A Decision Support System for the Production Scheduling of a Printing Plant

Horácio Magalhães Neri

Master's Dissertation

Supervisor: Bernardo Sobrinho Simões de Almada-Lobo
Co-supervisor: Luís Gonçalo Rodrigues Reis Figueira

U. PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Industrial e Gestão

2015-07-01

Abstract

Pressured by an increasingly competitive global market, production companies are more concerned than ever with operational optimization. One of the typical tasks performed in this type of companies is the production scheduling. Even for simple environments, scheduling tasks are known to be complex combinatorial problems. In a printing plant, the problem is especially complex due to its large flexibility. Contrarily to common scheduling problems, in a printing plant, the number of operations required to complete a job is not pre-defined.

Motivated by a real-world case of a Portuguese metal-sheet packaging company, a decision support system is developed to support the production scheduling of the printing plant. The decision support system is not just an optimization tool. It is designed to increase the efficiency of upstream and downstream processes by providing longer-term information, and also to allow making what-if analysis that can influence tactical-level decisions.

In this dissertation, the problem is decomposed into four subproblems to decrease the computational time required to find a solution. Firstly, the bottleneck stage is tackled separately from the other stages. Secondly, for each of the resulting subproblems, the machine assignment is done before the sequencing process. Every subproblem is solved with the aid of mixed-integer linear programming models.

Additional analysis of the current planning process of the company is also presented. This analysis aims to find improvement opportunities and also to address the lack of stored data related to some important parameters of the decision support system. Different concepts and techniques are used during this analysis, such as multiple linear regressions and the overall equipment efficiency.

At the end of this dissertation project, the validation of the results had not been completed. However, the solutions analyzed are motivating, since they show that the requirements of the problem are being met and that the information provided by the decision support system meets the company's needs.

Resumo

Pressionadas por um mercado global cada vez mais competitivo, as empresas industriais, mais do que nunca, procuram soluções de optimização operacional. Uma das tarefas típicas neste tipo de empresas é o escalonamento da produção. Mesmo em ambientes de pouca complexidade, o escalonamento é uma tarefa combinatória complicada. Numa fábrica de impressão, o problema é ainda mais complexo devido à sua grande flexibilidade. Ao contrário dos problemas de escalonamento comuns, numa fábrica de impressão, o número de operações necessárias para completar um trabalho não está previamente definido.

Motivado por um caso real de uma empresa de embalagens metálicas Portuguesa, um sistema de apoio à decisão é desenvolvido para ajudar o escalonamento da produção na fábrica de impressão. O sistema de apoio à decisão não será só uma ferramenta de optimização, mas tambem deverá permitir fazer análises de cenários para servir de apoio a decisões táticas.

Nesta dissertação, o problema é decomposto em quatro subproblemas para diminuir o esforço computacional necessário para chegar a uma solução. Primeiro, o gargalo do sistema é abordado em separado dos outros estágios. Segundo, para cada um dos subprblemas resultantes, a fase de alocação das tarefas às linhas de produção é processada antes da fase de sequenciamento. Todos os subproblemas são resolvidos através de modelos de programação linear inteira mista.

Também são apresentadas análises feitas ao atual processo de planeamento para encontrar possíveis oportunidades de melhoria, e para colmatar a falta de dados relacionados com parâmetros importantes para o sistema de apoio à decisão. São utilizados diferentes conceitos e técnicas durante estas análises, como por exemplo, regressões lineares múltiplas e a eficácia global dos equipamentos.

Na data de conclusão desta dissertação, a validação de resultados ainda não tinha sido concluída. Contudo, as soluções analisadas até ao momento são motivadoras, visto que mostram que os requisitos do problema estão a ser cumpridos e que a informação disponibilizada pelo sistema de apoio à decisão vai de encontro às necessidades da empresa.

Acknowledgments

First, I would like to thank my supervisor, Prof. Bernardo Almada-Lobo for the opportunity to work with him in such a challenging project. His expertise and experience were crucial to define the direction of the project and to help me grow, not only as a professional, but also as a person.

I would also like to thank Prof. Gonçalo Figueira, who closely followed the progress of the project. He helped me solve the many obstacles that appeared along the way. More than a co-supervisor, he has been a friend. As the project team is concerned, I would also like to address a word to Prof. Luís Guimarães, whose knowledge concerning operations research is impressive and a great motivation to make me want to improve.

I am also grateful to the team of Colep who gave us the access to the data needed to continue the development of this project and have always shown us confidence in the value of our work.

My colleagues from CEGI, who I met in the first day of this project, and soon became friends, also played a major role on motivating me and offering me wise advice to pursue my goals.

Because life is not all about work, I would like to thank my friends from FEUP, and my friends from high school for supporting me when I was in need and for not allowing me to spend the weekends sleeping at home.

Finally, a special "thank you" to my family, mainly my parents and sister. Mãe, Pai, Megui, thank you for always being there when I needed and for helping me getting where I am today.

Contents

Acronyms and Symbols

AIC	Akaike Information Criterion
CMYK	Cyan, Magenta, Yellow and Black
CO	Combination of Orders
CP	Constraint Programming
DSS	Decision Support System
EDD	Earliest Due Date first
FCFS	First-Come First-Served
FJSSP	Flexible Job Shop Scheduling Problem
GA	Genetic Algorithm
ILS	Iterative Local Search
KPI	Key Performance Indicators
LPT	Longest Processing Time first
MILP	Mixed-Integer Linear Programming
MP	Mathematical Programming
MS	Minimum Slack first
MTO	Make to Order
MTS	Make to Stock
OEE	Overall Equipment Efficiency
PAM	Partitioning Around Medoids
SBH	Shifting Bottleneck Heuristic
SPT	Shortest Processing Time first
TS	Tabu-Search
VNS	Variable Neighborhood Search
WIP	Work in Process

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The increasing complexity of companies and supply chains enlarges the number and the range of decisions to be made. Executives of top tier companies can not decide the future of a company alone without delegating short-term and mid-term planning to other employees. Increasing the number of people involved in the decision-making process creates issues related with the coherence of the decisions since each individual has different motivations that in some cases may neglect the bigger picture.

Hierarchical Production Planning (HPP), decomposes the overall planning problem into planning modules. Long-term, mid-term and short-term decisions are considered with different levels of detail to balance optimality and practicability. The operational level is the lowest level of decisions considered in the HPP. Daily decisions that have a short-term impact on the company are contemplated here. Ordering materials, production scheduling, distribution scheduling, and short-term sales planning are examples of operational decisions.

Despite their short-term impact, operational-level decisions play an important role on the efficiency and effectiveness of the company. Operational optimization is currently becoming more popular, as managers become aware of its potential value. In production environments, production scheduling is a typical short-term planning task that may have a large influence on the throughput of the plant and on customer loyalty by increasing the proportion of orders that meet their due dates.

While forecasting and simulation models are used in predictive analytics, in prescriptive analytics, where the goal is to support the decision making by suggesting possible solutions, optimization models are applied. Optimization models are often incorporated in decision support systems, which are interactive information systems designed to receive, process and return information in order to aid the decision-making process.

1.2 Scope of the Project

The study focuses on a Portuguese company that produces metal packages for different types of products. Metal packages are used by a wide range of sectors from food industry to hygiene products. This variety of sectors increases the spectrum of requirements of customers, demanding high levels of customization. As in other sectors, globalization opened new opportunities for growth, but also created new challenges to overcome as the number and economical power of competitors increased. Companies that can not compete in price have to find other competitive advantages such as customization or shorter lead times. To increase their flexibility the customers have been ordering more frequently but in less quantities than what was observed in the past, requiring short lead times to be able to fulfill their demand.

The company has several factories spread around Europe, the factory under study is located in Portugal and is divided into two facilities, the printing plant and the metal packaging plant. The focus of the project is the printing plant, more precisely the lithography stage where varnishes and inks are applied to the metal sheets. According to company managers, the lithography phase is the bottleneck of the company, mostly due to the large setup times that are highly dependent on the sequence of operations processed. Currently, scheduling is done manually and relies heavily on the experience of the scheduler.

In an effort to increase the overall throughput of the factory, this project focuses on the bottleneck of the production process. The company believes that the automation of the process associated with the usage of sophisticated optimization techniques will prove to be a major asset in the future. It should be noted that due to confidentiality agreements, some of the values present in this report do not correspond to reality. The schedule of the proposed project is presented in figure 1.1.

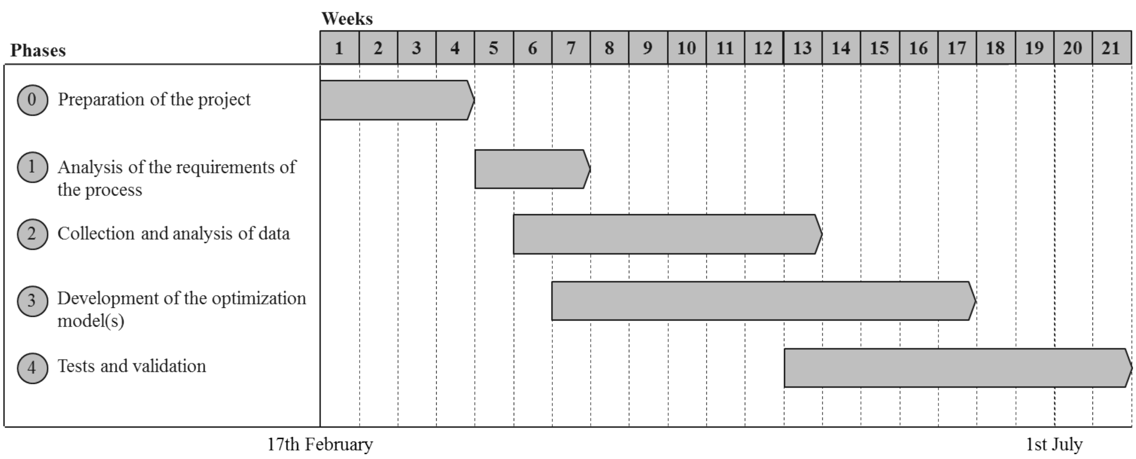

Figure 1.1: Schedule of the proposed project

1.3 Dissertation Structure

The dissertation is divided in six chapters that are organized as follows.

The case study is explained in chapter 2. Firstly, the production process in the printing plant is described and secondly, the current scheduling process is analyzed.

In chapter 3 it is possible to find a review of the literature on scheduling problems and common solutions methods used to solve them. The flexible job shop problem is studied in more detail since it is the closest machine environment to the problem under study.

In chapter 4 the decision support system developed is presented. A diagnosis of the current planning process and some analysis on the data that will be used as input to the system are reported. The rolling horizon strategy used to tackle some operational constraints is also explained. Finally, the interface of the system is disclosed, inputs, outputs and KPI are described.

In chapter 5 the decision model is presented. It begins by introducing some of the tested formulations, then the decomposition of the problem is justified and finally, each mathematical model is explained.

Chapter 6 presents the steps taken to validate the solutions generated by the decision support system.

In the last chapter conclusions are drawn and the future direction of the project is disclosed, as well as opportunities to improve the proposed methodology.

Chapter 2

The Challenge

This chapter is divided into two sections. In the first section the production process of the printing plant is described, detailing each production stage. In the second section the current scheduling process is depicted.

2.1 Production Process

The printing plant, which is the subject of this study, includes the stages illustrated in figure 2.1, namely the primary cut, the lithography and the secondary cut. The focus of this project is the production scheduling of lithography, which includes three main production phases: preparation, printing and finishing. In the first and last phases, varnishes are applied, in the printing stage, colors are printed into the metal sheet. It is also important to understand the requirements of upstream and downstream processes, since they constitute constraints that must be considered when scheduling the lithography stage.

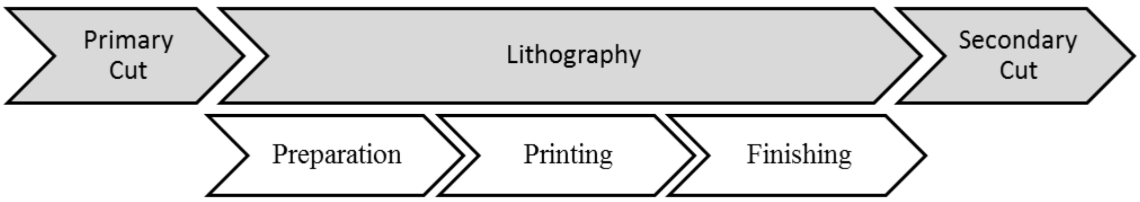

Figure 2.1: Production stages in the printing plant

As mentioned in the first chapter, setup times are a major concern in the printing plant. With clients ordering more times, but in less quantity, more setups are performed. In this paragraph, a brief description of setups is presented, later in chapter 4, a deeper analysis can be found.

In both varnish and printing machines, setup times depend on the previous and the following orders. Setups involve preparing the machine for new sheet formats, changing colors or varnishes, among other operations. The time spent changing these features depends both on the difference from one operation to the next and on the direction of the changes, since increasing the size of the format takes more time than decreasing, and changing from a dark color to a lighter color takes more time than the opposite. In other words, the setup matrix is asymmetric.

One last remark that is noteworthy is that setups in the printing stage can be split into two phases: static setup and dynamic setup, as named by the company. Static setups involve the operations that have been described so far. Dynamic setups include the adjustments made to ensure that the combination of color results exactly on what the client wants. The times related with this type of setups are highly dependent on the operator and are extremely hard to anticipate.

2.1.1 Primary Cut

The primary cut is the first processing stage in the printing plant. Here, coils are cut into metal sheets that will later be turned into packages of different sizes and formats.

Since raw materials are a major factor on the cost of the final product, sheets with different widths and/or thicknesses are produced using coils with different sizes to reduce waste. It is also important to mention that different coils might require different blades to be cut. There are three types of blades used in this facility: thin, thick and scroll.

The setup times between different coils and sheet formats are not a major concern in this stage, but when blades are changed, the changeover time involved is an issue. To avoid these setups, production is scheduled to use the same blade as much as possible, with cycles of up to four days.

This stage works in a make-to-stock (MTS) fashion, where production and procurement are based on forecasted demand. This fact, allied with the long production cycles pose a challenge for the next stages since orders that do not have enough sheets ready to be varnished or printed may need to wait a long time to begin production.

2.1.2 Varnishing: Preparation

This production phase involves two main types of operations: application of varnish to the internal side of the sheet and application of varnish to the external side of the sheet. The first is used as a layer of protection between the metal and the contained product, and/or for appearance purposes. The second is used to ensure that the colors printed in the next stage stick to the sheet and/or for appearance purposes as well.

Usually an order involves one internal varnish and one external varnish, but this is not always the case. Some orders need more than one internal or external varnish, and some need more than one application of the same varnish. These requirements are agreed with the client beforehand and when the order is created, the number of operations required in this stage is already given.

There are five machines that can be used to apply varnish, identified in the set $\{M_2, M_3, M_4, M_5, M_6\}$. Each of these machines can apply one varnish per operation. Even though the machines are extremely flexible, and in theory every machine can process every kind of varnish, in practice, due to the different processing times, some machines are specialized as stated in table 2.1.

Table 2.1: Specialization of varnish machines

Machine	Varnish
M_2	White varnishes
M_3	Finishing varnishes
M_4	Golden varnishes
M_5	Finishing varnishes
M_6	Flexible

This is an interesting idea since it takes advantage of machine efficiency, and also decreases the setup times by grouping similar products on the same machines. However, it is important to pay attention to the work load balance and adapt machines that are less demanded to help machines that are not being able to meet the due dates.

As stated previously in this section, varnishes are applied depending on client's requirements and therefore this stage works mostly under a make-to-order (MTO) policy. However, because some combinations of varnishes are very frequent, some operations are performed in a MTS fashion.

2.1.3 Printing

As the name suggests, in this phase of production colors are printed on the metal sheet. The patterns and colors printed depend on the product and client being considered, so, this stage follows a MTO strategy in all cases. Orders have a set of colors that must be performed, and the sequence of colors should also be followed to ensure that the final product is exactly as agreed with the client.

Contrarily to the varnish machines, in the printing stage, machines can print more than one color in the same operation, the number of colors a machine can print depends on the number of printing units it has. Printing machines are identified in the set $\{M_5, M_{11}, M_{13}, M_{15}\}$, different characteristics, including the number of printing units of each machine can be seen in table 2.2.

Table 2.2: Characteristics of the printing machines

Machine	Number of Printing Units	Nominal Speed	Average Setup Time (min)
M_5	2	6000 sheets/hour	46
M_{11}	4	6300 sheets/hour	74
M_{13}	2	6300 sheets/hour	56
M_{15}	7	9700 sheets/hour	86

One of the features that makes the printing plant such an interesting case study, and at the same time increases the complexity of the problem, is the fact that the number of operations required to complete each order depends on the machines chosen to process it. In table 2.3 it is presented an example of possible machine sequences for an order with 4 colors.

Table 2.3: Example of possible machine sequences for an order with four colors

Printing Stage	**Number of printing operations**
M_5 - M_5	2
M_{11}	1
M_{13} - M_{13}	2
M_{15}	1
M_{13} - M_5	2
M_5 - M_{13}	2

At the moment of the project, the maximum amount of colors allowed per order was eight. It is interesting to notice that an order with seven colors can be processed in any number of operations, from one (M_{15}) to four (ex: M_5-M_5-M_5-M_5).

Intuitively, it might seem obvious that the best approach is to use machines that minimize the number of operations needed, to decrease the number of setups performed. However, this is not always the case. Since, as explained in the beginning of this chapter, setup times depend on the difference of colors and formats between two consecutive operations, using more operations may be beneficial if they are similar to other operations performed on the same machine.

Jobs can be divided into two main categories depending on the type of colors they require. Some jobs use primary colors (cyan, magenta and yellow) and black to reproduce an almost infinite amount of colors, while others require the application of direct colors to achieve the desired result. To differentiate both types of jobs, they are called *CMYK* and *Pantones*, respectively.

2.1.4 Varnishing: Finishing

In the last stage of lithography, one or more finishing varnishes are applied on top of the colors to give a shiny or matte effect and/or to serve as a layer of protection to avoid direct contact between skin and paint.

As seen in subsection 2.1.2, machines M_3 and M_5 are used to apply finishing varnishes. It is important to notice that machine M_5 can also be used to print colors as stated in subsection 2.1.3. In figure 2.2 it is possible to see that the varnishing unit comes after the printing units, and that is the main reason why M_5 is used for finishing varnishes instead of preparation. This way, in the same operation it is possible to print two colors and apply one layer of finishing varnish.

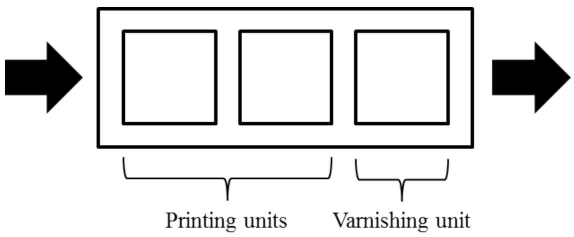

Figure 2.2: Representation of machine M5

2.1.5 Secondary Cut

In the secondary cut phase, sheets are cut into single bodies as explained in figure 2.3. Even though both cuts are made in a single operation, the cut sequence can not be disregarded as will be explained in subsection 2.2.3 due to the fact that every body in the same row of bodies falls into the same container C_x to be taken to the next production stage.

While the previous stages work every day of the week, all day long, the secondary cut does not work during the weekends because it has excess capacity. It is noticeable that at the beginning of the week there are many orders waiting to be cut, while at the end of the week the machines are waiting for jobs to perform. This unbalance creates issues that will be explained in the next section.

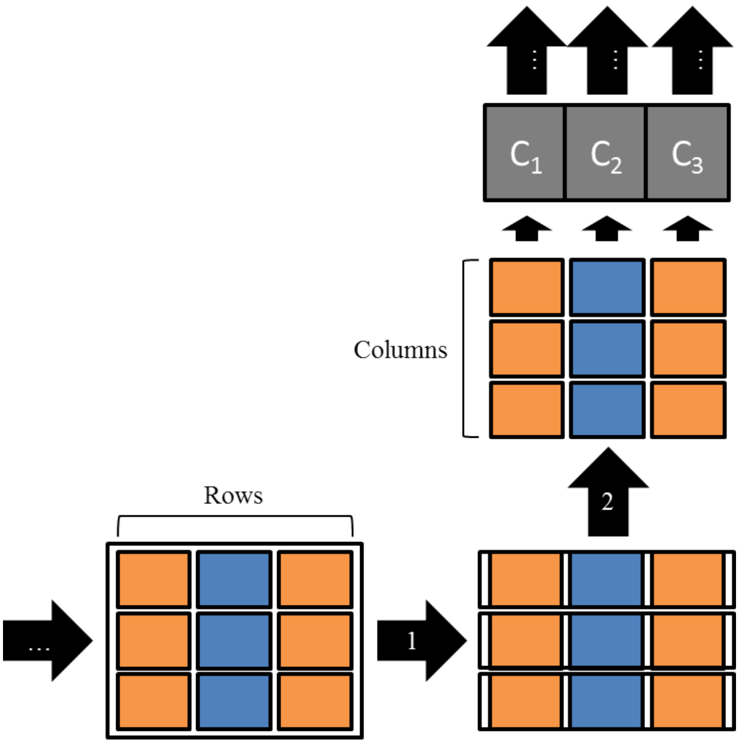

Figure 2.3: Cutting process in the secondary cut stage

2.2 Current Scheduling Process

The current scheduling process may be divided into different phases: machine assignment, combination of orders and sequencing. In this section all the phases are described. Furthermore, an initial introduction to the tactical-level planning is presented.

2.2.1 Planning Process

The main decision made in the planning stage are the orders that are due to the end of each week. Every Wednesday the planning department sends to the programming team the list of orders that should be completed before the end of the next week. This list is calculated using average producing capacities and predefined machine sequences.

Another decision that is made at this level relates to the orders that should be outsourced. Since the printing plant is the bottleneck of the production system and it is also the first step in the value creation chain, outsourcing this service for some orders increases the throughput of the factory and ensures that upstream machines are not restrained by the printing plant's throughput.

As stated before, the plan is delivered every Wednesday and it specifies what should be produced before the end of the following week. Weeks are counted from Monday to Sunday, but the secondary cut is not available during weekends, meaning that everything that is finished during Saturday and Sunday will not be able to meet the due date.

One last remark is that the team that assigns machines and schedules the tasks on an operational level has access to all the orders in the system and is not restrained to those that the planning department listed for a given week.

2.2.2 Machine Allocation

As seen in the previous section, varnish machines are specialized, meaning that most of the times, when an operation requiring a given varnish is scheduled, the machine is already known based on the varnish. When one resource is unable to fulfill its plan, another one is adapted to ensure that orders do not fall behind.

What requires more effort from the machine allocation phase is the printing stage. A series of factors must be considered to decide which machine sequence is the most favorable for each order: the number of colors required; the number of sheets needed; the similarities between the order being considered and other orders allocated to a machine.

Previously, in subsection 2.1.3, it was explained why the number of colors is an important factor to consider when assigning orders to machines. The need to examine similarities was demonstrated in the beginning of section 2.1, in the paragraph related to setup times. The number of sheets in the order is also an important factor because different machines have different processing speeds. Orders that require a larger number of sheets are assigned to machines with higher processing speeds.

The combination of these three factors to make the best decision is an extremely difficult job that depends heavily on the experience of the person performing the assignment. To avoid this dependency, the company created a set of predefined sequences that were considered satisfactory. The matrix can be found in appendix A.

2.2.3 Combination of Orders (COs)

There is the possibility to combine orders that respect a set of criteria: the orders must have the same format, use sheets of the same size and require the same varnishes. When two orders are combined, the same sheet will have two different products in different rows of bodies, as exemplified in figure 2.4 (here rows are defined as perpendicular to the processing direction). Theoretically, it would be possible to also have different products along the same row of bodies. Nevertheless, this is not done because at the end of the secondary cut, when sheets are transformed into bodies, different products would be mixed in the same container. The maximum number of orders that can be combined into the same CO is equal to the number of rows of its sheet.

Figure 2.4: Example of a combination of orders

When orders are combined, common colors can be applied on the same printing unit. If order A requires 5 colors and order B 3 colors, where 2 of them are different from any color required by order A, then the CO will require 7 colors. It is important to pay attention to this detail because it might not be beneficial to combine orders with many different colors since the expected number of operations to be performed also increases with the number of colors in the job.

Another critical point that should be considered is the number of sheets in a CO. The orders are combined in the same sheet and, as explained previously, a row can only have bodies that belong to the same order. So, when creating a CO the number of rows dedicated to each order has to be decided. If two orders of 1000 sheets, O_1 and O_2, are combined into CO_A and they use a sheet with 3 rows and 2 columns as exemplified in figure 2.5, then one of the orders will be assigned to two rows, while the other will be assigned to one row.

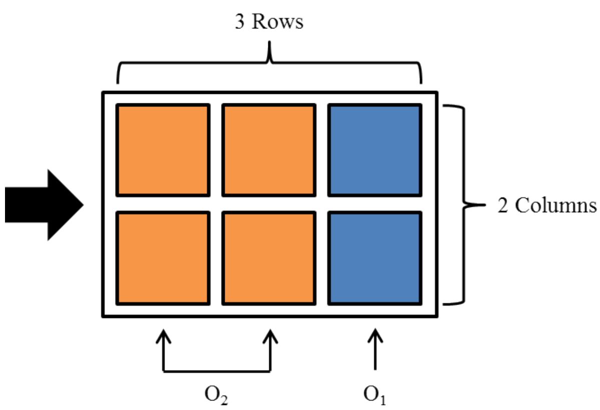

Figure 2.5: Example of a CO of 2 orders with three rows per sheet

In the previous example, it would take 1500 sheets of CO_A to fulfill the demand of O_2, since $2/3$ of CO_A consist of O_2 bodies. However, to fulfill the demand of O_1, 3000 sheets would be necessary, which is 50% more than the amount required to produce both orders separately (2000 sheets). Since the raw material represents the highest cost of the final product, the company established a gap limit between the number of sheets of a CO and the sum of sheets of the orders.

So far, all the examples presented considered the combination of two orders. In fact, the number of orders combined can go from 2 to the number of rows in the sheet. Using the example presented before, O_1 and O_2 could not be combined, but if a third order O_3, that met all the requirements to be combined with O_1 and O_2 and also had 1000 sheets, was available at the time, then the CO that would result of this combination would be acceptable and its number of sheets would be equal to the sum of sheets of each order combined.

2.2.4 Sequencing

Since the machines are already assigned, the optimization of throughput is dependent on the optimization of setups and minimization of idle time. Currently, idle times are not allowed in the lithography, so, grouping orders with similar colors and formats is one of the main concerns of the programmer. To be able to take advantage of the similarities between jobs, the programmer mixes the orders and COs that are due to the scheduled week with orders that still do not have a fixed due date. These readjustments benefit the throughput, but consume processing time that could be used to process orders that have a higher priority.

Presently, the programmer only schedules operations that are available to start production at the time the scheduling is done to guarantee that two operations of the same job are not performed simultaneously, which would be an impossibility. It should also be noted that every day, only the following day is sequenced.

Unlike the machine assignment task, in the sequencing stage, the trade-off is entirely a responsibility of the scheduler since there are no pre-defined rules to condition the programmer's choices. The setup times are not well documented and the experience of the programmer is the only way to anticipate the quality of the schedule. At the same time there is no indication of how many orders can be delayed to increase a certain amount of throughput. This lack of control and information makes the current scheduling phase hard to evaluate since there are no objective short-term indicators to compare different schedules.

Chapter 3

Literature Review

This chapter starts by presenting the generic scheduling problem and the notation used to identify different objectives and constraints. Then, different solution methods and approaches are introduced. Finally, a deeper review of the flexible job shop problem describes how researchers have been tackling this problem in recent years. In each section a brief explanation of how the concepts found in the literature apply to the problem under study is given.

3.1 Production Scheduling

As defined by Graves (1981), production scheduling is the process of allocating available production resources over time to optimize a given objective. According to Allahverdi et al. (2008), the first studies on scheduling problems were made in the mid-50's, and since then both the number and flexibility of the problems have been increasing. Nowadays, it is possible to find thousands of papers in the literature with a wide range of complexity.

In this chapter the notation proposed by Graham et al. (1979) will be explored in detail. It uses three fields $\alpha|\beta|\gamma$ to define the machine environment, job characteristics and optimality criteria, respectively.

3.1.1 Machine Environment (α):

In production scheduling problems, the machine environment plays a major role in defining the complexity and flexibility of the problem and is therefore critical when comparing methodologies and results. The notation used is based on the definitions of Allahverdi et al. (1999) and Pinedo (2005).

Single Machine (1) Tasks are processed, one at a time, by a single available resource.

Parallel-Machine (P,Q,R) This is a generalization of single machine problems. In this environment, tasks can be processed by any of the available machines. Machines are considered identical if the processing time of each task is the same on every machine. Uniform machines have different processing speeds, but their ratio is constant for every task. Finally, in unrelated machines, a relation between processing times can not be defined.

Flow Shop (F) Jobs have m operations that must be processed on m machines in the same order.

Flexible Flow Shop (FF) This is a generalization of the standard flow shop, where in at least one of the production stages more than one machine is available to process a job. In some cases, jobs may skip some stages, but they must always follow the same order.

Job Shop (J) There are m different machines and each job has a given machine route where some machines may be missing.

Flexible Job Shop (FJ) This is a generalization of job shop problems, where, instead of a fixed machine route, jobs can be processed on more than one machine in at least one production stage.

Open Shop (O) Jobs must be processed once on each of the m machines in any order.

3.1.2 Job Characteristics (β):

The second field characterizes the interaction between jobs and tasks. A brief and certainly not exhaustive list of some of the most common characteristics is presented and explained.

Precedence Constraints A job J_k requires that job J_i is completed before it can start.

Release dates A job can only start after a given date/time $t > 0$.

Preemption A job can be interrupted in favor of a higher priority job.

Setup Times and Costs Some approaches in the literature consider setup times as part of the processing time. However, this approximation is not possible when the time or cost depends on the sequence or on the machine where the job is processed. A setup is sequence-dependent if its duration or cost depends on the tasks that are processed before and after the setup.

Batch Setup Times and Costs In some industries jobs are produced in batches, and a setup time or cost takes place before the production of each batch starts. In problems with this characteristic, jobs are usually divided into families, and batches are created with jobs of the same family.

3.1.3 Optimality Criteria (γ):

The last aspect that is considered when defining a scheduling problem is the objective. The notation used to describe the most typical objectives is the following:

Makespan Defined as the completion time of the last task processed. It is very common to find this criteria in the literature since minimizing the makespan increases the throughput of the shop floor.

Lateness Difference between the completion time of a job C_j and its due date d_j, $L_j = C_j - d_j$. The maximum lateness is therefore used to minimize the worst divergence from the due dates.

Tardiness Defined as $T_j = max\{0, C_j - d_j\}$. It should be noted that if there is any late job, then T_{max} = L_{max}. However, total tardiness and total lateness are very different. In the literature, the minimization of total tardiness is one of the most popular objectives.

Earliness Being the opposite of tardiness, it is defined as $E_j = max\{0, d_j - C_j\}$. Although this criteria is less used than tardiness, in some cases it might be important to pay attention to this metric since it minimizes inventories of finished products.

Setup Time or Cost The minimization of setup times can be used to maximize throughput in some cases. Furthermore, in industries where setups may have a negative impact on machine reliability or efficiency, the explicit minimization of setup costs is considered.

Weighted criteria In real world cases, jobs and customers often do not have the same priority. Thus, it is common to minimize for instance the weighted tardiness defined as $wT_j = \sum_j w_j T_j$ instead of the total tardiness. This idea can be applied to other objectives.

3.1.4 Classification of the Problem under Study

In the studied problem a set of jobs J has to go through a set of operations O of varnishing $V \subset O$ and printing $P \subset O$. Each operation o can be performed on a subset of machines $M_o \subset M$. This formulation points to a flexible flow shop or flexible job shop problem. The main difference between these concepts relies on the order of the operations performed. If every job follows the same operation sequence, the environment is considered a flexible flow shop. If the sequence varies from job to job, it is a flexible job shop.

Following the description presented in chapter 2, the sequence of the stages preparation, printing and finishing is well defined and could therefore be characterized as a flexible flow shop environment. However, inside each stage the sequence and the number of operations are more flexible, approaching a flexible job shop setting. After further analysis of the literature related to this problem, some relevant differences between the printing plant under study and a common flexible job shop scheduling problem were found.

Firstly, reentrant processes are allowed, meaning that the same job can have more than one operation performed on the same machine. Secondly, in the printing stage the number of operations is not defined. Depending on the machine assignment, the number of operations may vary considerably, which increases the complexity of the problem.

In terms of job characteristics, precedence constraints between jobs are not considered, but for each job j, operation O_n can only start after operation O_{n-1}. Release dates are only considered in cases where there is not enough inventory of sheets ready to start production at the beginning of the time horizon. Preemption is not allowed.

As described in the beginning of chapter 2, setup times are sequence-dependent and should be considered explicitly since they constitute an important fraction of the time horizon.

The optimality criteria considered are the makespan and total tardiness. The minimization of makespan increases the throughput of the facility, while the minimization of total tardiness increases customer satisfaction and loyalty.

3.2 Solution Methods

In this section different solution approaches used in the scheduling literature are presented. The goal is to consider the advantages and disadvantages of each methods, to be able to make a conscious decision regarding the future direction of the project.

3.2.1 Dispatching Rules

Dispatching rules are usually greedy heuristics used to select which task should be processed next on a resource. In theory they are useful as constructive heuristics, or in highly uncertain and dynamic environments where more sophisticated (and thus more computationally demanding) methods can not be applied. In practice, when the schedules are programmed manually without the aid of a decision support system, planners often use these rules to build and update production schedules. Some of the most common rules used are briefly explained in this section following the definitions presented by Almada Lobo (2005).

Earliest Due Date first (EDD): Tasks are scheduled according to their due date. When a resource finishes the production of a task, the next task to be processed is the one with the earliest due date. This rule is used to minimize maximum lateness and tardiness.

First-Come, First-Served (FCFS): Tasks are sequenced according to their release dates. As Almada Lobo (2005) states, this rule minimizes the variation of the waiting time among tasks.

Minimum Slack first (MS): Related to the EDD, this rule sequences tasks based on their slack, which can be defined as $max\{d_i - p_i - t, 0\}$, where t is the current time and p_i is the processing time of the task. Unlike the other rules presented, MS is dynamic, meaning that the sequence of tasks may change over time. This rule is used to minimize criteria that involve due dates.

Shortest/Longest Processing Time first (SPT)/(LPT): Tasks are sequenced according to their processing time. While SPT is used to minimize the mean completion time, LPT is used to balance the load in problems with parallel machines, because at the end of the time horizon tasks with a shorter processing time can be used to adjust gaps created by larger tasks.

3.2.2 Mixed-Integer Linear Programming (MILP)

As computer processors evolve, mathematical programming formulations become more suitable to solve combinatorial problems, such as scheduling. However, scheduling problems are NP-hard (Garey and Johnson, 1979) and therefore, MILP formulations are used mostly for small and medium size instances.

As stated by Wilson and Morales (2012), MILP formulations for scheduling problems can be divided into continuous and discrete time models. Discrete models divide time into a finite number of periods, and each task is linked to one of those periods. Despite resulting in constraints that are simple and easily understood, according to Floudas and Lin (2005), discrete models have two disadvantages compared to continuous models. The first is that time is a continuous variable and as a result, any discrete representation is by definition an approximation. The second is related to the duration of each period: large periods will decrease solution quality, whereas short periods will increase computational requirements. Due to these issues, most of the scheduling literature prefers to use continuous-time models.

Continuous-time models can be further classified in two categories: models that use immediate precedence and models that use general precedence. Immediate precedence models derive from Wagner (1959) and Wilson (1989). Wilson's model uses the variables:

$$X_{ii'm} = \begin{cases} 1 & \text{if task i' is performed immediately after task i on machine m;} \\ 0 & \text{otherwise.} \end{cases}$$

$$\forall i, i' \in I,\ m \in M$$

On the other hand, global precedence models are based on Manne (1960) that uses the decision variables:

$$X_{ii'm} = \begin{cases} 1 & \text{if task i' is performed later than task i on machine m;} \\ 0 & \text{otherwise.} \end{cases}$$

$$\forall i, i' \in I \ :\ i' > i,\ m \in M$$

Pan (1997) compared the most well known models of the scheduling literature and concluded that Manne's model was the most efficient. Later, Pan and Chen (2005) stated that the formulation of

Liao and You (1992) based on Manne's model was able to reduce the number of constraints and achieve results faster.

3.2.3 Constraint Programming (CP)

Constraint programming started as a technique used in artificial intelligence. Recently, it has been used in operation research as an optimization method. As in mathematical programming (MP), in CP, constraints can be mathematical relations. However, in CP, nonlinear equations do not pose such a greater burden than linear equations, as they do in MP.

CP was first designed to find good, yet not necessarily optimal, solutions that respected a given set of constraints. Consequently, an objective function was not explicitly considered in this framework. The algorithm starts by finding an initial feasible solution disregarding the objective function. Once it finds it, a new constraint is created stating that the value of the objective function must be less (for minimization problems) or higher (for maximization problems) than that of the last solution found. Every time the algorithm finds a new feasible solution it adds a new constraint shrinking the solution space and ensuring that the objective value improves over time.

In Pinedo (2005) it is possible to find different applications of constraint programming for scheduling problems. One of them regards a job shop system for which a possible formulation is presented. It is also interesting to notice that some optimization software, as IBM'S ILOG, uses scheduling problems to exemplify how CP can be applied to optimization problems.

3.2.4 Metaheuristics

Metaheuristics are frameworks that combine heuristics to explore a solution space. These methods rely on two major phases: intensification (exploiting a specific region of the solution space - typically ends in a local optimum) and diversification (getting out of the local optimum and exploring new regions of the solution space).

Metaherusitcs may not be able to find the optimal solutions in many problems, and will never prove optimality, even when the obtained solution is optimal. Nevertheless, on large-instances and problems where exact methods take too much time to even find a feasible solution, these methods are a compelling alternative.

Some of the most commonly applied metaheuristics in scheduling problems are: Simulated Annealing (SA) introduced by Kirkpatrick (1983), Genetic Algorithms (GA) proposed by Goldberg et al. (1989), Tabu Search (TS) developed by Glover (1989) and Variable Neighborhood Search (VNS) proposed by Mladenović and Hansen (1997). According to the review published by Allahverdi et al. (2008), out of 300 scheduling papers published from 1999 to 2007, 35 used GA. TS was the second most common metaheuristic, followed by SA.

3.2.5 The Shifting Bottleneck Heuristic (SBH)

The SBH was first proposed by Adams et al. (1988), and it tries to decompose the problem into single-machine sub-problems that can be tackled in a reasonable amount of time. The idea behind this heuristic is that the bottleneck determines the throughput of the entire system and therefore should be optimized.

This heuristic uses the disjunctive graph representation, exemplified in figure 3.1, proposed by Roy and Sussmann (1964), where nodes are the tasks to be performed, conjunctive arcs connect operations of the same job that must be performed following a known sequence, and disjunctive arcs connect pairs or operations processed on the same resource.

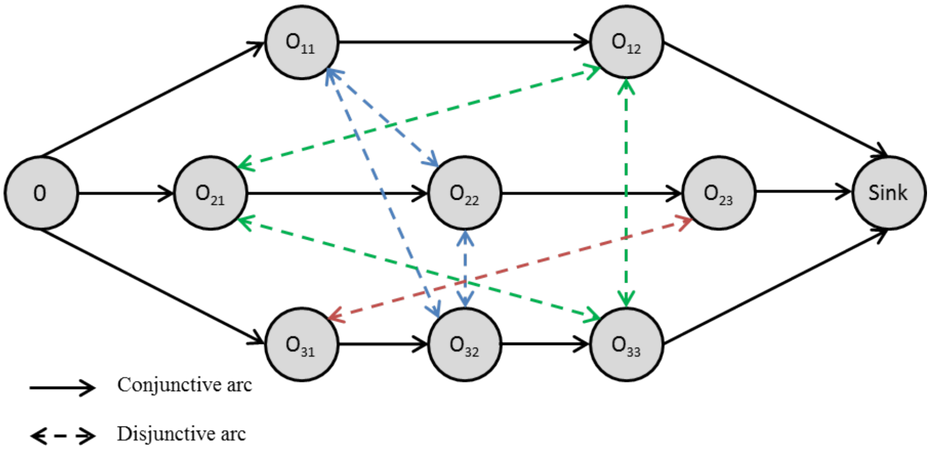

Figure 3.1: Disjunctive graph representation

SBH is mostly used for job shop problems since the machine where each operation should be processed is already given and therefore is not part of the decision process. It starts by identifying the most critical non-scheduled resource, optimizing that resource by fixing its disjunctive arcs directions.

Topaloglu and Kilincli (2009) solved a reentrant job shop problem using a method based on the shifting bottleneck heuristic (SBH). The author applied some modifications to the heuristic, both in the sequencing algorithm and in the identification of the bottleneck machine. The solution was tested on random instances and for a real world dyeing–finishing facility in a textile factory. The results show that the method outperforms the existing techniques used in that facility.

3.2.6 Proposed Approach

The approach followed is discussed in more detail in chapter 5. Here, only a succinct explanation is provided.

The problem is divided into smaller sub-problems, separating the scheduling of the varnish operations from the scheduling of printing operations, and the machine assignment from the sequencing phase. The sub-problems are solved sequentially through MILP models. MILP formulations allow to add new requirements easily, while in other methods as metaheuristics it is harder to add the same requirements after the design of the methodology has already started.

CP and SBH are also interesting techniques, specially for the sequencing of operations assigned to each machine. Due to the time available to complete the project these methodologies will not be tested in this dissertation, but should be considered in the future.

3.3 Flexible Job Shop Scheduling Problem

This section delves into the literature related to the flexible job shop scheduling problem (FJSSP). This problem is the closest to the real world case that is being tackled in this dissertation. Hence, this section seeks to understand what methodologies are more effective, and identify possible gaps in the literature.

In the FJSSP each job J_i in a set J of jobs has to go through a sequence $O_{i_1}, O_{i_2}, ..., O_{i_n}$ of operations. This sequence implies that every O_{i_k} must be processed before $O_{i_{k+1}}$. A set $M = \{M_1, M_2, ..., M_m\}$ of machines is available, and for each operation O_{i_k} a subset $M_{ik} \subseteq M$ of compatible machines is given. To solve the problem it is necessary to answer the questions:

- In which of the available machines will each operation be processed?
- What is the sequence of operations in each machine?

In the literature it is possible to find several papers tackling this problem. Most of them use metaheuristics to find solutions.

Ponnambalam et al. (2005) developed an Ant-Colony Optimization (ACO) algorithm to minimize makespan, and compared it against a CP formulation, showing that their algorithm is able to achieve much better results especially on larger instances. In their paper they also show how the problem can be formulated using MILP. In Pezzella et al. (2008) a GA is developed to minimize makespan. The results achieved for well-known instances are compared against other GAs and a TS and it is shown that their GA is able to get comparable results or even outperform them.

Bagheri and Zandieh (2011) developed a VNS algorithm with three neighborhood structures to minimize a bi-criteria objective function considering makespan and mean tardiness. In this study, setup times are explicitly considered and are sequence-dependent. The authors compared the results against a modified version of the GA created by Pezzella et al. (2008) and the parallel VNS proposed by Yazdani et al. (2010), and confirmed that their algorithm was able to find better solutions. Mousakhani (2013) also considers sequence-dependent setup times, and presents an iterative local search (ILS) algorithm and a MILP model. The results show that the MILP model is able to find better solutions than the model proposed by Fattahi et al. (2007). Moreover, they have shown that their ILS outperforms a TS and the VNS developed by Bagheri and Zandieh (2011).

In the last few years, the generalization of the FJSSP where reentrant processes are allowed is getting more attention due to the diverse application areas that require this type of flexibility. Reentrant processes are considered when more than one operation of the same job can be processed on the same machine. In most examples in the literature, reentrant processes do not consider the possibility of two consecutive operations of the same job being processed on the same machine. Chen et al. (2008) and later Chen et al. (2012) tackled this problem using a two stage algorithm. In the first stage machines are selected for each operation according to a grouping GA. The scheduling stage is tackled applying a different GA. The minimization of multiple objectives including makespan, total tardiness and total idle time is considered. The algorithm is tested in a real weapon production facility where it is able to outperform the current scheduling technique.

In our literature review, we were unable to find any example, where the number of operations was not previously defined, which is an important aspect of the studied problem, since it significantly increases the complexity of the problem.

Chapter 4

Decision Support System (DSS)

In this chapter, the developed DSS is introduced. First, a general overview of the advantages of the DSS will be given. Then, it is presented a study carried to identify improvement opportunities in the current scheduling process. Next, the analysis of the data required to feed the DSS is explained. Finally, it is described how the system can be used every day by the operational planners and what inputs are needed as well as the output given.

4.1 Overview

Operational optimization is becoming a trend in established companies, however, this is a complex area where inputs are constantly changing, and every detail of the process must be considered to guarantee that the suggested solution is feasible in practice .

The decision support system built aims to make the task easier for the user and at the same time consider a series of operational requirements that will have consequences on downstream planning processes as the preparation of components needed for the production of each operation.

Besides supporting the decisions at an operational level, one of the aims of this DSS is to allow what-if analysis to help tactical level decisions. Examples of scenarios that can be simulated with the developed DSS are listed below:

- How much would the throughput increase if one more machine was bought?
- How many more jobs would be delivered in time, if optimizing the setups was not a priority?
- How many more COs would be possible to create if the gap limit was increased?

Another implicit benefit of the implementation of the DSS is the standardization and collection of data. At the moment, setup times are not controlled in every machine, previous schedules are not stored in a digital format, and the information flow between the planner and the shop-floor workers is done personally. With the DSS these paradigms should be shifted benefiting the entire organization by granting improved information reliability and decreasing time consumed performing tasks of low value creation.

4.2 As-Is Analysis

In this subsection, the current planning process of the lithography is studied to find improvement opportunities that may help the company increase its productivity. At the same time, this diagnoses may be used as a basis to ensure that the DSS under development answers the current needs of the company.

4.2.1 Machine Allocation

To understand if the matrix displayed in appendix A was being followed or not, the historical data from the beginning of 2014 onward was analyzed. For each type of job, the percentage of times it went to each machine on different operations of the printing stage was computed. The results are shown in appendix B and it is possible to see that in reality, the programmer does not commit to the predefined sequences. For instance, an order with 6 direct colors (*pantones*) with over 4000 sheets was expected to be allocated to the sequences: M_5-M_5-M_5, M_{11}-M_5 or M_{11}-M_{13}. Instead, in $\approx 76\%$ of the cases, it is processed on machine M_{15}.

This discrepancy may be explained by the programmer's experience that adapts the criteria to the work load that is waiting for production at that moment. If done correctly, this adaptation might increase the throughput significantly. In weeks where there is a low number of CMYK jobs, using the machine M_{15} to perform more direct colors might be a good option. At the same time, in weeks where the average number of sheets per job is low, instead of changing the criteria at 4000/6000 sheets, perhaps changing it at 2500/3000 sheets would also increase the overall productivity of the plant. It is recognizable that these adjustments bring benefits in terms of throughput. However, a deep reorganization of the machine assignments may create conflicts with the capacities considered by the planning department. Besides, one of the main goals of the creation of the predefined sequences, to decrease the dependency on the programmer, is lost.

4.2.2 Sequencing

As stated in chapter 3, scheduling, even in simpler environments, is an extremely complex work even for an automated algorithm. When a person faces this type of problem, simplifications are the natural way to handle it. Currently, sequencing is done for each machine separately, and the programmer only looks to operations that are already available to start at the time the sequencing is done. If a job requires two operations, the second is scheduled only after the first has been performed.

The current approach makes the scheduling task easier since precedence constraints do not need to be taken into account. However, the time between operations increases. After analyzing the historical data it was possible to notice that the average time between operations in January and February of 2015 was around 2,4 days. The results can be seen in figure 4.1.

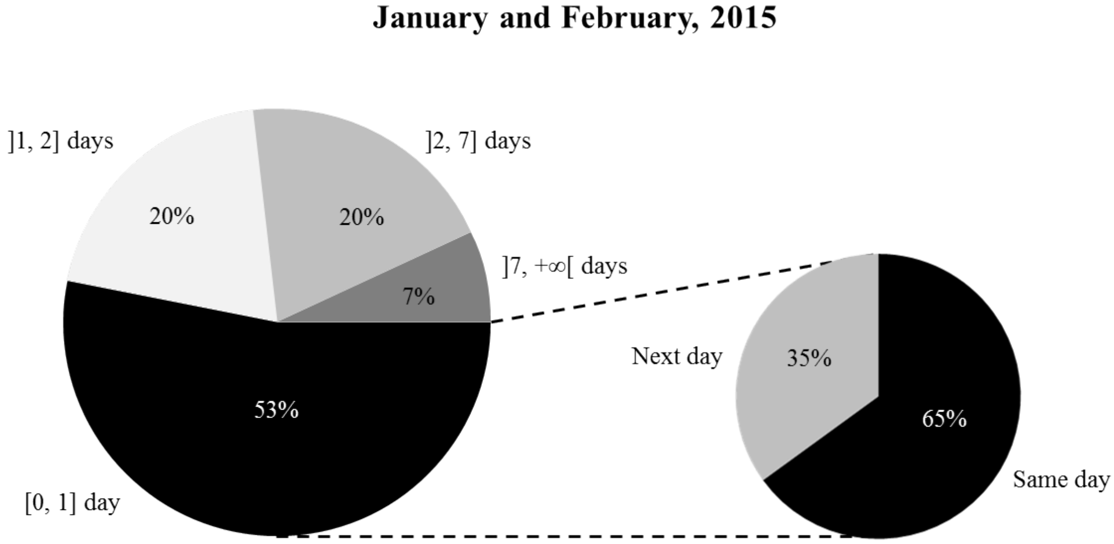

Figure 4.1: Distribution of time between operations

Another point that was analyzed was how time between operations evolved over time, the results are presented in figure 4.2. In one year (from January 2014 to February 2015), the time between operations increased by almost one day. This increment increases the work in process (WIP), jeopardizing the shop floor organization and stability, and also has a negative effect on the tardiness of the jobs. When time between operations increases, jobs with several operations spend a long time as WIP and one week might not be enough to perform all the operations required.

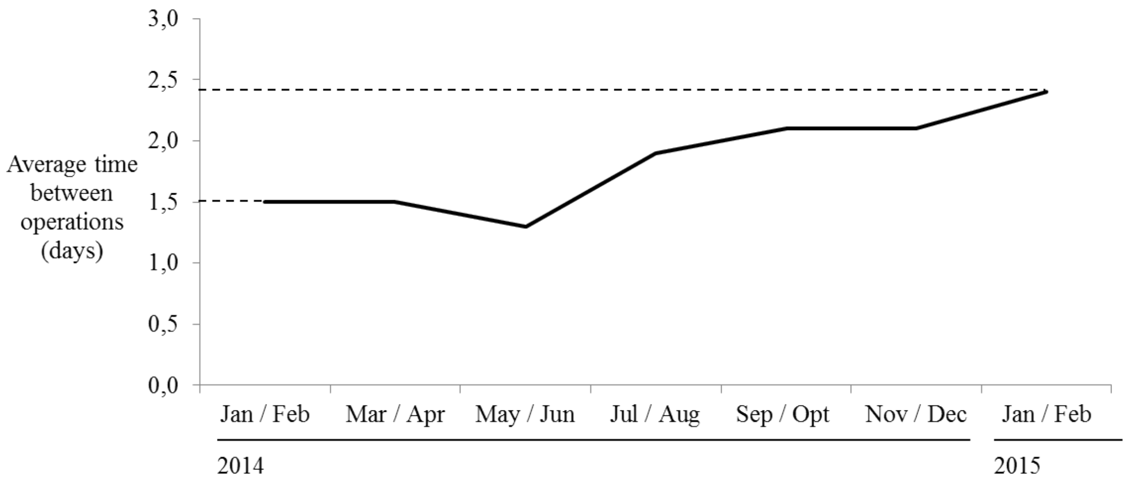

Figure 4.2: Time between operations over time

4.3 Data Analysis

Throughout the development of the DSS, different information was required that was not available. The identification of the bottleneck of the lithography was important to understand which stage should be focused more deeply. Furthermore, to model the reality of the plant, quantitative data related to the processing times and setup times was necessary. In this section, the different analysis carried to find the information needed are presented.

4.3.1 Bottleneck Identification

It should be noted that since idle times are strongly avoided in all the lithography stages to balance machine availability and work load the number of available shifts of each machine is adjusted. This makes the identification of the bottleneck stage more difficult since it may change over time. To identify the bottleneck, an analysis of the times between the last operation of a stage and the first operation of the next stage was carried out. The results are shown in figure 4.3.

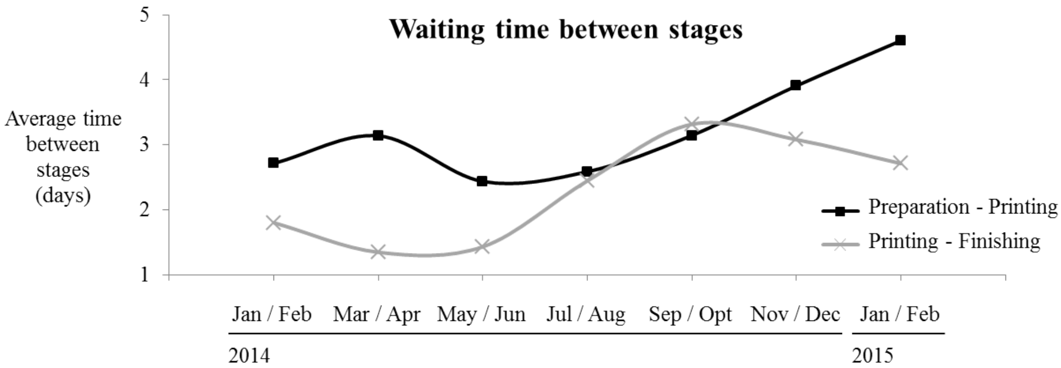

Figure 4.3: Variation of the average waiting time between stages in lithography

It is possible to see that in the last months the work in progress waiting to be printed increased while the work in progress after printing decreased. This confirms the managers' expectations that the printing stage is the current bottleneck of the plant.

4.3.2 Actual Machine Output

As can be seen in the matrix presented in appendix A, machine M_{15} is preferred for orders with a higher number of sheets and that use *CMYK* colors, even if the number of colors is inferior to the number of printing units of the machine. The reasoning behind this assignment is that machine M_{15} has a higher nominal speed than other machines (see table 2.2) and a higher average setup time, so keeping it producing for as long as possible is advantageous for the production system. However, the nominal speed is not a good indicator of the machine's performance, and the real speed should then be calculated. Following the approach used to calculate the Overall Equipment Efficiency (OEE) illustrated in figure 4.4 the actual processing speed was calculated.

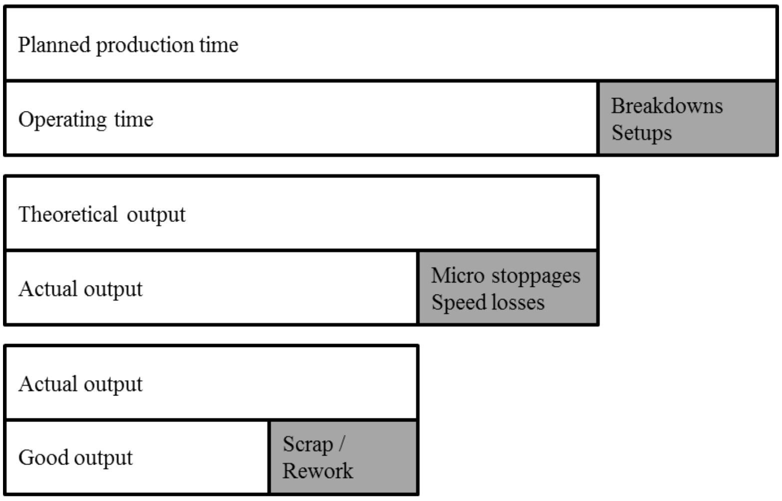

Figure 4.4: Representation of the overall equipment efficiency (OEE) calculation

The goal is to find the actual output per unit of time of each machine, so equation 4.1 results from the division of the actual output and the operating time, that is given by subtracting the time spent with breakdowns and setups from the planned production time.

$$v_r = \frac{\text{Number of sheets processed}}{\text{Planned production time - Time lost with failures - Setup times}} \tag{4.1}$$

The results are shown in table 4.1. The fastest machine is in fact M_5 despite having the lowest nominal speed.

Table 4.1: Real processing speeds of printing machines

Machine	**Real speed**
$M5$	3380 sheets/hour
$M11$	2500 sheets/hour
$M13$	2030 sheets/hour
$M15$	3100 sheets/hour

4.3.3 Setup Times

As previously stated one of the main aims of the DSS is to support the scheduling task that is highly focused on the optimization of setups. However, information regarding setup times is scarce and currently estimating the setup time between two operations O_1 and O_2 is mostly an empirical exercise. In fact, the planner does not try to anticipate the setup time between O_1 and O_2, he just tries to compare whether the setup between O_1 and O_2 is going to be longer or shorter than the setup between O_1 and O_3 (kind of pairwise comparison).

The current approach might be considered effective because it is done by an experienced planner without the aid of a software. Notwithstanding, to start using an automated tool, it is crucial to provide reliable quantitative data to increase the quality of the solutions to be generated. To be able to feed information about setup times to the DSS, historical data was analyzed. It is important to point that the data available about setups was considered by company managers unreliable, but it was the only source where it was possible to extract enough information to study the drivers of setups.

As pointed in chapter 2, managers and planners identified the main factors that influence the setup time: the dimensions of the sheets and the colors in each operation. They also stated that the time spent changing to a sheet with bigger dimensions was different from changing to a sheet with smaller dimensions and that switching to a darker color required less time than switching to a lighter color.

Considering this information, a set of possible drivers was created - see table 4.2.

Table 4.2: Possible setup time drivers

Symbol	**Description**	**Interval**	**Unit**
Δw^+	Increase in the width of the sheet	$[0, +\infty]$	mm
Δw^-	Decrease in the width of the sheet	$[0, +\infty]$	mm
Δl^+	Increase in the length of the sheet	$[0, +\infty]$	mm
Δl^-	Decrease in the length of the sheet	$[0, +\infty]$	mm
Δt^+	Increase in the thickness of the sheet	$[0, +\infty]$	mm
Δt^-	Decrease in the thickness of the sheet	$[0, +\infty]$	mm
N_c	Number of different colors	$[0, +\infty]$	colors
Δc^+	Sum of the differences between colors in the same printing unit (if colors are lighter)	$[0, +\infty]$	*
Δc^-	Sum of the differences between colors in the same printing unit (if colors are darker)	$[0, +\infty]$	*
C	Need to change at least one color	$\{0, 1\}$	-
V	Need to change the varnish	$\{0, 1\}$	-
F	Need to change the format of the sheet	$\{0, 1\}$	-

The difference between two colors is hard to understand conceptually, specially for someone who is not used to deal with the different available color scales. Most color scales use three dimensions to define a color, but the meaning of each dimension varies from scale to scale. To mathematically compare two colors it is common to use the Lab color space or similar scales. Since this information was not available at the time, the RGB codes were used and the difference of colors was computed as the euclidean distance between two points (see equation 4.2), where color $c_1 = \{R_1, G_1, B_1\}$ and color $c_2 = \{R_2, G_2, B_2\}$.

$$\Delta c = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \tag{4.2}$$

As listed in table 4.2 it is important to differentiate whether one color is darker or lighter than the previous color. The approximation used to discern between these two situations was based on the summation of the three dimensions $\Phi_c = R_c + G_c + B_c$. If $\Phi_1 > \Phi_2$, then c_1 is considered lighter than c_2, because $black = \{0,0,0\}$ and $white = \{255,255,255\}$.

With the aid of the software R Core Team (2013) a multiple linear regression was computed to establish the relationship between the drivers and the setup time. A stepwise regression with both forward selection and backward elimination was applied. Stepwise regressions are used to automatically choose the independent variables (the drivers) that explain the dependent variable (the setup time). Forward selection starts the model with no variables, and at each iteration chooses the variable that improves the model the most; this process is repeated until the point where no neglected variable improves the model. Backward elimination, on the contrary, starts the model with every variable and at each step rejects the variable that improves the model the most by being deleted; the process is repeated until the point where no remaining variable can be deleted without decreasing the quality of the model. The bidirectional approach that was used combines both options, deciding at each step what variables should be rejected, and what variables should be added. To compare the quality of the regression at each step, the software used the Akaike information criterion (AIC). AIC$= 2k - 2ln(L_k)$, where k is the number of independent variables selected, and L_k is the maximized value of the likelihood function. The minimization of this criterion benefits the goodness of fit, and penalizes the number of variables selected.

The data available was not uniform for every machine. For machines M_{11} and M_{15}, data about the sequence of the operations performed was available, but for machines M_5, M_{13}, and varnishing machines, it was only known what tasks were performed at each setup. For example, it was known if a printing unit was changed, but it was not possible to know how many were changed or what were the colors on the machine nor the colors that were going to be used next. As a result, in machines M_{11} and M_{15} the variables Δw^+, Δw^-, Δl^+, Δl^-, Δt^+, Δt^- and N_c were tested, while in the other machines, the binary variables C, V and F were used. In table 4.3 it is possible to see for each machine the value of the adjusted R^2 and what variables were considered important to estimate setup times.

Table 4.3: Results of the multiple linear regression for each machine

Machine	**Adjusted R^2**	**Variables**
M_5	0.16	C, V, F
M_{11}	0.43	Δw^+, Δw^-, Δl^+, Δl^-, N_c
M_{13}	0.35	C, F
M_{15}	0.20	Δw^+, Δw^-, Δl^+, Δl^-, N_c
M_2	0.75	V, F
M_3	0.39	V, F
M_4	0.16	V, F
M_6	0.62	V, F

The results are not as good as intended, and that can be due to a series of factors: firstly, the data that was used as input to compute the regressions was not reliable nor complete. Secondly there are tasks involved in a changeover that are not explained by any of the drivers considered in these regressions. In figures 4.5 and 4.6 a representation of the results is displayed. In the figures, for each value of real setup time, the average estimated setup time was calculated. If the regression was a perfect representation of reality, all points would be on the bissectrix, .

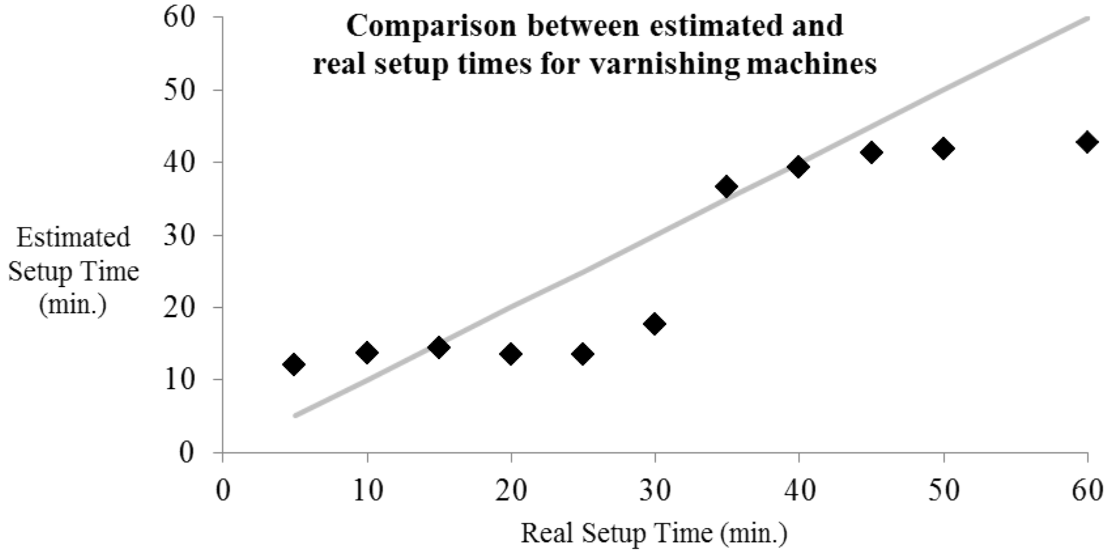

Figure 4.5: Comparison between the real setup time and the estimated setup times in varnishing machines

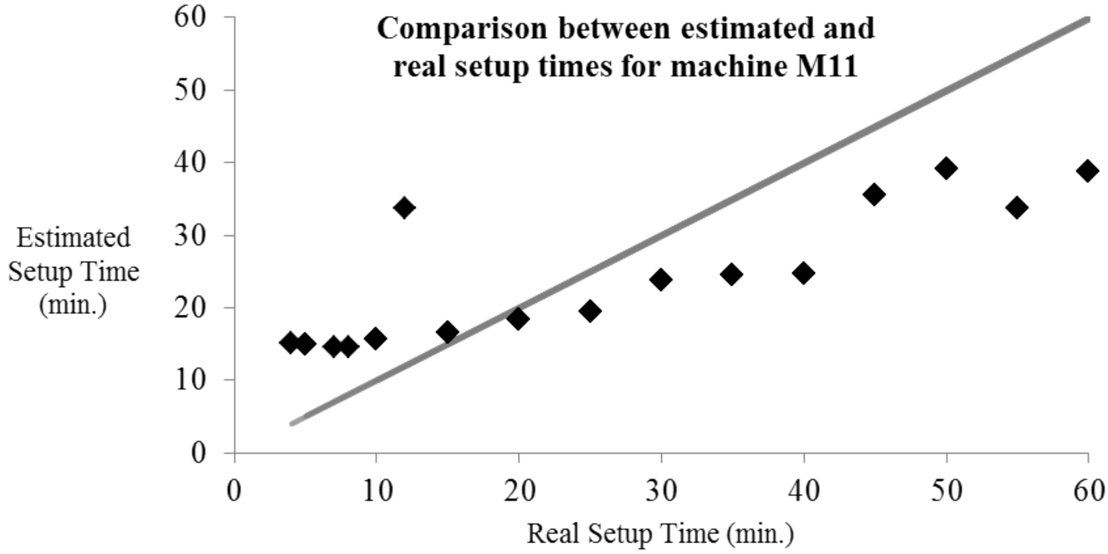

Figure 4.6: Comparison between the real setup time and the estimated setup times in machine M11

A deeper analysis of the setup times is recommended, yet, setup analysis is not the aim of this project, so these results will be used as an input to the DSS until better data is available.

4.4 Detailed Scheduling and Aggregated Scheduling

One of the improvements proposed by the implementation of the DSS is to increase the visibility of the schedule. At the moment, as stated in subsection 2.2.4, sequencing of operations is a daily task and the planner only decides what he is going to produce in the next day. This situation makes other tasks that are dependent on the schedule more difficult to plan and prepare in advance.

It is also important to recognize that operational tasks are subject to several uncertainty factors that influence the quality and feasibility of the schedule, so any attempt to plan for a long period of time might not be worth the extra computational time required to produce a good solution. The developed DSS divides then the operations into three groups: operations that are going to be performed in the next days, operations that should be performed until the end of the week and operations that will be performed later:

- The first group of operations are the ones that are going to be sequenced (the underlying planning horizon will be explained in the next subsection). For each operation of this group expected starting and finishing times are provided so that shop-floor workers know what should be produced, and the teams that provide the resources needed for each operation have everything available in advance. The schedule at this level takes into account every requirement of the process guaranteeing that all previous operations of the same job have already finished before the start of the next one.

- The second group of operations is important for the planning team. As stated in chapter 2, due dates are scheduled for the end of the week, so it is advantageous to have an idea, even if not detailed, of what operations will be performed until the end of the week and what orders will finish before their due date.

- Finally, for the last group of operations the DSS will only decide for each order what is the best machine sequence.

4.5 Rolling Horizon

The proposed DSS is built to be run every morning and reoptimize the allocation of the orders and the sequencing of operations. Due to the need to prepare the materials required to perform the printing operations one day in advance, it must be guaranteed that the schedule generated in the previous day will not change when the DSS reschedules the operations on a given morning.

In figure 4.7 it is possible to see how the DSS will take into account the operational constraints related with the preparation of the materials needed for the printing operations. Every day, the schedule suggested in the previous day is fixed to guarantee that the allocation and sequencing is not going to be reoptimized.

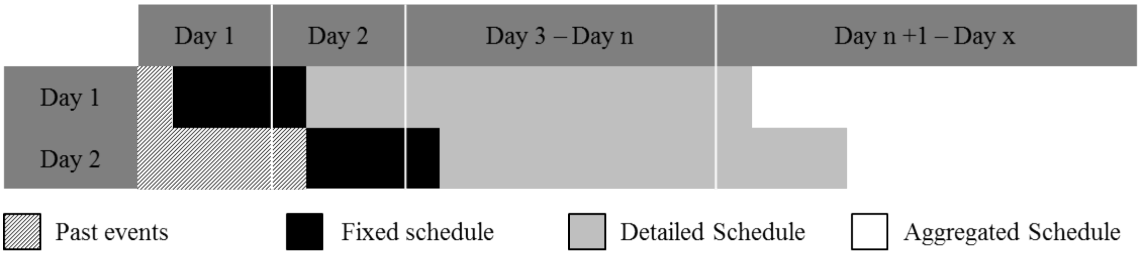

Figure 4.7: Rolling horizon scheme implemented in the DSS

This strategy of fixing 24 hours would work in case the team that prepares the materials and the scheduling team would work every day. In fact, none of the teams works during weekends meaning that the DSS will not be used on Saturdays and Sundays. To compensate this fact, the schedule created on Friday morning should be followed during Saturday, Sunday and Monday, which implies that the team that prepares the operations would have to do three days worth of work in one day. To correct this situation the schedule presented in table 4.4 was built in conjunction with the person responsible for the preparation of the materials. In this table it is defined for each day of the week how many fixed days should be considered, how many days should be optimized, both in detailed scheduling and aggregated scheduling.

Table 4.4: Fixed days, detailed schedule and aggregated schedule for each day of the week

Day of the Week	**Fixed days**	**Detailed Schedule**	**Aggregated Schedule**
Monday	1	3	3
Tuesday	1	3	2
Wednesday	1	3	1
Thursday	2	3	6
Friday	2	3	5

It is possible to see that while on Monday, only that day is fixed, on Thursday the Friday is already fixed so that the team can start preparing the materials needed sooner. On Wednesdays, the team can already start preparing the materials for Friday because they know that the schedule can not be changed. This distribution helps the team to split the extra work needed to prepare the materials for the weekend over a larger period of time.

As stated in the previous subsection, the aggregated schedule is supposed to give an idea of what operations can be performed until the end of the week. On Wednesdays afternoon, the planning department sends the list of orders that should be completed in the following week. On Thursdays, it would be helpful to estimate how many of those orders can actually be performed before their due date. In figure 4.8 it is possible to see how the different schedules would work together for a schedule generated on Thursday of week n.

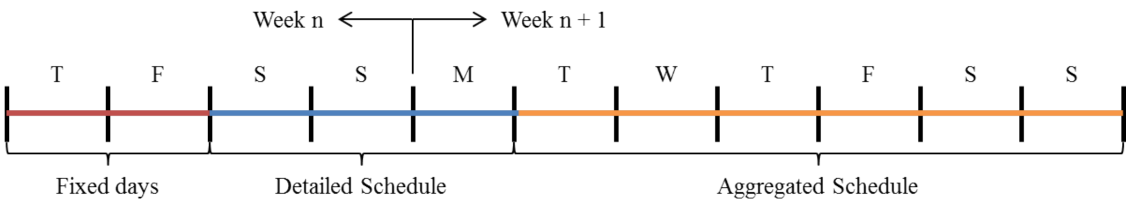

Figure 4.8: Example division of days of the result generated on a Thursday morning

4.6 Inputs and Output

For the DSS to work as intended it needs to receive a determined amount of data that should be delivered in a consistent way. Figure 4.9 shows a schematic representation of how the inputs and outputs of the DSS interact with each other. There are three main inputs, the data related with the work load that should be sent every day; the data related with tactical options that should be changed only when new features or requirements are added (also known as master data); and the schedule generated in the previous day.

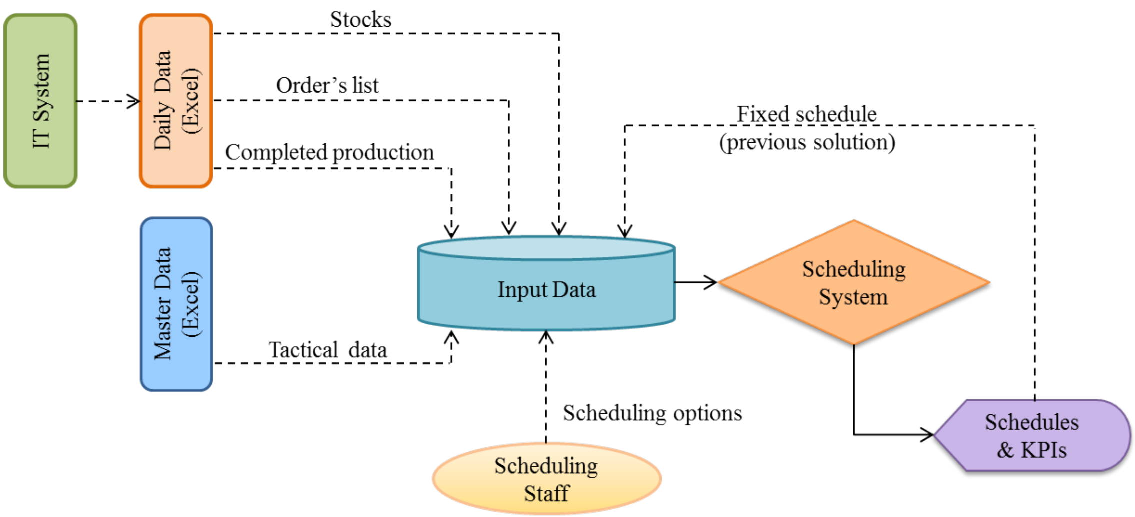

Figure 4.9: Scheme of the interface of the DSS

Every day the IT system generates an excel report with the information related to the orders that have not been completed yet. In this report it is stated what operations of these orders have been performed and what operations still need to be scheduled. It is also possible to extract information related with the stocks of sheet available to be used.

In a different file it is stored the information related with tactical decisions: machines performance and availability, the possible machine sequences for each type of job, the rolling horizon strategy, the limits related to the number of sheets of COs, etc. This information is stored in the DSS so that it does not need to be updated every time the software is run. There is the option to update it when needed, and it also allows to use a new file only for a specific run so that what-if analysis can be performed without changing the file used for everyday runs.

The output file, where the solution is represented, was developed in a way so that it can be used as an input file for the next run stating what orders and operations should be fixed. In addition, a list of key performance indicators (KPI) is presented so that the schedule can be evaluated and compared to previous plans. A list of important KPI was discussed and agreed with the scheduling and planning teams of the company and is presented in table 4.5

Table 4.5: Key performance indicators calculated by the DSS

Description
Number of COs created
Number of orders combined
Time needed to complete every operation required
Time needed to complete the orders that had been previously planned for the current week
Percentage of time spent on orders that had been previously planned for the current week
Number of sheets processed in each machine per hour
Average setup time in each machine

The balance between these KPI is essential to minimize the total tardiness and maximize the throughput of the lithography stage, which are the criteria that must be optimized to increase the productivity of the entire facility and ensure that upstream processes are not jeopardized by this bottleneck.

Chapter 5

Solution Approach

In the previous chapter, the interaction between the user and the software was analyzed as well as some of the information needed for the DSS to be effective. In this chapter, the goal is to understand how the schedules are generated with the information fed into the system. In the first part of the chapter, a brief explanation of the thinking process that led to the decision of what formulation should be used is presented. Later, the decomposition of the problem into smaller sub-problems is explained together with the algorithms and techniques used. Finally, the new MILP models developed are described.

5.1 Different Formulations

During the process of conceptualization of the problem, different possible formulations emerged. The main difference was on the way each model tackled the printing stage, mainly the division of jobs into operations.

The first proposed model divided each job j into N operations, and chose for each operation the machine m where the operation should be performed using the decision variable:

$$X_{jom} = \begin{cases} 1 & \text{if operation } o \text{ of job } j \text{ is performed on machine } m; \\ 0 & \text{otherwise.} \end{cases} \quad \forall j \in J, m \in M, o \in [1,N]$$

To be able to compute the setup time between operations, as stated in chapter 4, it is also needed to know which colors are applied in each operation. To answer this requirement the following decision variable was also used:

$$Z_{jco} = \begin{cases} 1 & \text{if color } c \text{ of job } j \text{ is applied in operation } o; \\ 0 & \text{otherwise.} \end{cases} \quad \forall j \in J, c \in C_j, o \in [1,N]$$

This formulation is extremely flexible being able to search through a wide solution space and representing the reality of the shop-floor environment accurately. However, the computational time required to find a good solution for a large instance proved to be operationally impracticable.

The second model considered was based on formulations used in lot-sizing and scheduling, where time is split into sub-periods for each machine and each operation is assigned to one sub-period. As in the previous case, in order to compute setup times the mathematical model must assign each color to a sub-period. As in the previous formulation, this model was unable to provide acceptable solutions in a reasonable amount of time.

The implemented formulation (third approach) trades flexibility for efficiency by decreasing the solution space and passing some computational effort from the mathematical model to a pre-processing phase. This pre-processing phase creates for each job j a set of possible machine sequences S_j, and for each sequence $s \in S_j$, a set of operations O_s, defining for each possible operation o the machine $m_o \in M$ where the operations is performed. Defining all the possible operations for each job also permits to compute the time of all the possible setups outside the model by deciding which colors are applied in each operation during the pre-processing stage. In this model the only set of decision variables required to define the assignment of operations to machines is:

$$X_{js} = \begin{cases} 1 & \text{if job } j \text{ is assigned to machine sequence } s; \\ 0 & \text{otherwise.} \end{cases} \quad \forall j \in J, s \in S_j$$

In figure 5.1, the three formulations described are illustrated.

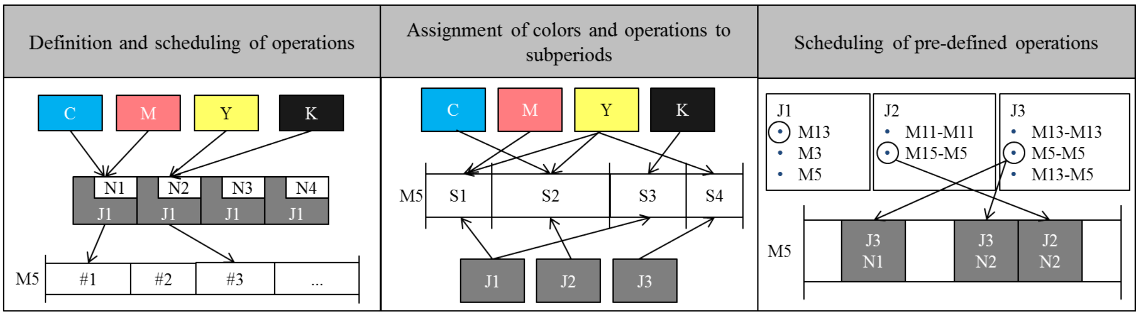

Figure 5.1: Illustrative representation of the different formulations

In terms of the sequencing of operations on each machine, the first and last formulations may use similar decision variables. For each pair of operations on the same machine, the variables decide which is performed first:

$$Y_{oo'} = \begin{cases} 1 & \text{if operation } o \text{ is performed before operation } o'; \\ 0 & \text{otherwise.} \end{cases}$$

On the other hand, in the second formulation, the sequencing is already evident in the assignment, since the decision variable must decide for each operation, the machine and the sub-period where the operation is performed:

$$Y_{osm} = \begin{cases} 1 & \text{if operation } o \text{ is performed in sub-period } s \text{ of machine } m; \\ 0 & \text{otherwise.} \end{cases}$$

5.2 Problem Decomposition

As stated in chapter 3 scheduling problems are complex combinatorial problems that are hard to solve using exact methods. In the studied problem, the complexity is even greater due to the flexibility of the shop-floor and the large number of operations that must be scheduled at a given time. To tackle these difficulties, the problem was divided into a series of smaller problems that should be solved sequentially.

Following the idea behind the Shifting Bottleneck Heuristic, that non-bottleneck stages should adapt their production according to the bottleneck, it was decided that the printing stage should be handled before the varnishing stages. This division must take into account two important details: machine M_5 can apply two colors and one varnish in one operation, and that orders can only enter the printing stage after the preparation varnishes have been applied. The strategies used to surpass these problems are explained in section 5.3.

In chapter 2, three decision processes were described: machine assignment, combination of orders and sequencing of each machine. Splitting these decision processes is not ideal, but it decreases the complexity of the global problem significantly and also promotes a more friendly validation process. In the methodology followed, a similar division was adopted. In figure 5.2 it is possible to see the different sub-problems and how they interact to find a solution. It is also depicted there the section where each of the blocks will be discussed.

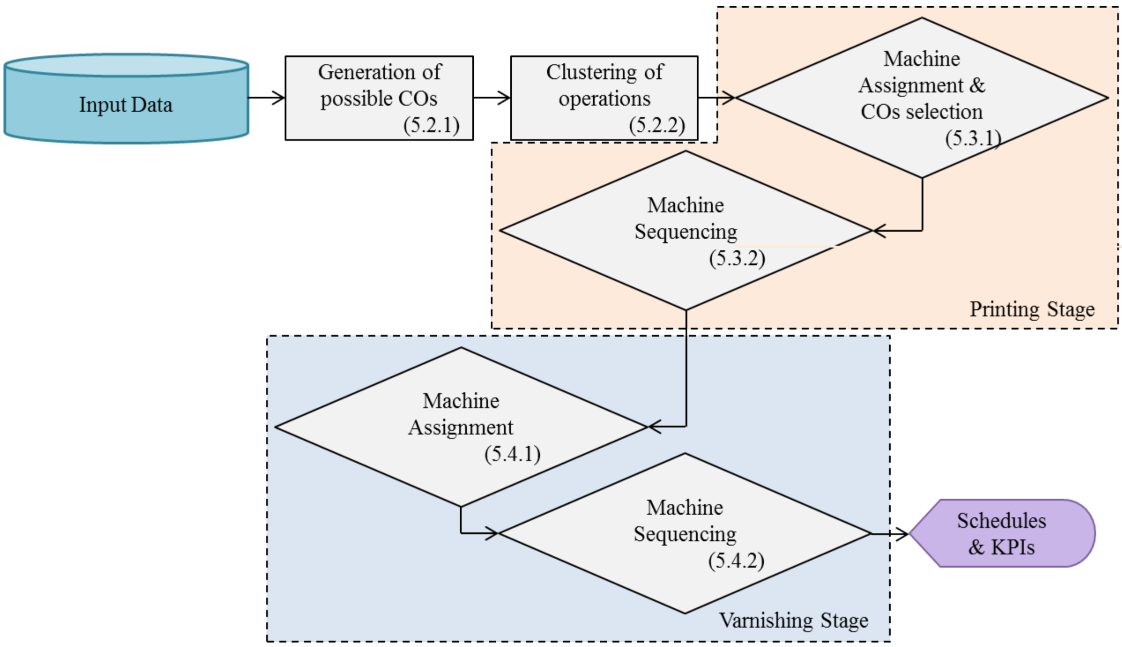

Figure 5.2: Scheme of the decomposition of the problem

It is recognized that decomposing the problem into so many sub-problems affects the quality of the solution negatively, this is a conscious trade-off between optimality and computational time. It is important to remember that in operational-level problems optimality can be lost very quickly, unexpected failures, speed losses, quality losses, and many other factors might make the scheduled plan impossible to accomplish. Being able to generate a new solution that adapts to the new conditions in a short period of time might be more advantageous than having an optimal plan.

5.2.1 Generation of possible COs

The first MILP model is going to select the COs that increase the throughput of the system. The orders should first be compared and a list of possible COs should be created to serve as input to the model. The pseudocode used to create this list of possible COs is presented in algorithm 1, where $\mathscr{C}$ is the set of COs and $\mathscr{O}$ denotes the set of orders.

Algorithm 1 Pseudocode used to create COs - Part 1

```
function CREATECOS(𝒞, 𝒪)
    𝒞 ← ∅
    for all o ∈ 𝒪 do
        for all o' ∈ 𝒪 : o' < o do
            if o' is compatible with o then
                𝒞.add(combine(o, o'))
            end if
        end for
        for all c ∈ 𝒞 do
            if c is compatible with o then
                𝒞.add(combine(o, c))
            end if
        end for
    end for
    for all c ∈ 𝒞 do
        if c violates the sheet limit then
            𝒞.delete(c)
        end if
    end for
end function
```

This algorithm allows to create all the possible COs by storing every feasible combination of orders. As exemplified in subsection 2.2.3, a CO with 3 orders might be possible even in cases where a CO with any 2 of those orders was impossible. The last step of the algorithm ensures that COs not meeting the requirements are deleted before the list is used as input for the decision models.

5.2.2 Clustering of operations

Setup times are sequence-dependent and the sequencing of operations is only done in the last step of the algorithm. To group similar operations in the same machine and time period, a clustering method is applied. Clustering techniques divide a set of objects into different groups, joining similar objects in the same group. In this case the goal is to create groups of operations in a way such that operations of the same group have similar characteristics. The indicator used to evaluate the similarity between different operations was the estimated setup time, calculated using the regressions presented in section 4.3.2.

The algorithm used to tackle this problem was the Partitioning Around Medoids (PAM) described in algorithm 2, $\mathscr{M}$ being the matrix with the setup times between operations and k the number of clusters. Medoids are the operations chosen to serve as the center of the cluster.

Algorithm 2 Partitioning Around Medoids algorithm - Part 1

function PAM($\mathscr{M}, k, \varepsilon$)
  for $c \leftarrow 1$ **to** k **do**
    select a random point to serve as $medoid_c$
  end for
  while $\frac{S_i - S_{i-1}}{S_i} \geq \varepsilon$ **do**
    for all $o \in Operations$ **do**
      $c^* \leftarrow argmin_c\{\mathscr{M}_{o,medoid_c} : c \in \{1, \cdots, k\}\}$
      $C_{c^*}.add(o)$
    end for
    for $c \leftarrow 1$ **to** k **do**
      $medoid_c \leftarrow argmin_{o_1}\{\sum_{o_2 \in C_c} \mathscr{M}_{o_1,o_2} : o_1 \in C_c\}$
    end for
    $S_i = \sum_{c \in \{1,\cdots,k\}} \sum_{o \in C_c} \mathscr{M}_{medoid_c,o}$
  end while
end function

The algorithm starts by choosing k random operations to serve as medoids of the k clusters. Then each of the other operations is assigned to the cluster, whose medoid is closest. Afterwards, new medoids are chosen based on the summation of the distances to the other points in the same cluster. When the previous step is finished, the operations are reassigned and new medoids are chosen. This iterative process ends when a given termination criteria is met (in this case, the convergence of the total distance is considered, where S_i is the total distance calculated in iteration i).

Since each operation is already assigned to one machine, this algorithm should be run for each machine and there is no need to calculate setups between operations in different machines, decreasing significantly the computational time required to perform this task.

5.3 Printing Stage

In this section each model will be explained thoroughly. Firstly, the general formulation of the problem is introduced, then each subsection will start by referring the decision variables, the model used is then described.

In the printing stage for each job j in the set of jobs J, a subset of sequences $S_j \subset S$ is given, each sequence s comprising a subset of operations $O_s \subset O$. Since printing and varnishing stages are considered in different models, here, the set O only considers printing operations. In the pre-processing, the machine where each operation o is performed is already defined and is therefore represented by $m_o \in M$, the job and the sequence of each operation can also be identified as j_o and s_o respectively. J includes both COs and Orders, but not every job must be completed. For each order i, only one job j can be chosen, so it is important to define the set of orders I and the subset of jobs that cover each order i, $J_i \subset J$. The subset of operations that are performed on machine m is given by $O_m \subset O$. Finally, to tackle the rolling horizon described in subsection 4.5 the subset of fixed sequences S_{fixed} and the subset of fixed operations O_{fixed} must be defined. In figure 5.3 a small instance containing two orders that can be combined is illustrated.

<table>
<tr><td colspan="8">I_1</td><td colspan="5">I_2</td></tr>
<tr><td colspan="5">J_1</td><td colspan="4">J_2 (CO)</td><td colspan="4">J_3</td></tr>
<tr><td>S_1</td><td colspan="2">S_2</td><td colspan="2">S_3</td><td colspan="3">S_4</td><td>S_5</td><td colspan="3">S_6</td><td>S_7</td></tr>
<tr><td>O_1</td><td>O_2</td><td>O_3</td><td>O_4</td><td>O_5</td><td>O_6</td><td>O_7</td><td>O_8</td><td>O_9</td><td>O_{10}</td><td>O_{11}</td><td>O_{12}</td><td>O_{13}</td></tr>
<tr><td>M_{11}</td><td>M_5</td><td>M_5</td><td>M_{13}</td><td>M_{13}</td><td>M_5</td><td>M_5</td><td>M_5</td><td>M_{15}</td><td>M_5</td><td>M_5</td><td>M_5</td><td>M_{15}</td></tr>
</table>

Figure 5.3: Example of a small instance for the printing stage with two orders (I_1 and I_2), where order I_1 can be produced alone in job J_1 in three alternative sequences, order I_2 can be produced alone in job J_3 in two sequences, or the orders can be combined into J_2 with two sequences.

To be able to represent the reality of the facility, some parameters are required. For each job the quantity of sheets required q_j, the due date d_j, the sheet used f_j and the priority level p_j must be defined. The actual speed of each machine v_m is also an input for every model. In the models presented, G represents a large integer number. Other parameters will be introduced during the next subsections since they are only used by a specific model.

5.3.1 Machine Assignment and Combination of orders in the Printing Stage

The main goals of this first model are to decide which machine sequences should be chosen for each job, and what COs should be processed. Furthermore, it should also indicate what operations are going to be performed during time interval H_1 defined by $[0, h_1]$ and what operations can be performed during time interval H_2 defined by $[h_1, h_2]$. An illustrative example of a possible output of this model is presented in figure 5.4.

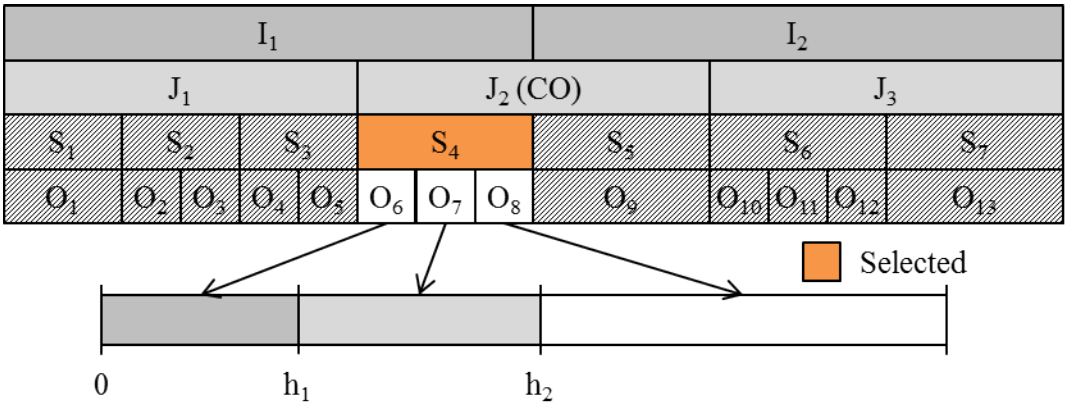

Figure 5.4: Schematic representation of the machine assignment model for the printing stage

The decision variables used are the following:

$X_s = \begin{cases} 1 & \text{if sequence s is going to be performed;} \\ 0 & \text{otherwise.} \end{cases}$

$Y_{1_o} = \begin{cases} 1 & \text{if operation } o \text{ performed during } H_1; \\ 0 & \text{otherwise.} \end{cases}$

$Y_{2_o} = \begin{cases} 1 & \text{if operation } o \text{ is performed during } H_2; \\ 0 & \text{otherwise.} \end{cases}$

$W_{cm} = \begin{cases} 1 & \text{if operations of cluster } c \text{ are performed on machine } m; \\ 0 & \text{otherwise.} \end{cases}$

$W_{1_{cm}} = \begin{cases} 1 & \text{if operations of cluster } c \text{ are performed on machine } m \text{ during time period } H_1; \\ 0 & \text{otherwise.} \end{cases}$

$W_{2_{cm}} = \begin{cases} 1 & \text{if operations of cluster } c \text{ are performed on machine } m \text{ during time period } H_2; \\ 0 & \text{otherwise.} \end{cases}$

Z: Makespan;

δ: Deviation between h_1 and the time required to perform every operation in H_1

L_f: Lack of sheet f to perform every job that requires sheet f in H_1

As stated previously, exactly one job must complete each order and only one sequence can be chosen for each job. To guarantee the satisfaction of these requirements, the constraint presented in (5.1) is used.

$$\sum_{j \in J_i} \sum_{s \in S_j} X_s = 1 \qquad \forall i \in I \tag{5.1}$$

Constraints (5.2) and (5.3) are created to ensure that the time required to perform operations during H_1 and H_2 does not exceed the duration of these intervals. It is important to note that since the sequencing is not done in this phase of the system, the expected setup times are not yet known, so an average setup time is used for each machine, $\overline{st_m}$.

$$h_1 + \delta \geq \sum_{o \in O_m} Y_{1_o} * (\frac{q_{j_o}}{v_m} + \overline{st_m}) \qquad \forall m \in M \tag{5.2}$$

$$h_1 + h_2 + \delta \geq \sum_{o \in O_m} (Y_{1_o} + Y_{2_o}) * (\frac{q_{j_o}}{v_m} + \overline{st_m}) \qquad \forall m \in M \tag{5.3}$$

When deciding the operations to be performed in each period, it is important to guarantee that the previous operations of the same job are also performed in that period, or in a previous period. These restrictions are represented by equations (5.4) and (5.5).

$$Y_{1_{o'}} \leq Y_{1_o} \qquad \forall j \in J, s \in S_j, o, o' \in O_s : o' > o \tag{5.4}$$

$$Y_{2_{o'}} \leq Y_{1_o} + Y_{2_o} \qquad \forall j \in J, s \in S_j, o, o' \in O_s : o' > o \tag{5.5}$$

The coherence between sequences and operations also needs to be taken into account. If a sequence is chosen, only operations belonging to that sequence can be performed, which is tackled by (5.6).

$$Y_{1_o} + Y_{2_o} \leq X_s \qquad \forall s \in S, o \in O_s \tag{5.6}$$

It must also be guaranteed that jobs that were fixed in the previous day are not changed and that operations that were fixed are performed in H_1, so equations (5.7) and (5.8) are used.

$$X_s = 1 \qquad \forall s \in S_{fixed} \tag{5.7}$$

$$Y_{1_o} = 1 \qquad \forall o \in O_{fixed} \tag{5.8}$$

Constraints to connect the decision variables with the variables used in the objective function are also required. Constraints (5.9)-(5.11) ensure that every different cluster of operations is considered in variables W, W_1 and W_2.

$$W_{c_o m_o} \geq X_s \qquad \forall s \in S, o \in O_s \tag{5.9}$$

$$W_{1_{c_o m_o}} \geq Y_{1_o} \qquad \forall o \in O \tag{5.10}$$

$$W_{2_{c_o m_o}} \geq Y_{2_o} \qquad \forall o \in O \tag{5.11}$$

The lack of sheet f to perform the appropriate jobs is represented by equation (5.12), where $J_f^{*^1}$ refers to the subset of jobs that use sheet f and that have not suffered any operation of color or varnish. The jobs that have already been through one or more operations do not need to use sheet from the stock because they already have specific sheets assigned to them.

$$L_f \geq \sum_{j \in J_f^{*1}} \sum_{s \in S_j} \sum_{o \in O_s} Y_{1_o} * q_j - stock_f \qquad \forall f \in F \tag{5.12}$$

The last constraint is the definition of the makespan. The completion time of each machine is obtained by the sum of the processing times of each operation assigned to that machine, and an average setup time for each operation assigned. The makespan corresponds the maximum of all the completion times of the machines.

$$Z \geq \sum_{o \in O_m} X_{s_o} * \left(\frac{q_{j_o}}{v_m} + \overline{st_m}\right) \qquad \forall m \in M \tag{5.13}$$

Finally, the objective function is:

minimize

$$\begin{aligned} & w_1 Z + w_2 \sum f \in F L_f + w_3 \sum_{c \in C} \sum_{m \in M} W_{cm} + w_4 \sum_{c \in C} \sum_{m \in M} W_{1_{cm}} + w_5 \sum_{c \in C} \sum_{m \in M} W_{2_{cm}} + \\ & \sum_{o \in O} (X_{s_o} - Y_{1_o}) * w_{6_{p_{jo}}} * \frac{q_{j_o}}{v_{m_o}} + \sum_{o \in O} (X_{s_o} - Y_{1_o} - Y_{2_o}) * w_{7_{p_{jo}}} * \frac{q_{j_o}}{v_{m_o}} + w_8 \sum_{j \in J} waste_j + \\ & w_9 \sum_{j \in J^{*2}} \sum_{s \in S_j^{*3}} X_s * q_j + w_{10} \delta \end{aligned} \tag{5.14}$$

The objective function presented has a large number of terms, so each term is explained individually. The minimization of makespan Z aims to balance the work load among machines. The minimum makespan is achieved in this case when all machines have very similar completion times. The second term refers to the number of operations performed in H_1 that do not have enough sheet ready to be produced. The minimization of the third term aims to group similar operations in the same machine, while terms 4 and 5 aim to group similar operations in the same time period.

The sixth and seventh terms penalize not performing each operation in time periods H_1 and H_2, the weights are indexed to the priority level of each job, so that jobs that are already late and jobs that are due to the current week are more penalized than jobs that have more time to be completed.

Term 8 is used to penalize the difference between the actual amount of sheets required by CO j and the sum of sheets required by each individual order contained in that CO. This difference is represented by the parameter $waste_j$, for jobs that are single orders, the parameter equals 0. In the ninth term, J^{*^2} is the subset of jobs that require finishing varnish and S^{*^3} is the subset of sequences that end in a machine with a varnishing unit, in the studied case, machine M_5. This term is used because machine M_5 can also apply finishing varnish and therefore is advantageous to apply colors and varnish in one operation rather than requiring two operations.

Finally, the last term penalizes δ. The duration of H_1 could be a hard constraint, however, to allow the programmer to fix as many operations as intended without turning the model infeasible, the deviation δ was added.

Having a large number of terms in the objective function requires an additional effort to balance the weights in such a way that the interaction between the different objectives produces viable solutions. To tackle this issue, the terms were divided into hierarchies according to their relative importance according to the decision-makers of the company. This topic is revisited in section 6.1.

5.3.2 Sequencing in the Printing Stage

With the operations already assigned, the resulting sub-problem has the characteristics of a reentrant job shop schedule, keeping in mind that consecutive operations can be performed on the same machine. In this sub-problem the subset of operations required for each job O_j is already known since the machine sequence was already chosen and the jobs that were discarded in the previous sub-problem are not considered here, so the set J only contains the jobs to be completed. Besides, the set of operations O in this model, only considers the operations that were chosen to be in H_1 in the previous model. The expected output of this sub-problem is the sequencing of the operations with a starting time for each operation o. An illustrative example of a possible solution of this sub-problem is presented in figure 5.5

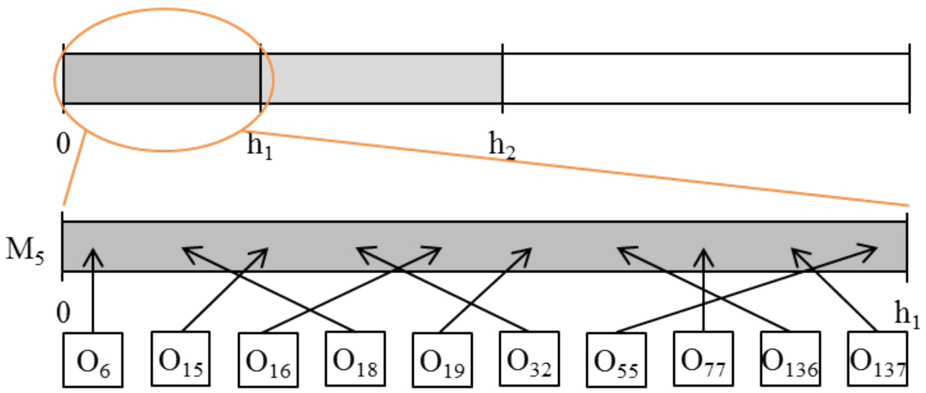

Figure 5.5: Schematic representation of the sequencing model for the printing stage

The decision variables present in this model are:

$$Y_{oo'} = \begin{cases} 1 & \text{if operation } o' \text{ is performed later than operation } o\text{ ;} \\ 0 & \text{if operation } o \text{ is performed later than operation } o'. \end{cases}$$

T_o: Starting time of operation o;

τ_j: Tardiness of job j;

Z_m: Makespan of machine m;

Typical requirements in scheduling problems are the no overlap constraints and the precedence constraints. The no overlap constraints represented in this case by (5.15) and (5.16) ensure that one operation does not start before previous ones have finished and the setup has been completed.

$$T_{o'} + (1 - Y_{oo'}) * G \geq T_o + \frac{q_{j_o}}{v_m} + st_{oo'} \qquad \forall m \in M,\, o, o' \in O_m : o' > o \tag{5.15}$$

$$T_o + (Y_{oo'}) * G \geq T_{o'} + \frac{q_{j_{o'}}}{v_m} + st_{o'o} \qquad \forall m \in M,\, o, o' \in O_m : o' > o \tag{5.16}$$

Note that for each pair of operations (o, o') performed on the same machine m, one of the previous restrictions will be ignored depending on the value of $Y_{oo'}$. If $Y_{oo'} = 1$ the first constraint is equivalent to $T_{o'} \geq T_o + \frac{q_j}{v_m} + st_{oo'}$ ensuring that o' only starts after the processing time of o, $\frac{q_j}{v_m}$, and the setup time between o and o', $st_{oo'}$. The second restriction will be non-active since $T_o + \infty$ will always be greater than the second term.

Precedence constraints guarantee that two operations of the same job do not occur at the same time. In the case under study it was decided to create two different precedence constraints, one for operations on the same machine and another for operations on different machines.

$$T_{o'} \geq T_o + \frac{q_j}{v_{m_o}} \qquad \forall j \in J, o, o' \in O_j : o' > o, m_o = m_{o'} \tag{5.17}$$

$$T_{o'} \geq T_o + \frac{q_j}{v_{m_o}} + gap_1 \qquad \forall j \in J, o, o' \in O_j : o' > o, m_o \neq m_{o'} \tag{5.18}$$

The first constraint for operations on the same machine only ensures that operation o is performed after operation $o-1$. The second constraint sets a gap between the ending of one operation and the beginning of the following. This gap is used to cover the time spent transporting the sheets from one machine to the next and as a safe factor to avoid that a delay in one machine spreads to other machines.

Some operational constraints must also be taken into account. If there is not enough sheet cut for the job, a gap time must be considered to allow the primary cut to prepare the sheet needed for production, ensured in (5.19) where J_{notCut} denotes the subset of jobs that do not have enough sheet ready to use.

$$T_o \geq gap_2 \qquad \forall j \in J_{notCut}, o \in O_j \tag{5.19}$$

A similar constraint is needed for jobs that still need to go through n preparation operations before they can enter the printing stage. Constraint (5.20) creates a buffer so that the varnishing stage has enough time to prepare the jobs scheduled in the printing. Here, $J_{notReady}$ is the subset of jobs that still have to go through at least one operation in the preparation stage and v_{min} the minimum processing speed of the machines used in the varnishing stage.

$$T_o \geq gap_3 + \frac{q_j}{v_{min}} * n \qquad \forall j \in J_{notReady}, o \in O_j \tag{5.20}$$

To fix the sequence of operations defined in the solution of the previous day, a list of fixed $Y_{oo'}$ denoted by $fix_{o,o'}$ is defined. The set of pairs of operations is denominated by Γ.

$$Y_{oo'} = fix_{oo'} \qquad \forall (o, o') \in \Gamma \tag{5.21}$$

Once all the operational constraints are defined, the variables that will be used in the objective functions must also be described. In this sub-problem, tardiness and makespan are considered. Tardiness is calculated as the difference between the end of production of the job and its due date, but it can not be lower than 0.

$$\tau_j + d_j \geq T_o + \frac{q_j}{v_{m_o}} \qquad \forall j \in J, o \in O_j \tag{5.22}$$

$$\tau_j \geq 0 \qquad \forall j \in J \tag{5.23}$$

Makepsan is computed for each machine as the completion time of the last operation performed on that machine and is therefore calculated as:

$$Z_{m_o} \geq T_o + \frac{q_j}{v_{m_o}} \qquad \forall j \in J, o \in O_j \tag{5.24}$$

Finally, the objective function in this sub-problem is straightforward and tries to minimize the weighted sum of makespan in each machine and tardiness in each job

$$minimize \qquad w_{11} \sum_{m \in M} Z_m + w_{12} \sum_{j \in J} \tau_j \tag{5.25}$$

5.4 Varnishing Stage

The varnishing stage is simpler to formulate since one varnish is applied in a single operation. Here, for each job j in a set of jobs J, a set of operations $O_j \subset O$ is given. Again, the set of operations O considered in these models only contains the varnishing operations. Each operation must be assigned to a machine $m \in M$. In the varnishing stage the set of jobs J only contains the jobs that were chosen during the assignment of the printing stage, so it is not important to define the set of orders.

The parameters introduced in section 5.3 will be used again.

5.4.1 Machine Assignment in the Varnishing Stages

The goal of this model is similar to the goal of the first model presented in subsection 5.3.1, the main difference relies on the fact that in the varnishing stages the number of operations to be performed is already known and the decision is on what machine each operation is going to be performed. As in the first model, it is also important to distinguish the operations that are going to be performed during time interval H_1, defined by$[0, h_1]$, H_2, defined by $[h_1, h_2]$ and later.

The decision variables in this model are:

$$X_{1_{om}} = \begin{cases} 1 & \text{if operation } o \text{ is performed on machine } m \text{ during } H_1; \\ 0 & \text{otherwise.} \end{cases}$$

$$X_{2_{om}} = \begin{cases} 1 & \text{if operation } o \text{ is performed on machine } m \text{ during } H_2; \\ 0 & \text{otherwise.} \end{cases}$$

$$X_{3_{om}} = \begin{cases} 1 & \text{if operation } o \text{ is performed on machine } m \text{ later than } H_2; \\ 0 & \text{otherwise.} \end{cases}$$

Z: Makespan;

δ: Deviation between h_1 and the time required to perform every operation in H_1

L_f: Lack of sheet f to perform every job that requires sheet f in H_1

The first constraint in this model,(5.26) guarantees that each operation is assigned to exactly one machine.

$$\sum_{m \in M} (X1_{om} + X2_{om} + X3_{om}) = 1 \qquad \forall o \in O \tag{5.26}$$

As in the printing stage, it is also needed to limit the number of operations that can be performed in H_1 and H_2. Similar constraints are then used:

$$h_1 + \delta \geq \sum_{o \in O_m} X1_{om} * (\frac{q_{j_o}}{v_m} + \overline{st_m}) \qquad \forall m \in M \tag{5.27}$$

$$h_2 + h_1 + \delta \geq \sum_{o \in O_m} (X2_{om} + X1_{om}) * (\frac{q_{j_o}}{v_m} + \overline{st_m}) \qquad \forall m \in M \tag{5.28}$$

Once again is also needed to ensure that operations only happen after or during the same time period of previous operations of the same job.

$$\sum_{m \in M} X1_{o'm} \leq \sum_{m \in M} X1_{om} \qquad \forall j \in J, o, o' \in O_j : o' > o \tag{5.29}$$

$$\sum_{m \in M} X2_{o'm} \leq \sum_{m \in M} X1_{om} + \sum_{m \in M} X2_{om} \qquad \forall j \in J, o, o' \in O_j : o' > o \tag{5.30}$$

Some operational constraints must also be taken into account, specially constraints created by the decision made during the printing stage. Some operations of the preparation stage must be finished in H_1 in order for the job to start printing, and some operations of the finishing stage can not start during H_1 because the printing has not been finished yet. These constraints are defined in (5.31) and (5.32), respectively. Here, J_α is the subset of jobs that start printing in H_1 and J_β is the subset of jobs that does not finish the printing stage during $h1$, O_p the subset of operations of preparation and O_f the subset of operations of finishing.

$$\sum_{m \in M} X1_{om} = 1 \qquad \forall j \in J_\alpha, o \in O_{p_j} \tag{5.31}$$

$$\sum_{m \in M} X1_{om} = 0 \qquad \forall j \in J_\beta, o \in O_{f_j} \tag{5.32}$$

Even though the specialization of machines should be a soft constraint, it was agreed with the company managers to keep it as an hard constraint during the initial implementation of the system. To do this, for each different type of varnish a compatibility matrix ζ_{om} must be inserted as a parameter. In this case: $\zeta_{om} = \begin{cases} 1 & \text{if operation } o \text{ can be performed on machine } m; \\ 0 & \text{otherwise.} \end{cases}$

$$X1_{om} + X2_{om} + X3_{om} \leq \zeta_{om} \qquad \forall o \in O, m \in M \tag{5.33}$$

Constraints (5.12) and (5.13) used in the first model are adapted for this model and presented in (5.34) and (5.35).

$$L_f \geq \sum_{j \in J_f^{*1}} \sum_{o \in O_s} \sum_{m \in M} X_{1_o} * q_j - stock_f \qquad \forall f \in F \tag{5.34}$$

$$Z \geq \sum_{o \in O_m} (X_{1_{om}} + X_{2_{om}} + X_{3_{om}}) * (\frac{q_{j_o}}{v_m} + \overline{st_m}) \qquad \forall m \in M \tag{5.35}$$

Finally, the objective function is represented in equation (5.36).

$$\begin{aligned} minimize \quad & w_{13}Z + w_{14} \sum_{f \in F} L_f + \sum_{o \in O} w_{15_{p_{jo}}} (1 - \sum_{m \in M} X_{1_{om}}) \frac{q_{j_o}}{v_m} + \\ & \sum_{o \in O} w_{16_{p_{jo}}} (1 - \sum_{m \in M} X_{1_{om}} - \sum_{m \in M} X_{2_{om}}) \frac{q_{j_o}}{v_m} + w_{17}\delta \end{aligned} \tag{5.36}$$

All the terms in this objective function have a correspondent term in the objective function of model 1, in subsection 5.3.1 and therefore they will not be detailed in this subsection.

5.4.2 Sequencing in the Varnishing Stages

The last model of this algorithm has the same goals as the model presented in subsection 5.3.2. Constrains presented in equations (5.15)-(5.18), (5.22) and (5.24) are also used in this model. A similar constraint to (5.19) delaying the start of jobs that do not have sheet ready for production is also considered.

To guarantee precedence constraints between the varnish stage and the printing stage it is important to define two additional parameters b_o and r_o. b_o is the deadline of operation o and r_o is the release date of operation o. As in the model presented in subsection 5.4.1 the subset of jobs that start the printing stage during H_1 is defined by J_α. O_p represents the subset of operations of preparation and O_f the subset of operations of finishing.

$$T_o + \frac{q_j}{v_{m_o}} + gap_4 \leq b_o \qquad \forall j \in J_\alpha, o \in O_{p_j} \tag{5.37}$$

$$T_o \geq r_o + gap_5 \qquad \forall j \in J, o \in O_{a_j} \tag{5.38}$$

The objective function considered is the same as the equation (5.25).

Chapter 6

Validation of the Proposed Approach

This chapter presents the process followed to validate the results of the developed DSS. As it is possible to see in the schedule illustrated in figure 1.1, at the end of this dissertation, the validation of the system was still ongoing.

In the beginning of this phase the aim was to adjust certain parameters to ensure that the DSS is robust and that the solutions generated are satisfactory. The adjustment of the weights of the objective functions and the running times of each model will be explained in more detail during this chapter.

The second part of the validation comprises the comparison between schedules generated by the DSS and the schedules used by the company. Since this phase had not been started at the end of this dissertation, it will not be discussed in this chapter.

6.1 Fine-tuning of the Objective Functions

In models with an objective function with more than one criteria, finding the optimal solution does not necessarily mean that the solution is good. In the case under study, generating a plan that minimizes the makespan, maximizing the throughput of the facility is not sufficient if the jobs' tardiness is neglected. To avoid these situations the weights used in each objective function must be carefully balanced.

A common technique to balance the objective function is to turn every term into the same unit of measurement, usually the cost. However, this is not always possible because not every cost is easily quantified, and sometimes, the information required is not available.

In the case under study, a different technique was chosen. Initial values of each weight were found based on the typical value of each term. Then, several schedules were generated. In each of the generated schedules, one or two weights were increased or decreased. Later, together with the decision makers of the company, the schedules were compared to understand which combination of weights led to the preferred results.

The value of each weight is important to serve as a basis of the DSS. Nevertheless, the user will have the ability to increase or decrease each value by a multiplying factor in order to compare different possibilities or even study if changing the current paradigms would be beneficial for the company.

6.2 Computational Tests and Generated Schedules

As previously stated in chapter 4, a balance between the quality of the solution and the time spent to achieve it must be found.

While on the machine assignment phases, the models are able to find a near-optimal, or even the optimal solution in a short period of time, in the sequencing phase, finding the optimal solution requires unpractical computational times. In figure 6.1, it is possible to see how the objective function's value behaves over time in the sequencing model of the printing stage.

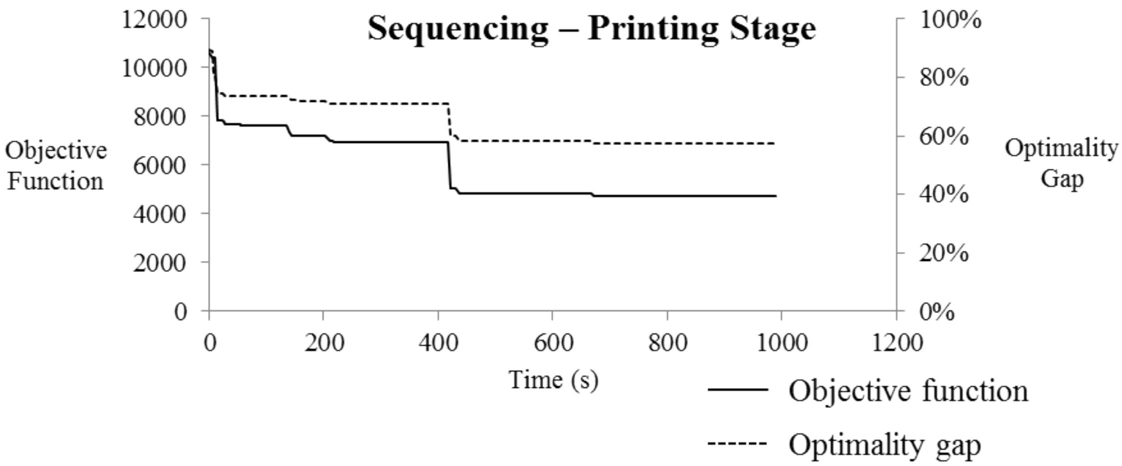

Figure 6.1: Behavior of the value of the objective value and optimality gap in the sequencing phase of the printing stage

In the previous figure, the solid line represents the value of the objective function, while the dashed line represents the optimality gap. The optimality gap is the relative difference between the current best value (upper bound), UB, and the current lower bound, LB ($gap = \frac{UB-LB}{UB}$).

The *gap* is noticeably high and suggests that the obtained solution quality is poor. Notwithstanding, this is not necessarily true, since the value of the lower bound may not be a realistic estimation of the optimal value.

For the instance used in the previous example, the value stabilizes after $\approx 450s$. However to guarantee that the system generates satisfactory results every day, it is important to continue this study with more instances.

The evolution of the optimality gap of each model is presented in table 6.1. The instance used during this tests was a real instance from the printing plant under study. It is possible to see that the machine assignment model of the varnishing stage is by far the fastest model, while the sequencing models are not able to find a near-optimal solution, or at least prove it, in the desired amount of time. In the table "Inf" means that the model was not able to find a feasible solution.

Table 6.1: Comparison of the gaps of each model over time

Model	**5s**	**10s**	**25s**	**50s**	**100s**	**250s**	**500s**	**1000s**
Machine Assignment - Printing	48%	27%	13%	3%	1.7%	1.3%	-	-
Sequencing - Printing	89%	75%	74%	74%	74%	71%	58%	57%
Machine Assignment - Varnishing	Optimality in $\approx 0.5s$							
Sequencing - Varnishing	Inf	Inf	Inf	Inf	Inf	70%	67%	67%

It is also possible to see that the sequencing in the varnishing stage has difficulties finding an initial solution, this may be explained by the release times and deadlines used in the model. This problem must be dealt with care, because if the model is not able to find a feasible solution, the confidence of the company in the DSS may be undermined. One of the methods that is being discussed to tackle this problem is to turn these constraints into soft constraints, with a large penalization in the objective function.

For the same instance, a sample of the solution generated for the detailed schedule of machine M_{11} is presented in figure 6.2. It is possible to see that the algorithm is able to group similar jobs as intended.

Order	Qt	O	Color1	Color2	Color3	Color4	Setup	Start Prod	Finish Prod	Delivery D	S Length	S Width
1000184726	854	1					-	18/06/2015 00:00	18/06/2015 00:30	05/07/2015	873	842
1000184629	600	1					01:24	18/06/2015 01:55	18/06/2015 02:16	05/07/2015	1075	830
1000184500	916	1					01:16	18/06/2015 03:33	18/06/2015 04:06	12/07/2015	873	842
1000184709	1053	1					01:43	18/06/2015 05:50	18/06/2015 06:27	05/07/2015	873	842
1000184623	600	1					01:36	18/06/2015 08:03	18/06/2015 08:25	05/07/2015	1106	850
1000182347	1200	1					01:12	18/06/2015 09:37	18/06/2015 10:21	05/07/2015	731,4	856
1000184434	550	1					01:28	18/06/2015 11:49	18/06/2015 12:09	28/06/2015	797	833
1000184454	550	1					00:38	18/06/2015 12:47	18/06/2015 13:07	28/06/2015	810	829
165	750	1					00:41	18/06/2015 13:48	18/06/2015 14:15	28/06/2015	873	842
1000183832	795	1					02:04	18/06/2015 16:19	18/06/2015 16:48	19/07/2015	759	931
1000184356	666	1					01:01	18/06/2015 17:49	18/06/2015 18:12	05/07/2015	759	931
1000184164	944	2				-	01:25	18/06/2015 19:38	18/06/2015 20:12	05/07/2015	740	931
1000184586	777	1					01:15	18/06/2015 21:27	18/06/2015 21:55	05/07/2015	740	931
1000184496	949	1					01:24	18/06/2015 23:19	18/06/2015 23:53	28/06/2015	758,9	833
1000184172	400	2					00:43	19/06/2015 00:37	19/06/2015 00:51	28/06/2015	873	842
1000184532	1713	1					00:40	19/06/2015 01:31	19/06/2015 02:33	28/06/2015	746	833
1000184415	600	2					01:09	19/06/2015 03:43	19/06/2015 04:04	05/07/2015	856	950
1000184453	1050	1					01:22	19/06/2015 05:26	19/06/2015 06:04	28/06/2015	810	829
1000184452	1050	1					00:37	19/06/2015 06:41	19/06/2015 07:19	28/06/2015	810	829
1000184219	500	1					00:40	19/06/2015 07:59	19/06/2015 08:17	05/07/2015	877	829
1000184509	600	1					00:39	19/06/2015 08:57	19/06/2015 09:18	05/07/2015	797	833
1000180980	134	2				-	00:38	19/06/2015 09:57	19/06/2015 10:02	29/03/2015	827,5	833

Figure 6.2: Example of a schedule of machine $M11$

Some of the KPI identified in section 4.6 were computed to be able to discuss the quality of the schedule generated with the decision-makers of the company. These results are presented in tables 6.2. In the next phase of the validation process, the goal is to compare these indicators with schedules generated manually by the company's scheduler.

Table 6.2: Example of KPI computed by the DSS

KPI - General				**Value**
Number of COs created				36
Number of orders combined				78
% time spent performing operations in the plan				78
KPI - Printing	$\mathbf{M_5}$	$\mathbf{M_{11}}$	$\mathbf{M_{13}}$	$\mathbf{M_{15}}$
Time needed to complete every operation required (h)	264	263	237	263
Time needed to complete every operation in the plan (h)	103	94	103	129
Number of sheets processed in each machine per hour	1949	1270	1120	1720
Average setup time in each machine (min)	69	113	73	95
KPI - Varnishing	$\mathbf{M_2}$	$\mathbf{M_3}$	$\mathbf{M_4}$	$\mathbf{M_6}$
Time needed to complete every operation required (h)	167	270	112	270
Time needed to complete every operation in the plan (h)	95	214	75	101
Number of sheets processed in each machine per hour	1805	1480	1700	20
Average setup time in each machine (min)	45	45	30	40

Chapter 7

Conclusions and Future Work

This dissertations focused on the production scheduling of the printing plant of a Portuguese company. The main goal was to increase the throughput of the facility, and at the same time guarantee that the due dates of each job were fulfilled. The project started with the assessment of the requirements of the problem. Then, shop floor data was analyzed to find improvement opportunities. Meanwhile, the development of the decision support system was initiated. The solution proposed is based on the decomposition of the problem into smaller sub-problems and the development of MILP models to tackle each of them.

This dissertation has not contributed just with an optimization-based DSS. Other prescriptive analysis was performed, as well as some descriptive analysis. For instance, being able to characterize and quantify setups, and therefore anticipate setup times, is a major benefit for a facility where time lost with setups can go up to almost 60% of the planned production time in some machines. Furthermore, confirming that the printing stage is the bottleneck of the lithography might encourage the company managers to focus their efforts on optimizing this stage in order to increase the productivity of the entire factory. Moreover, realizing the importance of using the actual output rather than the nominal speed might also correct some paradigms that were created based on wrong data.

Improvement opportunities were also found in different phases of the planning process. In the machine assignment process, the predefined machine sequences are often ignored because they do not adapt to the reality of the facility. This creates a mismatch between the plans generated by the planning department and the actual output of the printing plant. In the sequencing phase, it was possible to assess that the current process decreases the difficulty of the task at the expense of increasing the work in process. In fact, the time spent by a job between operations increased by almost one entire day during the last year.

The differentiation between detailed scheduling and aggregated scheduling answers the needs of the users increasing the visibility of the schedule from one to four days, and giving an estimation of what can be completed until the end of the week. This information may be useful to balance the amount of work planned for a week with the current reality of the facility.

Throughout this project different formulations were tested to decide the future of the project.

The formulation chosen shrinks the solution space by defining in a pre-processing phase the allowed machine sequences for each job. Moreover, in this pre-processing, it is possible to predefine the colors that are applied in each operation, decreasing even further the computational effort of the models.

The decomposition of the problem can be seen as a double decomposition. First, the bottleneck stage is separated from the other stages, in order to improve the throughput of the entire plant. Second, the machine assignment process is handled before the sequencing of the operations in each machine. The goal of this decomposition is to find a good balance between computational time and solution quality.

The results' validation is still an ongoing process. The computational tests show that there is still room to improve the solution quality of the sequencing models. However, fix-and-optimize heuristics can be used to increase the efficiency of these models. Another promising alternative is to use constraint programming. The developed MILP models can be easily adapted to this paradigm. Nevertheless, the increased flexibility of these reentrant job shop scheduling problems may impact CP's performance.

The problem considered in this dissertation has not been as well studied as other scheduling problems. The main difference between this printing plant and a common flexible job shop environment is the fact that the number of operations required to complete a job is not pre-defined. Even though metaheuristics are not considered in this dissertation, it would be interesting to see how well they would be able to tackle this problem. The flexibility of the problem and the size of the instances considered make the metaheuristics an appealing choice, although it would not be straightforward to define a smart search procedure.

Hopefully, our work will motivate other researchers to study the printing plant scheduling problem and to apply optimization techniques to real-world problems.

Bibliography

Adams, J., Balas, E., and Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34(3):391–401.

Allahverdi, A., Gupta, J. N. D., and Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *Omega*.

Allahverdi, A., Ng, C. T., Cheng, T. C. E., and Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985–1032.

Almada Lobo, B. (2005). Planeamento e Escalonamento da Produção - visão global e caso de estudo. *Faculdade de Engenharia da Universidade do Porto*.

Bagheri, A. and Zandieh, M. (2011). Bi-criteria flexible job-shop scheduling with sequence-dependent setup times—Variable neighborhood search approach. *Journal of Manufacturing Systems*, 30(1):8–15.

Chen, J. C., Chen, K. H., Wu, J. J., and Chen, C. W. (2008). A study of the flexible job shop scheduling problem with parallel machines and reentrant process. *The International Journal of Advanced Manufacturing Technology*, 39(3-4):344–354.

Chen, J. C., Wu, C.-C., Chen, C.-W., and Chen, K.-H. (2012). Flexible job shop scheduling with parallel machines using Genetic Algorithm and Grouping Genetic Algorithm. *Expert Systems with Applications*, 39(11):10016–10021.

Fattahi, P., Saidi Mehrabad, M., and Jolai, F. (2007). Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. *Journal of Intelligent Manufacturing*, 18(3):331–342.

Floudas, C. A. and Lin, X. (2005). Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Annals of Operations Research*, 139(1):131–162.

Garey, M. and Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. A Series of books in the mathematical sciences. W. H. Freeman.

Glover, F. (1989). Tabu Search - Part II. *ORSA journal on Computing*, 2 1(3):4–32.

Goldberg, D., Deb, K., and Korb, B. (1989). Messy genetic algorithms: motivation, analysis, and first results. *Complex Systems*, (3):493–530.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Kan, a. H. G. R. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey.

Graves, S. C. (1981). A Review on Production Scheduling. *JSTOR*, 29(4):646–675.

Kirkpatrick (1983). Optimization by SA. 220(4598):671–680.

Liao, C.-J. and You, C.-T. (1992). An improved formulation for the job-shop scheduling problem. *The Journal of the Operational Research Society*, 43(11):pp. 1047–1054.

Manne, A. S. (1960). On the job-shop scheduling problem. *Operations Research*, 8(2):219–223.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search.

Mousakhani, M. (2013). Sequence-dependent setup time flexible job shop scheduling problem to minimise total tardiness. *International Journal of Production Research*, 51(12):3476–3487.

Pan, C.-H. (1997). A study of integer programming formulations for scheduling problems. *International Journal of Systems Science*, 28(1):33–41.

Pan, J. C. H. and Chen, J. S. (2005). Mixed binary integer programming formulations for the reentrant job shop scheduling problem. *Computers and Operations Research*, 32(5):1197–1212.

Pezzella, F., Morganti, G., and Ciaschetti, G. (2008). A genetic algorithm for the Flexible Job-shop Scheduling Problem. *Computers and Operations Research*, 35(10):3202–3212.

Pinedo, M. L. (2005). *Planning and Scheduling in Manufacturing and Services*. Springer.

Ponnambalam, S. G., Jawahar, N., Girish, B. S., Ant, A., Optimization, C., and Job, F. (2005). An Ant Colony Optimization algorithm for Flexible Job shop scheduling problem. (Pinedo).

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Roy, B. and Sussmann, B. (1964). Les problemes d'ordonnancement avec contraintes disjonctives. *Note ds*, 9.

Topaloglu, S. and Kilincli, G. (2009). A modified shifting bottleneck heuristic for the reentrant job shop scheduling problem with makespan minimization. *International Journal of Advanced Manufacturing Technology*, 44(7-8):781–794.

Wagner, H. M. (1959). An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly*, 6(2):131–140.

Wilson, J. M. (1989). Alternative formulations of a flow-shop scheduling problem. *The Journal of the Operational Research Society*, 40(4):pp. 395–399.

Wilson, S. and Morales, G. (2012). Formulações matemáticas e estratégias de resolução para o problema job shop clássico.

Yazdani, M., Amiri, M., and Zandieh, M. (2010). Flexible job-shop scheduling with parallel variable neighborhood search algorithm. *Expert Systems with Applications*, 37(1):678–687.

Appendix A

Predefined Machine Allocation Matrix

N Colors	Type	N Sheets	1st Option	2nd Option	3rd Option	4th Option
1	Pantone	<4000	M13	M5	-	-
		>4000	M13	M5	-	-
2	Pantone	<4000	M13	M5	-	-
		>4000	M5	M13	-	-
3	Pantone	<4000	M13-M13	M5-M5	M13-M5	-
		>4000	M5-M5	M13-M13	M13-M5	-
4	Pantone	<4000	M13-M13	M5-M5	M11	-
		>4000	M5-M5	M11	M13-M13	-
	CMYK	>0	M11	M15	M5-M5	(M13)*2
5	Pantone	<4000	M11-M13	(M13)*3	(M5)*3	-
		>4000	M11-M13	M11-M5	(M5)*3	-
	CMYK	<4000	M13-M11	M15	M11-M5	-
		>4000	M15	M11-M5	M13-M11	-
6	Pantone	<4000	M11-M5	M11-M13	(M13)*3	-
		>4000	M5-M5-M5	M11-M5	M11-M13	-
	CMYK	<4000	M11-M5	M11-M13	M1 5	-
		>4000	M15	M11-M5	M13-M13	-
7	Pantone	<6000	M11-M13-M13	M11-M5-M5	M11-M11	(M13)*4
		>6000	M11-M13-M13	M11-M5-M5	M15	(M13)*4
	CMYK	<6000	M11-M11	M15	-	-
		>6000	M15	M11-M11	-	-
8	Pantone	<8000	M11-M5-M5	M11-M13-M5	M11-M11	(M13)*4
		>8000	M11-M5-M5	M11-M13-M5	M15-M13	(M13)*4
	CMYK	<8000	M11-M11	M15-M5	-	-
		>8000	M15-M5	M11-M11	-	-

Appendix B

Real Machine Allocation Frequency Matrix

N Colors	Type	N Sheets	Machine	O_1 (%)	O_2 (%)	O_3 (%)	O_4 (%)
1	Pantone	<4000	M5	80			
			M11				
			M13	20			
			M15				
		>4000	M5	100			
			M11				
			M13				
			M15				
2	Pantone	<4000	M5	47			
			M11				
			M13	53			
			M15				
		>4000	M5	84			
			M11				
			M13	13			
			M15	3			
3	Pantone	<4000	M5	16	18		
			M11	14			
			M13	70	68		
			M15				
		>4000	M5	17	23		
			M11	70			
			M13	12	5		
			M15	1			

N Colors	Type	N Sheets	Machine	O_1 (%)	O_2 (%)	O_3 (%)	O_4 (%)
4	Pantone	<4000	M5	7	8		
			M11	30			
			M13	63	62		
			M15	1			
		>4000	M5	11	14		
			M11	81			
			M13	7	4		
			M15	1			
	CMYK	>0	M5				
			M11	98			
			M13				
			M15	2			
5	Pantone	<4000	M5	4	25	4	
			M11	23	1		
			M13	54	54	52	
			M15	19			
		>4000	M5	6	50	9	
			M11	46	1		
			M13	13	13	10	
			M15	36			
	CMYK	<4000	M5		13		
			M11	21	23		
			M13	22	7		
			M15	57			
		>4000	M5		27		
			M11	27	3		
			M13	3			
			M15	70			

N Colors	**Type**	**N Sheets**	**Machine**	O_1 (%)	O_2 (%)	O_3 (%)	O_4 (%)
6	Pantone	<6000	M5		23	9	
			M11	16	2		
			M13	48	48	44	
			M15	36			
		>6000	M5		30	6	
			M11	17	1		
			M13	7	5	5	
			M15	76			
	CMYK	<6000	M5		6	1	
			M11	19	39		
			M13	44	18	6	
			M15	37			
		>6000	M5		14		
			M11	11	9		
			M13	9			
			M15	90			
7	Pantone	<6000	M5	1	78	3	2
			M11	4	3	1	
			M13	14	14	12	
			M15	81			
		>6000	M5		88	2	
			M11	9	9		
			M13				
			M15	91			
	CMYK	<6000	M5		30	1	
			M11	24	14	6	
			M13	14	23	22	
			M15	62			
		>6000	M5		38		
			M11	25	25		
			M13				
			M15	75			

N Colors	**Type**	**N Sheets**	**Machine**	O_1 (%)	O_2 (%)	O_3 (%)	O_4 (%)
8	Pantone	>0	M5		87		1
			M11				
			M13	8	8	8	7
			M15	92			
	CMYK	>0	M5		100		
			M11				
			M13				
			M15	100			