Hybrid flow shop scheduling with prescription constraints on jobs

Nicolas SIMONNEAU

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Industrial and Systems Engineering

December 5, 2003
Nantes, France

Dr. John P. Shewchuk, Chair, Co-advisor
Dr. Stéphane Dauzère-Pérès, Co-advisor
Dr. Michael P. Deisenroth

Keywords: Two-stage Hybrid Flow Shop, Prescription Constraints, Optimization, Integer Programming Software, Heuristic Software, Aerospace Industry

Hybrid flow shop scheduling with prescription constraints on jobs

Nicolas SIMONNEAU

ABSTRACT

The sponsor of the thesis is the Composite Unit of AIRBUS Nantes plant, which manufactures aircraft composite. The basic process to manufacture composite parts is to lay-up raw composite material on a tool and involves very costly means and raw material.

This process can be modeled as a two-stage hybrid flow shop problem with specific constraints, particularly prescription constraints on the jobs.

This thesis restates the practical problem as a scheduling problem by doing hypotheses and restrictions. Then, it designs a mathematical model based on time-indexed variables. This model has been implemented in an IP solver to solve real based scenarios. A heuristic algorithm is developed for obtaining good solutions quickly. Finally, the heuristic is used to increase the execution speed of the IP solver. This thesis concludes by a discussion on the advantages and disadvantages of each option (IP solver vs. heuristic software) for the sponsor.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Dr. John P. Shewchuk for his support and guidance throughout the research process. Given the geographical distance between us, the circumstances were not very easy but Dr. Shewchuk has always been available when his expertise and his professionalism were required. I would like also to thank him for his acceptance of being chair of my committee, especially in this particular context of dual degree program with Ecole des Mines de Nantes. I am aware that it was sometimes difficult to deal with this aspect.

I express my sincere gratitude to Dr. Stéphane Dauzère-Pérès for being co-advisor in my committee. Dr. Dauzère-Pérès initiated the design and management of the dual degree program, which enabled me to carry out the Master of Science in Industrial and Systems Engineering at Virginia Tech. His advice and encouragement during the writing of this thesis were very helpful, and I would like to thank him very much for this.

I would like to thank Dr. Deisenroth for having agreed to be a member of my committee. Again, the dual degree program has been a particular aspect of this thesis that he handled in a very professional way.

Finally, I would like to express my gratitude to Michel Rouby, who was my boss at AIRBUS and who gave me, during my internship, the opportunity to perform some research on the composite manufacturing scheduling problem.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CHAPTER 1. INTRODUCTION

1.1. Sponsor presentation and project background

The sponsor of the thesis is the composite unit of the **AIRBUS Nantes plant**. This part describes the AIRBUS group, then Nantes plant and finally explains the context of the research project.

1.1.1. AIRBUS group

The EADS[1] group is born in July 2000 from the fusion of three aeronautical companies: Aérospatiale-Matra in France, DASA[2] in Germany and CASA[3] in Spain.

AIRBUS represents a large part of the EADS aeronautical business activities. AIRBUS is possessed by EADS at 80 % and by the English group BAE Systems[4] at 20 % (Figure 1). Including four major European nations, AIRBUS is the example of a successful collaboration within Europe.

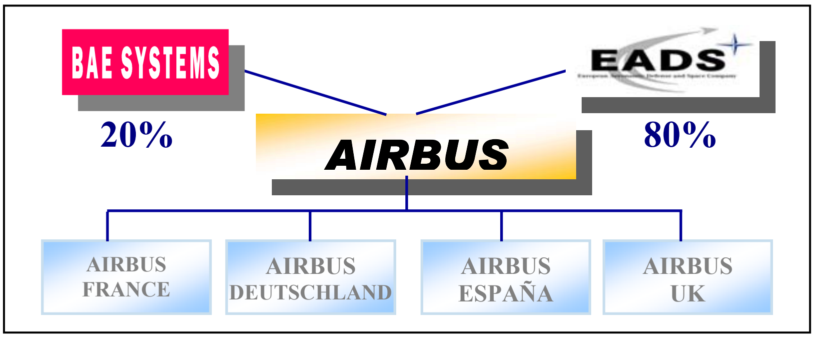

Figure 1: AIRBUS organization

AIRBUS owns **16 industrial sites**, representing 46,000 employees all over the world. The turnover for 2002 was **€ 19.5 billion**, which represents **303 aircrafts delivered**.

Aircraft manufacturing is an international process, structured around key manufacturing units: each one is responsible for producing a complete section of the aircraft, which is then deliver to the final assembly lines (Figure 2).

[1] European Aeronautic Defense and Space Company
[2] Daimler-Chrysler Aerospace SA
[3] Construcciones Aeronaúticas SA
[4] British Aerospace Systems

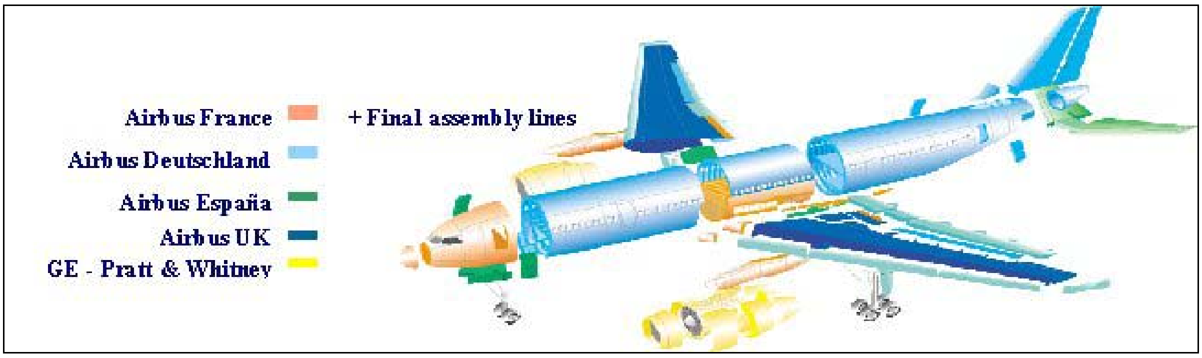

Figure 2: A300 industrial competencies

AIRBUS has always developed new aircrafts following the **family concept** (Figure 3). The idea is to manufacture an integrated family of aircrafts with some common characteristics (cockpit ergonomics, procedures, handling characteristics, pilots adapted training). The result is considerable savings in operating costs for the airlines.

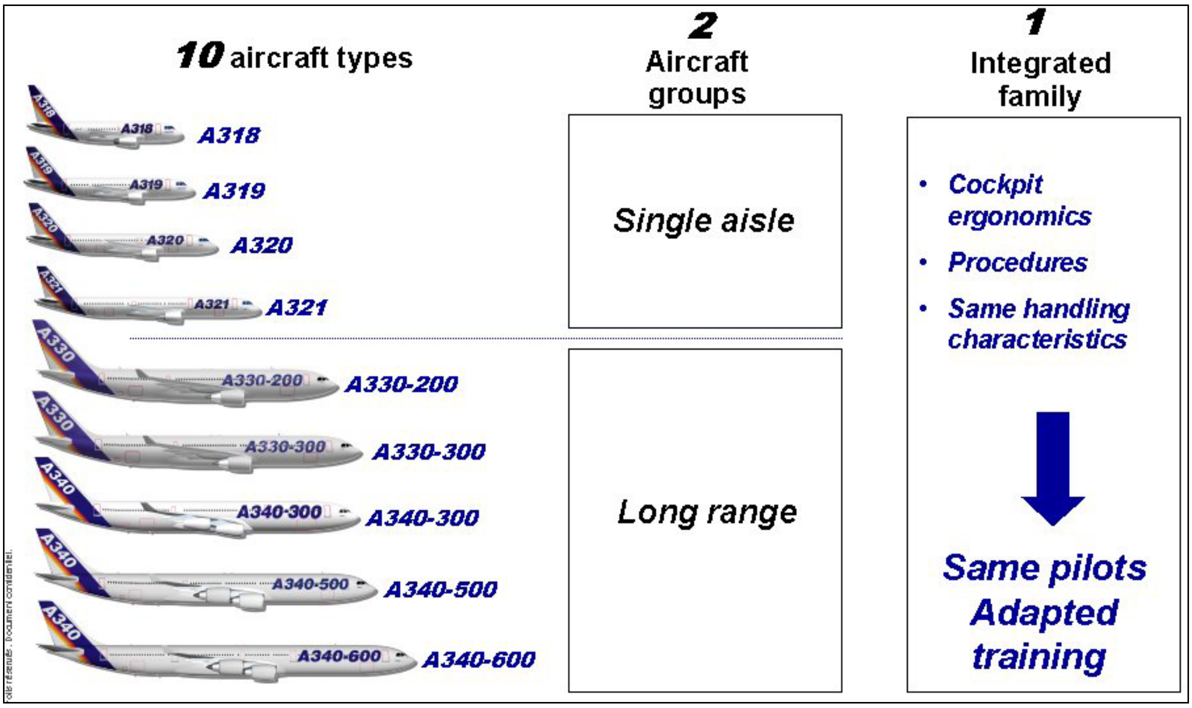

Figure 3: AIRBUS family

1.1.2. AIRBUS Nantes plant

In the AIRBUS organization, Nantes plant is referred as MUN[5]. **Nantes plant** counts approximately **1800 employees** and is specialized in:

- **Assembly of center wing box** (for the whole AIRBUS range of aircrafts),
- Keel beams,
- Ailerons,
- Inner cool,
- Landing gear doors.

[5] Manufacturing fUselage Nantes

The plant activities can be separated into 4 main parts:

- **Mechanical** part: milling on numerically controlled machine-tool ;
- **Composite** part: composite-made part manufacturing
- **Chemical treatment** part: chemical machining and surface treatments of large-size mechanical part in aluminum
- **Assembly** part: assembling of subsets of the center wing box, the fuselage and aerofoil boxes

In Nantes, many parts are in CFRP (Composite Fiber Reinforced Plastic). The two main advantages of a composite part are its lightness and its strength, which explains why it is largely used in the aeronautics industry. The skills of Nantes plant as regards the **composite technology** are recognized inside AIRBUS.

1.1.3. Context and subject

AIRBUS started to manufacture the **A380 aircraft** in January 2002: the largest airliner ever designed with a capacity of 550 seats for a 15,000-km range (Figure 4).

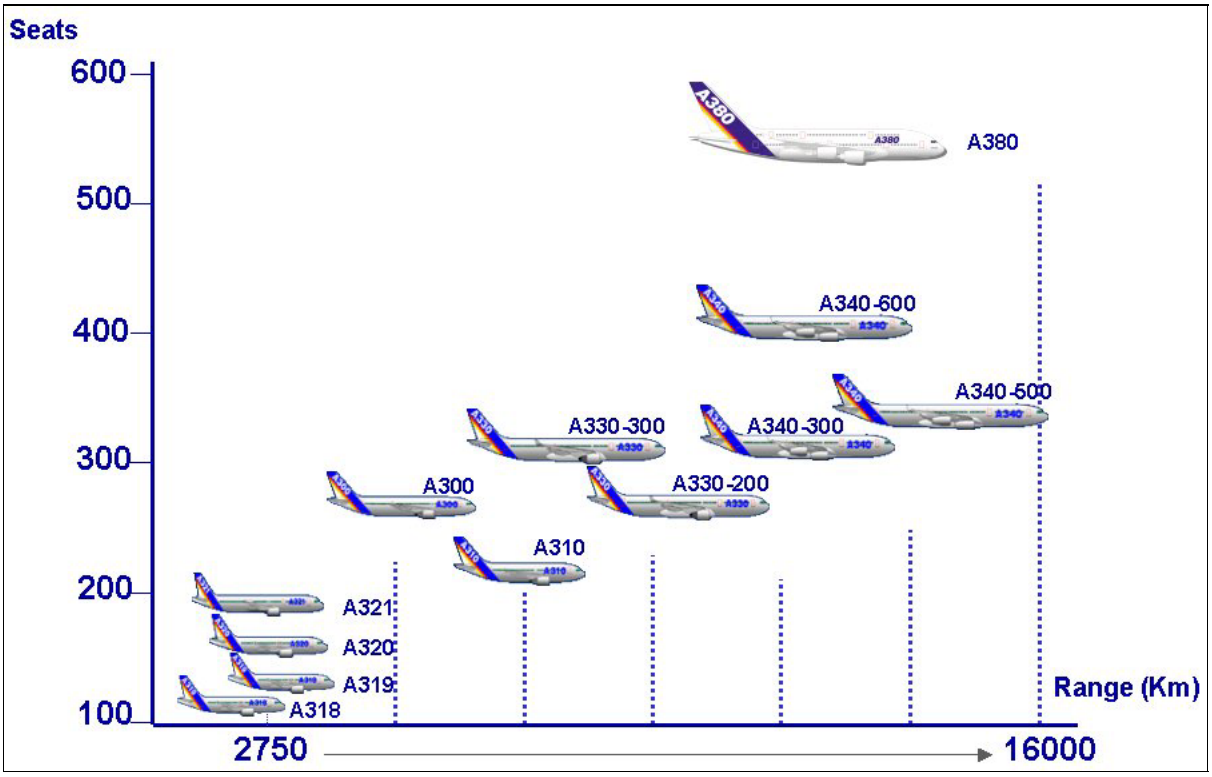

Figure 4: A complete range of aircraft

Nantes plant has been chosen to be responsible for manufacturing the center wing box of the A380. **60% of the center wing box is in composite,** which is a dramatic increase in proportion and in quantity, compared with the previous aircrafts.

To be able to respect these new production rates, it was necessary to buy **new production means** to manufacture composite parts. The **Shared Means line** of the **Composite Unit** manages these machines and builds parts for different manufacturing lines. Consequently, scheduling the parts on these machines is tricky: it has to take into account the activity of these different lines.

Furthermore, the **price** (approximately 40 $/m²) and the **life duration** (1 year at –18°C and 30 days at room temperature) of composite raw material have **strong economic consequences on the scheduling decisions** of the workshop management.

This is why the Shared Mean line wishes to have **decision support software** to help operators and managers to optimize the workshop management. It focused on scheduling by taking into account the specific constraints of the process, especially the prescription of the raw materials.

1.2. Problem statement

The basic process to manufacture composite parts is to lay-up raw composite material on a tool. There are two main types of machines needed: the cutting and the laying machines. To schedule the parts on the laying machines, the cutting operation must have been done and a prepared tool must be available.

The process can be modeled as a **hybrid flow shop system with two steps**:

1. The first step is the cutting operation (there are several identical cutting machines).
2. The second step is the laying operation (there are several identical laying machines). There is a release date on the laying operation (which depends on the tool availability) and a precedence constraint with the cutting operation.

Moreover, when the cutting operation begins, a **prescription constraint** appears on the part manufactured, due to the material used. It means there are deadlines on the parts manufactured.

This research project is of first interest for the company and the results will be used in **decision support software** for the workshop management. Furthermore, this research project is also of interest for the research community because it deals with a problem very few studied, where the deadlines depend on some decision variables: the start dates.

1.3. Research objectives

The goal of this research is to study a hybrid flow shop scheduling problem with prescription constraints on the jobs. The objectives of the thesis are the following:

1. **Analyze a scheduling problem** in a real manufacturing process, by identifying the specific criteria and constraints
2. **Develop a suitable model** for the problem and develop heuristics or exact methods to solve it
3. **Analyze the method used** and how it can be implemented in a decision support tool
4. **Evaluate the research results** and what additional research can be done

1.4. Outline of the document

This document is divided into four chapters as follows:

- Chapter 1 presents the sponsor of the thesis and the background, and also succinctly describes the research problem and objectives

- Chapter 2 focuses on the problem statement: the composite laying process organization and the scheduling problem characteristics

- Chapter 3 gives a literature review for the general scheduling problem and the hybrid flow shop problem, and then focuses on the thesis contributions for the research community.

- Chapter 4 restates the research problem and proposes an approach to solve it (define scenarios, model, resolution)

- Chapter 5 presents the mathematical model developed. Two models are given: a simple model and then the complete model.

- Chapter 6 explains how the optimal solutions are obtained and how the scenarios are defined. After that, a specific heuristic is described.

- Chapter 7 explains how the scenarios are solved with the heuristic and with IP software. Then, it comments on the results of these experiments by giving the heuristic quality, the weights setting influence and the impact of the machine configurations.

- Chapter 8 discusses on the results of this research: the benefits to use IP or heuristic software, the choice the company faces, the savings for the company.

- Chapter 9 concludes the thesis and suggests additional research on the topic.

CHAPTER 2. PROBLEM STATEMENT

2.1. Composite laying process organization

This part firstly explains the composite manufacturing organization, then focuses on the lay-up machines workshop and finally explains the prescription constraints on the composite raw material and the components manufactured.

All the specific terms are highlighted in bold and are defined in APPENDIX 1: GLOSSARY.

2.1.1. Composite manufacturing organization

The process to manufacture an **aircraft part** is separated into two main steps:
1. In the **Composites Unit**, manufacture all the composite **elementary parts** needed for this aircraft part
2. In the **Assembly Unit**, assemble the elementary parts to obtain the aircraft part.

The Composites Unit is composed of elementary parts manufacturing lines, referred as **EP line**. Each EP line is responsible for a specific aircraft part, which means the manufacturing of the whole range of elementary parts in composite needed for this aircraft part. Figure 5 and Figure 6 give examples of specific aircraft parts manufactured at Nantes plant. For example, there is an EP line responsible for the manufacturing of all the elementary parts of the A340 ailerons.

Figure 5: Internal and external ailerons (A340)

Figure 6: Keel beam (A340)

EP line organization
Each EP line manages its production depending on the Assembly Unit needs. A strong constraint for them is that a **tool** is needed to manufacture a specific elementary part. The EP lines have a limited number of tools so they can launch an elementary part in production only if a tool adapted to this elementary part is available. There can be multiple copies of a tool to manufacture a given elementary part.

Shared Means line

Furthermore, there is a specific line: the **Shared Means line**, which is in charge of all the resources needed by several EP lines. These resources are, among others:

- the lay-up machines (composite cutting and laying machines) managed by **the Lay-Up Machines Workshop (LUMW)** ;
- the autoclaves managed by the Polymerization Workshop.

Manufacturing process

The manufacturing process of an elementary part is given on Figure 7. The blue operations are processed by the EP line and the yellow operations are processed by the Shared Means line (LUMW or Polymerization Workshop). The process shows only the operations where the tool is required and where a prescription constraint exists.

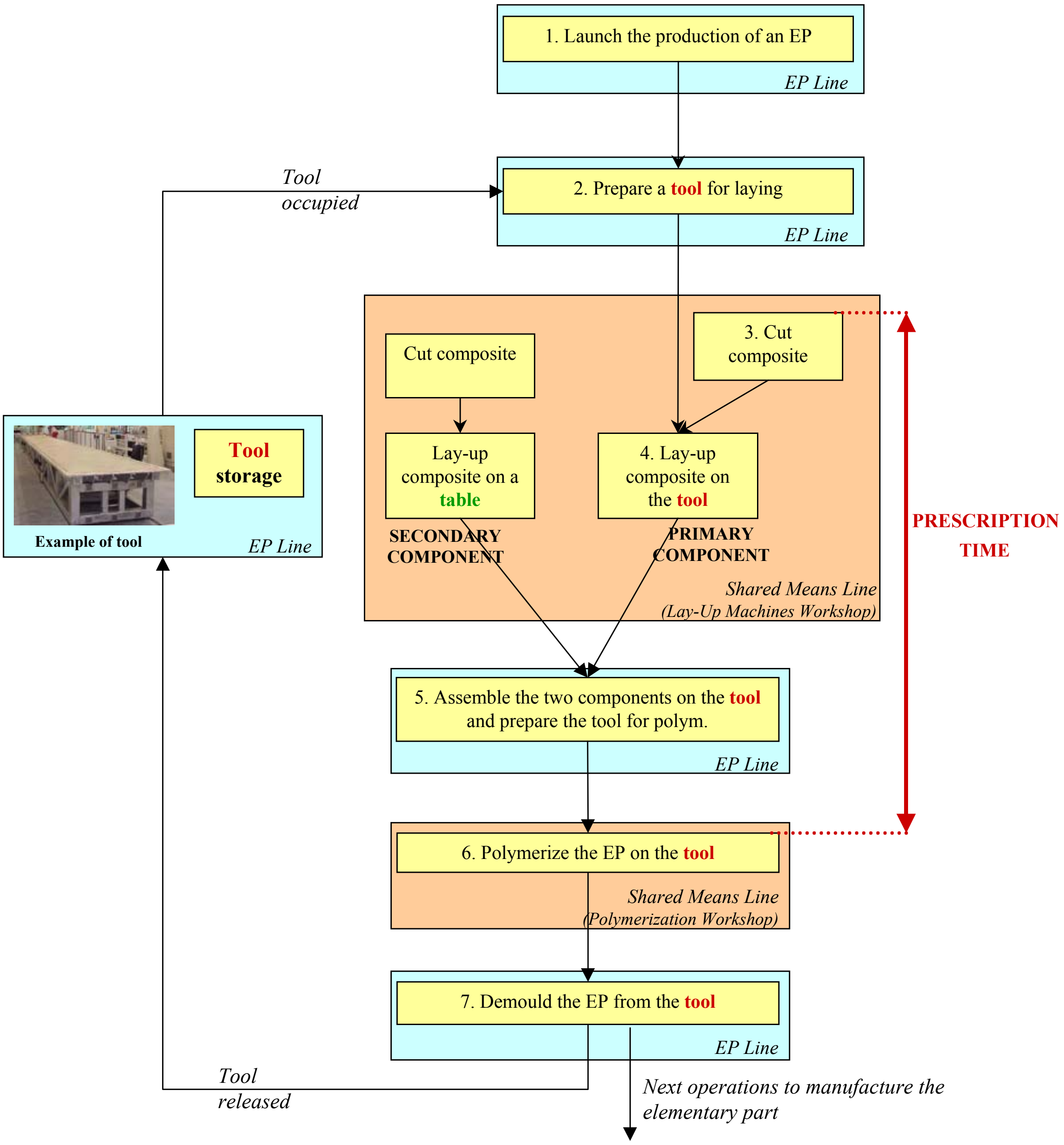

Figure 7: Process to manufacture an elementary part when a tool is required

Some elementary parts manufactured are made of **two components**: a primary component and a secondary component. The **primary component** needs a **prepared tool** to be laid while the **secondary component** only needs a table to be laid. One can consider there are as many tables as needed. Some elementary parts are just made of a single primary component, which needs a prepared tool to be laid. Notice also that for some primary or secondary components, there are no cutting operations required. The two components, once laid, are assembled on the tool.

As soon as one of the two cutting operations begins, a prescription constraint appears on the elementary part manufactured: the polymerization of this elementary part has to begin before a given delay. Part 2.1.3 explains in detail the prescription constraint.

2.1.2. Lay-up machines workshop (LUMW)

Inside the Shared Means line, the **LUMW** manages the **cutting machines (ACCESS)** and the **laying machines (ATLAS)**.

The ACCESS machine cuts some composite material in order to obtain complex plies. These complex plies are stored in **cassettes**. The laying operation on ATLAS consists in deposing some **plies** on a tool in a specific order. The plies are juxtaposed in order to obtain a **layer**, and the superposition of the layers on the tool gives the elementary part. The simple plies are cut directly on the ATLAS machine and the complex plies are laid from the cassettes. Figure 8 sums up the process to **lay-up** a component.

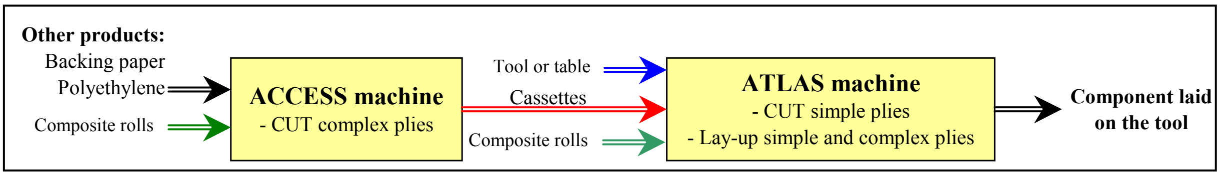

Figure 8: Process to lay up a component

The ACCESS and ATLAS machines use computer-aided manufacturing systems. Therefore, to begin the cutting or laying operation of an elementary part on an ACCESS or an ATLAS machine, the worker has just to download the right numerical control program from a computer. Then, the machine can begin to cut or to lay up the composite raw material.

More details about the ACCESS and ATLAS machines are given on APPENDIX 2: ACCESS and ATLAS SPECIFICATION.

2.1.3. Composite raw material and prescription

The **composite raw material** (conditioned in **rolls**, width: 0.15m, length: 600m) is composed of impregnated **fibers** (very long carbon unit element, with a small diameter). The fibers are linked together with **resin** in a single direction. Some backing paper and polyethylene is added around the impregnated fibers to condition them into rolls (Figure 9).

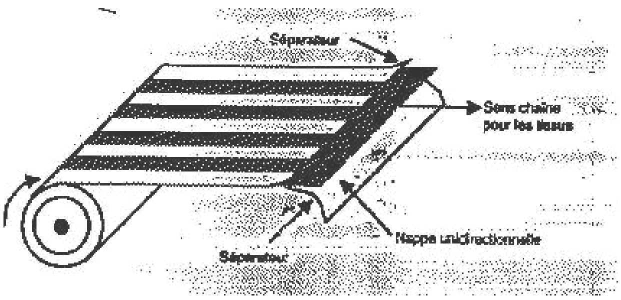

Figure 9: Raw composite roll

Due to the resin, the rolls are **perishable**. Formally, there are two prescription constraints:

- A **life duration at controlled room temperature** (approximately 19°C), which is about 30 days
- A **life duration at frozen temperature** (approximately –18°C), which is about 1 year

Consequently, the rolls are stored in freezers. To use the rolls in the LUMW, it is necessary to defrost the rolls.

Due to the prescription constraint on the raw material used, there is a prescription constraint on the elementary parts manufactured. An elementary part has a prescription date until the polymerization begins. The polymerization removes the prescription constraint.

To know the prescription date of a given elementary part, one needs to know all the rolls used to do the cutting and laying operations of this part. Each roll used has its own prescription date. The prescription date of the elementary part is the earliest prescription dates of the rolls used.

Consequently, the best way to manage the prescription constraint is to use, at the beginning of the cutting operation, some rolls whose life duration at 19°C is maximal. Because the cutting operation is always the first of the process, the life duration of a component is initialized to the life duration of the first roll. The life duration decreases continuously with time. It can be suddenly shortened if a roll with a remaining duration life less than the component remaining duration life is used.

2.2. Problem restrictions and objectives

2.2.1. Hypotheses

- **The setup times on the jobs are not significant and are ignored.**

The setup times are sequence dependent and depend on the raw material used. If the raw material used for a part is the same as the one used for the previous part processed on the machine, there is no setup time; otherwise, there is a setup time: the time needed to change a roll on the machine. Because this time is really small compared to the processing time, the setup times are ignored.

- **The raw material availability is not managed.**

It is assumed that the raw material available (rolls) always has the maximum life duration (static and known). Practically, the first roll to use for the cutting operation has to have a maximal life duration, which initializes the life duration of the elementary part manufactured to the maximal life duration. Then, the next rolls to use have to have remaining life duration greater than or equal to the remaining life duration of the elementary part manufactured.

- **The move times for the tools are not managed**

The problem focuses on the scheduling decisions and the move times for the tools are ignored. The tools have to be moved from the EP line where the tool is prepared to the machine which lays the part in the LUMW and once the part is laid, the tool has to be moved again to the EP line. All these move times are ignored: there are managed without any consequences on the scheduling decisions.

- **All the cutting machines are identical. All the laying machines are identical.**

The machines are really the same but they have not been installed at the same date. Thus, the cutting or laying processing times can be considered to be the same whatever the machine used.

- **All the processing times are deterministic.**

The processing times are set to 75% of the maximum time values observed. This allows being more conservative, without an over-estimation of performance.

- **All the scenarios to solve have a feasible solution.**

Let considering it is always possible to find a solution which respects the prescription constraint for all the jobs to schedule (for more explanations, see part 2.3.6 - Prescription). This implies for all scenarios, there is at least one solution which respects the prescription constraint. This is a reasonable assumption because the cutting and laying processing times are small compared to the prescription delay. So it is always possible to find a date where the part can be scheduled without waiting time between the cutting and the laying operation, meaning the prescription will be necessarily respected.

2.2.2. Problem scope

The LUMW is the actual sponsor of the thesis. They wish to have a tool to help them to schedule the jobs on the cutting and laying machines. This means the research problem will have to be focused on scheduling at the LUMW, and thus will not schedule the operations before or after the LUMW, such as the tool preparation or polymerization. Formally, the following assumptions transcribe the scope of the research problem, stated by the sponsor:

- **Tools release dates are given by the EP line and are not decision variables.**

The tool preparation for laying is managed by the EP line and they have their own way to manage them. So the dates when the tools have to be prepared are not planned. It is assumed the dates when the tools will be prepared and available for the laying operation are known with certainty: these dates are inputs of the scheduling problem. In practice, the tool release can be set to the earliest possible date when the tool could be prepared. Depending on the scheduling decision: the tool cannot be required for laying just after the release date. In this case, the EP line responsible for the tool preparation has freedom to adapt its organization so that the tool will be prepared only when needed by the LUMW.

- **The polymerization operation is not scheduled. The time a part needs from the end of the last laying operation until the beginning of the polymerization is known, assuming normally distributed arrivals to the autoclave.**

The plant has a given number of autoclaves. Consequently, when the tool has finished to be prepared for polymerization, it could wait until a compatible autoclave becomes available to polymerize it. In other words, queue can exist at autoclave.
To take into account this waiting time, the arrivals to the autoclave are assumed normally distributed, and a confidence value is set for the queuing delay. Even with this security delay, there always being a chance that a feasible solution will not work out in the real workshop, because the security delay will not be greater than the time the tool actually wait before entering the autoclave. So the workshop managers will have to take care the parts with a weak prescription margin are favored in the autoclave queue.

Application & Example
The sponsor chose to have an 80% confidence. The mean and standard deviation of observed data, for the time a specific part need from the end of the last laying operation until the beginning of the polymerization, are computed. Then the value used to have an 80% confidence is computed accordingly. For example:
Mean: $\mu = 5.5$ days
Standard deviation: $\sigma = 1.9$
Time used to have an 80% confidence: $T(80\%) = \mu + z(80\%) \times \sigma = 5.5 + 0.65 \times 1.9 \approx 6.7$ days

2.2.3. Optimization objectives

The optimization objectives are those of the workshop management. They have two objectives to optimize when they solve their scheduling problems:

1. **Minimize the flow time of the parts**. The flow time is the time between the tool release date and the tool laying finishing date. The goal is to occupy the machines optimally and to keep the tools in the workshop the shortest possible time. However, because the cutting machines do not need the tool to process the part, this criterion is not relevant for the cutting operations. This is why a second criterion is used.
2. **Minimize the prescription time for the parts**. The prescription time is the time a part is submitted to prescription in the LUMW (time between the beginning of the first cutting operation until the last laying operation of the two parts). This objective aims to synchronize the cutting and laying operations and to synchronize the manufacturing of the primary and secondary jobs.

The second criterion is less important than the first one.

2.3. LUMW scheduling problem

Knowing when the tools (prepared by the EP lines) will be delivered to the LUMW, the scheduling problem consists in scheduling the primary and secondary components on the cutting and laying machines. This part describes the general scheduling problem and then focuses on the hypotheses, the constraints, the decision variables and the dynamic considerations of the problem.

2.3.1. Manufacturing planning and scheduling problem

Planning process
The manufacturing planning is managed by an Enterprise Resource Planning (ERP). The ERP computes the manufacturing starting dates of the elementary parts (EP) and dynamically gives the sequencing of the jobs to process on the workshops.

Scheduling process
Nevertheless, the ERP does not take into account the tool availability. As a consequence, each EP line adapts this manufacturing plan depending on the tools available. The scheduling decisions to start an elementary part are taken by the EP lines.

The LUMW receives tools from the EP lines and has to schedule the primary and secondary components on the laying and cutting machines. More precisely, the **laying operation of a primary component has a release date**, which is the date when the tool will be prepared by the EP line. The secondary component does not have a release date.

For each primary or secondary component to schedule in the LUMW, the cutting operation has to be done before the laying operation. Thus, the scheduling problem is a flow shop. Furthermore, several machines can do the cutting and the laying operations**.** Consequently, this problem can be classified in the **hybrid flow shop** category.

To understand the whole scheduling process of this hybrid flow shop, the following paragraphs will describe in detail:

- For a given elementary part to manufacture, how is defined a job and the two operations for this job?
- What are the decision variables of the scheduling problem?
- What are the constraints of the problem?
- What are the static data of the scheduling problem?
- For a given elementary part to manufacture, what is the prescription constraint?
- What are the solving considerations of the problem?
- What are the criteria to optimize?
- What are the original aspects of this scheduling research problem?

2.3.2. Definitions: elementary part, jobs and operations

Two different types of operations will be considered: the **cutting operation** done on one of the cutting machines and **the laying operation** done on one of the laying machines. A **job** is defined as the succession of one cutting operation and one laying operation**.** To manufacture an elementary part, the **primary job** will refer to the primary component to manufacture while the **secondary job** will refer to the secondary component to manufacture.

For a given elementary part to manufacture, there are two jobs to schedule: a primary job and a secondary job. There is a **release date only on the laying operation of the primary job** (not on the secondary job because it needs a table to be laid and not a specific prepared tool).

The primary and secondary jobs are ideally scheduled on two parallel laying machines, so that they are available at the same time for the assembling operation before the polymerization.

2.3.3. Decision variables

For each of the primary and secondary jobs of the elementary parts to manufacture, the decision variables are:

- The start date of the cutting operation
- The cutting machine which processes the cutting operation
- The start date of the laying operation
- The laying machine which processes the laying operation

2.3.4. Constraints

The constraints of the scheduling problem are the following:

- **Machines:** the machines work a given number of hours per shift, a given number of shifts per day and a given number of days per week.
- **Set of machines to process a specific elementary part:** a primary job of a specific elementary part can be processed only on a subset of the cutting and laying machines. This is the consequence of operational constraints of the workshop (e.g. some tools cannot be moved or do not fit to some machines).
- **Operation precedence:** the cutting operation has to be finished before the beginning of the laying operation.
- **Tool availability:** to start the laying operation of a primary job, the tool must be available and prepared, which results in a release date for the laying operation of the primary job.
- **Processing times:** the processing times are known and deterministic.

2.3.5. Static problem data

- **For each elementary part manufactured in the LUMW:**
  - The prescription delay (deterministic because the raw material availability is not managed)
  - The time to prepare the tool for polymerization and the security delay to take into account the time the tool have to wait for the autoclave
  - The number of jobs (one if there is only a primary component to manufacture, two if there are a primary and a secondary component to manufacture)
  - The data concerning the **primary job**:
    - Processing time for the cutting operation, which can be zero if there is no required cutting operation
    - Processing time for the laying operation
    - Subset of cutting and laying machines where this component can be processed
    - The release date
  - The data concerning the **secondary job**:
    - Processing time for the cutting operation, which can be zero if there is no required cutting operation
    - Processing time for the laying operation
    - Subset of cutting and laying machines where this component can be processed

NB: There is no component where the sum of the cutting and laying processing times plus the time to prepare the tool for polymerization is greater than the prescription delay. In other words, all the components can be manufactured in order to respect the prescription delay.

- **For the machines:**
  - The number of days per week the machine is working
  - The number of shifts per day the machine is working
  - The number of hours per shift the machine is working.

2.3.6. Prescription

For each elementary part to manufacture, the following data is known with certainty:
- ✓ The **prescription delay**: maximal time between the beginning of the first cutting operation and the beginning of the polymerization operation
- ✓ The **assembling and polymerization processing times** (the assembling time includes the security delay for the autoclave queuing)

As soon as this elementary part is scheduled on the LUMW, one can define:
- ✓ The **prescription starting date**: scheduled starting date of the first cutting operation of an elementary part
- ✓ The **prescription date**: prescription delay added to the prescription starting date. The prescription date is the latest starting date for the polymerization operation.
- ✓ The **prescription stopping date**: scheduled starting date of the polymerization operation

The key issue is to schedule the elementary parts so that no prescription date is outstripped. If the prescription stopping date of an elementary part happens after its prescription date, the prescription is outstripped. Otherwise, the prescription is respected (Figure 10).

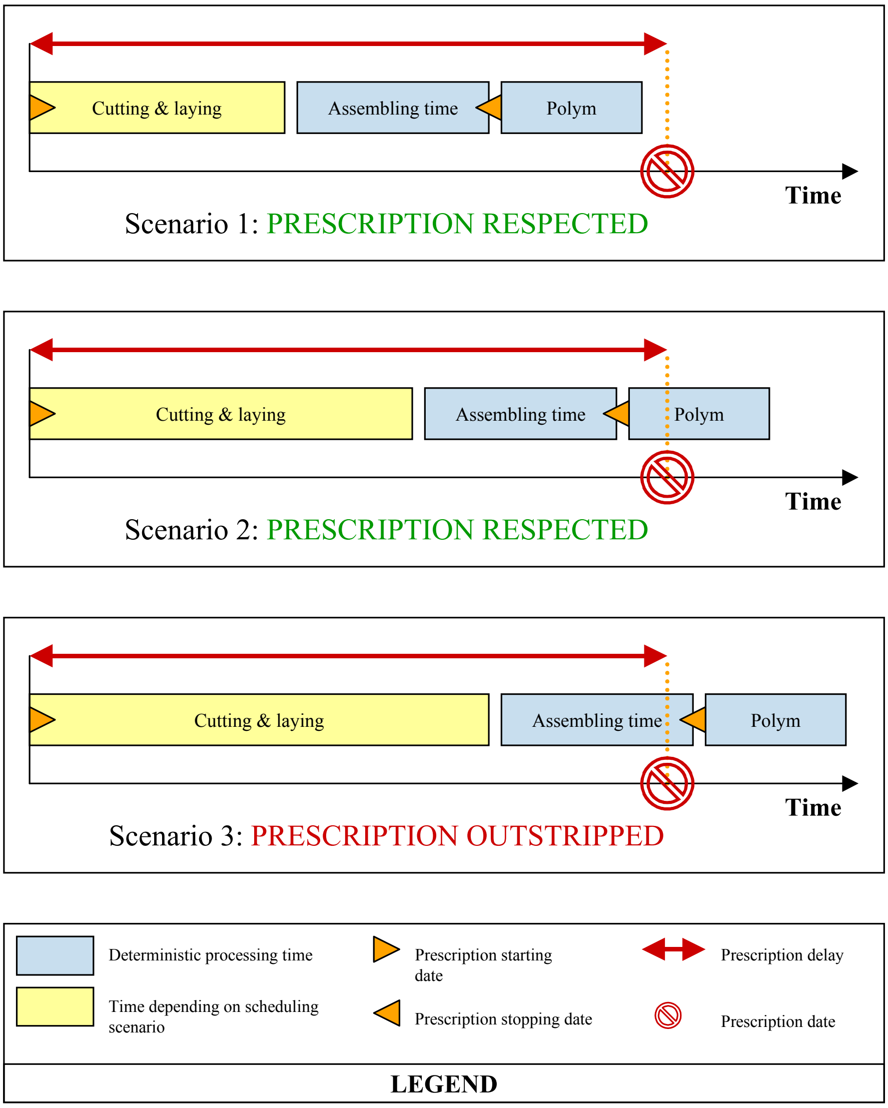

Figure 10: Prescription – three scheduling scenarios

Consider the extreme scenario, where the polymerization starts on the prescription date. In this case, the finishing date of the laying operation is equal to the prescription date minus the assembling time.

From this extreme case, a **deadline** for each elementary part is defined, which is the latest finishing date for the laying operations (Figure 11). The deadline is computed by subtracting the assembling time to the prescription date. It adapts the prescription constraint to the LUMW scheduling problem.

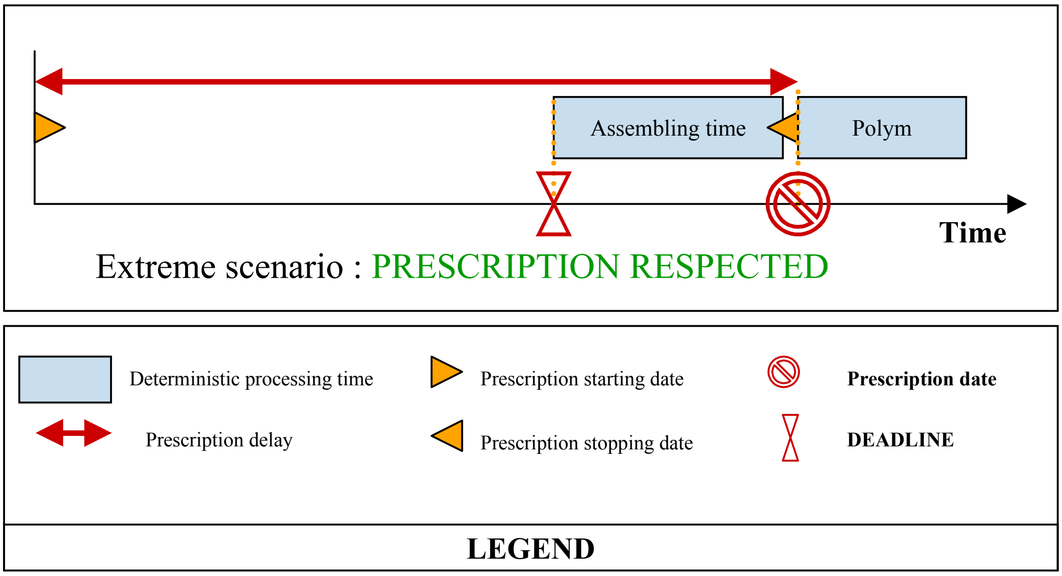

Figure 11: Deadline definition from the prescription extreme scenario

Notice that:

- **A deadline is defined for a single elementary part** (Figure 12).
  The deadline is the same for the primary and secondary jobs of an elementary part. The laying operation of the primary job and the laying operation of the secondary job have to be finished before the deadline. Otherwise, the prescription is outstripped

- If **no cutting operation has begun, the deadline is not defined** but depends on when the earliest cutting operation (primary or secondary component) is scheduled.

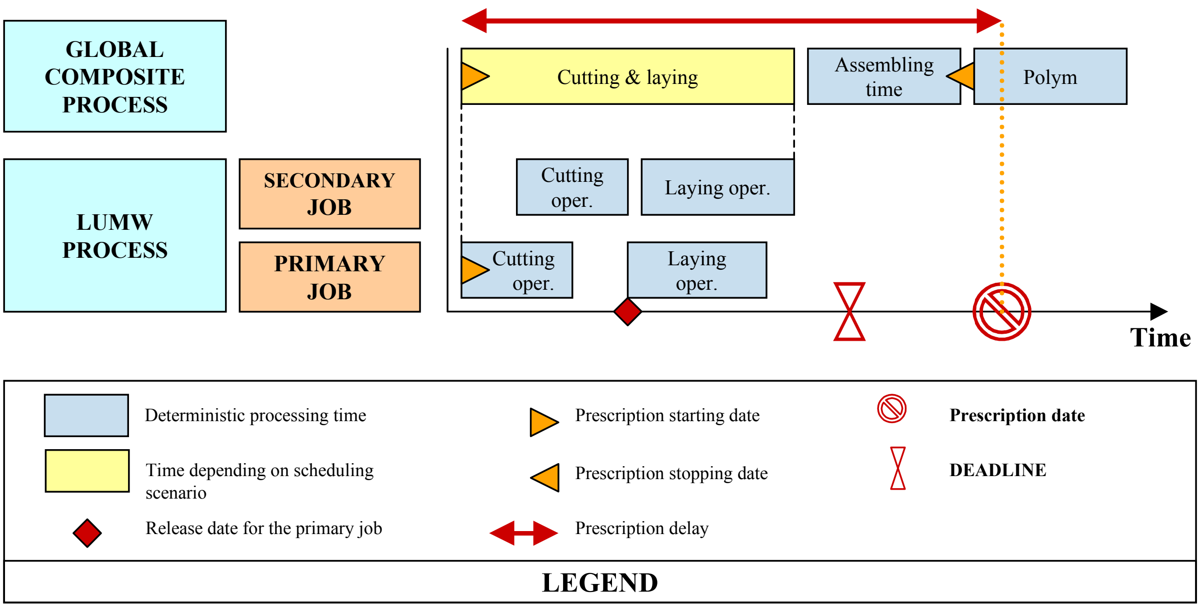

Figure 12: Detailed scheduling scenario for a given elementary part

2.3.7. Solving considerations

The scheduling problem has to be solved twice a week at the LUMW. Because the scheduling choices result in other decisions, such as raw material supply, the first jobs scheduled on the machines cannot be changed. As a consequence, at the beginning of each scheduling problem, the **date availability of each machine** is updated.

Furthermore, for a given elementary part to manufacture, if the cutting operation of one of the two jobs is finished or cannot be changed, the deadline no more depends on decision variables but is known with certainty.

2.3.8. Criteria to optimize

As stated before, there are two criteria to optimize. The **first one** is to **minimize the average flow time for the elementary parts**. The completion time of an elementary part is defined as the latest completion time between the primary and the secondary component of this job. Thus, the flow time of an elementary part is the difference between the completion time of the elementary part and the release date of the primary job of this elementary part. Consequently, the objective is to minimize the flow time of a part, by trying to finish the latest component (between the primary and the secondary) the earliest possible. Given all the elementary parts to manufacture, the goal is to minimize the average of the flow times of these elementary parts.

The **second criterion is** to **minimize the average time a part is submitted to prescription** in the LUMW. For each part, the prescription time is defined: this is the difference between the prescription stopping date and the prescription starting date. The goal is to minimize the average prescription time.

The second criterion is less important than the first one. To model the research problem, it will be needed to analyze how to handle this bi-criteria scheduling problem.

2.3.9. Original aspects of the problem

The two main original aspects of this research problem are the modeling of a real problem and the elementary part prescription.

Real problem
Because the research problem comes from a real workshop management issue, the research results are of great interest for the sponsor. The goal is to provide enough input to the sponsor so that a decision support tool can be specified to help the workshop management in their scheduling process.

Elementary part prescription
The job prescription can be modeled by a deadline constraint. Nevertheless, the key point is that the start date of an operation (which is a decision variable) results in the appearance of

the prescription date. In this problem, the constraint does not refer to a due date (given before solving the problem), but to a deadline, which depends on the decision variables.
Moreover, the prescription results in other constraints, such as the prescription link between the jobs of a same elementary part, or the solving considerations of the problem.

2.3.10. A small example problem

This part develops a small example problem and then compares two possible scheduling solutions of this problem.

Let considering 3 elementary parts to schedule (A, B and C):

- part A is composed of primary job Ap and secondary job As;
- part B is composed of primary job Bp and secondary job Bs;
- part C is just composed of a single primary job Bp.

The jobs can be processed on one cutting machine (CM1) and two laying machines (LM1 and LM2). All the machines are available at *t*=1. The jobs can be processed on all the machines (no restriction on a subset of machines to process a specific elementary part).

Table 1 gives the problem data and restrictions. All data is imaginary and is expressed with the same time unit.

Job	Cutting processing time	Laying processing time	Release for laying	Prescription delay	Assembling time
Ap	4	4	5	30	5
As	2	5	*Do Not Apply*	30	5
Bp	4	3	10	30	5
Bs	-	3	*Do Not Apply*	30	5
Cp	6	6	8	30	1

Table 1: Small example problem data

Figure 13 and Figure 14 give two different schedules for this problem.

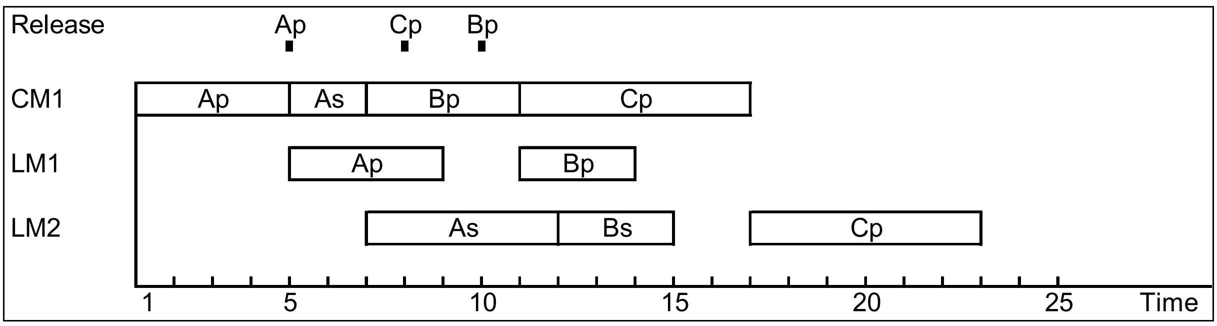

Figure 13: First possible schedule for the example problem

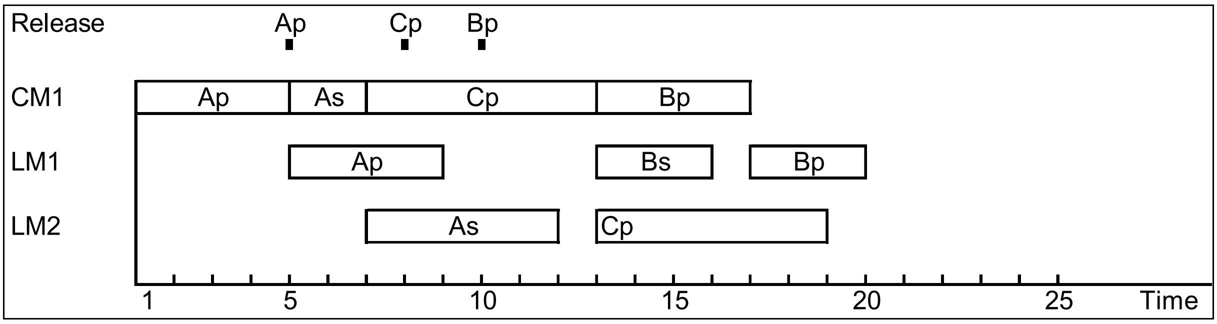

Figure 14: Second possible schedule for the example problem

For each elementary part, the following data is computed for each schedule (Table 2):

- The last job finishing date: latest date between the primary job and secondary job laying finishing dates.
- The prescription starting date.
- The prescription finishing date: sum of the last job finishing date plus the assembling time (the assembling time includes the security delay for the autoclave queuing)
- The flow time: difference between the last job finishing date and the release for the primary job laying operation.
- The prescription time: difference between the prescription finishing date and the prescription starting date.

FIRST SCHEDULE							
Part	Release for laying	Prescription delay	Last job finishing date	Prescription starting date	Prescription stopping date	Flow time	Prescrip-tion time
A	5	30	12	1	17	7	16
B	10	30	15	7	20	5	13
C	8	30	23	11	24	15	13
					Average:	**9**	**14**

SECOND SCHEDULE							
Part	Release for laying	Prescription delay	Last job finishing date	Prescription starting date	Prescription stopping date	Flow time	Prescrip-tion time
A	5	30	12	1	17	7	16
B	10	30	20	13	25	10	12
C	8	30	19	7	20	11	13
					Average:	**9.33**	**13.67**

Table 2: Results for the two schedules of the example problem

<u>Comments</u>

Notice the prescription constraint is respected for all the parts, for the two schedules (the prescription time is lower than the prescription delay).
As regards the optimization objectives, the first criteria (average flow time) is better for the first schedule and the second criteria (average prescription time) is better for the second schedule. So it will be necessary to define precisely the objective function of the problem and to develop strategies to obtain optimal solutions for such problems.

CHAPTER 3. LITERATURE REVIEW

This research problem is a hybrid flow shop (HFS) with prescription constraints on parts. One part corresponds to one or two jobs to schedule, with a single prescription date. The two criteria to optimize are: firstly to minimize the average flow time for the parts and secondly to minimize the average prescription time for the parts.

This section aims to present the research done on the scheduling of the HFS (two stages and *k*-stages). Extensive work has been done on the HFS scheduling. Nevertheless, most of the research works explores the performance of algorithms to find minimum makespan schedules. Moreover, the prescription constraints on the jobs were very few studied (which can be modeled as a deadline for the jobs).

This literature review reminds the research done around the problem and argues how relevant this problem is. This section is divided into three categories:

- Part 3.1 reminds the main notations of operations scheduling problems
- Part 3.2 reviews the state of art of the HFS scheduling problems (especially the 2-stage HFS) and discuss their contributions
- Part 3.3 points out why this problem is relevant.

3.1. Operations scheduling generalities

3.1.1. Introduction

The scheduling of manufacturing systems has been the subject of extensive research for over forty years. The main focus is on the **efficient allocation of one or more resources to operations** over time. Adopting manufacturing terminology, a **job** consists of one or more **operations**, and a **machine** is a resource that can perform at most one operation at a time. The problem concerns deterministic machine scheduling, where all data are known with certainty.

The objective is to find an **optimal schedule**, relatively to one or more **criteria**. The two most well-know simple problems for which the optimal solutions are known are:

- Scheduling a single machine to minimize the maximum lateness of the jobs, for which an optimal solution is obtained by sequencing the jobs following the earliest due date (EDD) rule of Jackson (Sipper & Bulfin, 1997 – Hax & Candea, 1984).
- Scheduling a single machine to minimize the sum of weighted completion times of the jobs, for which an optimal solution is obtained by sequencing the jobs following the shortest weighted processing time (SWPT) rule of Smith (Sipper & Bulfin, 1997– Hax & Candea, 1984).

Every scheduling problem can be described as follows (Chen et al., 1998):

- There are m machines to process n jobs.
- A schedule specifies, for each machine m ($m = 1,\dots,M$) and each job j ($j = 1,\dots,n$), one or more time intervals where processing is performed by machine m on job j.

A schedule is **feasible** if there is no overlapping of time intervals corresponding to the same job (so that a job cannot be processed by two machines at once), or of time intervals corresponding to the same machine (so that a machine cannot process two jobs at the same time). Furthermore, there are various requirements relating to the specific problem that makes the schedule feasible: the machine environment, the jobs characteristics and an optimality criterion.

3.1.2. Machine environment

Each job may require several operations. An operation refers to a specific period of processing by some machine type. The scheduling problems can be classified into two families: the single-stage systems (each job requires one operation) and the multi-stage systems (each job may require several operations).

Single-stage system – Machine configurations:

- Single machine
- *m* machines operating in parallel with the same function:
  - identical parallel machines (processing time is independent of the machine)
  - uniform parallel machines (machines operates at different speed but are otherwise identical)
  - unrelated parallel machines (processing time depends on the machine)

Multi-stage systems (*k* stages with different functions) – Machine configurations:

- Flow shop: processing of each job goes through the stages 1 to *k* in that order
- Open shop: processing of each job goes also goes one through each stage, but the routing can differ between the jobs
- Job shop: each job has a prescribed routing through the stages, and the routing may differ from job to job

3.1.3. Jobs characteristics

The processing requirements of each job are given. Furthermore, depending on the problems, the jobs may have or not some of these characteristics:

- Availability constraints (the jobs cannot start before a release date)
- Finishing constraints (the jobs must finish before a deadline)
- Precedence constraints between the jobs
- Preemption (the processing of any operation may be interrupted and resumed at a later time on the same or on a different machine)

3.1.4. Optimality criteria

At this point, some **notations** have to be adopted for the terms introduced:

$j = 1,...N$ subscript for the job

$i = 1,...,O_j$ subscript for the operation i of job j

$m = 1,...,M$ subscript for the machine

r_j release date for job j

d_j due date of job j

p processing time (will take various subscripts)

w_j positive integer weight for job j

f_j non decreasing cost function

Given a schedule, one can compute

C_j completion time for job j

$F_j = C_j - r_j$ flow time for job j

$L_j = C_j - d_j$ lateness for job j

$E_j = \max\{d_j - C_j; 0\}$ earliness for job j

$T_j = \max\{C_j - d_j; 0\}$ tardiness for job j

$U_j = \begin{cases} 1 \text{ if } C_j > d_j \\ 0 \text{ otherwise} \end{cases}$ the unit penalty for job j

$f_j(C_j)$ the cost of job j

Some commonly used **criteria** are the minimization of:

- The maximum completion time (makespan): $C_{\max} = \max_j C_j$
- The maximum lateness: $L_{\max} = \max_j L_j$
- The maximum earliness: $E_{\max} = \max_j E_j$
- The total (weighted) completion time: $\sum_j w_j C_j$
- The total (weighted) or average flow time: $\left(\frac{1}{n}\right)\sum_j w_j F_j$
- The total (weighted) earliness: $\sum_j w_j E_j$
- The total (weighted) tardiness: $\sum_j w_j T_j$
- The (weighted) number of late jobs: $\sum_j w_j U_j$
- The total cost: $\sum_j f_j$

3.1.5. Solving methods

The methods to solve scheduling problems can be classified into two main groups:

- **Exact methods**: The exact methods solve the problem optimally (it is not always possible). They include **explicit enumerative methods** or **implicit enumerative methods** such as branch & bounds methods, dynamic programming, integer linear programming, column generation, constraint programming…

- **Heuristics methods**: Heuristic methods aim to give good solutions, but not optimal. They include Lagrangean relaxation, greedy heuristics, local neighborhood search, **metaheuristics** (simulated annealing, tabu search, genetic algorithm). Heuristics are used instead of exact methods for four main reasons:
  - Solution time: one needs to obtain quickly a solution
  - Cost: it is usually very costly to obtain the optimal solution
  - Complexity: sometimes the problem is too complex so that it is not possible to find the optimal solution
  - Robustness: the solution has to be robust if new constraints are added.

3.2. Hybrid Flow Shop scheduling problems

3.2.1. Introduction

A Hybrid Flow Shop (HFS) consists of a series of production stages. Each stage has several machines operating in parallel. Some stages may have only one machine, but at least one stage must have multiple machines. The flow of jobs through the shop is unidirectional. Each job is processed by one machine in each stage and it must go through one or more stage. Machines in each stage can be identical, uniform or unrelated. There has been a significant amount of research done on the HFS scheduling problem since its first attempt in 1971 (Arthanari and Ramaurthy, 1971).

3.2.2. General HFS scheduling problem

Little work has been done on systems with more than two stages. Research on the *k*-stages HFS problems (*k*>3) is limited to identical machines in each stage. The most common used approaches are heuristic and branch-and-bound, in order to minimize makespan and queuing in the buffers. The basic approach is to decompose the problem into three sub-problems: machine allocation, sequencing and timing.

3.2.3. Two-stage HFS scheduling problem

The two-stage HFS scheduling problem has been proven to be **NP-complete**, even with only two stages and multiple machines on only one of the two stages (Gupta, 1988). Research work has been done on the different machine configurations. The objective function is usually the minimization of the makespan. Table 3 sums up some surveys done on the exact methods and heuristics approaches developed for the two-stage HFS (Vignier et al., 1999). The problem type is identified by using the **three-field representation** ($\alpha|\beta|\gamma$) for HFS.

- α represents the machine configuration: $\alpha = \alpha_1, \alpha_2$ where:
  - α_1 = "HF" + *k* (*k* is the number of stages of the HFS)
  - $\alpha_2 =$ $(\text{type of parallel machines at stage } l + \text{number of machines at stage } l)_{l=1}^{k}$
- β represents the jobs characteristics
- γ represents the optimization criterion or criteria

Example:
Problem: 2 stages HFS, m_u uniform parallel machines at stage 1, m_i identical parallel machines at stage 2, identical jobs, makespan minimization
Three field representation: $HF2,(Qm_u, Pm_i) \mid p_j = p \mid C_{max}$

Problem type	**Approach**	**Paper**
$HF2,(1,Pm) \mid \mid C_{max}$	Dedicated heuristics Dedicated heuristic	Gupta & Tunc (1994) Sriskandarajah & Sethi (1989)
$HF2,(Pm,1) \mid \mid \max_j E_j$	MILP and heuristics	Vignier et al. (1996)
$HF2,(Qm,Qm) \mid p_j = 1 \mid C_{max}$	Dedicated heuristic	Dessouky et al. (1998)
$HF2,(Pm,Pm) \mid \mid C_{max}$	MILP and heuristics B&B Simulated Annealing Simulated Annealing Tabu Search GA	Guinet et al. (1996) Rajendran & Chaudhuri (1992) Haouari & M'Hallah (1997) Riane et al. (1999) Haouari & M'Hallah (1997) Portmann et al. (1998)
$HF2,(Pm,Pm) \mid \mid \sum U_j$	Dedicated heuristic	Gupta & Tunc (1998)
$HFk,(Pm^l)_{l=1}^k \mid \mid costs$	MILP	Aghezzaf & Artiba (1998)
$HFk,(Pm^l)_{l=1}^k \mid \mid C_{max}$	Dedicated heuristic (B&B crossed with GA) Dedicated heuristic (B&B) Tabu Search Dedicated heuristic	Portmann et al. (1998) Moursli & Pochet (2000) Nowicki & Smutnicki (1998) Guinet & Solomon
$HFk,(Pm^l)_{l=1}^k \mid \mid \sum C_j$	Dedicated heuristic B&B	Brah & Loo (1999) Azizoglu et al. (2001)

Table 3: Survey on the HFS problems and approaches

3.2.4. Scheduling criteria

Scheduling criteria are the measures from which the schedules are evaluated. There are two classes of criteria:
Schedule costs: fixed cost associated with production, overtime cost, inventory holding cost, etc.
Schedule performance: utilization level of the production resources, average or maximum tardiness and earliness for a set of tasks, makespan, etc.

In theoretical research, most of the literature addresses single criterion problems. The **most widely used criteria** are the minimization of:
1) the makespan
2) the maximum tardiness
3) the average interval
4) the total or average flow time
5) the sum or average of completion time

Only few studies consider bi-criteria optimization (Chou & Lee, 1999). Nevertheless, multiple objectives are usually involved in the real world workshop management. Two main classes of performance measures considered are:

- Those based on flow time: sum of setup times, work-in-process inventory…
- Those based on due dates: number of tardy jobs, average time of tardiness to reduce, sum of tardiness…

3.2.5. Approaches to HFS scheduling

Heuristic and **branch-and-bound** (B&B) are the two mostly employed approaches to solve HFS scheduling problems. Because the HFS is NP-complete, it is always hard to find a good algorithm. One usually resorts to heuristics that may give reasonably good schedules. Solutions that are optimal for the simple problems often turn out to be good heuristics for the more complicated NP-complete problems. For example, Vignier et al. (1996) give a heuristic based on the EDD rule to solve 2-stage HFS problem. Guinet et al. (1996) propose a sequence-first, allocate-second heuristic approach in which SPT rule and LPT rule are adopted in the sequencing phase.

When a more accurate solution is required, a more sophisticated technique such as B&B can be adopted. B&B consists in the calculation of lower bound, branching and node elimination. However, the B&B approaches have been **few tried** in **real world application**. The reason is that the existing B&B algorithms are developed in simplified machine environment; they could not be mounted to real world case with much more complicated environment. On the other hand, when the machine environment is complicated, it is difficult to develop a B&B algorithm to obtain optimal solution because the algorithms are **time consuming** even for a moderate number of parts.

Instead, the **dispatching rule based heuristics** are the most common approach to HFS scheduling problems in practice. The popularity of dispatching rules in practice is due to their low computational requirements; the ease of interfacing them with computer aided manufacturing systems and their intuitive nature making them easy to sell to management. However, one shortcoming lies here is that no proof has been shown whether or not the schedules they make are optimal, even near optimal.

3.2.6. Theory-application gap

It is apparent from the literature that there is a **gap** between theories developed and practical applications (Fortemps et al., 1996 – Riane et al., 2001). Treatments of hybrid flow shops in the scientific literature are rather simplistic and abstract in nature. They ignore many of the **real life restrictions** that must be taken into consideration, and which sometimes complicate the problem well beyond the capabilities of the standard methodologies of Operations Research.

3.3. Thesis contributions

According to this literature review, one can conclude this research problem has not been studied by the research community. The two main original aspects and interests of this research are:

- The problem is based from a **real issue** and little research work has been done to address real HFS scheduling problems.
- The problem has **original constraints** (prescription constraints) and **criteria** (bi-criteria optimization), which have not been well studied in the literature.

CHAPTER 4. APPROACH

4.1. Restate problem

The problem consists in scheduling a set of jobs, where a job is composed of two operations. Those operations have to be realized sequentially, without overlapping between stages. Job preemption and job splitting are not allowed. Each operation has a given processing time at each stage. All data of the problem are assumed integer.

This part details the prescription constraints on the parts, the solving considerations and the problem feasibility.

4.1.1. Prescription constraints

Some jobs have a prescription link: a part to schedule is thus composed of a primary job and a secondary job. Each primary job has a release date at the **second** stage (laying operation). Each job has:

- a **given prescription delay**: this is the maximal time between the beginning of the first cutting operation of this job (either the cutting operation of the primary job or the secondary job) and the finishing of the last laying operation (either the laying operation of the primary job or the secondary job),
- a **prescription time**, which **depends on the decision variables**: this is the actual time between the beginning of the first cutting operation of this job and the finishing of the last laying operation.

Scheduling consists in assigning a specific machine to each operation of every job as well as sequencing all operations assigned to each machine, so that successive operations of a job do not overlap and so that each machine processes at most one job at a time. Furthermore, the prescription constraints have to be respected for all the parts, meaning **the prescription time is lower than the prescription delay**.

4.1.2. Solution considerations

For each primary or secondary job, the cutting or laying operations could have already been processed in the LUMW or could have previously been scheduled with no possibility to change these decisions. To tackle these solving considerations, two statuses are defined for each job:

- ✓ the **cutting status** (“no change” or “can be changed”)
- ✓ the **laying status** (“no change” or “can be changed”)

In case the operation is in progress or processed in the LUMW, the status is “no change”. In case the scheduling dates of this operation cannot be modified, the status is also “no change”. Once one operation of an elementary part has its status to “no change”, the deadline is determined because the prescription starts with the first scheduled operation. Table 4 sums up the different cases of solving status of an elementary part.

Before solving the problem, one needs to set the start dates of the "no change" operations: these dates are no more decision variables but become parameters of the problem.

ELEMENTARY PART				Number of decision variables	Deadline is variable?
PRIMARY JOB		SECONDARY JOB			
Cutting	Laying	Cutting	Laying		
CBC	CBC	CBC	CBC	4	**YES**
NC	CBC	CBC	CBC	3	No
NC	**NC**	CBC	CBC	2	No
CBC	CBC	**NC**	CBC	3	No
NC	CBC	**NC**	CBC	2	No
NC	**NC**	**NC**	CBC	1	No
CBC	CBC	**NC**	**NC**	2	No
NC	CBC	**NC**	**NC**	1	No
NC	**NC**	**NC**	**NC**	0*	No

LEGEND

CBC = "can be changed"
NC = "no change"

*: in this case, the elementary part is removed from the problem

Table 4: Different solving status of an elementary part

4.1.3. Solving feasibility

Let assume it is always possible to find a schedule where the prescription delay is respected for all the jobs. This is a reasonable assumption because the number of jobs to schedule is small, the processing times are small compared to the prescription delay, and the problem is solved frequently.

4.2. Proposed approach

The proposed approach can be decomposed into four steps:

- Define scenarios,
- Model the problem with integer variables,
- Solve the scenarios exactly with IP solver,
- Develop heuristic to solve the problem and test its performance,
- Use the heuristic solution to increase the time needed by the IP solver to find the optimal solution.

4.2.1. Scenarios definition

A scenario is fully defined by a set of jobs to schedule (with all the associated data) and the number of cutting and laying machines. To give an idea of the size of the problem, one scenario corresponds to about 10 to 15 jobs to schedule, on a one-week horizon, with one or two cutting machines and three or four laying machines.

The total number of scenarios examined will be from 20 to 30. The scenarios will be obtained from actual scheduling problems at the LUMW (3 to 5 scenarios) and the other scenarios will be defined by modifying the actual scenarios, keeping the data as realistic as possible.

4.2.2. Problem modeling

This problem is a short-term scheduling problem: the jobs have to be produced within a short period of time (about one week). Therefore, the problem is small enough to infer the optimal solution could be found by an **integer-linear programming** (ILP) **solver**. Three different modeling approaches are possible, in term of integer variables to use:

- Sequencing variables: $x_{ij} = 1$ if job *i* is before job *j*.

  This model is usually not very efficient for solving with standard ILP software because of the big-M coefficient (loose LP relaxation).
- Positional variables: $x_{ik} = 1$ if job *i* is at position *k*.

  This model is difficult to use in multi-machine problems. Moreover, some constraints also require a big-M coefficient.
- **Time-indexed variables**: $x_{it} = 1$ if job *i* starts at period *t*.

  This model appears promising for this problem because the number of variables is not very large (small time horizon and number of jobs). Furthermore, this model does not use a big-M coefficient so that the **LP relaxation** is more efficient (Dauzère-Pérès, 1995).

Therefore, an integer-linear programming model will be developed based on a discrete time formulation (the time unit will be the hour). The main focus will be to find a good model for the problem so that the capacity of a standard ILP solver is enough to obtain the optimal solution in a reasonable amount of time.

4.2.3. Optimal solutions

Once the mathematical model will be defined, it will be coded with an IP solver. Thus, the optimal solutions of the scenarios will be obtained in a reasonable amount of time.

4.2.4. Heuristic solutions

Even if it is expected to solve the problem optimally for the set of scenarios, two points are still unsatisfying for its use in a decision support tool:

- The exact solution may require a **long CPU time** to be obtained by the solver (several hours). Yet, the managers want to have the possibility to add a new job, to change the release dates or to update the job processing times, and then to solve again the problem. Consequently, the result must be obtained quickly so that they can change a little bit the data and see what the consequences on the solution are.
- It is **costly** to buy and to use solver software. If a heuristic could be used and gives good solution (meaning very close to the optimal solution), then the gain obtained by finding the optimal solution is not justified compared to the cost of the solver.

Therefore, a dedicated **heuristic** will be developed to solve the problem. The exact method will allow testing the performance of the heuristic on the scenarios defined.

The literature review has shown B&B approaches have been **few tried** in **real world application** because the algorithms need a very long time to give the solutions (even for a moderate number of jobs) and are developed in simplified machine environment. As a consequence, it will rather tried to use a **dispatching rule based heuristic** or a metaheuristic (such as tabu search, simulated annealing or genetic algorithms), depending on the model characteristics. This is the most common approach to HFS scheduling problems in practice so it is expected to be a promising method.

4.2.5. Optimal solving speed improvement

The last step in the approach will be to use the solution given by the heuristic in the IP solver in order to increase the time needed to obtain the optimal solution. Indeed, the value of the objective function of the solution given by the heuristic is an upper bound of the objective function of the optimal solution. Thus, it is expected to obtain the optimal solutions much more quickly with the IP solver by adding a cutoff to the value of the objective function. The cutoff value will be set to the value of the objective function of the solution given by the heuristic.

CHAPTER 5. MATHEMATICAL MODEL

This chapter presents the mathematical model developed. Firstly, a simple model is given (without secondary jobs) and then the complete model is presented.

5.1. Simple model

In this first model, N parts have to be scheduled, and each part is composed of a single primary job (no secondary job). Each of the N jobs to schedule is composed of one cutting operation and one laying operation.

The model presented is based on time-indexed variables without solving considerations: none of the operations have a "no change" status, which means all the operations of all the jobs given have to be scheduled. All data are assumed to be binary or integer

Parameters

N : set of jobs
U : set of cutting machines
V : set of laying machines

N : number of jobs
U : number of cutting machines
V : number of laying machines
T : number of periods

n : subscript for a job, $n \in \mathsf{N}$, $n = 1...N$
u : subscript for a cutting machine, $u \in \mathsf{U}$, $u = 1...U$
v : subscript for a laying machine, $v \in \mathsf{V}$, $v = 1...V$
t : subscript for a period, $t = 1...T$

pc_n : cutting processing time for job n, $n \in \mathsf{N}$
pl_n : laying processing time for job n, $n \in \mathsf{N}$
C_n : subset of cutting machines which can process the cutting operation of job n, $n \in \mathsf{N}$
L_n : subset of laying machines which can process the laying operation of job n, $n \in \mathsf{N}$

r_n : release date for laying operation of job n, $n \in \mathsf{N}$
d_n : prescription delay for job n, $n \in \mathsf{N}$

ac_u : availability date of cutting machine u, $u \in \mathsf{U}$
al_v : availability date of laying machine v, $v \in \mathsf{V}$

α : weight for the average flow time in the objective function
β : weight for the average prescription time in the objective function

Variables

$C_{n,t,u}$ $= 1$ if the cutting operation of job n starts at t on cutting machine u
$L_{n,t,v}$ $= 1$ if the laying operation of job n starts at t on laying machine v

Auxiliary variables

F_n : laying flow time for job $n \in \mathsf{N}$
PT_n : prescription time for job $n \in \mathsf{N}$

Constraints

$$\sum_{u \in \mathsf{C}_n} \sum_{t=ac_u}^{T} C_{n,t,u} = 1 \qquad \forall n \in \mathsf{N} \qquad (1)$$

$$\sum_{v \in \mathsf{L}_n}^{V} \sum_{t=\max\{al_v, r_n\}}^{T} L_{n,t,v} = 1 \qquad \forall n \in \mathsf{N} \qquad (2)$$

$$\sum_{n=1}^{N} \sum_{s=t-pc_n+1}^{t} C_{n,s,u} \le 1 \qquad \forall u \in \mathsf{U}, \forall t \ge ac_u \qquad (3)$$

$$\sum_{n=1}^{N} \sum_{s=t-pl_n+1}^{t} L_{n,s,v} \le 1 \qquad \forall v \in \mathsf{V}, \forall t \ge al_v \qquad (4)$$

$$\sum_{v \in \mathsf{L}_n} \sum_{s=1}^{t} L_{n,s,v} \le \sum_{u \in \mathsf{C}_n} \sum_{s=1}^{t-pc_n} C_{n,s,u} \qquad \forall n \in \mathsf{N}, \forall t \qquad (5)$$

$$F_n \ge pl_n + \sum_{v \in \mathsf{L}_n} \sum_{t=al_v}^{T} t * L_{n,t,v} - r_n \qquad \forall n \in \mathsf{N} \qquad (6)$$

$$PT_n \ge pl_n + \sum_{v \in \mathsf{L}_n} \sum_{t=al_v}^{T} t * L_{n,t,v} - \sum_{u \in \mathsf{C}_n} \sum_{t=ac_u}^{T} t * C_{n,t,u} \qquad \forall n \in \mathsf{N} \qquad (7)$$

$$PT_n \le d_n \qquad \forall n \in \mathsf{N} \qquad (8)$$

Objective function

$$\min\left\{\alpha * \frac{1}{N}\sum_{n=1}^{N} F_n + \beta * \frac{1}{N}\sum_{n=1}^{N} PT_n\right\}$$

Additional constraints

$$\sum_{n=1}^{N}\sum_{t=1}^{ac_u - 1} C_{n,t,u} = 0 \quad \forall u \in \mathsf{U} \quad (9)$$

$$\sum_{n=1}^{N}\sum_{t=1}^{al_v - 1} L_{n,t,v} = 0 \quad \forall v \in \mathsf{V} \quad (10)$$

$$\sum_{v \in \mathsf{L}_n}\sum_{t=1}^{r_n - 1} L_{n,t,v} = 0 \quad \forall n \in \mathsf{N} \quad (11)$$

$$\sum_{u \notin \mathsf{C}_n}\sum_{t=1}^{T} C_{n,t,u} = 0 \quad \forall n \in \mathsf{N} \quad (12)$$

$$\sum_{v \notin \mathsf{L}_n}\sum_{t=1}^{T} L_{n,t,v} = 0 \quad \forall n \in \mathsf{N} \quad (13)$$

Comments

- Cutting and laying operations of all the jobs have to be scheduled: (1) and (2)
- Each cutting or laying machine processes at most one job at a time: (3) and (4)
- For a given job, the cutting operation is done before the laying operation: (5)
- For a given part, the laying flow time corresponds to the time between the release date of the laying operation and the finishing date of the laying operation: (6)
- For a given part, the prescription time is the time between the start date of the cutting operation and the finishing date of the laying operation: (7)
- For all the parts, the prescription time has to be lower than the prescription delay: (8)
- The operations cannot be scheduled before the availability date of the machines: (9) and (10)
- The laying operation of a job cannot be processed before the release date of the laying operation: (11)
- Some jobs can be scheduled on a given subset of machines: (12) and (13)

5.2. Complete model

This model takes into account all the constraints of the problem. N jobs have to be scheduled, decomposed of P primary jobs and S secondary jobs. It means P parts have to be manufactured (because the secondary job of a part is assembled on the primary job before the polymerization). There is not necessarily one secondary job for each primary job. Each job (primary or secondary) is composed of one cutting operation and one laying operation.

The release date applies only on the primary jobs. The prescription delay, the flow time and the prescription time are defined for a single part, i.e. a couple primary/secondary parts. They are defined the model with the subscript of the primary job.

Some operations have a “no change” status: they are already scheduled with no possibility to change them (the start date and the machine of these operations are known). All data are assumed to be binary or integer

Parameters

N : set of jobs ($\mathsf{N} = \mathsf{P} \cup \mathsf{S}$)
P : set of primary jobs
S : set of secondary jobs
U : set of cutting machines
V : set of laying machines

N : number of jobs ($N = P + S$)
P : number of primary jobs
S : number of secondary jobs
U : number of cutting machines
V : number of laying machines
T : number of periods

n : subscript for a job, $n \in \mathsf{N}$, $n = 1...N$
j : subscript for a primary job, $j \in \mathsf{P}$, $j = 1...P$
k : subscript for a secondary job, $k \in \mathsf{S}$, $k = P + 1...N$
u : subscript for a cutting machine, $u \in \mathsf{U}$, $u = 1...U$
v : subscript for a laying machine, $v \in \mathsf{V}$, $v = 1...V$
t : subscript for a period, $t = 1...T$

pc_n : cutting processing time for job n, $n \in \mathsf{N}$
pl_n : laying processing time for job n, $n \in \mathsf{N}$
C_n : subset of cutting machines which can process the cutting operation of job n, $n \in \mathsf{N}$
L_n : subset of laying machines which can process the laying operation of job n, $n \in \mathsf{N}$

$$b_{j,k} \begin{cases} = 1 \text{ if job } j \in \mathsf{P} \text{ is the primary job of secondary job } k \in \mathsf{S} \\ = 0 \text{ otherwise} \end{cases}, j \in \mathsf{P}, k \in \mathsf{S}$$

r_j : release date for laying operation of job j, $j \in \mathsf{P}$
d_j : prescription delay for part whose primary job is job j, $j \in \mathsf{P}$

ac_u : availability date of cutting machine u, $u \in \mathsf{U}$
al_v : availability date of laying machine v, $v \in \mathsf{V}$

$$cs_n \begin{cases} = 1 \text{ if the cutting operation of job } n \text{ cannot be changed} \\ = 0 \text{ otherwise} \end{cases}, \ n \in \mathsf{N}$$

$$ls_n \begin{cases} = 1 \text{ if the laying operation of job } n \text{ cannot be changed} \\ = 0 \text{ otherwise} \end{cases}, \ n \in \mathsf{N}$$

cb_n : index of the period where the cutting operation of job n begins, $n \in \mathsf{N}$ s.t. $cs_n = 1$
lb_n : index of the period where the laying operation of job n begins, $n \in \mathsf{N}$ s.t. $ls_n = 1$
$cm_n \in \mathsf{U}$: index of the cutting machine which processes job n, $n \in \mathsf{N}$ s.t. $cs_n = 1$
$lm_n \in \mathsf{V}$: index of the laying machine which processes job n, $n \in \mathsf{N}$ s.t. $ls_n = 1$

α : weight for the average flow time in the objective function
β : weight for the average prescription time in the objective function

Variables

$C_{n,t,u} = 1$ if the cutting operation of job n starts at t on cutting machine u
$L_{n,t,v} = 1$ if the laying operation of job n starts at t on laying machine v

Auxiliary variables

F_j : laying flow time for the part whose primary job is job $j \in \mathsf{P}$
PT_j : prescription time for the part whose primary job is job $j \in \mathsf{P}$

Constraints

$$\sum_{u \in \mathsf{C}_n} \sum_{t=ac_u}^{T} C_{n,t,u} = 1 \qquad \forall n \in \mathsf{N} \qquad (1)$$

$$\sum_{v \in \mathsf{L}_k}^{V} \sum_{t=al_v}^{T} L_{k,t,v} = 1 \qquad \forall k \in \mathsf{S} \qquad (2)$$

$$\sum_{v \in \mathsf{L}_j}^{V} \sum_{t=\max\{al_v, r_j\}}^{T} L_{j,t,v} = 1 \qquad \forall j \in \mathsf{P} \qquad (3)$$

$$\sum_{n=1}^{N} \sum_{s=t-pc_n+1}^{t} C_{n,s,u} \leq 1 \qquad \forall u \in \mathsf{U}, \forall t \geq ac_u \qquad (4)$$

$$\sum_{n=1}^{N} \sum_{s=t-pl_n+1}^{t} L_{n,s,v} \leq 1 \qquad \forall v \in \mathsf{V}, \forall t \geq al_v \qquad (5)$$

$$\sum_{v \in \mathsf{L}_n} \sum_{s=1}^{t} L_{n,s,v} \leq \sum_{u \in \mathsf{C}_n} \sum_{s=1}^{t-pc_n} C_{n,s,u} \qquad \forall n \in \mathsf{N}, \forall t \qquad (6)$$

$$F_j \geq pl_j + \sum_{v \in \mathsf{L}_j} \sum_{t=al_v}^{T} t * L_{j,t,v} - r_j \qquad \forall j \in \mathsf{P} \qquad (7)$$

$$F_j \geq pl_k + \sum_{v \in \mathsf{L}_j} \sum_{t=al_v}^{T} t * L_{k,t,v} - r_j \qquad \forall j \in \mathsf{P} \text{ and } k \text{ s.t. } b_{j,k} = 1 \qquad (8)$$

$$PT_j \geq pl_j + \sum_{v \in \mathsf{L}_j} \sum_{t=al_v}^{T} t * L_{j,t,v} - \sum_{u \in \mathsf{C}_j} \sum_{t=ac_u}^{T} t * C_{j,t,u} \qquad \forall j \in \mathsf{P} \qquad (9)$$

$$PT_j \geq pl_k + \sum_{v \in \mathsf{L}_k} \sum_{t=al_v}^{T} t * L_{k,t,v} - \sum_{u \in \mathsf{C}_k} \sum_{t=ac_u}^{T} t * C_{k,t,u} \qquad \forall j \in \mathsf{P} \text{ and } k \text{ s.t. } b_{j,k} = 1 \qquad (10)$$

$$PT_j \geq pl_j + \sum_{v \in \mathsf{L}_j} \sum_{t=al_v}^{T} t * L_{j,t,v} - \sum_{u \in \mathsf{C}_k} \sum_{t=ac_u}^{T} t * C_{k,t,u} \qquad \forall j \in \mathsf{P} \text{ and } k \text{ s.t. } b_{j,k} = 1 \qquad (11)$$

$$PT_j \geq pl_k + \sum_{v \in \mathsf{L}_k} \sum_{t=al_v}^{T} t * L_{k,t,v} - \sum_{u \in \mathsf{C}_j} \sum_{t=ac_u}^{T} t * C_{j,t,u} \qquad \forall j \in \mathsf{P} \text{ and } k \text{ s.t. } b_{j,k} = 1 \qquad (12)$$

$$PT_j \leq d_j \qquad \forall j \in \mathsf{P} \qquad (13)$$

Objective function

$$\min\left\{\alpha * \frac{1}{P} \sum_{j=1}^{P} F_j + \beta * \frac{1}{P} \sum_{j=1}^{P} PT_j\right\}$$

Additional constraints

$$\sum_{n=1}^{N}\sum_{t=1}^{ac_u-1} C_{n,t,u} = 0 \quad \forall u \in \mathsf{U} \qquad (14)$$

$$\sum_{n=1}^{N}\sum_{t=1}^{al_v-1} L_{n,t,v} = 0 \quad \forall v \in \mathsf{V} \qquad (15)$$

$$\sum_{v\in \mathsf{L}_j}\sum_{t=1}^{r_j-1} L_{j,t,v} = 0 \quad \forall j \in \mathsf{P} \qquad (16)$$

$$\sum_{u\notin \mathsf{C}_n}\sum_{t=1}^{T} C_{n,t,u} = 0 \quad \forall n \in \mathsf{N} \qquad (17)$$

$$\sum_{v\notin \mathsf{L}_n}\sum_{t=1}^{T} L_{n,t,v} = 0 \quad \forall n \in \mathsf{N} \qquad (18)$$

$$C_{n,cb_n,cm_n} = 1 \quad \forall n \in \mathsf{N} \text{ s.t. } cs_n = 1 \qquad (19)$$

$$L_{n,lb_n,lm_n} = 1 \quad \forall n \in \mathsf{N} \text{ s.t. } ls_n = 1 \qquad (20)$$

Comments

- Cutting and laying operations of all the jobs have to be scheduled: (1), (2) and (3)
- Each cutting or laying machine processes at most one job at a time: (4) and (5)
- For a given job, the cutting operation is done before the laying operation: (6)
- For a given part, the laying flow time corresponds to the time between the release date of the laying operation of its primary job and the finishing date of the last laying operation (primary or secondary job): (7) and (8)
- For a given part, the prescription time is the time between the start date of the first cutting operation (primary or secondary job) and the finishing date of the last laying operation (primary or secondary job): (9), (10), (11) and (12)
- For all the parts, the prescription time has to be lower than the prescription delay: (13)
- The operations cannot be scheduled before the availability date of the machines: (14) and (15)
- The laying operation of a primary job cannot be processed before the release date of the laying operation: (16)
- Some jobs can be scheduled on a given subset of machines: (17) and (18)
- Some operations have already been scheduled with no possibility to change them: (19) and (20)

Additional comments

- The parts composed by a single primary job j (without secondary job) verify $\sum_{k=P+1}^{N} b_{j,k} = 0$

- There are at most one secondary job for a given primary job: $\sum_{k=P+1}^{N} b_{j,k} \leq 1 \quad \forall j \in \mathsf{P}$
- For all the secondary jobs, there is an associated primary job: $\forall k \in \mathsf{S}, \exists\, j \in \mathsf{P}$ such that $b_{j,k} = 1$
- The availability date of a machine is always the finishing date of the last operation done on this machine for a job removed from the problem. Thus, even a "no change" operation has to be scheduled after the availability date of the machine.
- The start dates and machines of the "no change" operations are assumed to be compatible with the constraints of the problem

CHAPTER 6. ALGORITHMS

This chapter explains how the optimal solutions are obtained. After that, a specific heuristic algorithm is described.

6.1. Optimal solutions

To obtain the optimal solutions, the mathematical model is implemented using the *Xpress-MP* software. This IP software uses its own branch and bound technique to find the optimal solutions. For each scenario, it allows to know:

- The node where the optimal solution has been encountered and the time it takes
- The total number of nodes and the total time it takes to prove that the solution was optimal

After having solved the first scenario with different configurations, it is noticed that the solving speed decreases if **integer weights** and **total values** are used instead of average values in the objective function. This may be due to the algorithm of *XPRESS-MP* software which takes advantages of the fact that all the parameters are integer so the objective function, which is a weighted sum of integer values, has to take integer values. Thus, it helps the algorithm to improve by integer steps the lower bound of the objective function when it applies the branch and bound technique.
Of course, this does not change the values of the optimal solutions, because the objective function is only divided by a constant (*P*, the number of primary jobs). Thus, the new objective function used in the model is the weighted sum of the total flow time and the total prescription time of the primary jobs:

$$\min\left\{\alpha * \sum_{j=1}^{P} F_j + \beta * \sum_{j=1}^{P} PT_j\right\} \quad (21)$$

It will now be considered as the objective function for the IP software, even if the average values will be presented in the results.

6.2. Heuristic

6.2.1. Framework of the algorithm

General strategy

Heuristics for two stages hybrid flow shop problems in the literature mainly consists in **successively scheduling the jobs on stage one and two**. The basic idea is, in each stage, to define a **priority list** and to assign first the job with highest priority, based on an **assignment rule**. The assignment rule determines which machine will process the job. To sum up, the general framework of the algorithm is detailed in Algorithm 1.

```
For the 2 stages (cutting and laying)
     Define a priority list
     Assign the jobs in this order to the machines
End for
```

Algorithm 1: General framework of the heuristic algorithm

Finding a first feasible solution

The **main goal of this heuristic is to minimize the total flow time of the jobs**, which is the criteria with the largest weight in the objective function. The flow time corresponds to the difference between the finish date of the last laying operation (between the primary and secondary jobs) and the release date of the job. Thus, the heuristic tries to schedule the laying operations of the primary jobs just after their release date. To do that, the cutting operations have to be scheduled so that they are finished before the release dates. This is why the cutting priority list is the jobs sorted by increasing release dates. Then the cutting assignment rule consists in assigning the job with higher priority to the first cutting machine available.

By doing that, it is expected that the total time between the finish date of the cutting operations and the release dates of the jobs will be minimized. Consider ***er* the earliest ready date a job can start its laying operation**, i.e. the latest date between the cutting finish date and the release date. The goal is now to schedule the jobs as soon as possible after their earliest ready date. So the laying priority list is the jobs sorted by increasing earliest ready dates. The laying assignment rule is a little bit more sophisticated than the cutting one. Indeed, it is not profitable to schedule a job on the first available machine if this machine is available before *er*, because the job laying operation cannot start before *er*. So the machine would not work until *er*. Thus, the laying assignment rule considers two cases:

1. One machine is available before *er*. In this case, the job is scheduled on the **last** machine available before *er*.
2. No machine is available before *er*. In this case, the job is scheduled on the **first** machine available after *er*.

Improving this solution

This heuristic gives a feasible solution and is built in order to minimize the total flow time. Nevertheless, it is possible to improve the total prescription time of this solution, without declining the total flow time. The idea is to try to **bring forward the cutting operations of the jobs** so that they finish at the start date of the laying operations. This is done only if it does not affect the flow time of the other jobs.

Similarly, the total prescription time of the solution is improved by **bringing forward the laying operation of the secondary jobs**, which do not have the constraint to start after the release date. They are brought forward so that they finish before the laying operation of the primary job, but they start the latest possible so that they do not penalize the prescription time of the job. Once again, the laying operation is brought forward only if it does not affect the other jobs (the total flow time does not have to be modified).

Local search stage

Once this first feasible solution is obtained, an improved solution is sought via local search around the neighborhood of this solution. The strategy consists in changing the order of the jobs in the laying priority list and then to apply again the assignment rule. With this strategy, it is neither sure to improve the solution nor to find the optimal solution if all the jobs permutations are explored as priority lists. Indeed, it is not proved that there is always one job permutation that can be taken as laying priority list and will give the optimal solution if the assignment rule is applied. Furthermore, it is not sure the cutting operations are scheduled optimally. But this is the essence of a heuristic: it is expected to find a good solution but it is not sure to find the optimal solution.

More precisely, two different kinds of local search methods are explored:

1. By **swapping two adjacent jobs** on the laying priority list
2. By **moving the secondary jobs to all the possible positions** in the laying priority list.

6.2.2. Detailed algorithm

This section details exactly how works the heuristic algorithm. It uses the same notations as in the mathematical model. It mixes standard algorithm vocabulary, such as "*if, then, else… end if*" or "*for all … end for*" and full text, for example to explain how the priority list is defined.

Let defining:

LC_n : priority list of the jobs for the cutting operation, $n \in \mathsf{N}$

LL_n : priority list of the jobs for the laying operation, $n \in \mathsf{N}$

Some parts of the algorithm are labeled with circled numbers (from ① to ⑥). This corresponds to improvements that have been brought to the algorithm and that will be specifically commented on the section 6.2.3.

1. Initializing stage (Algorithm 2**)**

This stage takes into account the **jobs already scheduled**, with no possibility to change them. The availability dates of the machines are updated by setting them to the latest date a "no change" job finishes on this machine.

```
For all n s.t. (cs_n = 1)
     If (ac_cmn < cb_n + pc_n)
          ac_cmn = cb_n + pc_n
     End if
End for

For all n s.t. (ls_n = 1)
     If (al_lmn < lb_n + pl_n)
          al_cmn = lb_n + pl_n
     End if
End for
```

Algorithm 2: Heuristic initializing stage

2. Cutting stage (Algorithm 3**)**

First, this stage defines the **cutting priority list** (LC) and then assigns the jobs in the priority list order to the **first machine available**. At each step, the availability date of the machine is updated. Notice that if a job does not have a cutting operation, the pseudo-cutting operation starts and finishes at 1 on any cutting machine so that this is not restricting for the laying stage. This will be improved later (during the improving stage) by setting the cutting start and finish date of these jobs to the laying start date of these jobs.

Define LC:

Sort the primary jobs by increasing release dates r_n.

① *If some primary jobs have the same release date, sort first the job with the shortest cutting processing time.*

Insert the secondary jobs in the list before their primary job if their laying processing time is greater than the one of their primary job, after otherwise.

```
For all n ∈ LC
     If (pc_n = 0)
          cb_n = 1
          cm_n = any m ∈ C_n
     End if
     If (cs_n = 0)
          cm_n = u s.t. ac_u = min{ac_m} where m ∈ C_n
          cb_n = ac_u
     End if
     ac_cmn = cb_n + pc_n
End for
```

Algorithm 3: Heuristic cutting stage

3. Laying stage (Algorithm 4**)**

First, this stage computes the **earliest ready dates of the jobs** (er). For the secondary jobs with a cutting operation, the earliest ready date is their cutting finish date; for the others, this is the cutting start date of their primary job.
This stage next defines the **laying priority list** (LC) and then assigns the jobs in the priority list order to the last machine available before er (if there is one machine available before er) or to the first machine available after er.

```
For all n
     If n ∈ P
          er_n = max{r_n ; cb_n + pc_n}
     Else
          If (pc_n = 0)
               er_n = cs_p where (b_p,n = 1)
          Else
               er_n = cb_n + pc_n
          End if
     End if
End for


Define LL:
     Sort the primary jobs by increasing earliest ready dates er_n.
②    If some primary jobs have the same earliest ready date, sort first the job with the longest laying processing time.
     Insert the secondary jobs in the list before their primary job if their laying processing time is greater than the one of their primary job, after otherwise.


For all n in LL
     If (∃ m ∈ L_n s.t. (al_m < er_n))
          lm_n = v s.t. al_v = max{al_m} where ((m ∈ L_n) and (al_m < er_n))
          lb_n = al_v
     Else
          lm_n = v s.t. al_v = min{al_m} where (m ∈ L_n)
          lb_n = al_v
     End if

     al_lmn = lb_n + pl_n
End for
```

Algorithm 4: Heuristic laying stage

③ **4. Local search stage – swapping adjacent jobs (**Algorithm 5**)**

Let defining `CurrentFT` as the **value of the total flow time of the current solution**. This value will be tried to be improved by **doing iteratively stages 1 to 3 except that the adjacent jobs in the laying priority list of stage 3 are swapped**.

```
Store BestFT = CurrentFT

Do while BestFT is improved

     For (pos = 1 to (N-1))
          Do stages 1 to 3 but swap job at position pos with the next job in LL
          Compute CurrentFT
          If (CurrentFT < BestFT)
               BestFT = CurrentFT
               Store LL in nextLL
          End if
     End for

     Set LL = nextLL

Loop
```

Algorithm 5: Heuristic local search stage – swapping adjacent jobs

④ **5. Local search stage – moving secondary jobs (**Algorithm 6**)**

Let defining `CurrentFT` as the value of the total flow time of the current solution. This value will be tried to be improved by doing iteratively stages 1 to 3 **except that the secondary jobs in the laying priority list of stage 3 are moved** to all the possible positions in the list.

```
Store BestFT = CurrentFT

For all s in S

     For (pos = 1 to N)
          Do stages 1 to 3 but move job s at position pos in LL
          Compute CurrentFT
          If (CurrentFT < BestFT)
               BestFT = CurrentFT
               Store LL in nextLL
          End if
     End for

     Set LL = nextLL

End for
```

Algorithm 6: Heuristic local search stage – moving secondary jobs

6. Prescription time improving stage (Algorithm 7**)**

First, this stage defines a priority list (LV) by sorting the jobs by decreasing laying start dates. Then, it tries **to bring forward the laying operations of the secondary jobs.** This is possible only if the laying operation does not overlap with a previously scheduled operation. That is why the **machine date needed for laying** (*dl*) is updated as each step: a job cannot finish after this date.
After that, it tries similarly to **bring forward the cutting operation** of the job with highest priority so that it finishes when the laying operation starts. For each job, if the laying operation begins later than the finish date of the cutting operation, it is tried to bring it forward. This allows the cutting operations to not decrease the prescription time of the jobs, by starting it at the latest.

$$\forall v,\ dl_v = \max_{n \text{ s.t. } lm_n = v} \{lb_n + pl_n\}$$

```
⑤ For all v in V
      Define LV:
      Sort the jobs processed on laying machine v by decreasing laying start date
      For all n in LV
         If n ∈ S
            If (lbp+plp > lbn+pln) where (bp,n = 1)
               If (dlv > lbp+plp)
                  lbn = lbp+plp-pln
               Else
                  lbn = dlv-pln
               End if
            End if
         End if
         dlv = lbn
      End for
   End for
```

$$\forall u,\ dc_u = \max_{n \text{ s.t. } cm_n = u} \{cb_n + pc_n\}$$

```
⑥ For all u in U
      Define LU:
      Sort the jobs processed on cutting machine u by decreasing cutting start date
      For all n in LU
         If (cbn+pcn < lbn)
            If (dcu > lbn)
               cbn = lbn - pcn
            Else
               cbn = dcu - pcn
            End if
         End if
         dcu = cbn
      End for
   End for
```

Algorithm 7: Heuristic prescription time improving stage

6.2.3. Method used to find the algorithm

This heuristic has been built by focusing on **minimizing the total flow time of the primary jobs**. This analysis gives the main framework of the heuristic. Then, the total prescription time of the solution is improved, without modifying the best total flow time found. This results in the addition of stage 6 in the heuristic (improvements ⑤ and ⑥).

After having solved optimally the scenarios with IP software, the solutions have been carefully observed and compared with those of the heuristic. This allows modifying the heuristic in order to guide it toward the optimal solution. This result in two improvements (① and ②) in the priority lists definitions. Finally, this helps to define the local search stages to try to come as close as possible of the optimal solutions (improvements ③ and ④).

The fact the optimal solutions of the scenarios are known was of great interest to develop the heuristic. This is a major reason of the quality of the heuristic. The results of the heuristic will be extensively described in CHAPTER 7. The quality of the heuristic and how it increases with each of theses improvements will be extensively described.

CHAPTER 7. EXPERIMENTAL DESIGN AND RESULTS

This chapter explains how the scenarios are defined. Then, it details how they are solved with the heuristic and with IP software. Finally, it comments on the results of these experiments by giving the heuristic quality, the weights setting influence and the impact of the machine configurations.

All the experiments (optimal and heuristic solutions) have been processed on a 2 GHz workstation with 512 Go of RAM.

7.1. Scenarios definition[6]

7.1.1. Data collection

6 real scheduling scenarios the sponsor faced are used.

- ♦ The processing times are set to 75% of the maximum values observed.
- ♦ The time a part need from the end of the last laying operation until the beginning of the polymerization operation is computed assuming normally distributed arrivals at the autoclave with an 80% confidence value.
- ♦ The number of cutting machines and laying machines are those currently owned by the plant.

7.1.2. Time period and horizon

The current "by hand" scheduling process considers the shift as the time period. It means the "by hand" scheduling process gives, for each operation, the shift during which the operation is supposed to start. If several operations are scheduled to start during the same shift, then the operations are sequenced.

In our model, the time period: $t = 1 \ldots T$ is not defined. The duration of the time period is a compromise between the precision of the schedule and the complexity, i.e. the number of variables of the model. The time period is supposed sufficiently small in order to have a good precision in our schedule. It is at least a subdivision of a shift and the number of periods in a horizon is about 300 on average.

The time discretization results in approximation of the actual schedules but is of great interest for the optimal resolution. The consequences on the plant operation are:

- Jobs could be run earlier than the schedule (solution) during operation, but the sequence will be the same (semi-active scheduling).
- Use of discrete time periods hence underestimates actual performance.

[6] Actual data is confidential and hance cannot be reproduced in the document

7.1.3. Weight setting

The objective function (21) considers two criteria: the total flow time and the total prescription time, with respectively the weights α and β.

The sponsor wants the total flow time to be the most important criterion, which means $\alpha > \beta$. A simple way to incorporate this to set $\alpha = N.\beta$, which indicates one can loose one hour on the total flow time only if this results in *N* hours savings on the total prescription time.

An extreme case would be to find the best solution in order to minimize the total flow time and then, without the possibility to decline the value of the total flow time, to minimize the total prescription time. To do that, α has to be set to the maximum value that can take the total prescription time, whatever the solution is. For example constraint (13) ensures whatever the solution is, the prescription time for a job is lower or equal than the prescription delay for this job ($PT_j \leq d_j \quad \forall j \in \mathsf{P}$).

Thus, whatever the solution is: $\sum_{j=1}^{P} PT_j \leq \sum_{j=1}^{P} d_j$. In other words, $\sum_{j=1}^{P} d_j$ is an upper bound of the total prescription time. By setting $\alpha = \sum_{j=1}^{P} d_j$ and $\beta = 1$, the solution with the minimum possible average flow time will be found.

To see how the weights setting impacts on the solutions, two scenarios are solved optimally by changing the weights α and β. Table 5 and Table 6 give for each weight setting, the values of the average flow time and the average prescription time of the optima solution. The time needed by IP software to find the optimal solution is also given.

		OPTIMAL SOLUTION		
α	β	$\frac{1}{P}\sum_{j=1}^{P} F_j$	$\frac{1}{P}\sum_{j=1}^{P} PT_j$	Total time (s)
4000 ([7])	1	48.25	53.25	293
500	1	48.25	53.25	202
20	1	48.25	53.25	657
9	1	48.25	53.25	123
8	1	48.25	53.25	437
5	1	48.25	53.25	1345
3	1	48.25	53.25	1035
2	1	48.25	53.25	2859
1	1	48.5	53	616
1	2	48.75	52.75	943
1	3	48.75	52.75	2224

Table 5: Weight settings influence on the optimal solution (scenario 1)

[7] $\alpha = \sum_{j=1}^{P} d_j$

		OPTIMAL SOLUTION		
α	β	$\frac{1}{P}\sum_{j=1}^{P} F_j$	$\frac{1}{P}\sum_{j=1}^{P} PT_j$	Total time (s)
2500 ([8])	1	63.2	105.2	4
500	1	63.2	105.2	8
20	1	63.2	105.2	25
9	1	63.2	105.2	98
8	1	63.2	105.2	75
5	1	63.2	105.2	630
3	1	63.2	105.2	29
2	1	65	99.8	33
1	1	65	99.8	29

Table 6: Weight settings influence on the optimal solution (scenario 2)

The optimal solution does not change when $\alpha \geq 2\beta$ for the first scenario and when $\alpha \geq 3\beta$ for the second scenario. Notice the time required to obtain the optimal solution vary a lot when the weights change. For the first scenario, the fastest optimal solution found is when $\alpha = 9$ and $\beta = 1$. For the second scenario, the fastest optimal solution found is when $\alpha = \sum_{j=1}^{P} d_j = 2500$ and $\beta = 1$. Nevertheless, a relation between the weights chosen and the time needed to find the optimal solution cannot be found from these two scenarios.

Because it is time-consuming to solve all the scenarios with these different weight settings, **two scheduling modes** are defined:

1. The total flow time criterion dominates strongly the average prescription time criterion: $\alpha = 9\beta$
2. The total flow time criterion dominates slightly the average prescription time criterion: $\alpha = 3\beta$

Thus, one can test if the same optimal solutions are found for $\alpha = 3\beta$ and $\alpha = 9\beta$, for all the scenarios. It will also allow comparing the time it takes to find the optimal solutions in these two cases.

The two modes characteristics are sum up in Table 7.

Mode	α	β	**Comments**
1	9	1	$\alpha = 9\beta$: FT[9] dominates strongly PT[10]
2	3	1	$\alpha = 3\beta$: FT[9] dominates slightly PT[10]

Table 7: The two scheduling criteria modes

[8] $\alpha = \sum_{j=1}^{P} d_j$

[9] FT: flow time

[10] PT: prescription time

7.1.4. Alternative scenarios

From the 6 initial scenarios, alternative scenarios are defined by modifying some parameters:

- **The two scheduling criteria modes** ($\alpha = 9\beta$ and $\alpha = 3\beta$)
- **The machine configurations**: two machine configurations are defined; the first one is the current machine configuration and the second one take into account investments in new machines, scheduled next year.

Consequently, 24 scenarios have to be solved: 6 initial scenarios multiplied by 4 alternative configurations, as shown in Table 8.

Scenarios numbers	Weights modes	Machines configuration
1 to 6	$\alpha = 9\beta$	Current
7 to 12	$\alpha = 3\beta$	Current
13 to 18	$\alpha = 9\beta$	Future
19 to 24	$\alpha = 3\beta$	Future

Table 8: Scenario numbers and configurations

7.2. Experiments

For each scenario, three different types of experiments are done:

1. Solve it optimally with an IP solver,
2. Solve it with the heuristic,
3. Solve it optimally again with the IP solver by adding a cutoff to the objective function, which is the value of the objective function of the heuristic solution.

7.2.1. Optimal solutions

The optimal solutions are obtained with the *Xpress-MP* software. For each scenario, it allows to know:

- The node at which the optimal solution has been found and the time it takes
- The total number of nodes and the total time it takes to prove that the solution is optimal

7.2.2. Heuristic solutions

The heuristic algorithm has been developed with Microsoft Excel 2002 and Microsoft Visual Basic for Applications 6.3. This choice has been recommended by the company because they already used Excel to define their schedules so it will be easier to integrate the heuristic algorithm. To be able to compare the heuristic solution with the optimal solution, the time it takes to find the solution is stored.

The results given by the heuristic for each improvement done are stored (improvements ① to ⑥ described in section 6.2). Thus, it is possible to **measure the improvement** on the heuristic quality related to each improvement.

7.2.3. Optimal solutions with cutoff

The scenarios are solved again with the *Xpress-MP* software by adding a cutoff to the objective function. The cutoff is set to the value of the objective function of the solution given by the heuristic.

7.3. Results

7.3.1. Optimal solutions

Table 9 gives the optimal solutions for the 16 scenarios and details the node and the time it takes to find the optimal solution and to prove this solution is optimal.

Sc.	α	β	Mach. Config.	Optimal solution Av FT	Av PT	Optimal found Node	Time (s)	Optimal proved Node	Time (s)
1	9	1	Current	48.25	53.25	165	48	178	49
2	9	1	Current	53.8	105.2	438	344	612	344
3	9	1	Current	79.56	91.89	68	265	68	266
4	9	1	Current	65.5	76.83	6	34	983	566
5	9	1	Current	67.5	76.12	2	4	3	4
6	9	1	Current	126.86	95.43	2	47	3	48
7	3	1	Current	48.25	53.25	600	170	675	170
8	3	1	Current	53.8	105.2	702	775	846	775
9	3	1	Current	79.78	91	3	132	3	132
10	3	1	Current	65.5	76.83	3	33	10009	3582
11	3	1	Current	67.5	76.12	3	4	5	4
12	3	1	Current	126.87	95.43	2	361	3	362
13	9	1	Future	46.5	52.75	755	327	861	328
14	9	1	Future	48.8	99.8	4	13	5	13
15	9	1	Future	62.33	86.89	2	4	3	4
16	9	1	Future	55	71.5	15	18	39	21
17	9	1	Future	51.87	75.37	462	119	503	120
18	9	1	Future	85.71	93.71	2	6	3	7
19	3	1	Future	46.5	52.75	231	122	284	122
20	3	1	Future	48.8	99.8	532	373	592	374
21	3	1	Future	62.33	86.89	2	4	3	4
22	3	1	Future	55	71.5	3	26	5	27
23	3	1	Future	51.87	75.37	4	9	7	9
24	3	1	Future	85.71	93.71	1	0.1	1	1

Table 9: Scenarios optimal solutions obtained with the IP solver

The total time to obtain the optimal solution and prove this solution is optimal is 5.1 minutes on average with a standard deviation of 12.1 minutes. The total time observed varies from 1 second to 59.7 minutes. The first interesting point is that the **time it takes to find the optimal solution varies a lot** from one scenario to another: there is a large deviation on the time needed to obtain the optimal solution. This deviation is not due to some specific parameters but to a combination of factors which are difficult to isolate. This is due to the specific algorithms implemented in IP software to find the optimal solution: they may require different techniques from one scenario to another with different probabilities of success. Nevertheless, it was possible to find the optimal solution within a **reasonable amount of time** with the IP solver for all the scenarios.

In Table 9, the total time is decomposed in the time to find the optimal solution and the time to prove this solution is optimal. For example, software needs 48 seconds to find the optimal solution of scenario 1 and after having found this solution, only 1 second to prove it is the optimal solution (total time is 49 seconds). For 67% of the scenarios, the time needed to prove the solution is optimal is less than 2% of the total time. This means it is generally **better to wait the end of the algorithm** rather than stopping it by expecting to have determined the best solution and "losing" time to prove this is the optimal solution. Indeed, when the optimal solution is found, it is usually fast to prove this is the optimal. In practice, a maximum time after which the algorithm stops is usually set, even if an optimal solution has not been found. For the 24 scenarios, if it had been set to one hour, the optimal solutions would have been obtained for all the scenarios.

7.3.2. Heuristic solutions and comparison

Table 10 gives the heuristic and optimal solutions for the 16 scenarios and details the time it takes to find the optimal solution and to prove this solution is optimal. The heuristic quality is defined as follow:

$$\text{heuristic quality} = \left(1 - \frac{\text{heuristic objective function value - optimal value}}{\text{optimal value}}\right) \times 100\%$$

				Optimal solution			Heuristic solution			
Sc.	α	β	Mach. Config.	Av FT	Av PT	Time (s)	Av FT	Av PT	Time (s)	Heuristic quality
1	9	1	Current	**48.25**	53.25	49	**48.25**	53.5	10	99.9 %
2	9	1	Current	**53.8**	**105.2**	344	**53.8**	**105.2**	5	100 %
3	9	1	Current	79.56	91.89	266	80.222	98.778	6	98.4 %
4	9	1	Current	65.5	76.83	566	68.833	71.667	4	96 %
5	9	1	Current	67.5	76.12	4	74.125	75.75	4	91.3 %
6	9	1	Current	**126.86**	**95.43**	48	**126.86**	**95.43**	5	100 %
7	3	1	Current	**48.25**	53.25	170	**48.25**	53.5	10	99.9 %
8	3	1	Current	**53.8**	**105.2**	775	**53.8**	**105.2**	5	100 %
9	3	1	Current	79.78	91	132	80.22	98.78	6	98.4 %
10	3	1	Current	65.5	76.83	3582	68.83	71.67	4	96 %
11	3	1	Current	67.5	76.12	4	74.12	75.75	4	91.3 %
12	3	1	Current	**126.86**	**95.43**	362	**126.86**	**95.43**	5	100 %
13	9	1	Future	**46.5**	52.75	328	**46.5**	53.25	9	99,9%
14	9	1	Future	48.8	99.8	13	49	118	5	96,3%
15	9	1	Future	62.33	86.89	4	63.67	107.78	7	94,9%
16	9	1	Future	55	71.5	21	55.67	81.17	5	97,2%
17	9	1	Future	**51.87**	75.37	120	**51.87**	76.87	5	99,7%
18	9	1	Future	85.71	93.71	7	87.86	99.28	5	97,1%
19	3	1	Future	**46.5**	52.75	122	**46.5**	53.25	9	99,7%
20	3	1	Future	48.8	99.8	374	49	118	5	92,4%
21	3	1	Future	62.33	86.89	4	63.67	107.78	7	90,9%
22	3	1	Future	55	71.5	27	55.67	81.17	5	95,1%
23	3	1	Future	**51.87**	75.37	9	**51.87**	76.87	5	99,4%
24	3	1	Future	85.71	93.71	1	87.86	99.29	5	96,6%

Table 10: Scenarios heuristic solutions and heuristic quality

Heuristic quality

The heuristic quality is very good: 97.2% on average with a standard deviation of 2.3%. Furthermore, the heuristic quality is always greater than 90%, and greater than 95% for 19 scenarios out of 24. The quality is even better if the only criteria considered is the most important one: the average flow time. In this case, the average flow time quality is greater than 95% for 20 scenarios out of 24.

Time to obtain the solution with the heuristic

The time to obtain the heuristic solution is very satisfying: 5.8 seconds on average with a standard deviation of 1.9 seconds. This time is always less than 10 seconds. The variations can be explained by the local search stage, which is efficient (so lengthy) for some scenarios and not for the others. Anyway, this time is very short compared to the time needed for the IP solver to find the optimal solution, which is about 5.1 minutes.

Heuristic improvements

On average, the heuristic does a **good job of improvement**: the heuristic quality increases on average from 76.6% initially to 97.2%.

Figure 15 shows the evolution of the **average flow time quality** of the first six scenarios with the heuristic improvements. The improvements considered in the graph are:

- Improvement ① - New cutting priority list
- Improvement ② - New laying priority list
- Improvement ③ - Addition of the local search stage where adjacent jobs are swapped in the priority list
- Improvement ④ - Addition of the local search stage where the secondary jobs are moved in the priority list

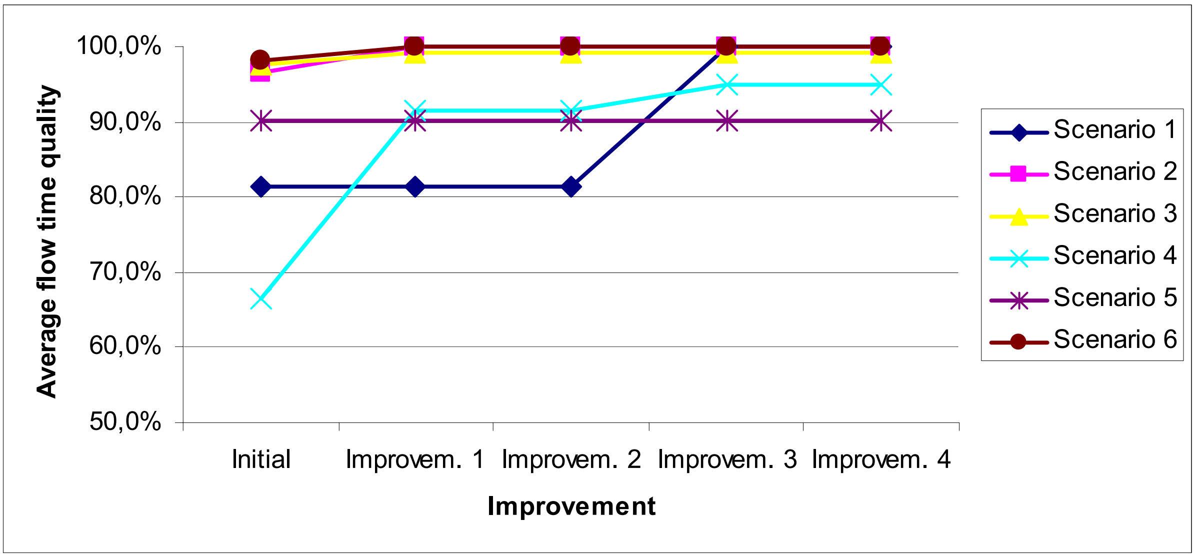

Figure 15: Evolution of the average flow time quality with the heuristic improvements (Scenarios 1 to 6 only)

The two efficient improvements are improvement 1 and 3. Improvement 1 allows sorting first the job with the shortest cutting processing time in case two jobs have the same release date. Improvement 3 is the local search stage where adjacent jobs are swapped and also results in very good improvements.

For these 6 scenarios, the improvements 2 and 4 have not led to average flow time improvements, which do not mean there are unnecessary. They are less efficient than improvements 1 and 3 but they may result in improvements for some specific scenarios:

- Scenarios where some jobs have the same earliest ready dates: in this case, improvement 3 will allow sorting first the job with the longest laying processing time.
- Scenarios where processing times of the secondary jobs are much more short or long than their primary jobs: in this case, it will be probably worthy to try to move the secondary jobs in the laying priority list.

Figure 16 shows the evolution of the **average prescription time quality** of the first six scenarios with the heuristic improvements. The improvements considered in the graph are:

- ➢ Improvement ⑤ - Bring forward the laying operations of the secondary jobs
- ➢ Improvement ⑥ - Bring forward the cutting operations

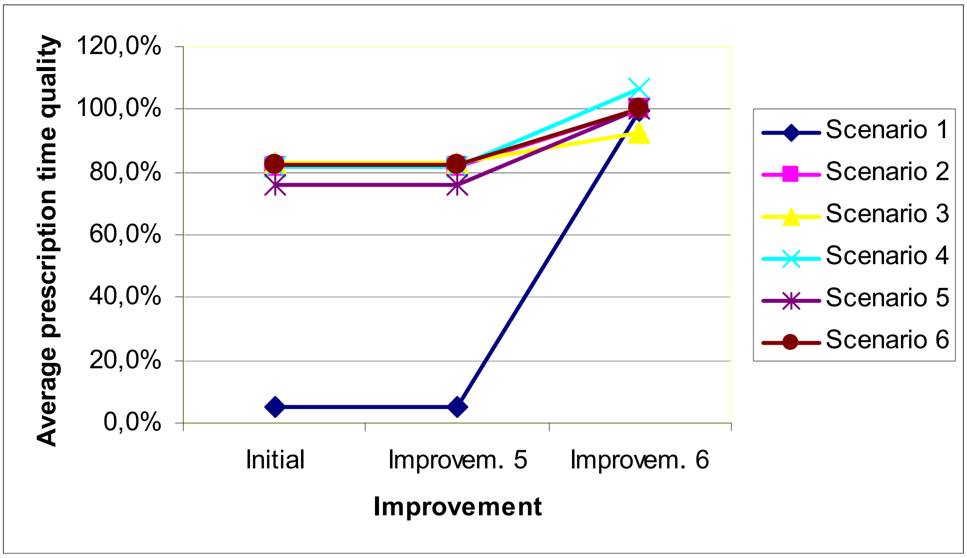

Figure 16: Evolution of the average prescription time quality with the heuristic improvements (Scenarios 1 to 6 only)

The most efficient improvement is improvement 6. It allows bringing forward the cutting operations of the jobs so that they finish just before the start date of their laying operation. This results in dramatic improvements because the heuristic is built in this way. Indeed, in the first step, it is tried to schedule as soonest as possible the cutting operations so that the job could be available when they are released. At the end, it is logical to bring these jobs forward, once their laying operations are set on the schedule.

Notice that for some scenarios, the prescription time quality is greater than 100% after improvement 6. This means the heuristic solution has an average prescription time greater than the optimal solution even if its objective function value is lower (the objective function is a weighted sum of the average flow time and the average prescription time).

As regards improvement 5, it has not led to average flow time improvements for the first six scenarios. Nevertheless, it may result in improvements for some specific scenarios, where the secondary jobs have been scheduled too early compared to their primary jobs.

7.3.3. Optimal solutions with cutoff

Table 11 gives the time needed by the IP solver to obtain the optimal solutions without and with cutoff. It details the time it takes to find the optimal solution and the total time it takes to find it and to prove this is the optimal solution.

				TIME NEEDED **WITHOUT CUTOFF**		TIME NEEDED **WITH CUTOFF**	
Sc.	α	β	Mach. Config.	Optimal found (s)	Optimal proved (s)	Optimal found (s)	Optimal proved (s)
1	9	1	Current	48	49	56	56
2	9	1	Current	344	344	131	131
3	9	1	Current	265	266	128	129
4	9	1	Current	34	566	18	442
5	9	1	Current	4	4	3	4
6	9	1	Current	47	48	51	51
7	3	1	Current	170	170	32	32
8	3	1	Current	775	775	237	237
9	3	1	Current	132	132	174	176
10	3	1	Current	33	3582	29	2095
11	3	1	Current	4	4	3	4
12	3	1	Current	361	362	99	100
13	9	1	Future	327	328	311	311
14	9	1	Future	13	13	124	125
15	9	1	Future	4	4	3	3
16	9	1	Future	18	21	6	7
17	9	1	Future	119	120	4	4
18	9	1	Future	6	7	3	4
19	3	1	Future	122	122	121	121
20	3	1	Future	373	374	116	117
21	3	1	Future	4	4	0.1	1
22	3	1	Future	26	27	35	36
23	3	1	Future	9	9	50	51
24	3	1	Future	0.1	1	0.1	1

Table 11: Time needed by the IP solver to obtain the optimal solutions (with and without the cutoff)

Table 12 gives statistics about the time needed by the IP solver to obtain the optimal solutions without and with cutoff. By addition of a cutoff in the model, the solving time is significantly decreased (43% saving on the average time) such as the deviation of the data (41% saving on the average standard deviation).

	TIME NEEDED **WITHOUT** CUTOFF		TIME NEEDED **WITH** CUTOFF	
	Optimal found (min)	Optimal proved (min)	Optimal found (min)	Optimal proved (min)
Average	2.2	5.1	1.2	2.9
Standard deviation	3.1	12.1	1.4	7.1
Minimum value	0.1	0.1	0.1	0.1
Maximum value	12.9	59.7	5.2	35.0

Table 12: Comparison of time needed by the IP solver with and without cutoff

7.4. Comments on the alternative scenarios

7.4.1. Weights setting influence on the optimal solutions

The scenarios have been solved with **two different scheduling criteria modes** ($\alpha = 9\beta$ and $\alpha = 3\beta$). What can be observed from Table 9 is that all scenarios except scenario 3 have the same results for the two criteria modes. The optimal solutions of scenario 3 are presented on Table 13. The difference is only a 0.22 hour loosing on the average flow time and a 0.89 hour saving on the average prescription time.

α	β	Av FT	Av PT
9	1	79.56	91.89
3	1	79.78	91

Table 13: Optimal solutions of scenario 3 for the two scheduling criteria modes

There is not a difference significant enough between the two modes to conclude on a best weight setting for the company: the two modes are comparable in term of efficiency. As a consequence, the only crucial factor to advise to set one of the two modes is the time needed to solve the scenarios. The preferred mode will be the one where the optimal solution is obtained the faster.

For the scenarios solved with $\alpha = 9\beta$, the average time to obtain the optimal solution is 2.5 minutes while for the scenarios solved with $\alpha = 3\beta$, it is equal to 7.7 minutes. Thus, if the sponsor wants to really concentrate on the average flow time, it is advised to the sponsor to choose the mode where $\alpha = 9\beta$ to solve the scenarios, so that the results are obtained faster.

7.4.2. Impact of the machine configurations (current vs. future) on the heuristic

Table 14 sums up the heuristic average quality for the two machine configurations (current vs. future).

	Heuristic Average Quality	
Machine config.	Objective function	Average Flow Time
CURRENT	97.9%	97.4%
FUTURE	96.6%	99.0%

Table 14: Machine configurations and heuristic quality

Notice the global quality is better for the current machine configuration but if the average flow time is the only criteria considered, i.e. the most important criteria for the company, the heuristic provides better results for the future machine configuration. Thus, one can argue the heuristic is really well designed in order to optimize the average flow time, even with the future machine configuration (investments in new machines).

CHAPTER 8. DISCUSSION

This chapter comments the results of this research: the benefits to use IP or heuristic software, the choice the company has to do now, the savings for the company.

8.1. IP software contributions

In this research, the use of IP software was of great interest for two reasons:

- It gives the optimal solutions of the scenarios so that it is possible to know how good was the results of the heuristic
- It helps to improve the heuristic by looking at the characteristics of the optimal solutions and try to lead the heuristic toward these solutions.

Furthermore, the use of IP software allows proving the mathematical model was right. By testing the model with small instances of the problem, the mathematical model was inferred to work and to use it for the real scenarios.

8.2. Heuristic software contribution

The heuristic software also contributes to the success of this research. Firstly, the sponsor wanted to have a decision support tool to give them very quickly a "good" schedule. The heuristic will provide schedules not far from optimal schedules. Secondly, the heuristic helps the IP solver to find the optimal solutions faster. By adding the heuristic solution as a cutoff value in the model, the solving time and its deviation significantly decrease.

8.3. Comparison with current "by hand" scheduling process

The concept of scheduling quality has been introduced by this research project in the plant. Thus, it is difficult to know with precision the current "by hand" scheduling quality. Unfortunately, this thesis cannot handle precisely how high the "by hand" scheduling quality is. It can be roughly estimated to 70% (+/- 10%), so that the use of the heuristic could result in a 25% quality increase.

8.4. A choice for the company

Right now, the company has to choose between two options:

- Use IP software to solve their day-to-day scenarios
- Use the heuristic to solve their day-to-day scenarios.

Table 15 gives the advantages and disadvantages of each option. For practical considerations, it is recommended to the company to firstly use the heuristic. Indeed, it will be easy to develop and adapt with the current system and it will provide good results. Furthermore, the heuristic will be needed anyway to improve the solving speed of the IP solver.

In the future, it is advised to use IP software in order to obtain optimal schedules.

Option	Advantages	Disadvantages
IP SOFTWARE	• Provides optimal solution • Provides robust solution (new constraints can be added in the model) • Mathematical model already implemented	• Cost • Time to obtain the solutions • Needs development to adapt with current system • Needs expertise to adapt with current system
HEURISTIC	• Solutions obtained quickly • Algorithm already developed with Microsoft Excel and Microsoft Visual Basic for Applications	• Solutions are not optimal • Solutions are not robust (the algorithm has to be modified if new constraints need to be added) • Needs development to adapt with current system

Table 15: Advantages and disadvantages of IP software vs. heuristic

8.5. Savings for the company

Whatever the choice done by the sponsor (use the heuristic or use IP software), this will result in savings for the company:

- They will have an easy to use tool to schedule the jobs on the machines (decision support tool).
- They will not spend time to do the schedule "by hand".
- It will ensure the prescription constraints are respected.
- It will optimize the average flow time and average prescription time of the jobs.
- It will result in better occupation of the machines and the tools.

CHAPTER 9. CONCLUSION AND FUTURE WORK

This chapter firstly presents the conclusions of this research and then points out some future research directions.

9.1. Conclusions

This research problem was based from a real issue and little research has been done to address real two stage hybrid-flow shop scheduling problems. The original constraints of the real problem, for example the prescription constraints, the bi-criteria optimization and the fact that there are "already taken" decisions, well complicate the classical research results.

Lot of research work has been done to address simple two stage hybrid flow shop scheduling problems but few of them address problems based on real configurations. An original approach is proposed: a mathematical model is given based on time-indexed variables and is implemented with IP software. Because the company also wants a fast procedure to solve the problem, a specific heuristic is developed, after having carefully observed the characteristics of the optimal results given by IP software. The heuristic algorithm gives solutions quickly and they are not far from the optimal solutions of IP software.

Of course, more work could be done to improve the heuristic, for example by developing metaheuristic techniques. However, the first goal for the company was to have a quick procedure which gives good results.

9.2. Future work

One can imagine future work to improve the quality of the heuristic, by using metaheuristic techniques such as simulated annealing, tabu search or genetic algorithm. This techniques are the most accurate to solve problems with particular constraints and are of great interest for the research community. Another improvement can be done by choosing a smaller time period, but attention should be brought to the consequence on the optimal resolution time.

An exhaustive additional study is also needed to figure out how efficient is the current "by hand" scheduling process and to compare it with the heuristic and optimal solutions. It will be very important to have a good idea of how much benefit is to be gained from this approach vs. the present one in order to convince the plant management to support actively this project.

One can also imagine a more integrated scheduling system in the composite unit, which could take into account the tool occupation optimization and the autoclaves activity optimization. This can be a future research project for the sponsor.

REFERENCES

Aghezzaf, E. H., & Artiba, A. (1998). Aggregate planning in hybrid flowshops. *International Journal of Production Research*.

Arthanari, T. S., & Ramamurthy, K. G. (1971). An extension of two machines sequencing problem. *Operations Research Letters*, Vol. 8, 10-22

Azizoglu, M., Cakmak, E., Kondakci, S. (2001). A flexible flowshop problem with total flow time minimization. *European Journal of Operational Research*, Vol. 132, 528-538.

Brah, S. A., & Loo, L. L. (1999). Heuristics for scheduling in a flow shop with multiple processors. *European Journal of Operational Research*, Vol. 113, 113-122.

Chen, B., Potts, C. N., and Woeginger, G. J. (1998). A review of machine scheduling: Complexity, algorithms and approximability. *Handbook of Combinatorial Optimization.* Kluwer Academic Publishers

Chou, F. D., & Lee, C. E. (1999). Two-machine flowshop scheduling with bicriteria problem. *Computers & Industrial Engineering*, Vol. 36, 549-564.

Dessouky, M. M., Dessouky, M. I., Verma, S. K. (1998). Flowshop scheduling with identical jobs and uniform parallel machines. *European Journal of Operational Research*, Vol. 109, 620-631.

Dauzère-Pérès, S. (1995). Minimizing late jobs in the general one machine scheduling problem. *European Journal of Operational Research*, Vol. 81, 134-142.

Fortemps, P., Ost, C., Pirlot, M., Teghem, J., Tuyttens, D. (1996). Using metaheuristics for solving a production scheduling problem in a chemical firm. *International Journal of Production Economics*, Vol. 46-47, 13-26.

Guinet, A., Solomon, M. M., Kedia, P. K., Dussauchoy, A. (1996). A computational study of heuristics for two-stage flexible flowshops. *International Journal of Production Research*.

Guinet, A., & Solomon, M. M. (1996). Scheduling hybrid flowshops to minimize maximum tardiness or maximum completion time. *International Journal of Production Research*.

Gupta, J. N. D. (1988). Two-Stage, Hybrid Flowshop Scheduling Problem. *Journal of the Operational Research Society*, Vol. 39, No.4, 359-364.

Gupta, J. N. D., & Tunc, E. A. (1994). Scheduling a two-stage hybrid flowshop with separable setup and removal times. *European Journal of Operational Research*, Vol. 77, 415-428.

Gupta, J. N. D., & Tunc, E. A. (1998). Minimizing tardy jobs in a two-stage hybrid flowshop. *International Journal of Production Research*.

Haouari, M., & M'Hallah, R. (1997). Heuristic algorithms for the two-stage hybrid flowshop problem. *Operations Research Letters*, Vol. 21, 43-53

Hax, A. C., & Candea, D. (1984). *Production and Inventory Management*. Prentice Hall, Inc.

Moursli, O., & Pochet, Y. (2000). A branch-and-bound algorithm for the hybrid flowshop. *International Journal of Production Economics*, Vol. 64, 113-125.

Nowicki, E., & Smutnicki, C. (1998). The flow shop with parallel machines: A tabu search approach. *European Journal of Operational Research*, Vol. 106, 226-253.

Portmann, M. C., Vignier, A., Dardilhac, D., and Dezalay, D. (1998). Branch and bound crossed with GA to solve hybrid flowshops. *European Journal of Operational Research*, Vol. 107, 389-400.

Rajendran, C., & Chaudhuri, D. (1992). A multi-stage parallel-processor flowshop problem with minimum flowtime. *European Journal of Operational Research*, Vol. 57, 111-122.

Riane, F., Raczy, C., Artiba, A. (1999). Hybrid auto-adaptable simulated annealing based heuristic. *Computers & Industrial Engineering*, Vol. 37, 277-280.

Riane, F., Artiba, A., and Iassinoviski, S. (2001). An integrated production planning and scheduling system for hybrid flowshop organizations. *International Journal of Production Economics*, Vol. 74, 33-48.

Sipper, D., & Bulfin, R. L. (1997). *Production Planning, Control, and Integration*. The McGraw-Hill Companies, Inc.

Sriskandarajah, C., & Sethi, S. P. (1989). Scheduling algorithms for flexible flowshops: worst and average case performance. *European Journal of Operational Research*, Vol. 43, 143-160.

Vignier, A., Billaut, J.-C., and Proust, C. (1999). Les problèmes d'ordonnancement de type flowshop hybride : état de l'art. *RAIRO/RO*, 33(2): 117-183.

Vignier, A., Billaut, J.-C., Proust, C., T'Kindt, V. (1996). Resolution of Some 2-stage Hybrid flowshop Scheduling Problems. *IEEE International Conference on Systems Man and Cybernetics, Information Intelligence and Systems*, 4, 2934-2941.

APPENDICES

APPENDIX 1: GLOSSARY

Composite manufacturing organization

Aircraft part: Subdivision of an aircraft, itself composed of different **elementary parts** assembled in the **Assembly Unit**.

Composites Unit: In charge of the manufacturing of all the **elementary parts** in composite of the **aircrafts parts** made at Nantes plant. The Composites Unit is organized in **EP lines**.

Assembly Unit: In charge of the assembling of the **aircraft parts**.

EP line: In charge of the manufacturing of the whole range of Elementary Parts needed for a specific **aircraft part**.

Shared Means line: In charge of all the manufacturing means needed by **several EP lines**. The Shared Means line is composed of two main workshops: the **lay-up machines workshop** (LUMW) and the **polymerization workshop**.

LUMW (Lay-Up Machines Workshop): In charge of the cutting and laying machines.

Polymerization Workshop: In charge of polymerize an elementary part on a tool (chemical reaction, which combines several elementary molecules: the monomers).

Elementary part: Composite part of an aircraft part, made of one primary **component** (which needs a specific prepared **tool** to be laid), and sometimes a secondary **component** (which not need a specific tool to be laid). The primary and secondary components are assembled in order to obtain the elementary part, which is then polymerized.

Component: Composite subset of an elementary part. It is a set of layers stacked and positioned in accordance with lay-up drawing.

Tool: Metallic structure, which gives the shape of the manufactured primary component. The tool has to be prepared before the beginning of the laying operation.

Composite raw material

CFRP: Composite Fiber Reinforced Plastic

Roll: composite raw material format, the composite fibers are oriented in a single direction, they are protected by a separator and are rolled to obtain the roll.

Fiber: Small diameter, very long unit textile element. The fibers are discontinuous but the word is frequently used to designate filaments (continuous unit textile element).

Cutting and laying process

ACCESS: Advanced Composite Cassette Edit/Shear System: machine which cut the composite material from rolls and replace it between two protective components in a cassette.

ATLAS: Advanced Tape Laying System: machine which receives cassettes with complex cutouts and lays up them.

Cassette: Same dimensions as the roll, the content of the cassette is the geometrical cutouts, ready to lay-up.

Layer: Surface of material made of one or more plies, the superposition of which, in the order defined in the lay-up table, constitutes the elementary part.

Lay-up: Operation of assembling plies which respect the position, orientation and stacking sequence defined by the lay-up drawing.

Orientation: Angle of the direction of the fibers in a layer with respect to the orientation definition "rose" (0 to 180°).

Ply: Single surface defined by a geometrical shape and a single fiber. A ply is defined geometrically by an external contour and possibly by one or more internal contours.

APPENDIX 2: ACCESS and ATLAS SPECIFICATIONS

ACCESS specifications (complex plies)

ACCESS is the machine used to realize the complex plies. Figure 17 explains the process:

- The raw material rolls are on the left, the protective components are removed
- The ACCESS machine cuts the carbon fibers, following the plies description (geometrical shapes)
- The plies are then rolled around a new protective component to obtain a cassette.

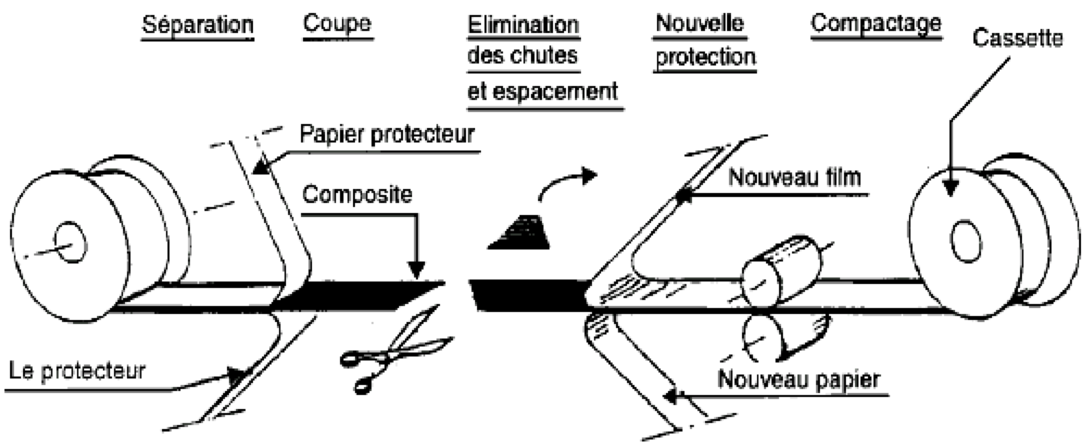

Figure 17: ACCESS process

ATLAS specifications (simple plies and lay-up)

The ATLAS machine allows simple plies (from rolls) to be cut and to lay up these simple plies and the complex plies on a tool.

Simple vs. Complex plies

Consider a simple component, made of three superposed squared layers. Each layer is composed of some juxtaposed plies, as drawn on Figure 18. The geometrical shape of each ply determines if the ply is simple and complex and thus if it has to cut by the ACCESS or the ATLAS machine.

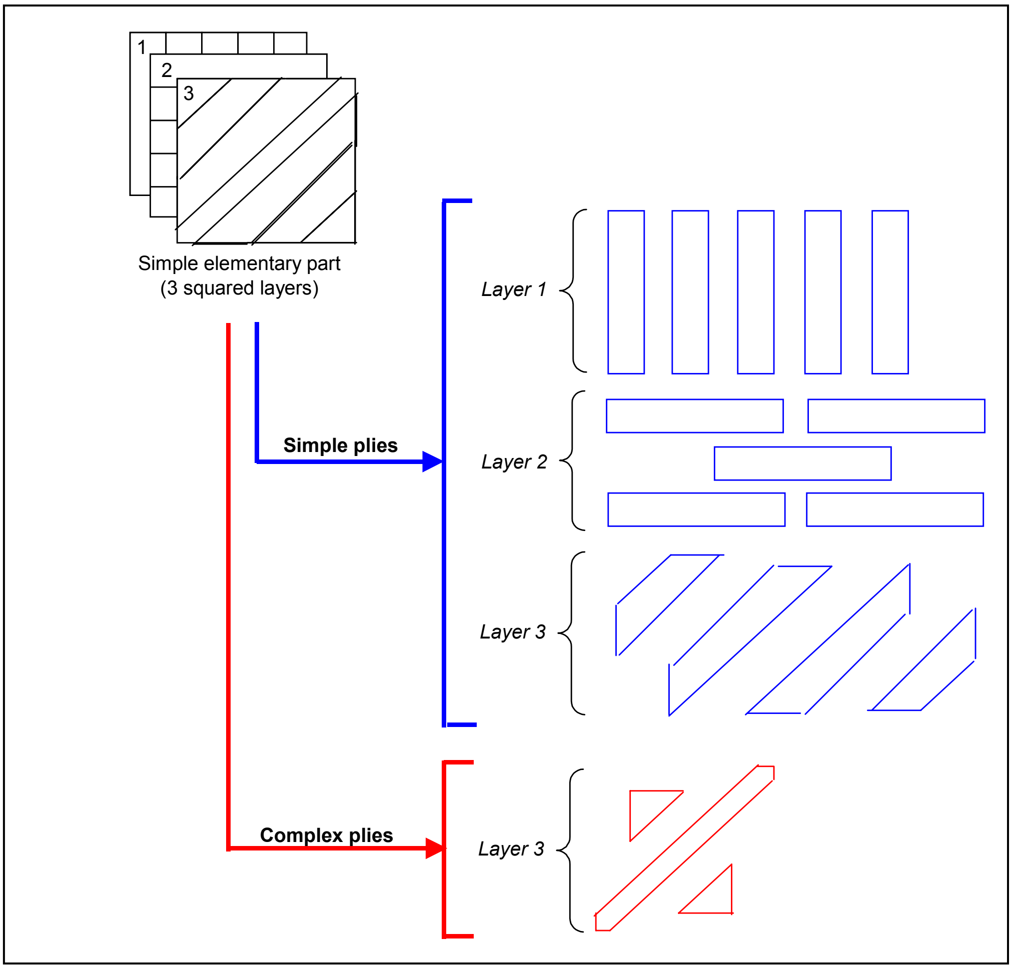

Figure 18: Simple vs. Complex plies

VITA

Nicolas Simonneau was born in France on May 7, 1980. He began engineering studies at the Ecole des Mines de Nantes in 1999 and decided to join the dual degree program running with Virginia Tech in order to specialize in Manufacturing and Operations Research techniques within the Industrial and Systems Engineering department. He obtained his French engineering degree in July 2003.

He spent six months as an intern in AIRBUS Nantes plant, where he worked on composite manufacturing scheduling problems. During this internship, he wrote his Master's thesis on a specific two stage hybrid flow shop problem with prescription constraints on jobs.

He is now working in France as a logistic engineer for HEULIEZ, subcontractor of the automobile industry.