Percentage of machines' occupation in Version 1 (%)

	Period					
Machine	**1**	**2**	**3**	**4**	**5**	**6**
1	99.17	100	68.33	87.50	68.33	68.33
2	93.33	98.33	52.50	99.17	97.50	53.33
3	100	92.50	100	98.33	100	89.17
4	93.33	100	83.33	93.33	100	69.17

Item J can be produced in machines 2 and 4, and O can be produced in machine 3. The demand of J was not satisfied in the first period, and the demand of O was not satisfied in the first and last periods. Observing the occupation of both machines, in these periods, where these items can be produced, we can conclude that in this first period, machines 2 and 4 are with almost 93% of occupation. The machine 3 in the first period is totally occupied, but in the last period has available time to produce. We applied the Version 2 of the final model, with the penalization, but we could not obtain better results in terms of backlogging for this instance. As we can see in the table 5.15, the amount of demand not satisfied is small, and its sales revenue would not compensate the costs incurred for its production.

Table 5.15: Total demand, D_{pt}, and backlogging, b_{pt}, of each item in each period

		Period					
	Item	**1**	**2**	**3**	**4**	**5**	**6**
Total Demand	A	1137	1109	1165	1171	1156	1143
Backlogging	A	0	0	0	0	0	0
Total Demand	B	3692	3640	3594	3660	3610	3576
Backlogging	B	0	0	0	0	0	0
Total Demand	C	10202	10296	10266	10434	10158	10174
Backlogging	C	0	0	0	0	0	0
Total Demand	D	39161	38891	38958	38981	39324	39354
Backlogging	D	0	0	0	0	0	0
Total Demand	E	24587	24703	24411	24467	24845	24719
Backlogging	E	0	0	0	0	0	0
Total Demand	F	12816	12886	12748	12760	12886	12836
Backlogging	F	0	0	0	0	0	0
Total Demand	G	24878	24776	24648	24926	24746	24916
Backlogging	G	0	0	0	0	0	0
Total Demand	H	17360	17700	17294	17520	17388	17390
Backlogging	H	0	0	0	0	0	0
Total Demand	I	9454	9596	9708	9554	9696	9720
Backlogging	I	0	0	0	0	0	0
Total Demand	J	3288	3438	3368	3282	3448	3582
Backlogging	J	**3288**	0	0	0	0	0
Total Demand	K	4778	4730	4772	4728	4912	4806
Backlogging	K	0	0	0	0	0	0
Total Demand	L	0	4426	4518	4612	4500	4488
Backlogging	L	0	0	0	0	0	0
Total Demand	M	75	75	82	0	64	0
Backlogging	M	0	0	0	0	0	0
Total Demand	N	6112	6098	0	5854	5892	6082
Backlogging	N	0	0	0	0	0	0
Total Demand	O	3310	3202	3296	3356	3296	3298
Backlogging	O	**3310**	0	0	0	0	**3298**

Chapter 6

Conclusions and Future Work

The goal of this project was the development of a decision support tool, able to assist the Planning and Logistic Department of an industry producing by means of an extrusion process in building optimal production plans.
The first step of this work was the detailed analysis of the constraints of the extrusion process. Even though exist a wide range of lot-sizing approaches, it was a challenge to model this problem, since to the best of our knowledge its production and sales processes are different from what has been studied previously.
As the computational results show, the developed model fits the requirements imposed by the company. All the results have been analyzed in detail and discussed with the coordinators of this project, allowing progressive adjustments until reaching the final model. The possibility of adding an extra penalization to the backlog cost provides a good way for the user to better understand whether it is viable or not, in terms of profit and costs, to produce some quantity that would be delayed in the model without penalization. As has been pointed out, this is a decision support tool; thus, the user can analyze the results with and without the penalization and choose the most convenient production plan at the moment.
The model answers to the requests, but being an extension of the *Capacitated Lot-sizing Problem* (CLSP), it also belongs to the NP-hard complexity class. Since we used even more binary variables than CLSP, the difficulty to solve has strongly increased, which led to added difficulty in solving the problem. The MIP solver [Gurobi Optimization, 2012] was able to solve small instances to optimality, with lower GAP values. However, with the growth of the instances, if the GAP values start to increase, the development of a heuristic specific to this problem would be a good further research.
To obtain even better results with the model developed in this project, an important improvement would be the implementation of a sales forecast, which, based on history, would compute a demand forecast for the planning horizon.

Bibliography

[Bitran and Yanasse, 1982] Bitran, G. R. and Yanasse, H. H. (1982). Computational complexity of the capacitated lot size problem. *Management Science*, 28(10):1174–1186.

[Bruggeman and Jahnke, 2000] Bruggeman, W. and Jahnke, H. (2000). The discrete lot-sizing and scheduling problem: complexity and modification for batch availability. *European Journal of Operational Research*, 124(3):511–528.

[Caserta and Voß, 2013] Caserta, M. and Voß, S. (2013). A mip-based framework and its application on a lot sizing problem with setup carryover. *Journal of Heuristics*, 19(2):295–316.

[Damas, 1999] Damas, L. (1999). *SQL*. FCA - Editora de Informática LDA.

[Drexl and Haase, 1995] Drexl, A. and Haase, K. (1995). Proportional lotsizing and scheduling. *International Journal of Production Economics*, 40(1):73–87.

[Drexl, 1997] Drexl, K. (1997). Lot sizing and scheduling-survey and extensions. *European Journal of Operational Research*, 99(2):221–235.

[Gurobi Optimization, 2012] Gurobi Optimization (Inc (2012)). Gurobi optimizer reference manual, version 5.0. http://www.gurobi.com.

[Haase, 1994] Haase, D. (1994). *Lotsizing and scheduling for production planning*. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag.

[Harris, 1913] Harris, F. (1913). How many parts to make at once. *The Magazine of Management*, 10(2):135–136.

[Jordan and Drexl, 1996] Jordan, C. and Drexl, A. (1996). Discrete lotsizing and scheduling by batch sequencing. *Management Science*, 44(5):698–713.

[Karimi et al., 2003] Karimi, B., Fatemi Ghomi, S. M. T., and Wilson, J. M. (2003). The capacitated lot sizing problems: a review of models and algorithms. *Omega*, 31(5):365–378.

[Karmarkar et al., 1987] Karmarkar, U. S., Kekre, S., and Kekre, S. (1987). The dynamic lot-sizing problem with startup and reservation costs. *Operations Research*, 35(3):389–398.

[Kock, 1998] Kock, R. (1998). *The 80/20 Principle - The Secret of Achieving More with Less*. Nicholas Brealey Publishing.

[Langtangen, 2009] Langtangen, H. P. (2009). *A Primer on Scientific Programming with Python*. Texts in Computational Science and Engineering. Springer-Verlag.

[Mood and Graybill, 1974] Mood, A. M. and Graybill, F. A., B. D. C. (1974). *Introduction to the Teory of Statistics*. McGraw-Hill, 3rd edition edition.

[Pochet and Wolsey, 2006] Pochet, Y. and Wolsey, L. A. (2006). *Production Planning by Mixed Integer Programming*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag.

[Python Software Foundation, 2014a] Python Software Foundation (2014a). Python documentation. https://docs.python.org.

[Python Software Foundation, 2014b] Python Software Foundation (2014b). Python language reference, version 2.7.9. http://www.python.org.

[Quadt, 2004] Quadt, D. (2004). *Lot-Sizing and Scheduling for Flexible Flow Lines*. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag.

[Sox and Gao, 1999] Sox, C. R. and Gao, Y. (1999). The capacited lot sizing problem with setup carry-over. *IIE Transactions*, 31(2):173–181.

[Suerie and Stadler, 2003] Suerie, C. and Stadler, H. (2003). The capacitated lot-sizing problem with linked lot sizes. *Journal Management Science*, 49(8):1039–1054.

[Wagner and Whitin, 1958] Wagner, H. M. and Whitin, T. M. (1958). Dynamic version of the economical lot size model. *Journal Management Science*, 50(12):1770–1774.

Appendix A

Demand Matrices Generation

Fersil provided a sales history for each item. With this information, we generated the two demand matrices needed for the model: the customer's demand and the demand for inventory replenishment. It follows the description of the complete process.

When we analyzed the sales history, we verified that many items are not sold every week. This was the start point to build the demand generator. The first step consists in the computation, for each item, of the probability of having demand in each period. Based on this information, for each product, we randomly selected the number of periods of the planning horizon with positive sales. Finally, the demand values are generated for these periods. The remaining periods have no sales.

The distribution of sales data provided by the company is not known. Our goal was the generation of random values with the mean m and the variance υ of this sample. As the data are non-negative and have high variance, we chose to use a log-normal distribution.

Let us consider m as the mean and υ as the variance of a log-normal variable, and μ and σ as the parameters of a log-normal distribution. Then, m and υ can be written as the following function of μ and σ ([Mood and Graybill, 1974]):

$$m = e^{(\mu+\sigma^2/2)} \tag{A.1a}$$

$$\upsilon = e^{(2\mu+\sigma^2)}(e^{\sigma^2} - 1) \tag{A.1b}$$

A log-normal distribution with mean m and variance υ has the following parameters:

$$\mu = ln(\frac{m}{\sqrt{1+\frac{\upsilon}{m^2}}}) \tag{A.2a}$$

$$\sigma = \sqrt{ln(1+\frac{\upsilon}{m^2})} \tag{A.2b}$$

In our case, we considered m and υ as the mean and variance of data in the sales history, which is a non-logarithmized sample. We fitted this sample with a log-normal distribution. Thus, the demand values are generated following a log-normal distribution with parameters μ and σ.

The pseudo code of the developed demand generator is presented below.

Figure A.1: Demand matrices generator

```
DMGenerator()
    for p ∈ 𝒩:
        compute the probability of having sales for each period
    for t ∈ 𝒯:
        if p ∈ ClassA or p ∈ ClassB:
            O_pt = 0
            if t is a period with positive sales:
                I_pt follows a log-normal(μ, σ)
        if p ∈ ClassC:
            I_pt = 0
            if t is a period with positive sales:
                O_pt follows a log-normal(μ, σ)
    return I, O
```

Appendix B

The Branch-and-Bound Algorithm

The branch-and-bound is the optimization algorithm present on Gurobi and on others black-box solvers to mixed integer programming (MIP) problems. The main idea behind this algorithm is the relaxation of the original MIP, replacing the integer variables by continuous variables restricted by their limits. The resultant problem is called the *linear-programming relaxation of the original MIP* and is easier to solve.
Looking for the simplest lot sizing model presented in this dissertation, the *multi-item single-level capacitated lot-sizing problem*, the branch-and-bound algorithm replaces the binary variable y_{pt} by a continuous variable restricted by $0 \leq y_{pt} \leq 1$. This is called *branching variable* and this is the branching part of the algorithm.
After that, two situations may occur. If the obtained result with the relaxation satisfies all the integrality constraints, even these are not imposed on the model, then this is the optimal solution of the original MIP and the process stops. However, this is a rare case. In the second and more usually situation, the binary variables take fractional values and therefore they do not represent an optimal solution of the problem. In this last case, the original MIP is split into two sub-MIPs. Looking to the example presented above, and denoting the original problem by P_0, this is divided into two problems: P_1, where $y_{pt} \leq 1$ and P_2, where $y_{pt} \geq 0$. It can be proved that if we can compute optimal solutions of both sub-problems, the best of those is an optimal solution of P_0. The next step consists in the application of the relaxation to both sub-MIPs and, if necessary, in the selection of branching variables.
Considering P_0 as the root node, and each sub-problem as a node, it is easy to see that this algorithm builds a *search tree*, as represented in the figure below.

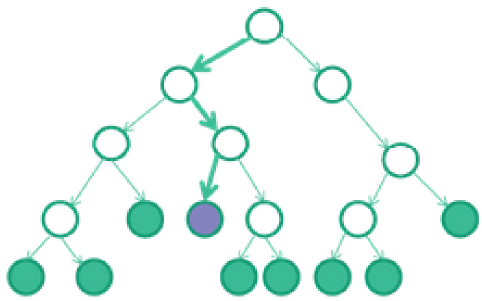


Figure B.1: Search tree of the branch-and-bound algorithm

Until now we present the branching part of the algorithm. As the name suggests, next we explain the bounding part, which is split into *upper bounding* and *lower bounding* part.
As seen before, when we achieve a node, if the resultant solution satisfies all the integrality restrictions, this node transforms into a leaf and no more splits can be done. The solution found in this node is a feasible solution of the original MIP. The next step consists in determining if this is or not the best solution found until the moment. To do this, at each iteration, the algorithm defines the *incumbent solution*. This is the best solution found in all the leaves of the search tree until the present iteration. When we find a new feasible solution of the problem, if this leads to a better objective function value

than the present incumbent solution, then it is done an update and this last value will occupy the incumbent solution position. If we suppose that we are on a minimizing problem, the incumbent solution consists in an upper bound of the problem, since we know that any other solution above this value that would be found in the search tree will be considered. This is the *upper bounding part*. In addition, in each step of the algorithm, a lower bound is also found. This is achieved by the lower value of all the solutions of all the leaves. This is the *lower bounding part*.
Finally, we can compute a measure of solution quality called GAP. This is given by the next expression.

$$GAP = \frac{U - L}{U} \times 100\% \tag{B.1}$$

where U and L are the best upper bound and lower bound, respectively, found in the whole process. If this value is equal to zero, the solution achieved is the optimal solution of the original problem. The further away from zero this value is, the worst quality solution we have. The GAP measures the deviation from optimality of the best feasible solution found by the algorithm.

Appendix C

Results of All the Classes of Instances

In this appendix are presented all the results in terms of runtime average and GAP average of all the instances, grouped by class.

Table C.1: Runtime for the two versions of the model applied on the instances of the class A (*seconds*)

Class A		
Instance	**Version 1**	**Version 2**
1	0.059	0.135
2	0.038	0.035
3	0.024	0.035
4	0.033	0.051
5	0.026	0.057
6	0.073	0.086
7	0.075	0.146
8	0.061	0.084
9	0.054	0.107
10	0.038	0.062

Table C.2: GAP for the two versions of the model applied on the instances of the class A (%)

Class A		
Instance	**Version 1**	**Version 2**
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0.0079
10	0	0

Table C.3: Runtime for the two versions of the model applied on the instances of the class B (*seconds*)

Class B		
Instance	**Version 1**	**Version 2**
1	0.253	0.299
2	0.6	0.927
3	0.235	0.287
4	0.179	0.429
5	0.106	0.323
6	0.834	1.32
7	0.175	0.304
8	0.226	0.42
9	0.356	1.05
10	0.218	0.369

Table C.4: GAP for the two versions of the model applied on the instances of the class B (%)

Class B		
Instance	**Version 1**	**Version 2**
1	0	0
2	0.0069	0.0074
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0.0053
8	0	0
9	0.0034	0.0069
10	0	0

Table C.5: Runtime for the two versions of the model applied on the instances of the class C (*seconds*)

Class C		
Instance	**Version** 1	**Version 2**
1	103	619
2	39.2	106
3	61.4	475
4	72.3	270
5	7.02	53.7
6	1.58	11.8
7	4.15	62.5
8	3.39	74
9	3.58	9.81
10	10.2	63.1

Table C.6: GAP for the two versions of the model applied on the instances of the class C (%)

Class C		
Instance	**Version 1**	**Version 2**
1	0.01	0.01
2	0.01	0.0099
3	0.01	0.01
4	0.01	0.0099
5	0.009	0.01
6	0.008	0
7	0.0099	0.01
8	0.0096	0.01
9	0	0
10	0.0091	0.01

Table C.7: Runtime for the two versions of the model applied on the instances of the class D (*seconds*)

Class D		
Instance	**Version 1**	**Version 2**
1	528	3600
2	1137	3600
3	725	3600
4	3600	3600
5	768	3600
6	976	3600
7	3600	3600
8	746	3600
9	1048	3600
10	3600	3600

Table C.8: GAP for the two versions of the model applied on the instances of the class D (%)

Class D		
Instance	**Version 1**	**Version 2**
1	0.01	0.283
2	0.01	0.637
3	0.01	0.262
4	0.08	0.449
5	0.01	0.231
6	0.01	0.239
7	0.189	0.456
8	0.01	0.232
9	0.01	0.321
10	0.0697	0.359

Appendix D

Detailed Data and Results for Instances of Subsection 5.3.1

D.1 The First Example

D.1.1 Data

Table D.1: Setup cost, S_{pm}

Item/Machine	**1**	**2**	**3**
A	0	101	128
B	196	0	0
C	199	0	0

Table D.2: Setup time, P_{pm}

Item/Machine	**1**	**2**	**3**
A	0	0.48	0.67
B	0.63	0	0
C	0.7	0	0

Table D.3: Performance loss factor (F_p), unit inventory cost (H_p), percentage of lost demand (ρ_p) and unit price (R_p)

Item	F_p	H_p	ρ_p	R_p
A	0.00155	0.0111	0.00398	0.591
B	0.0169	0.0327	0.0434	1.735
C	0.00694	0.04	0.0178	2.117

Table D.4: Time limit to setup, L_t

Period	L_t
1	5
2	5
3	5
4	5
5	5
6	5

Table D.5: Unit production time, U_{pm}

Item/Machine	**1**	**2**	**3**
A	0	0.00187	0.00211
B	0.00243	0	0
C	0.00311	0	0

Table D.6: Machines' capacity, C_{mt}

Machine/Period	**1**	**2**	**3**	**4**	**5**	**6**
1	15	15	15	15	15	15
2	15	15	15	15	15	15
3	15	15	15	15	15	15

D.1.2 Detailed Results

Table D.7: Machines' occupation

Version 1						
	Period					
Machine	**1**	**2**	**3**	**4**	**5**	**6**
1	2	12	3	3	0	3
2	2	0	0	0	0	0
3	0	0	0	0	0	0

Version 2						
	Period					
Machine	**1**	**2**	**3**	**4**	**5**	**6**
1	11	3	3	3	0	3
2	2	0	0	0	0	0
3	0	0	0	0	0	0

Table D.8: Production detail of Version 1 (first part)

Version 1								
			Period					
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	A	1	0	0	0	0	0	0
y	A	1	0	0	0	0	0	0
z	A	1	0	0	0	0	0	0
x	A	2	563	0	0	0	0	0
y	A	2	1	1	0	0	0	0
z	A	2	0	1	0	0	0	0
x	A	3	0	0	0	0	0	0
y	A	3	0	0	0	0	0	0
z	A	3	0	0	0	0	0	0
D	A		563	0	0	0	0	0
O	A		0	0	0	0	0	0
I	A		563	0	0	0	0	0
b	A		0	0	0	0	0	0
i	A		0	0	0	0	0	0
x	B	1	0	2428	1158	1252	0	1198
y	B	1	0	1	1	1	1	1
z	B	1	0	0	1	1	1	1
x	B	2	0	0	0	0	0	0
y	B	2	0	0	0	0	0	0
z	B	2	0	0	0	0	0	0
x	B	3	0	0	0	0	0	0
y	B	3	0	0	0	0	0	0
z	B	3	0	0	0	0	0	0
D	B		1241	1241	1158	1252	0	1198
O	B		0	0	0	0	0	0
I	B		1241	1241	1158	1252	0	1198
b	B		1241	0	0	0	0	0
i	B		0	0	0	0	0	0

Table D.9: Production detail of Version 1 (second part)

Version 1								
Period								
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	C	1	433	1652	0	0	0	0
y	C	1	1	1	0	0	0	0
z	C	1	0	1	0	0	0	0
x	C	2	0	0	0	0	0	0
y	C	2	0	0	0	0	0	0
z	C	2	0	0	0	0	0	0
x	C	3	0	0	0	0	0	0
y	C	3	0	0	0	0	0	0
z	C	3	0	0	0	0	0	0
D	C		433	395	418	426	413	411
O	C		0	0	0	0	0	0
I	C		433	395	418	426	413	411
b	C		0	0	0	0	0	411
i	C		0	1257	839	413	0	0

Table D.10: Production detail of Version 2 (first part)

Version 2								
Period								
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	A	1	0	0	0	0	0	0
y	A	1	0	0	0	0	0	0
z	A	1	0	0	0	0	0	0
x	A	2	563	0	0	0	0	0
y	A	2	1	1	0	0	0	0
z	A	2	0	1	0	0	0	0
x	A	3	0	0	0	0	0	0
y	A	3	0	0	0	0	0	0
z	A	3	0	0	0	0	0	0
D	A		563	0	0	0	0	0
O	A		0	0	0	0	0	0
I	A		563	0	0	0	0	0
b	A		0	0	0	0	0	0
i	A		0	0	0	0	0	0

Table D.11: Production detail of Version 2 (second part)

			Version 2					
			Period					
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	B	1	1241	1241	1158	1252	0	1198
y	B	1	1	1	1	1	1	1
z	B	1	0	1	1	1	1	1
x	B	2	0	0	0	0	0	0
y	B	2	0	0	0	0	0	0
z	B	2	0	0	0	0	0	0
x	B	3	0	0	0	0	0	0
y	B	3	0	0	0	0	0	0
z	B	3	0	0	0	0	0	0
D	B		1241	1241	1158	1252	0	1198
O	B		0	0	0	0	0	0
I	B		1241	1241	1158	1252	0	1198
b	B		0	0	0	0	0	0
i	B		0	0	0	0	0	0
x	C	1	2085	0	0	0	0	0
y	C	1	1	0	0	0	0	0
z	C	1	0	0	0	0	0	0
x	C	2	0	0	0	0	0	0
y	C	2	0	0	0	0	0	0
z	C	2	0	0	0	0	0	0
x	C	3	0	0	0	0	0	0
y	C	3	0	0	0	0	0	0
z	C	3	0	0	0	0	0	0
D	C		433	395	418	426	413	411
O	C		0	0	0	0	0	0
I	C		433	395	418	426	413	411
b	C		0	0	0	0	0	411
i	C		1652	1257	839	413	0	0

D.2 The Second Example

D.2.1 Data

Table D.12: Setup cost, S_{pm}

Item/Machine	1	2	3
D	199	0	0
E	0	277	290
F	0	265	377

Table D.13: Setup time, P_{pm}

Item/Machine	1	2	3
D	0.7	0	0
E	0	1.23	1.46
F	0	1.09	1.95

Table D.14: Performance loss factor (F_p), unit inventory cost (H_p), percentage of lost demand (ρ_p) and unit price (R_p)

Item	F_p	H_p	ρ_p	R_p
D	0.00694	0.04	0.0178	2.117
E	0.0842	0.00924	0.216	0.485
F	0.196	0.0117	0.502	0.616

Table D.15: Time limit to setup, L_t

Period	L_t
1	5
2	5
3	5
4	5
5	5
6	5

Table D.16: Unit production time, U_{pm}

Item/Machine	1	2	3
D	0.00311	0	0
E	0	0.00143	0.00182
F	0	0.00162	0.00212

Table D.17: Machines' capacity, C_{mt}

Machine/Period	1	2	3	4	5	6
1	15	15	15	15	15	15
2	15	15	15	15	15	15
3	15	15	15	15	15	15

D.2.2 Detailed Results

Table D.18: Machines' occupation

Version 1						
	Period					
Machine	**1**	**2**	**3**	**4**	**5**	**6**
1	2	1	1	1	1	1
2	15	15	15	15	15	15
3	10	9	9	9	9	9

Version 1						
	Period					
Machine	**1**	**2**	**3**	**4**	**5**	**6**
1	2	1	1	1	1	1
2	15	15	15	15	15	15
3	12	12	12	12	12	12

Table D.19: Production detail of Version 1

Version 1								
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	D	1	433	395	418	426	413	411
y	D	1	1	1	1	1	1	1
z	D	1	0	1	1	1	1	1
x	D	2	0	0	0	0	0	0
y	D	2	0	0	0	0	0	0
z	D	2	0	0	0	0	0	0
x	D	3	0	0	0	0	0	0
y	D	3	0	0	0	0	0	0
z	D	3	0	0	0	0	0	0
D	D		433	395	418	426	413	411
O	D		0	0	0	0	0	0
I	D		433	395	418	426	413	411
b	D		0	0	0	0	0	0
i	D		0	0	0	0	0	0
x	E	1	0	0	0	0	0	0
y	E	1	0	0	0	0	0	0
z	E	1	0	0	0	0	0	0
x	E	2	0	0	0	0	0	0
y	E	2	0	0	0	0	0	0
z	E	2	0	0	0	0	0	0
x	E	3	4956	4977	5136	5143	5089	5013
y	E	3	1	1	1	1	1	1
z	E	3	0	1	1	1	1	1
D	E		4956	4977	5136	5143	5089	5013
O	E		0	0	0	0	0	0
I	E		4956	4977	5136	5143	5089	5013
b	E		0	0	0	0	0	0
i	E		0	0	0	0	0	0
x	F	1	0	0	0	0	0	0
y	F	1	0	0	0	0	0	0
z	F	1	0	0	0	0	0	0
x	F	2	8572	9243	9243	9243	9243	9243
y	F	2	1	1	1	1	1	1
z	F	2	0	1	1	1	1	1
x	F	3	0	0	0	0	0	0
y	F	3	0	0	0	0	0	0
z	F	3	0	0	0	0	0	0
D	F		11822	11822	11655	11744	11764	11638
O	F		0	0	0	0	0	0
I	F		11822	11822	11655	11744	11764	11638
b	F		3250	4198	4502	4743	4883	4827
i	F		0	0	0	0	0	0

Table D.20: Production detail of Version 2

Version 2								
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	D	1	433	395	418	426	413	411
y	D	1	1	1	1	1	1	1
z	D	1	0	1	1	1	1	1
x	D	2	0	0	0	0	0	0
y	D	2	0	0	0	0	0	0
z	D	2	0	0	0	0	0	0
x	D	3	0	0	0	0	0	0
y	D	3	0	0	0	0	0	0
z	D	3	0	0	0	0	0	0
D	D		433	395	418	426	413	411
O	D		0	0	0	0	0	0
I	D		433	395	418	426	413	411
b	D		0	0	0	0	0	0
i	D		0	0	0	0	0	0
x	E	1	0	0	0	0	0	0
y	E	1	0	0	0	0	0	0
z	E	1	0	0	0	0	0	0
x	E	2	0	0	0	0	0	0
y	E	2	0	0	0	0	0	0
z	E	2	0	0	0	0	0	0
x	E	3	5760	6561	6561	6561	6561	6561
y	E	3	1	1	1	1	1	1
z	E	3	0	1	1	1	1	1
D	E		4956	4977	5136	5143	5089	5013
O	E		0	0	0	0	0	0
I	E		4956	4977	5136	5143	5089	5013
b	E		0	0	0	0	0	0
i	E		804	2388	3814	5232	6704	8252
x	F	1	0	0	0	0	0	0
y	F	1	0	0	0	0	0	0
z	F	1	0	0	0	0	0	0
x	F	2	8572	9243	9243	9243	9243	9243
y	F	2	1	1	1	1	1	1
z	F	2	0	1	1	1	1	1
x	F	3	0	0	0	0	0	0
y	F	3	0	0	0	0	0	0
z	F	3	0	0	0	0	0	0
D	F		11822	11822	11655	11744	11764	11638
O	F		0	0	0	0	0	0
I	F		11822	11822	11655	11744	11764	11638
b	F		3250	4198	4502	4743	4883	4827
i	F		0	0	0	0	0	0

Appendix E

Detailed Data of the Instance of the Subsection 5.3.3

E.1 Data

Table E.1: Setup cost, S_{pm}

Item/Machine	1	2	3	4
A	0	175	189	0
B	375	0	191	0
C	0	277	0	290
D	0	265	0	377
E	0	171	0	136
F	0	0	199	0
G	564	0	221	0
H	345	0	558	0
I	380	0	0	0
J	0	348	0	423
K	0	0	0	299
L	0	269	0	146
M	0	944	209	0
N	0	296	187	0
O	0	0	251	0

Table E.2: Setup time, P_{pm}

Item/Machine	1	2	3	4
A	0	0.98	0.84	0
B	1.17	0	0.84	0
C	0	1.23	0	1.46
D	0	1.09	0	1.95
E	0	0.72	0	0.75
F	0	0	0.86	0
G	1.95	0	0.96	0
H	1.11	0	2.17	0
I	1.18	0	0	0
J	0	1.51	0	2.5
K	0	0	0	1.44
L	0	1.25	0	0.9
M	0	5.33	0.95	0
N	0	1.55	0.81	0
O	0	0	1.19	0

Table E.3: Machines' capacity, C_{mt}

Machine/Period	1	2	3	4	5	6
1	120	120	120	120	120	120
2	120	120	120	120	120	120
3	120	120	120	120	120	120
4	120	120	120	120	120	120

Table E.4: Performance loss factor (F_p), unit inventory cost (H_p), percentage of lost demand (ρ_p) and unit price (R_p)

Item	F_p	H_p	ρ_p	R_p
A	0.0109	0.0140	0.0280	0.739
B	0.0346	0.0235	0.0886	1.248
C	0.0977	0.00924	0.25	0.485
D	0.373	0.0117	0.955	0.616
E	0.235	0.0147	0.601	0.777
F	0.122	0.0223	0.313	1.172
G	0.236	0.0269	0.606	1.427
H	0.166	0.0392	0.426	2.146
I	0.0917	0.0491	0.235	2.607
J	0.0324	0.0223	0.0831	1.167
K	0.0456	0.00941	0.117	0.498
L	0.0358	0.0211	0.092	1.13
M	0.00047	0.0951	0.00121	5.23
N	0.0477	0.0338	0.122	1.79
O	0.0314	0.0694	0.08	3.68

Table E.5: Unit production time, U_{pm}

Item/Machine	**1**	**2**	**3**	**4**
A	0	0.0026	0.00197	0
B	0.002	0	0.00271	0
C	0	0.00143	0	0.00182
D	0	0.00162	0	0.00212
E	0	0.00189	0	0.00261
F	0	0	0.00256	0
G	0.00229	0	0.00294	0
H	0.00282	0	0.00363	0
I	0.00325	0	0	0
J	0	0.00258	0	0.00367
K	0	0	0	0.00176
L	0	0.00268	0	0.00384
M	0	0.0121	0.00895	0
N	0	0.00438	0.00349	0
O	0	0	0.00704	0

Table E.6: Time limit to setup, L_t

Period	L
1	35
2	35
3	35
4	35
5	35
6	35

Table E.7: Indicative of the subset of machines of each item, Z_{pm}

Item/Machine	**1**	**2**	**3**	**4**
A	0	1	1	0
B	1	0	1	0
C	0	1	0	1
D	0	1	0	1
E	0	1	0	1
F	0	0	1	0
G	1	0	1	0
H	1	0	1	0
I	1	0	0	0
J	0	1	0	1
K	0	0	0	1
L	0	1	0	1
M	0	1	1	0
N	0	1	1	0
O	0	0	1	0

E.2 Detailed Results

Table E.8: Production detail of Version 1 (first part)

Version 1								
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	A	1	0	0	0	0	0	0
y	A	1	0	0	0	0	0	0
z	A	1	0	0	0	0	0	0
x	A	2	6881	0	0	0	0	0
y	A	2	1	0	0	0	0	0
z	A	2	0	0	0	0	0	0
x	A	3	0	0	0	0	0	0
y	A	3	0	0	0	0	0	0
z	A	3	0	0	0	0	0	0
x	A	4	0	0	0	0	0	0
y	A	4	0	0	0	0	0	0
z	A	4	0	0	0	0	0	0
D	A		1137	1109	1165	1171	1156	1143
O	A		1137	1109	1165	1171	1156	1143
I	A		0	0	0	0	0	0
b	A		0	0	0	0	0	0
i	A		5744	4635	3470	2299	1143	0

Table E.9: Production detail of Version 1 (second part)

			Version 1					
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	B	1	10926	0	0	10846	0	0
y	B	1	1	0	0	1	0	0
z	B	1	0	0	0	0	0	0
x	B	2	0	0	0	0	0	0
y	B	2	0	0	0	0	0	0
z	B	2	0	0	0	0	0	0
x	B	3	0	0	0	0	0	0
y	B	3	0	0	0	0	0	0
z	B	3	0	0	0	0	0	0
x	B	4	0	0	0	0	0	0
y	B	4	0	0	0	0	0	0
z	B	4	0	0	0	0	0	0
D	B		3692	3640	3594	3660	3610	3576
O	B		0	0	0	0	0	0
I	B		3692	3640	3594	3660	3610	3576
b	B		0	0	0	0	0	0
i	B		7234	3594	0	7186	3576	0
x	C	1	0	0	0	0	0	0
y	C	1	0	0	0	0	0	0
z	C	1	0	0	0	0	0	0
x	C	2	0	0	0	0	0	0
y	C	2	0	0	0	0	0	0
z	C	2	0	0	0	0	0	0
x	C	3	0	0	0	0	0	0
y	C	3	0	0	0	0	0	0
z	C	3	0	0	0	0	0	0
x	C	4	10558	20206	0	10919	19847	0
y	C	4	1	1	0	1	1	0
z	C	4	0	1	0	0	1	0
D	C		10202	10296	10266	10434	10158	10174
O	C		0	0	0	0	0	0
I	C		10202	10296	10266	10434	10158	10174
b	C		0	0	0	0	0	0
i	C		356	10266	0	485	10174	0

Table E.10: Production detail of Version 1 (third part)

Version 1								
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	D	1	0	0	0	0	0	0
y	D	1	0	0	0	0	0	0
z	D	1	0	0	0	0	0	0
x	D	2	39161	38891	38958	38981	39324	39354
y	D	2	1	1	1	1	1	1
z	D	2	0	0	1	1	0	1
x	D	3	0	0	0	0	0	0
y	D	3	0	0	0	0	0	0
z	D	3	0	0	0	0	0	0
x	D	4	0	0	0	0	0	0
y	D	4	0	0	0	0	0	0
z	D	4	0	0	0	0	0	0
D	D		39161	38891	38958	38981	39324	39354
O	D		0	0	0	0	0	0
I	D		39161	38891	38958	38981	39324	39354
b	D		0	0	0	0	0	0
i	D		0	0	0	0	0	0
x	E	1	0	0	0	0	0	0
y	E	1	0	0	0	0	0	0
z	E	1	0	0	0	0	0	0
x	E	2	0	0	0	0	0	0
y	E	2	0	0	0	0	0	0
z	E	2	0	0	0	0	0	0
x	E	3	0	0	0	0	0	0
y	E	3	0	0	0	0	0	0
z	E	3	0	0	0	0	0	0
x	E	4	24587	24703	24411	24467	24845	24719
y	E	4	1	1	1	1	1	1
z	E	4	0	0	0	1	0	0
D	E		24587	24703	24411	24467	24845	24719
O	E		0	0	0	0	0	0
I	E		24587	24703	24411	24467	24845	24719
b	E		0	0	0	0	0	0
i	E		0	0	0	0	0	0

Table E.11: Production detail of Version 1 (fourth part)

			Version 1					
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	F	1	0	0	0	0	0	0
y	F	1	0	0	0	0	0	0
z	F	1	0	0	0	0	0	0
x	F	2	0	0	0	0	0	0
y	F	2	0	0	0	0	0	0
z	F	2	0	0	0	0	0	0
x	F	3	12816	25634	0	17107	8539	12836
y	F	3	1	1	0	1	1	1
z	F	3	0	1	0	0	1	0
x	F	4	0	0	0	0	0	0
y	F	4	0	0	0	0	0	0
z	F	4	0	0	0	0	0	0
D	F		12816	12886	12748	12760	12886	12836
O	F		0	0	0	0	0	0
I	F		12816	12886	12748	12760	12886	12836
b	F		0	0	0	0	0	0
i	F		0	12748	0	4347	0	0
x	G	1	0	21843	0	0	0	0
y	G	1	0	1	0	0	0	0
z	G	1	0	0	0	0	0	0
x	G	2	0	0	0	0	0	0
y	G	2	0	0	0	0	0	0
z	G	2	0	0	0	0	0	0
x	G	3	27811	0	24648	24926	24746	24916
y	G	3	1	0	1	1	1	1
z	G	3	0	0	0	1	0	1
x	G	4	0	0	0	0	0	0
y	G	4	0	0	0	0	0	0
z	G	4	0	0	0	0	0	0
D	G		24878	24776	24648	24926	24746	24916
O	G		0	0	0	0	0	0
I	G		24878	24776	24648	24926	24746	24916
b	G		0	0	0	0	0	0
i	G		2933	0	0	0	0	0

Table E.12: Production detail of Version 1 (fifth part)

			Version 1					
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	H	1	22433	12627	17294	17520	17388	17390
y	H	1	1	1	1	1	1	1
z	H	1	0	0	1	0	1	0
x	H	2	0	0	0	0	0	0
y	H	2	0	0	0	0	0	0
z	H	2	0	0	0	0	0	0
x	H	3	0	0	0	0	0	0
y	H	3	0	0	0	0	0	0
z	H	3	0	0	0	0	0	0
x	H	4	0	0	0	0	0	0
y	H	4	0	0	0	0	0	0
z	H	4	0	0	0	0	0	0
D	H		17360	17700	17294	17520	17388	17390
O	H		0	0	0	0	0	0
I	H		17360	17700	17294	17520	17388	17390
b	H		0	0	0	0	0	0
i	H		5073	0	0	0	0	0
x	I	1	9454	9596	9708	9554	9696	9720
y	I	1	1	1	1	1	1	1
z	I	1	0	1	0	1	0	1
x	I	2	0	0	0	0	0	0
y	I	2	0	0	0	0	0	0
z	I	2	0	0	0	0	0	0
x	I	3	0	0	0	0	0	0
y	I	3	0	0	0	0	0	0
z	I	3	0	0	0	0	0	0
x	I	4	0	0	0	0	0	0
y	I	4	0	0	0	0	0	0
z	I	4	0	0	0	0	0	0
D	I		9454	9596	9708	9554	9696	9720
O	I		0	0	0	0	0	0
I	I		9454	9596	9708	9554	9696	9720
b	I		0	0	0	0	0	0
i	I		0	0	0	0	0	0

Table E.13: Production detail of Version 1 (sixth part)

					Version 1			
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	J	1	0	0	0	0	0	0
y	J	1	0	0	0	0	0	0
z	J	1	0	0	0	0	0	0
x	J	2	0	9821	0	10312	0	0
y	J	2	0	1	0	1	0	0
z	J	2	0	0	0	0	0	0
x	J	3	0	0	0	0	0	0
y	J	3	0	0	0	0	0	0
z	J	3	0	0	0	0	0	0
x	J	4	0	0	0	0	0	0
y	J	4	0	0	0	0	0	0
z	J	4	0	0	0	0	0	0
D	J		3288	3438	3368	3282	3448	3582
O	J		0	0	0	0	0	0
I	J		3288	3438	3368	3282	3448	3582
b	J		3288	0	0	0	0	0
i	J		0	3368	0	7030	3582	0
x	K	1	0	0	0	0	0	0
y	K	1	0	0	0	0	0	0
z	K	1	0	0	0	0	0	0
x	K	2	0	0	0	0	0	0
y	K	2	0	0	0	0	0	0
z	K	2	0	0	0	0	0	0
x	K	3	0	0	0	0	0	0
y	K	3	0	0	0	0	0	0
z	K	3	0	0	0	0	0	0
x	K	4	14280	0	0	14446	0	0
y	K	4	1	0	0	1	0	0
z	K	4	0	0	0	0	0	0
D	K		4778	4730	4772	4728	4912	4806
O	K		0	0	0	0	0	0
I	K		4778	4730	4772	4728	4912	4806
b	K		0	0	0	0	0	0
i	K		9502	4772	0	9718	4806	0

Table E.14: Production detail of Version 1 (seventh part)

Variable	Item	Machine	1	2	3	4	5	6
Version 1								
x	L	1	0	0	0	0	0	0
y	L	1	0	0	0	0	0	0
z	L	1	0	0	0	0	0	0
x	L	2	0	0	0	0	0	0
y	L	2	0	0	0	0	0	0
z	L	2	0	0	0	0	0	0
x	L	3	0	0	0	0	0	0
y	L	3	0	0	0	0	0	0
z	L	3	0	0	0	0	0	0
x	L	4	0	4426	9130	0	4500	4488
y	L	4	0	1	1	0	1	1
z	L	4	0	0	1	0	0	1
D	L		0	4426	4518	4612	4500	4488
O	L		0	0	0	0	0	0
I	L		0	4426	4518	4612	4500	4488
b	L		0	0	0	0	0	0
i	L		0	0	4612	0	0	0
x	M	1	0	0	0	0	0	0
y	M	1	0	0	0	0	0	0
z	M	1	0	0	0	0	0	0
x	M	2	0	0	0	0	0	0
y	M	2	0	0	0	0	0	0
z	M	2	0	0	0	0	0	0
x	M	3	296	0	0	0	0	0
y	M	3	1	0	0	0	0	0
z	M	3	0	0	0	0	0	0
x	M	4	0	0	0	0	0	0
y	M	4	0	0	0	0	0	0
z	M	4	0	0	0	0	0	0
D	M		75	75	82	0	64	0
O	M		75	75	82	0	64	0
I	M		0	0	0	0	0	0
b	M		0	0	0	0	0	0
i	M		221	146	64	64	0	0

Table E.15: Production detail of Version 1 (eighth part)

Version 1								
Variable	**Item**	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**
x	N	1	0	0	0	0	0	0
y	N	1	0	0	0	0	0	0
z	N	1	0	0	0	0	0	0
x	N	2	6112	6098	0	5854	11974	0
y	N	2	1	1	0	1	1	0
z	N	2	0	1	0	0	1	0
x	N	3	0	0	0	0	0	0
y	N	3	0	0	0	0	0	0
z	N	3	0	0	0	0	0	0
x	N	4	0	0	0	0	0	0
y	N	4	0	0	0	0	0	0
z	N	4	0	0	0	0	0	0
D	N		6112	6098	0	5854	5892	6082
O	N		0	0	0	0	0	0
I	N		6112	6098	0	5854	5892	6082
b	N		0	0	0	0	0	0
i	N		0	0	0	0	6082	0
x	O	1	0	0	0	0	0	0
y	O	1	0	0	0	0	0	0
z	O	1	0	0	0	0	0	0
x	O	2	0	0	0	0	0	0
y	O	2	0	0	0	0	0	0
z	O	2	0	0	0	0	0	0
x	O	3	0	6294	6603	0	3296	0
y	O	3	0	1	1	0	1	0
z	O	3	0	0	1	0	0	0
x	O	4	0	0	0	0	0	0
y	O	4	0	0	0	0	0	0
z	O	4	0	0	0	0	0	0
D	O		3310	3202	3296	3356	3296	3298
O	O		0	0	0	0	0	0
I	O		3310	3202	3296	3356	3296	3298
b	O		3310	0	0	0	0	3298
i	O		0	49	3356	0	0	0