



HAL
open science

Planning & Scheduling flow shop organizations with constraints – towards a decision engineering system for production planning & production activity control

Valérie Botta-Genoulaz

► **To cite this version:**

Valérie Botta-Genoulaz. Planning & Scheduling flow shop organizations with constraints – towards a decision engineering system for production planning & production activity control. Operations Research [math.OA]. Université Claude Bernard Lyon 1, 1996. English. NNT: . tel-03849864

HAL Id: tel-03849864

<https://hal.science/tel-03849864>

Submitted on 12 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée

devant l'UNIVERSITE CLAUDE BERNARD - LYON I

pour l'obtention

du **DIPLÔME DE DOCTORAT**

(arrêté du 30 mars 1992)

label européen

Spécialité : INGENIERIE INFORMATIQUE

par

Valérie BOTTA-GENOULAZ

**PLANIFICATION & ORDONNANCEMENT D'UNE
SUCCESSION D'ATELIERS AVEC CONTRAINTES**

-

***VERS UN ATELIER DE GENIE DECISIONNEL POUR LE PILOTAGE
DES SYSTEMES DE PRODUCTION***

soutenu le 16 décembre 1996 devant le jury composé de:

Président: Christian PROUST
Rapporteurs: Abdelhakim ARTIBA
François ROUBELLAT
Marino WIDMER
Examineurs: Jean Luc DANIEL
Alain DUSSAUCHOY
Alain GUINET
Henri VIALLO

REMERCIEMENTS

Le travail présenté dans ce mémoire a été effectué dans le cadre d'un contrat Région - Université - Entreprise, au laboratoire Productique et Informatique des Systèmes Manufacturiers de l'université Claude Bernard - Lyon I et de l'INSA de Lyon que dirige Monsieur le Professeur Joël Favrel. Je tiens à le remercier pour l'accueil qu'il m'a réservé au sein de son laboratoire.

J'adresse mes sincères remerciements à Monsieur Christian Proust, Professeur à l'E3I de Tours, qui a accepté de juger mes travaux et d'être président du jury.

Je tiens à remercier Monsieur Alain Dussauchoy, Professeur à l'Université Claude Bernard - Lyon I, pour sa participation à la commission d'examen. Je lui suis particulièrement reconnaissante de m'avoir donné le goût de la recherche lors de mes études et de m'avoir orienté, par ses conseils, lors de mon retour dans le monde universitaire.

Je suis profondément reconnaissante à Messieurs Jean Luc Daniel, Directeur industriel d'INOSETA, et Henri Viallon, Directeur Informatique de DMC pour avoir accepté de faire partie de ce jury. Ils m'ont permis de réaliser une interface fructueuse entre le monde professionnel et celui de la recherche, et offert le support expérimental de cette thèse. Je les remercie ainsi que l'ensemble de l'équipe d'INOSETA pour le temps qu'ils m'ont consacré.

Je remercie tout particulièrement Monsieur Abdelhakim Artiba, Professeur aux Facultés Universitaires Catholiques de Mons en Belgique, qui a montré un grand intérêt pour ce travail et a accepté de le rapporter. Qu'il trouve ici l'expression de ma profonde gratitude pour m'avoir accueilli dans son laboratoire de recherche, pour l'écoute patiente et le soutien qu'il m'a accordé et pour la confiance qu'il m'a témoignée.

J'adresse mes vifs remerciements à Messieurs François Roubellat, Directeur de Recherche au LAAS à Toulouse, et Marino Widmer, Professeur à l'Université de Fribourg en Suisse, pour avoir accepté d'être rapporteurs de cette thèse, pour le temps qu'ils m'ont consacré et pour les conseils et remarques qu'ils m'ont adressés.

Monsieur Alain Guinet, Professeur à l'INSA de Lyon, m'a fait confiance en acceptant d'encadrer mes travaux et je l'en remercie vivement. Il a parrainé mon entrée dans la communauté scientifique, et les conseils qu'il m'a prodigués durant ces trois années ont été très précieux.

Mes remerciements vont également à Mesdames Janel Boulogne et Vanessa Sherlock qui ont relu cette thèse et ont validé la rédaction en Anglais de ce mémoire.

Je remercie les membres de l'équipe de la FUCaM pour leurs encouragements et leur accueil lors de mes trois mois en Belgique. J'ai une pensée particulière pour Fouad qui m'a accompagné dans des moments difficiles. Une partie de cette thèse leur est dédiée dans le cadre du label Européen que j'ai pu obtenir grâce à leur compétence et notre travail commun.

Je remercie également tous les amis et collègues qui m'ont encouragée et plus particulièrement Hamid, Véronique, Claire, Viviane, Francine...

Enfin, une pensée particulière est adressée à ma famille pour le soutien moral et logistique qu'ils m'ont prodigués durant ces trois années.

Il me tient particulièrement à coeur de remercier mon mari Philippe pour sa patience, sa compréhension, et surtout pour sa présence à mes cotés dans les moments difficiles, ainsi que ma fille Caroline qui a su, à sa manière, contribuer à ma réussite.

Ph.D. THESIS

presented

at **UNIVERSITE CLAUDE BERNARD - LYON I**

to obtain

DIPLÔME DE DOCTORAT

(arrêté du 30 mars 1992)

label européen

by

Valérie BOTTA-GENOULAZ

**PLANNING & SCHEDULING FLOW SHOP
ORGANIZATIONS WITH CONSTRAINTS**

-

*TOWARDS A DECISION ENGINEERING SYSTEM FOR
PRODUCTION PLANNING & PRODUCTION ACTIVITY CONTROL*

defense: December 16, 1996

JURY: A. ARTIBA
J.L. DANIEL
A. DUSSAUCHOY
A. GUINET
C. PROUST
F. ROUBELLAT
H. VIALON
M. WIDMER

CONTENTS

1.	<u>INTRODUCTION</u>	2
1.1.	EVOLUTION OF THE INDUSTRIAL AND ECONOMIC ENVIRONMENT	2
1.2.	THE PLACE OF SCHEDULING IN AN ORGANIZATION	3
1.3.	SCHEDULING PROBLEMS IN PROCESS INDUSTRIES	5
1.4.	THESIS OUTLINE	8
2.	<u>HYBRID FLOW SHOP ORGANIZATIONS</u>	14
2.1.	DESCRIPTION OF HYBRID FLOW SHOPS	14
2.1.1.	DEFINITION	14
2.1.2.	GENERAL CHARACTERISTICS	15
2.1.3.	COMPLEMENTARY CHARACTERISTICS	16
2.2.	A CLASSIFICATION SCHEME FOR HYBRID SHOP PROBLEMS	19
2.2.1.	RESOURCE ENVIRONMENTS	19
2.2.2.	MEASURE OF SCHEDULE PERFORMANCE	21
2.2.3.	A CLASSIFICATION SCHEME FOR HYBRID SHOP SCHEDULING PROBLEMS	23
2.3.	PROBLEM FORMULATION	26
2.3.1.	GENERAL PROBLEM FORMULATION	27
2.3.1.1.	Parameters	27
2.3.1.2.	Variables	27
2.3.1.3.	Model	27
2.3.1.4.	Comments on problem complexity	28
2.3.2.	EXTENSION TO COMPLEMENTARY CHARACTERISTICS	29
2.3.2.1.	Complementary parameters	29
2.3.2.2.	Modeling of additional characteristics	30
2.4.	SUMMARY AND CONCLUSION	35
3.	<u>SCHEDULING PROBLEMS: A CONCISE SURVEY</u>	38
3.1.	TYOLOGY OF SOLUTION METHODS	39
3.2.	PARALLEL MACHINES	40
3.3.	FLOW SHOPS	43
3.4.	HYBRID FLOW SHOPS	45
3.4.1.	2-STAGE HYBRID FLOW SHOPS	45

3.4.2.	V-STAGE HYBRID FLOW SHOPS	47
3.4.3.	MAIN RESULTS	49
3.5.	SUMMARY AND CONCLUSION	50
<u>4.</u>	<u>A SUITABLE METHOD FOR PRODUCTION SYSTEM ANALYSIS AND MODELING</u>	<u>54</u>
4.1.	ANALYSIS AND DESIGN METHODOLOGIES: A SURVEY	54
4.1.1.	THE STRUCTURED APPROACH	54
4.1.2.	THE SYSTEMIC APPROACH	55
4.1.3.	THE OBJECT-ORIENTED APPROACH	55
4.1.4.	PRODUCTION ENGINEERING-ORIENTED METHODOLOGIES	56
4.2.	A NEW METHODOLOGY: OASISS	56
4.2.1.	TOP-DOWN ANALYSIS OF RESOURCES	58
4.2.1.1.	A resource approach	58
4.2.1.2.	A graphic language support	59
4.2.2.	BOTTOM-UP RECOGNITION OF RESOURCES	61
4.2.2.1.	Object classes	61
4.2.2.2.	The recognition process	63
4.2.3.	INSTANTIATING AND INTEGRATION OF OBJECTS	63
4.2.3.1.	Specification tools for object behavior	63
4.2.3.2.	Dynamic modeling	67
4.2.3.3.	Selection of scheduling tools	69
4.2.4.	IMPLEMENTATION OF OBJECTS	70
4.3.	SUMMARY AND CONCLUSION	70
<u>5.</u>	<u>APPLICATION IN THE TEXTILE INDUSTRY</u>	<u>74</u>
5.1.	INTRODUCTION	74
5.2.	DESCRIPTION & ANALYSIS OF THE PRODUCTION SYSTEM	76
5.2.1.	THE GREY MANUFACTURING PROCESS (P1)	76
5.2.1.1.	The weft twisting process	78
5.2.1.2.	The warping process	79
5.2.1.3.	The weaving process	80
5.2.1.4.	Grey manufacturing process characteristics and constraints	80
5.2.2.	THE FINISHING PROCESS (P2)	82
5.3.	IDENTIFICATION OF SCHEDULING PROBLEMS	83
5.4.	SIMULATION MODELS	86
5.5.	CONCLUSION	87

6.	<u>NEW DEVELOPMENTS IN HYBRID FLOW SHOP SCHEDULING</u>	90
6.1.	PRINCIPLES	90
6.2.	THE SEQUENCING STEP: PREVIOUS RESULTS	91
6.2.1.	$N/2, M_1=M_2=1/F/C_{MAX}$: JOHNSON [1954]	92
6.2.2.	$N/V, M_V=1/F/C_{MAX}$: CAMPBELL, DUDEK & SMITH [1970]	92
6.2.3.	$N/V, M_V=1/F, A_{1,V}/C_{MAX}$: SZWARC [1983]	92
6.2.4.	$N/V, M_V=1/F/C_{MAX}$: NAWAZ, ENSCORE & HAM [1983]	93
6.2.5.	$N/V, M_V=1/F/L_{MAX}$: TOWNSEND [1977]	93
6.3.	THE SEQUENCING STEP: NEW DEVELOPMENTS	94
6.3.1.	INTEGRATION OF VARIOUS WORKSHOP CALENDARS	94
6.3.2.	INTEGRATION OF NON-SEQUENCE DEPENDENT SETUP AND REMOVAL TIMES	95
6.3.3.	INTEGRATION OF JOB-PRECEDENCE CONSTRAINTS AND MINIMUM TIME LAGS	95
6.3.4.	FROM MAKESPAN TO MAXIMUM LATENESS	97
6.3.5.	A FAST HEURISTIC FOR NUMEROUS JOB-PRECEDENCE CONSTRAINTS: VBA	98
6.4.	THE ASSIGNMENT STEP	99
6.5.	SUMMARY: FIVE HEURISTICS	100
6.6.	LOWER BOUNDS AND ANOTHER HEURISTIC: VBB	103
6.7.	CONCLUSION	104
7.	<u>EXPERIMENTS & RESULTS</u>	108
7.1.	GENERAL CASE STUDY	108
7.1.1.	DATA	109
7.1.2.	RESULTS FOR C_{MAX} CRITERION	110
7.1.2.1.	Flow shop configurations	110
7.1.2.2.	Hybrid flow shop configurations	113
7.1.3.	RESULTS FOR L_{MAX} CRITERION	115
7.1.3.1.	Flow shop configurations	116
7.1.3.2.	Hybrid flow shop configurations	117
7.1.4.	CONCLUSIONS	121
7.2.	INOSETA'S CASE STUDY	122
7.2.1.	DATA	122
7.2.2.	THE WARPING / WEAVING STAGE (WWP)	124
7.2.3.	THE FINISHING STAGE (FP)	125
7.3.	CONCLUSIONS & RECOMMENDATIONS	127
8.	<u>CONCLUSIONS & PERSPECTIVES</u>	130
8.1.	CONCLUSIONS	130

8.2.	PERSPECTIVES	131
8.3.	TOWARDS A DECISION ENGINEERING SYSTEM FOR PRODUCTION PLANNING & PRODUCTION ACTIVITY CONTROL	131
9.	<u>BIBLIOGRAPHIC REFERENCES</u>	<u>133</u>
9.1.	BIBLIOGRAPHIC REFERENCES	134
9.2.	COMMERCIAL SOFTWARE REFERENCES	153
10.	<u>APPENDIX</u>	<u>155</u>
10.1.	N/2/F, J-PREC, OUT-TREE/C_{MAX}	156
10.2.	N/1//S_{NSD}, R_{NSD}/L_{MAX}	158
10.3.	ENGLISH-FRENCH GLOSSARY OF TEXTILE TERMS	160
10.4.	ABBREVIATIONS & NOTATIONS	161

FIGURES

<i>Figure 1-A: Information flow diagram in a manufacturing system (from [Pinedo; 1995]).</i>	6
<i>Figure 2-A: Process flow in a hybrid flow shop organization.</i>	15
<i>Figure 2-B: Job release date $r(j)$.</i>	17
<i>Figure 2-C: Job minimum time lag between two workshops v and $v+1$: $a(j,v)$.</i>	17
<i>Figure 2-D: Non-sequence dependent setup and removal times on any machine of workshop v: $S(j,v)$ and $R(j,v)$.</i>	18
<i>Figure 2-E: Job-precedence constraints (i precedes j, $i < j$).</i>	18
<i>Figure 2-F: In-tree, out-tree and general precedence constraints.</i>	19
<i>Figure 2-G: Typology of resource environments.</i>	21
<i>Figure 2-H: Typology of performance criteria.</i>	23
<i>Figure 4-A: Overall architecture of OASISS methodology.</i>	57
<i>Figure 4-B: IDEF0 function box and interface arrows.</i>	60
<i>Figure 4-C: IDEF0 model decomposition.</i>	60
<i>Figure 4-D: Specialization and aggregation tree.</i>	62
<i>Figure 4-E: Specification of simple object classes using Petri nets</i>	65
<i>Figure 4-F: Specification of simple object classes using ARENA.</i>	67
<i>Figure 4-G: Object unification with Petri nets.</i>	68
<i>Figure 4-H: Object unification with ARENA.</i>	69
<i>Figure 5-A : 3-level production management approach.</i>	75
<i>Figure 5-B: 2-level production management approach.</i>	76
<i>Figure 5-C: Grey manufacturing process (P1).</i>	77
<i>Figure 5-D: Resourcegram of the grey manufacturing process (P1).</i>	77
<i>Figure 5-E: Resourcegram of weft twisting process (P1.1).</i>	78
<i>Figure 5-F: Resourcegram of the warping process (P1.2).</i>	79
<i>Figure 5-G: Finishing manufacturing process (P2).</i>	82
<i>Figure 5-H: Specification of the warping process with Petri nets.</i>	86
<i>Figure 5-I: Specification of the warping process using ARENA.</i>	87
<i>Figure 6-A: Nsd setup and removal times changed into minimum time lags.</i>	95
<i>Figure 7-A: Configuration A: problems introducing negative time lags.</i>	113
<i>Figure 7-B: Configuration B: problems introducing time lags closed to zero.</i>	113
<i>Figure 7-C: Configuration C: problems introducing positive time lag.</i>	113
<i>Figure 8-A : Architecture of GEPETO.</i>	132

CHAPTER 1

INTRODUCTION

1.1. EVOLUTION OF THE INDUSTRIAL AND ECONOMIC ENVIRONMENT	2
1.2. THE PLACE OF SCHEDULING IN AN ORGANIZATION	3
1.3. SCHEDULING PROBLEMS IN PROCESS INDUSTRIES	5
1.4. THESIS OUTLINE	8

1. INTRODUCTION

In this introductory chapter, we define the framework of the present thesis, and present some requirements:

Section 1.1 shows the evolution of the current industrial and economical environment, and the importance of good production environment scheduling strategies in today's competitive markets. Section 1.2 introduces the reader to production planning and control, and situates the scheduling function in the decision-making hierarchy. Section 1.3 presents the application field of the scheduling problems under study, i.e. process industries, and introduces specific requirements that have to be taken into account when developing computer-integrated production planning and scheduling systems. Finally, section 1.4 presents the thesis outline.

1.1. EVOLUTION OF THE INDUSTRIAL AND ECONOMIC ENVIRONMENT

Competitive firms are operating today in global and worldwide markets. Productivity improvement is thus a major objective for these manufacturing companies. Manufacturers are currently experiencing fluctuating market demand for their products, with ever shorter lead times, smaller but more frequent order quantities, and more frequent changes in product specifications. Product life cycles are shorter, and forecasting of demand is becoming increasingly difficult. The customer loyalty to trademarks or the fashion attachment no longer exist. Customers are today looking for products or services that are easy-care and with a good quality/price ratio.

All this requires agility and flexibility on the part of factory, which runs counter to its built-in-inertia, its gravitational force to « conduct business as usual », and the limited ability of management to reconsider decisions almost « in real time » [Artiba & Elmaghraby; 1996].

To achieve the rationalization of manufacturing management with the desired attributes (agility, flexibility, and continuous improvement in productivity), many methods, tools, and techniques have emerged over the past two decades. Traditional Material Requirements Planning (MRP I) methods are recognized to be efficient for the handling of stock management and purchasing functions, but inadequate for the management of the manufacturing shop-floor. The limitation of MRP I gave rise to manufacturing resource

planning (MRP II), whose potential seemed unlimited. This approach did not come up to expectations however, and manufacturing firms are once again seeking to a better solution; it has become apparent that push-based approaches MRP (I or II) are based on assumptions that contradict one another [Stevenson; 1993]. For instance, the assumption that the market can be predicted contradicts the assumption that demand should be managed. On the other hand, just-in-time (JIT) and total quality control (TQC) are complementary philosophies which aim to gradually eliminate intermediate stock buffers between workstations, wasteful processes, and production disruptions. The assumption of stability of demand required by the JIT approach is in contradiction with the inherent variability of current market behavior.

Production planners are therefore looking for alternative support methodologies for the coordination of their activities. Production planning and control literature which addresses the areas of production management and scheduling problems, is diverse and prolific. The theory of scheduling has received much attention from practitioners, management scientists, production and operations research workers, and mathematicians since the early 1950s. Tools and techniques vary from single-machine sequencing algorithms to large-scale computer-integrated manufacturing systems. Despite this fact, the theoretical methods which have been developed are still far from being widely put into practice. MacCarthy and Liu [1993] offer two explanations:

- The theory and solution methods are unknown or not properly understood by the practitioners.
- The ideal situations assumed by the theory are not sufficiently close to those found in practice.

Several comparative studies highlight the importance of selecting the right technique for the right problem as there is no global best technique [Artiba & Elmaghraby; 1996].

1.2. THE PLACE OF SCHEDULING IN AN ORGANIZATION

After the previous section, we can conclude that improvements in relevant scheduling tools are desperately needed. The present thesis deals with production scheduling in the process industry, especially in flow shop organizations, and proposes scheduling tools to help production planners and material managers in their decision making. Before discussing production scheduling problems, we introduce the reader to production planning and control, and situate the scheduling function in the decision-making hierarchy.

Production is the process of transforming raw materials into pre-designed finished products. However, before beginning the actual production, several control activities have to be carried out. Bertrand & *al.* [1990] define the **production control** as « the coordination of supply and production activities in manufacturing systems to achieve a specific delivery reliability at minimum cost ». A production control system is the set of functions and activities responsible for production control. The **decision-making hierarchy** of a production control system is made up of all the decisions of the production control system and includes the decision order and the decision methods; the decision order specifies the order in which the decisions are made, and the decision method specifies the way in which each decision is made. Terms often used with regard to the decision-making hierarchy are production planning and production scheduling.

Production planning is the process of allocating production quantities over the next several planning periods and making the relevant decisions concerning the availability of resources, such as personnel or machines, and materials. In a decision-making hierarchy, we can distinguish several planning levels. The classical decision hierarchy of Anthony [1965] distinguishes three planning levels: a strategic level, a tactical level, and an operational level. The strategic level concerns decisions which, over the long term, determine the output of a company, such as investments in new plants or new production resources, and the selection of new product-market combinations. The tactical planning level deals with decisions connected to the use of production resources, for example the determination of target production levels, and seasonal inventory levels. The operational level is concerned with decisions about week-to-week or day-to-day production, for instance the allocation of production orders (which can consist of several customer orders) to short planning periods, the allocation of production orders to resources, and the sequencing of production orders. At this level, both planning and scheduling activities can be carried out. This thesis will mainly focus on operational level decisions.

Graves [1981] defines **production scheduling** as the allocation of available production resources to tasks over time in order to best satisfy a given set of criteria. In a production process, resources can represent workshops, machines, or equipment, and tasks may be production orders or operations. Each task may have a different priority level, earliest possible starting time, and due-date. It is a decision-making process that has as a goal the optimization of one or more objectives. Generally speaking, the objective of scheduling is to reach a compromise between conflicting goals, which include efficient utilization of staff,

equipment, and facilities, as well as minimization of customer waiting time, inventories, and process times.

In the decision-making hierarchy, scheduling decisions are the final step in the transformation process before actual output occurs. Many system design and operation decisions have to be made long before scheduling decisions such as system capacity, equipment selection, selection and training of workers, products design, and so on. Consequently, scheduling decisions must be made within the constraints imposed by many other decisions, making them fairly narrow in scope and latitude. Scheduling also receives input from shop floor control. Unexpected events such as machine breakdowns, processing times that are longer than anticipated, or late delivery of raw materials have to be taken into account as they may have a major impact on the schedules. Figure 1-A depicts a diagram of the information flow in a manufacturing system.

1.3. SCHEDULING PROBLEMS IN PROCESS INDUSTRIES

In this section, we present the field of application of the scheduling problems under study, i.e. the process industry, and introduce specific requirements that have to be taken into account when developing computer-integrated production planning and scheduling systems.

Process industries are businesses that add value to materials by mixing, separating, forming or by chemical reaction. Processes may be either continuous or batch, and usually require rigid process control, and high capital investment [Wallace; 1984]. The process industry includes companies from a wide range of different sectors: chemical, pharmaceutical, oil, food, tobacco, textile, paper, and metallurgical...

In recent years, under market pressure, the number of products in the process industry has been increased, order sizes have been reduced, and delivery times have been shortened. As a result, companies recognize that it is difficult for material managers to make good production schedules within a reasonable time delay, and it is practically impossible to appreciate the consequences of a planning oversight. Therefore, there is an increasing demand by these companies for improved scheduling tools.

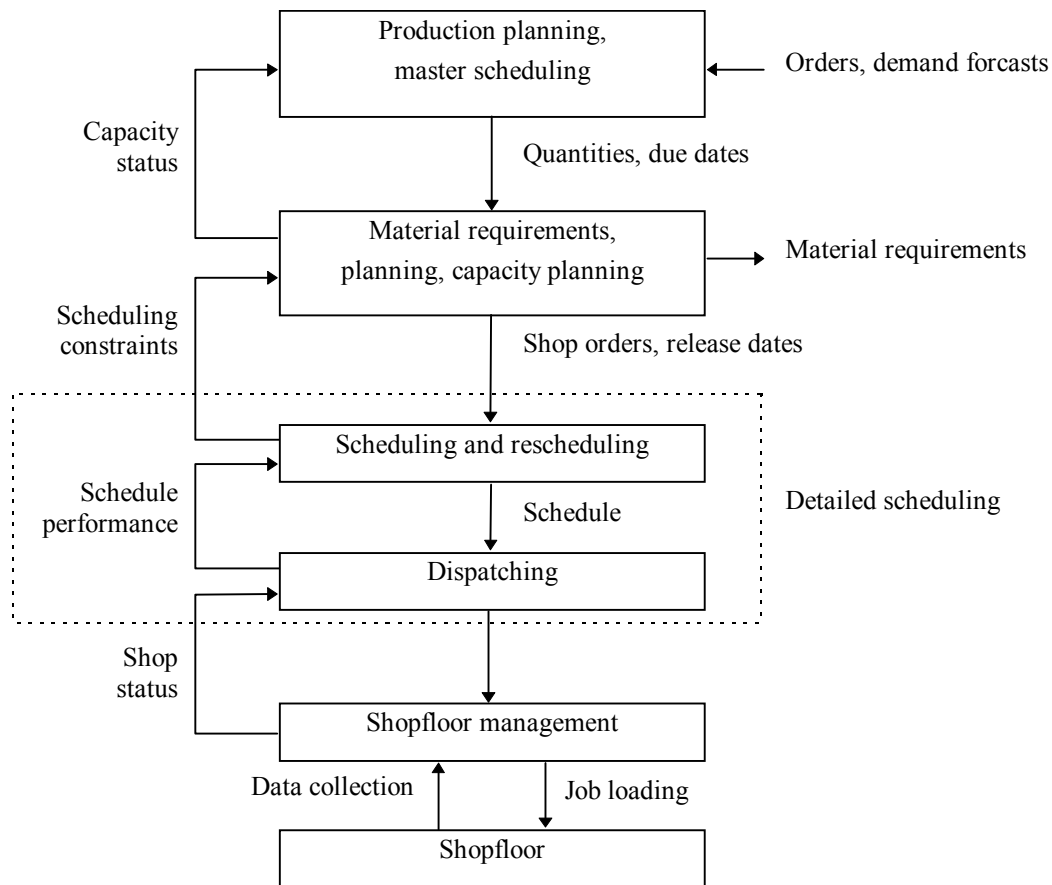


Figure 1-A: Information flow diagram in a manufacturing system (from [Pinedo; 1995]).

In the process industry, there are a variety of production systems. Production planning and control and, as a result, detailed scheduling are especially difficult and complex problems. Many research activities are concerned with general scheduling problems [Baker; 1974], [French; 1982], [GOTHA; 1993], [Pinedo; 1995], or with specific problems in process industries like the chemical industry [Artiba & al.; 1996-a], the electronics industry [Bitran & Tirupati; 1988], [Feo & al.; 1995], the textile industry [Serafini & Speranza; 1990], [Echalier; 1991], [Guinet; 1991], [Elmaghraby & Karnoub; 1995], the food industry [Harrison; 1996], the glass industry [He & al.; 1996], the packaging industry [Van Dam; 1995], [Tahmassebi & Hindi; 1996], and the metallurgical industry [He & Kusiak; 1992], [Hendry & al.; 1996]. Production activity control tools (planning functions, scheduling algorithms, resource assignment rules ...) are often dedicated to only one class of these systems [Bossink; 1992], [Carvalho & al., 1990], [Fransoo; 1996], [Slomp, 1993].

Stevenson [1993] states that the manner in which tasks are scheduled depends largely on the volume of the system output. High-volume systems can be characterized by standardized equipment and activities that provide identical or highly similar operations on

products as they pass through the system. Consequently the goal is to obtain a smooth rate of goods' flow through the system in order to get a high utilization of labor and equipment. These systems are often referred to as **flow systems**, and scheduling in these systems is referred to as **flow shop scheduling**. In low-volume systems, products are made to order, and orders usually differ considerably in terms of processing requirements, materials needed, processing time, and processing sequence and setups. Scheduling in these systems is referred to as **job shop scheduling**. High-volume systems require approaches that differ substantially from those required by low-volume systems.

In our work, we have considered scheduling for high-volume systems in the process industry. Because of the highly repetitive nature of these systems, many of the loading and sequencing decisions can be determined during the design of the system. In fact, few flow systems are completely devoted to a single product; most must handle a variety of sizes and models. Each change involves different inputs, such as different parts and materials and different processing requirements that must be scheduled. If a system is to operate smoothly, the flow of materials, and work activities must be coordinated. This requires scheduling of input, processing, and output, as well as scheduling purchases. Furthermore, it is important to avoid excessive buildup of inventory, while having sufficient materials in stock has for effect the reduction of the delivery time.

Again, each variation in size or model will tend to have quite different inventory requirements, so that additional scheduling efforts will be needed to avoid unmarketable inventories. Another source of scheduling requirements is system disruptions that result in a less than desired amount of outputs. These disruptions can be caused by equipment failure, material shortage, quality problems, accidents, or absence.

Allweyer & *al.* [1996] outline the following aspects which should be taken into consideration when scheduling in process industries:

- Complexity of the production structure: The difficulty of finding a good schedule depends on the complexity of production structure inside the plant and on the number of interdependencies with other plants.
- Changeover times and costs: The reduction of changeover times and costs is always an important scheduling activity. Changeovers include equipment setup, mounting, dismounting and, typical for the process industry, cleaning procedures which may be dependent on product sequencing.

- Campaign scheduling: One strategy to minimize changeover times is the creation of campaigns. A campaign consists of several different kind of jobs which are ordered according to a pre-defined sequence.
- Resource-dependent parameters and times: In many cases, it is possible to use alternative resources. Process parameters and scheduling restrictions may change when using a different resource, and the processing time of an operation may depend on the type of resource used.
- Changeable configuration of equipment: Sometimes, primary resources need to be equipped with additional technology before a given process can be executed. The feasibility of a schedule may therefore not only depend on the availability of main resources, but also on the availability of secondary equipment.
- Changeable unit connections: Units can be directly connected by pipelines, and existing pipelines determine the possible paths in a plant. Scheduling can have as an objective the reduction of the effort necessary for changing such connections (when their modification is possible).
- Unstable products: Unstable products can only be stored for a limited time; this is an important restriction for job scheduling and execution.
- Quality tests: Most production processes require a thorough and continuous quality control, i.e. by frequent quality tests. Sometimes, decisions about the further treatment of a material can only be made after a quality test has been carried out at a certain step. This may imply the rescheduling of other jobs, and it may also be necessary to schedule these quality tests, even when the necessary equipment is a bottleneck.
- Shared resources: Some plants use shared resources, which by their nature, have a maximum capacity. When this fact is not taken into account explicitly for scheduling, jobs may be delayed if the resources are used elsewhere. Considering shared resources increases scheduling complexity.

1.4. THESIS OUTLINE

We present here a study of flow shop organization scheduling of in process industries. We will illustrate our main results using an example of a textile company. According to the

definition given above, the textile industry is a process industry: it involves both batch and continuous processes, and value is added to raw materials mainly by mixing (weaving,...) and using chemical reactions (dyeing, ...).

As it has been emphasized in section 1.1, when we speak about scheduling, it is important to select the right technique for the right problem. The specification of tools for production activity control and scheduling requires the identification of the main characteristics of the production system in order to identify the problem(s). These tools generally only take into account a subset of the production system characteristics. The evaluation of these tools, which requires the modeling of the production system, often takes place after the implementation of the scheduling system.

In this thesis, we propose a new approach to identify the scheduling problem(s), and to design dynamic models for the evaluation and validation of scheduling decisions. The aim of this identification is to select pre-existing solution tools; and the dynamic model will enable us to evaluate these tools, and validate scheduling decisions. This approach associates discrete event simulation with classical scheduling theory.

Scheduling problems in the textile industry have been identified and modeled, taking into account technological constraints on jobs, resource constraints, and various scheduling objectives. We assume that there are a finite number of jobs to be scheduled and a single objective to be minimized at a time. The flow pattern is a hybrid flow shop: V serial workshops with $M(v)$ parallel machines. The following restrictions have been considered: release dates, job precedence constraints, time lags (transport time, waiting time), setup and removal times, limited intermediate storage. Criteria to be minimize can be either maximum completion time (or makespan) in the aim of optimizing the utilization of resources, or maximum lateness with respect to prescribed deadlines. This kind of problem is also encountered in other process industries such as chemical, food, paper or electronics industries.

The problem of scheduling hybrid flow shop organizations may be described by two characteristics: scheduling and resource assignment. In other industrial situations, lot-sizing can be considered. Given that the problem is NP-hard [Gupta; 1988] and because of the industrial computational requirements, we have developed a set of constructive methods to solve it. We have selected previous works according to their flexibility and their capability to solve pure flow shop problems minimizing such criteria, and we have adapted and improved

them to take into account both objectives and additional constraints. They have been tested on theoretical and industrial data.

This thesis is organized as follows:

- In Chapter 2 we define and characterize hybrid flow shop organizations, and propose a classification scheme for these problems. We present a mathematical formulation for the general problem, and also for some constrained problems that can be encountered in process industries. This formal model will be useful to explain and assess the solution approach. This chapter presents our contribution to modeling.
- Chapter 3 consists of a review of scheduling problems and solution tools. Several types of resource layouts are presented: parallel machine organizations, flow shop organizations and hybrid flow shop organizations, the latter being a generalization of the two previous organizations.
- Before introducing a practical application, we submit in Chapter 4 a resource-oriented methodology to analyze and model a production system and to propose a suitable production activity control system. This methodology has been designed according to the previous classification scheme. This framework enables the analyst to identify scheduling problem(s), to find appropriate scheduling tools for these problem(s), and helps him/her to design dynamic model of the production system, in the aim to validate the schedules obtained with previous algorithms. This chapter represents our contribution to analysis and design.
- Chapter 5 presents an application of this methodology to analyze a real problem from the textile industry, which leads to the identification and specification of scheduling problems under interest. The process of dynamic model design will be illustrated.
- In Chapter 6, we propose heuristic scheduling algorithms that have been developed and/or adapted to solve hybrid flow shop scheduling problems. Numerous characteristics have been taken into account: various workshop calendars, release dates, job-precedence constraints, minimum time lags, and non-sequence dependent setup and removal times. These algorithms can be used either for make-to-stock or make-to-order environments. In both cases, lower bounds on the optimal solution have been calculated to assess the quality of these algorithms. This chapter presents our contribution to scheduling.

- The quality and the robustness of these algorithms are illustrated in Chapter 7 by the results of numerous experiments on a large range of both theoretical and real data. Recommendations are given for several general cases, and also for the specific industrial case from the textile industry which we have studied.
- In Chapter 8, we conclude this work and present some perspectives in the field of Operational Research. We also propose to generalize our approach in the design of a decision engineering system for production planning and production activity control.

This thesis can be read in different ways according to the reader's point of view: The reader especially interested in theoretical issues concerning scheduling problems may read Chapters 2, 3, 6, and 7; people who are concerned with industrial applications may be more interested in Chapter 4, 5, and 7.

This work has been carried out as part of a collaborative project between the PRISMa laboratory (PProductique et Informatique des Systèmes Manufacturiers) and INOSETA company, with a research grant from the Région Rhône-Alpes [Botta; 1994-a, 1995-c, 1996].

CHAPTER 2

HYBRID FLOW SHOP ORGANIZATIONS

2.1. DESCRIPTION OF HYBRID FLOW SHOPS	14
2.1.1. Definition	14
2.1.2. General Characteristics	15
2.1.3. Complementary Characteristics	16
2.2. A CLASSIFICATION SCHEME FOR HYBRID SHOP PROBLEMS	19
2.2.1. Resource Environments	19
2.2.2. Measure of Schedule Performance	21
2.2.3. A Classification Scheme for Hybrid Shop Scheduling Problems	23
2.3. PROBLEM FORMULATION	26
2.3.1. General Problem Formulation	27
2.3.1.1. Parameters	27
2.3.1.2. Variables	27
2.3.1.3. Model	27
2.3.1.4. Comments on problem complexity	28
2.3.2. Extension to Complementary Characteristics	29
2.3.2.1. Complementary parameters	29
2.3.2.2. Modeling of additional characteristics	30
2.4. SUMMARY AND CONCLUSION	35

2. HYBRID FLOW SHOP ORGANIZATIONS

In section 2.1, we define and characterize hybrid flow shop organizations in general (§2.1.1 & 2.1.2), and introduce the reader to many restrictions or special features which may be encountered in process industries (§2.1.3).

Most of the time, scheduling tools are defined according to a classification of machine scheduling problems. This classification is a key, firstly, to judge the complexity of a problem and secondly, to find an existing tool. In section 2.2, after a brief review of the main resource environments and the principal scheduling objectives (§2.2.1 and §2.2.2 respectively), we propose a convenient classification scheme for hybrid shop problems in §2.2.3.

Section 2.3 deals with the mathematical formulation of hybrid flow shop scheduling problems. We present, in §2.3.1, a mixed integer linear programming formulation for the hybrid flow shop scheduling problem with makespan or maximum lateness criteria for the general case:

$$N/V, M(v) \geq I, R/F/C_{max} \text{ or } L_{max}.$$

Then, in §2.3.2, we show how complementary characteristics can be taken into account in the MILP modeling and in the calculation of job completion times.

Section 2.4 summarizes and concludes this chapter.

2.1. DESCRIPTION OF HYBRID FLOW SHOPS

In this section, we define the framework of hybrid flow shops. A definition of a hybrid flow shop is given in §2.1.1, and the general assumptions for the scheduling problem in such environments are exposed in §2.1.2. We then introduce the reader to many restrictions or special features that can be encountered in process industries. A number of restrictions will be illustrated with examples. In the literature, **hybrid flow shops** can be referred to as **flow shops with multiple processors** [Hunsucker & Shah; 1994], as **multi-stage systems** [Egbelu; 1991], or as **flexible flow shops** [Sriskandarajah & Sethi; 1989], [Pinedo; 1995].

2.1.1. DEFINITION

A hybrid flow shop consists of a series of production stages or workshops, each of which has several facilities in parallel [Elmaghraby & Karnoub; 1995]. Some stages may have

only one facility, but for the plant to be qualified as a hybrid flow shop, at least one stage must have several facilities. The flow of products in the plant is unidirectional. Each product is processed at only one facility in each stage and at one or more stages before it exits the plant. An illustration of a three-stage hybrid flow shop with three machines in the first stage, four machines in the second, and two machines in the third is given in Figure 2-A.

In Operational Research literature, the notion of hybrid flow shop emerged in the 70s, with the study of a two-stage problem with identical parallel machines per stage [Arthanary & Ramaswamy; 1971], [Shen & Chen; 1972] and a no-wait hybrid flow shop problem [Salvador; 1973]. But, as far as we know, the expression « hybrid flow shop » was first employed in 1987 [Narasimhan & Mangiameli; 1987]. We remark that authors who use the denomination « flexible flow shop » imply that the machines of each center are identical [Sriskandarajah & Sethi; 1989], [Pinedo; 1995]. More information about previous work in hybrid flow shop scheduling will be given in Chapter 3, section 3.4. Similar problems may be found in multi-project scheduling with resource constraints [Artigues & *al.*; 1996].

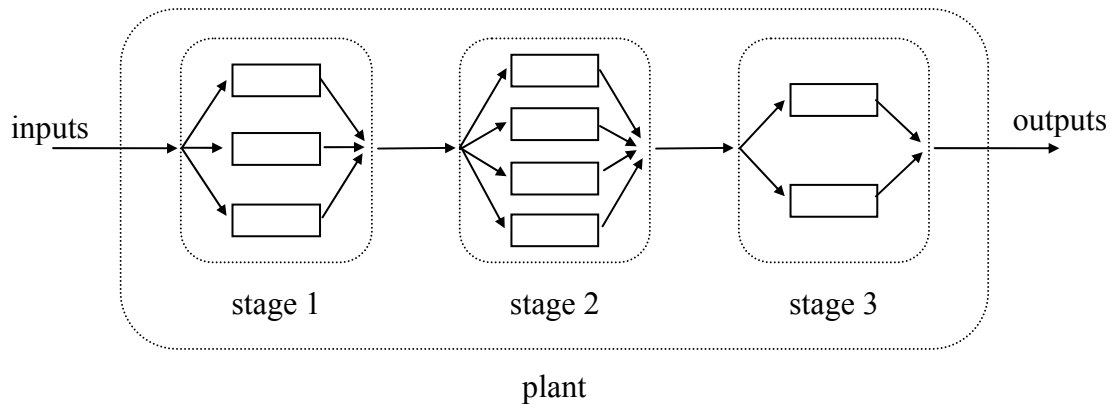


Figure 2-A: Process flow in a hybrid flow shop organization.

2.1.2. GENERAL CHARACTERISTICS

In scheduling flow shop organizations, the basic assumptions are as follows:

- All the jobs follow the same path through the V workshops: each job must be processed first in shop 1, then in shop 2, and so on.
- Each workshop v is composed of $M(v)$ parallel machines, with $M(v) \geq 1$. In a workshop, a job can be processed on any of the machines, but the processing time

required may be different from one machine to another. Machines in the same workshop can be either identical, uniform, or unrelated. Identical machines imply that the processing of a job will have exactly the same consequences (processing time, ...). Uniform machines are characterized by different speeds: the processing time of any job is proportional to the speed of the machines. In case of unrelated machines, the processing time of a job depends on both the job and the machine.

- Each machine can process a maximum of one operation at a time.
- The processing time of each job on a machine is known, fixed and independent of the order in which jobs are processed.
- Jobs are independent from one another.
- All jobs and machines are available simultaneously at the beginning of the scheduling horizon.
- Each job may have its own due date, $d(i)$, which represents the date the job is promised to the customer. The completion of a job after its due date is allowed, but a penalty is incurred.
- Preemption is not allowed: a job, once started on a machine, continues in processing until it is completed.
- Splitting of any particular job is not allowed.
- Transportation time between machines or workshops is negligible.
- Jobs are allowed to wait between two workshops. When this does not occur, we speak about a no-wait environment.
- Intermediate storage is unlimited, i.e. the buffer capacity between two consecutive workshops is unlimited.
- Machines never break down and setup times are negligible or included in processing times.

2.1.3. COMPLEMENTARY CHARACTERISTICS

Many restrictions to the previous description can be encountered in process industries. Industrial cases studies have enabled us to identify many of them, principally from textile,

carpet, and chemical industries. Some of these characteristics will be illustrated with Gantt diagrams, where f , i , and j represent job indices.

- In some cases, jobs may have different priority levels; we introduce the job weight, basically a priority factor, denoting the priority of one job relative to the others in the system. These weights may represent a cost function, such as the cost of keeping the job in the system, the inventory cost, or the amount of value already added to the job.
- Availability constraints may occur when jobs are subject to release dates. An illustration is given in Figure 2-B.

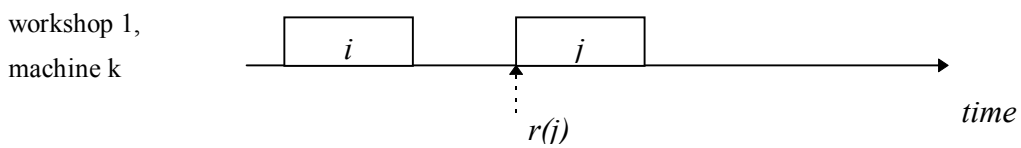


Figure 2-B: Job release date $r(j)$.

- Job processing may involve minimum time lags between two successive operations, such as transportation time, drying time, cooling time, etc. An illustration of minimum time lag is given in Figure 2-C.

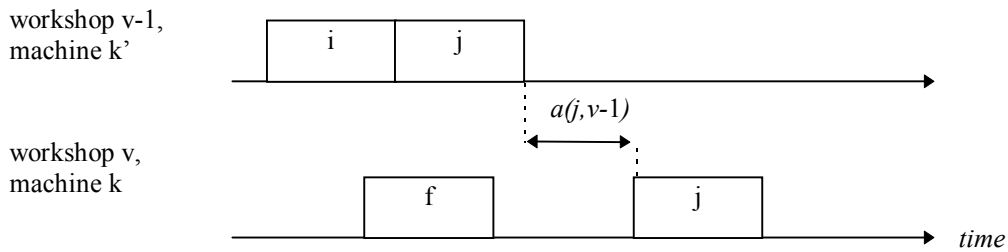


Figure 2-C: Job minimum time lag between two workshops v and $v+1$: $a(j, v)$.

- Job processing can also involve maximum time lags between two consecutive operations; for example in the textile industry, after certain dyeing treatments, a fabric must be dried within a ten hours.
- The processing of a job on a machine can require a setup (or removal) time which does or does not depend on the previous (or next) job processed on that machine. More often, these times can not be included in job processing times. For instance, a cleaning operation on a machine often involves a sequence-dependent setup time; the dismantling of a tool implies a non-sequence-dependent removal time. An illustration of non-sequence dependent setup and removal times is given in Figure 2-D. Moreover, we have encountered specific setup

times in the textile industry: the preliminary operation for the mounting of a loom is a setup operation which immobilizes only the warp, i.e. the job.

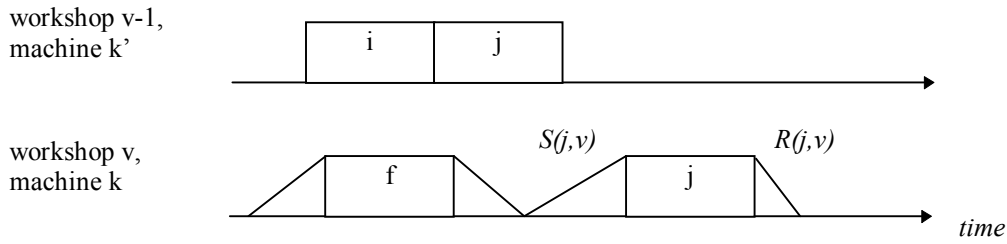


Figure 2-D: Non-sequence dependent setup and removal times on any machine of workshop v : $S(j,v)$ and $R(j,v)$.

- Limited buffer between two successive workshops may arise. This characteristic may be due to limited space between workshops or may result from strategic decisions.
- The processing of jobs may require auxiliary resources during one or more successive operations. When these auxiliary resources exist in a limited quantity, they have to be taken into account. Examples of these kinds of resource are pallets or beams in the textile industry and containers in the chemical industry.
- Workshops in the same plant may not have the same working calendar (the same number of working hours per day or week): there may be 3 daily shifts or only 2 or 1; some workshops work on the week-end, some do not... .
- Another restriction can be found: precedence constraints between jobs. This concept was first defined by Rinnooy Kan [1976], and will be defined as **job-precedence** constraints in opposition to **operation-precedence** constraints. This latter characteristic concerns precedence relations between the operations of two different jobs [Monma & Sidney; 1979]. With job-precedence constraints, the job i precedes the job j , noted « $i \prec j$ », meaning that the last operation of job i must be finished before the first operation of job j can begin. Job-precedence constraints are illustrated in Figure 2-E.

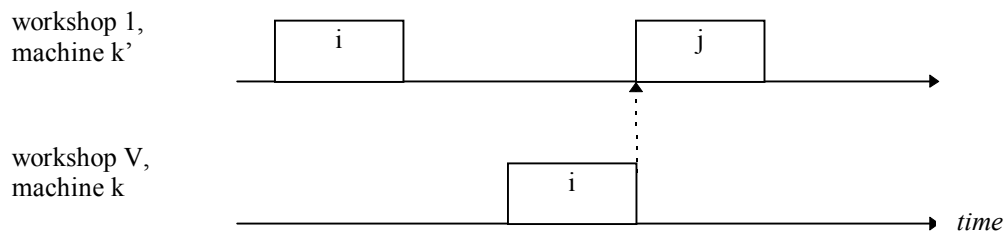


Figure 2-E: Job-precedence constraints (i precedes j , $i \prec j$).

This constraint can be used to take into account routings that are not completely linear in an organization. For example a job i whose the routing is « A-B-A-B » will be replaced by two jobs i_1 and i_2 whose routings are « A-B », linked by the constraint « $i_1 \ll i_2$ ». In the problems studied, we have encountered **tree-like** precedence constraints. When each job has one immediate successor at most, we speak of **in-tree** precedence constraints; when each job has one immediate predecessor at most, we speak of **out-tree** precedence constraints. The number of levels in tree-like precedence constraints represents the maximum number of jobs that can be linked in a series by precedence relations. Figure 2-F illustrates an in-tree precedence with three levels, an out-tree precedence with four levels and a more general case of precedence constraints. In-tree precedence constraints may model assembly operations, whereas out-tree precedence constraints can be used to model different processing units in a flow process, to transform a job shop into a flow shop [Guinet & Botta; 1995-b, -c], or as an alternative to job-splitting.

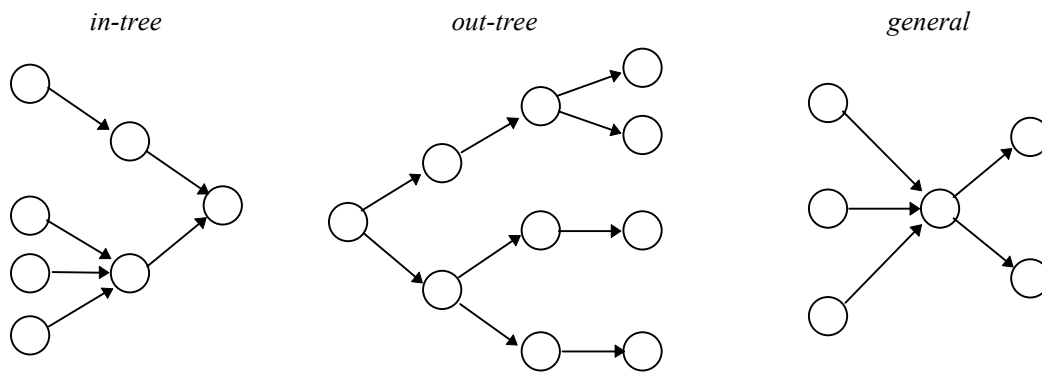


Figure 2-F: In-tree, out-tree and general precedence constraints.

2.2. A CLASSIFICATION SCHEME FOR HYBRID SHOP PROBLEMS

In defining a scheduling problem, both the technological constraints on jobs and the scheduling objectives must be specified. Technological constraints are determined by the flow pattern of the jobs on the machines (routings). After a brief presentation of the main resource environments and their relationships, we outline principal scheduling objectives. We then propose a convenient classification scheme for hybrid shop problems.

2.2.1. RESOURCE ENVIRONMENTS

Several flow patterns can be encountered, depending on the number of operations required to process a job and the number of available machines per operation. When a job

requires only one operation for its completion, we characterize it as **single-operation**; otherwise, we call it **multi-operation**. In the latter case, the concept of routing may be introduced.

- 1) Single machine shop: all jobs are single-operation and only one machine is available.
- 2) Flow shop (F): all jobs are multi-operation; there are V machines in series, and each job has to be processed on each machine in the same order (following the same routing).
- 3) Permutation flow shop (P): in such a flow shop the order of job processing is constrained to be the same for all machines; this characteristic is also referred as no-passing, and may be due to handling necessities.
- 4) Job shop (G): all jobs are multi-operation; there are V machines, but each job has its own specific routing through the V machines.
- 5) Open shop (O): jobs are multi-operation, there are V machines, but there is no specified flow pattern for any job.

When processing stages are considered instead of machines, the following definitions are useful.

- 6) Parallel machine shop: all jobs are single-operation, and several machines are available in a single stage to process the jobs. Machines can be either identical (I), uniform (U), or unrelated (R).
- 7) Hybrid flow shop: generalization of the flow shop and parallel machine environments; instead of V machines in series, there are V workshops in series with a number of machines $M(v)$ in parallel at each stage v ; each job must be processed on one machine at most in each workshop.
- 8) Job shop with duplicate machines (or flexible job shop): generalization of the job shop and parallel machine environments; instead of V machines, there are V workshops, with a number of machines $M(v)$ in parallel at each stage v ; each job must be processed on one machine at most in each workshop.

The diagram in Figure 2-G illustrates schematically the relationships between the different machine environments. On this diagram, an arrow from a environment P1 to an

environment P2 means that P2 is a special case of P1, or in other terms, that P1 is a generalization of P2. The arrow's label defines the restriction.

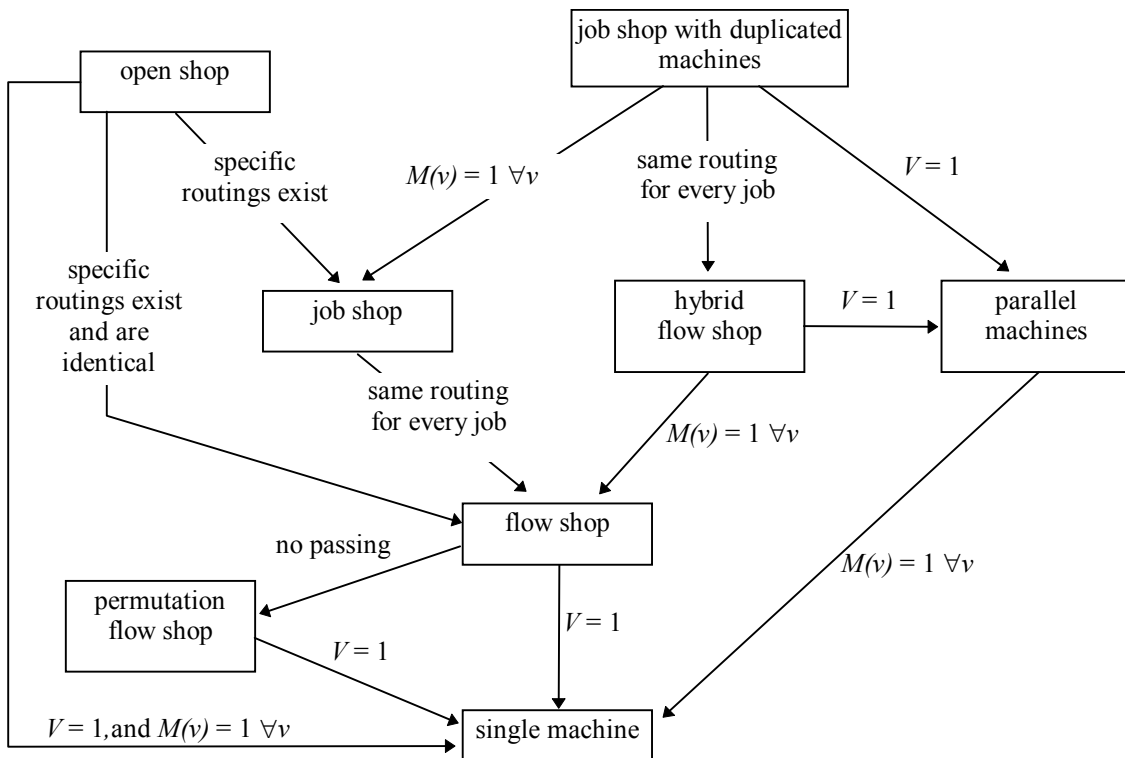


Figure 2-G: Typology of resource environments.

As we have seen, hybrid flow shops are a generalization of the flow shop and parallel machine environments. When the number of stages is equal to one, we have a parallel machine environment or a single machine environment ($M(v)=1$). When the machine number of each stage is equal to one, we have a flow shop environment. In a hybrid flow shop, all job routings are identical; we can also say that the hybrid flow shop is a particular case of a job shop with duplicate machines.

2.2.2. MEASURE OF SCHEDULE PERFORMANCE

Within any of these environments, a schedule may be made with respect to various objectives. Mellor [1966] lists 27 different objectives. Criteria can be based on completion times, on due dates, on inventory costs and utilization, or on a combination of these parameters. We will only present here the main or more frequently used criteria, after a definition of the required parameters.

If $C(i)$ is the completion time of the job i , $d(i)$ its due date, $r(i)$ its release date, and $p(i,k,v)$ its processing time on the machine k of workshop v , we can define the following measures:

$$\text{Flow time, } F(i) = C(i) - r(i)$$

$$\text{Waiting Time, } W(i) = C(i) - r(i) - \sum_{v=1}^V p(i, k_{i,v}, v)$$

$$\text{Lateness, } L(i) = C(i) - d(i)$$

$$\text{Tardiness, } T(i) = \text{Max}[0, L(i)]$$

$$\text{Earliness, } E(i) = -\text{Min}[0, L(i)]$$

$$\text{Tardiness unit penalty, } U_t(i) = 0 \quad \text{if } C(i) \leq d(i), \quad \text{and } U_t(i) = 1 \quad \text{otherwise}$$

$$\text{Earliness unit penalty, } U_e(i) = 0 \quad \text{if } C(i) \geq d(i), \quad \text{and } U_e(i) = 1 \quad \text{otherwise}$$

Baker [1974] notes three kinds of decision-making goals prevalent in scheduling, and indicates commonly used measures of performance that are associated with them:

- Efficient utilization of resources: maximum completion time or schedule length (makespan) C_{max} .
- Rapid response to demands, or minimization of work in process: mean completion time \bar{C} , mean flow time \bar{F} , or mean waiting time \bar{W} .
- Close conformity to prescribed deadlines: mean tardiness \bar{T} , maximum tardiness T_{max} , and the number of tardy jobs N_t .

In the current economic environment, we can add further criteria: maximum earliness E_{max} , mean earliness \bar{E} , maximum absolute lateness ET_{max} , mean absolute lateness \overline{ET} , or the number of early jobs N_e .

If $A(i)$ is a performance measure of job i , we define the sum ΣA , the mean \bar{A} and the maximum A_{max} as:

$$\Sigma A = \sum_{i=1}^N A(i), \quad \bar{A} = \frac{1}{N} \sum_{i=1}^N A(i) \quad \text{and} \quad A_{max} = \max_{i=1, \dots, N} [A(i)]$$

$$\bar{T} = \frac{1}{N_t} \sum_{i=1}^N T(i), \quad \bar{E} = \frac{1}{N_e} \sum_{i=1}^N E(i), \quad \overline{ET} = \frac{1}{N_t + N_e} \sum_{i=1}^N |L(i)|$$

with the exception:

$$N_t = \sum_{i=1}^N U_t(i), \quad N_e = \sum_{i=1}^N U_e(i)$$

One can also be interested in the study of weighted criteria. In that case, measure of performance is a function of job weight $\omega(i)$:

$$\sum A\omega = \sum_{i=1}^N \omega(i)A(i) \quad \text{and} \quad A\omega_{\max} = \max_{i=1,\dots,N} [\omega(i)A(i)]$$

Rinnooy Kan [1976] has examined the relations between criteria. If criterion C1 is **equivalent** to criterion C2 (denoted $C1 \Leftrightarrow C2$), a schedule that is optimal with respect to one of them is also optimal with respect to the other(s).

$$\begin{aligned} \sum C \Leftrightarrow \sum F \Leftrightarrow \sum W \Leftrightarrow \sum L \Leftrightarrow \bar{C} \Leftrightarrow \bar{F} \Leftrightarrow \bar{W} \Leftrightarrow \bar{L} \\ \bar{T} \Leftrightarrow \sum T \\ \sum C\omega \Leftrightarrow \sum F\omega \Leftrightarrow \sum W\omega \Leftrightarrow \sum L\omega \end{aligned}$$

A schedule that is optimal with respect to L_{\max} is also optimal with respect to T_{\max} .

The diagram in Figure 2-H illustrates the relationships between the different criteria in term of specialization / generalization schematically. Arrows have the same meaning as in the diagram presented in Figure 2-G.

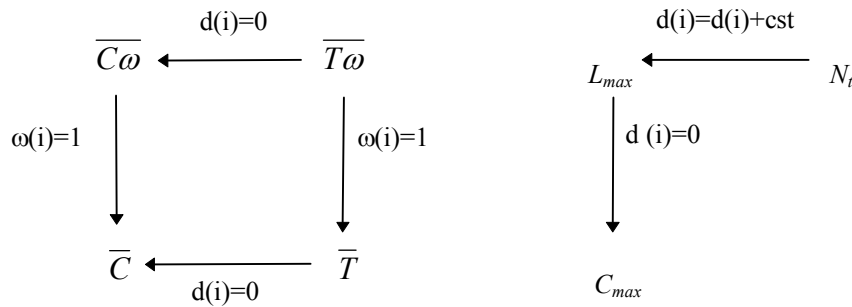


Figure 2-H: Typology of performance criteria.

2.2.3. A CLASSIFICATION SCHEME FOR HYBRID SHOP SCHEDULING PROBLEMS

Most of the time, scheduling tools or algorithms are defined according to the classification of machine scheduling problems. This is a key, firstly, to judge the complexity

of a problem, and secondly, to find an existing tool. In the literature, we have noticed that many authors use their own notation scheme. After an overview of the different classification schemes encountered in the literature, we propose a new notation system for scheduling problems which takes into account the « hybrid » dimension.

The first classification was proposed by Conway, Maxwell and Miller [1967], and was extended by Lenstra [1976]. It consists of five parameters:

$$A / B / C, D / E$$

where A represents the number of jobs, B the number of resources, C the type of resource organization (I, U, or R for parallel machines, or F, P, G, or O in case of multi-operations jobs, as described in §2.2.1), D the possible assumptions or restrictions of the organization (see §2.1.3), and E the optimality criterion to minimize (see §2.2.2). This notation scheme has been widely used for a long time and is familiar to most manufacturing and scheduling researchers. In this classification, a resource represents a machine, and consequently it can not be used as it stands to characterize hybrid problems.

Another classification system was proposed by Graham & al. [1979] and extended by Lawler & al. [1989]. It is based on three fields: $\alpha/\beta/\gamma$, where α represents the machine environment, β the possible restrictions, and γ the optimality criteria. The first field α is equal to $\alpha_1\alpha_2$, with α_1 set to ‘o’ for single machine, ‘P’ for identical, ‘Q’ for uniform, ‘R’ for unrelated parallel machines, ‘F’ for flow shop, ‘J’ for job shop, or ‘O’ for open shop, and α_2 set to the number of machines. As in precedent example, this classification system is not suitable for hybrid organizations. Sriskandarajah & Sethi [1989] propose to extend Graham & Lawler’s notation scheme by changing the meaning of the α and β fields: $\alpha = Fk$ for flow shop environment with k machine centers, and $\beta = \{m_1, m_2, \dots, m_k\}$ the number of machines at each center. In this system, additional constraints are not taken into account.

Vignier & al. [1995] are also inspired by Lawler’s scheme (previously expanded by Blazewicz & al. [1994-a]), and give more information in the first field α ; β and γ keep initial meanings.

$$\alpha = [\alpha_1\alpha_2], \left\{ \left(\alpha_3 \alpha_4^{(l)} \right) \right\}_{l=1}^{\alpha_2},$$

with $\alpha_1 = FH$ (hybrid flow shop), $\alpha_2 =$ workshop number, $\alpha_3 = \emptyset$ (single machine), P, Q, R, and $\alpha_4 =$ machine number in workshop l . This notation scheme uses a triple index: one upper

index (l) for workshops, and two lower indexes, i and j , for jobs and machines respectively. These triple indexes are used in both the α and β fields; it makes the notation laborious to read, and difficult to understand quickly.

For clarity and understanding, it seems important that authors adopt a standard, user-friendly notation scheme, based on an existing classification system, which is capable of describing all the scheduling problems studied. Therefore, we propose an extension of the classification of Conway & *al.* in order to show the « hybrid » dimension, and the introduction of an additional parameter B' :

$$A / B, B' / C, D / E.$$

B represents the number of workshops, and B' the machine layout in the workshops, i.e. the number of machines in each workshop and their characteristics (I, U, or R). C is now taken to indicate the workshop layout ('F' for flow shop, 'G' for job shop or 'O' for open shop). The other parameters A , D and E keep the same definition as in Conway's classification scheme. For further information see [Artiba & *al.*; 1997].

The following possible organization assumptions or restrictions will be noted:

- $r(i) \geq 0$: job release dates exist,
- *job-spl*: job splitting is allowed,
- *pmt*: job preemption is allowed,
- $a_{i,v}$: minimum time lag,
- $A_{i,v}$: maximum time lag,
- S_{sd}, S_{nsd} : sequence-dependent or non-sequence-dependent job setup times,
- R_{nsd} : non-sequence-dependent job removal times,
- $b_{v,v+1}$: limited buffer storage capacity between workshops v and $v+1$,
- $B_{v,v+\beta}$: limited number of auxiliary resources usable during $\beta+1$ successive operations of a job,
- *no-wait*: no waiting time is allowed between two successive operations of a job,
- H_v : existence of various workshop calendars on the scheduling horizon (for each workshop, the available time is constant for each calendar period and each machine),

- *j-prec*: general job-precedence constraints⁽¹⁾,
- *o-prec*: general operation-precedence constraints,
- *in-tree*: in-tree precedence constraints,
- *out-tree*: out-tree precedence constraints.

This notation is illustrated by the four following examples:

- $N/1, m(1)=1/Nt$ means the scheduling of N jobs on a single machine while minimizing the number of tardy jobs;
- $N/1, m(1)>1, U/pmt/C_{max}$ means the scheduling of N jobs on $m(1)$ uniform parallel machines while minimizing the makespan, when preemption is allowed;
- $N/V, m(v)=1/J/C_{max}$ means the scheduling of N jobs on a V -machines job shop while minimizing the makespan;
- $N/V, m(v)\geq 1, I/F, r(i)\geq 0/L_{max}$ means the scheduling of N jobs subject to release dates on a V -stage hybrid flow shop with identical parallel machines while minimizing the maximum lateness.

2.3. PROBLEM FORMULATION

In this section, we propose a mathematical formulation for the general hybrid flow shop scheduling problem and show how complementary characteristics can be taken into account.

A mixed integer linear programming (MILP) formulation for the general hybrid flow shop scheduling problem is given in §2.3.1, with either makespan or maximum lateness criteria, which are currently the most commonly used:

$$N / V, M(v)\geq 1, R / F / C_{max}, \text{ or } L_{max}$$

The §2.3.2 presents how the restrictions or complementary characteristics defined in §2.1.2 can be taken into account in the MILP formulation, and/or in the calculation of performance criteria.

⁽¹⁾ job-precedence constraints are only considered when jobs are multi-operations; otherwise, these constraints are equivalent to operation-precedence constraints.

2.3.1. GENERAL PROBLEM FORMULATION

Before presenting the model, we define the parameters and variables required. We then comment briefly on model complexity.

2.3.1.1. Parameters

N	number of jobs;
V	number of workshops;
$d(i)$	due date of job i ;
$M(v)$	number of machines in the workshop v ;
$p(i,k,v)$	processing time of the v^{th} operation of job i processed on machine k in workshop v ;
HV	a scalar chosen to be larger than the maximum completion time of any feasible solution of an active schedule.

2.3.1.2. Variables

$x(i,h,k,v)$	bivalent variable equal to 1 if operation v of job i is processed in rank h on machine k in workshop v , and to 0 otherwise;
$c(i,v)$	completion time of job i in workshop v ;
C_{max}	maximum of job completion time;
L_{max}	maximum of the job lateness.

Note that indexes i and j correspond to jobs, that r and h represent scheduling ranks, and that machines are indexed by k and workshops by v .

2.3.1.3. Model

$$\text{MINIMIZE}(Z) = C_{\max}, \text{ or } L_{\max} \quad (2.1)$$

subject to:

$$\sum_{k=1}^{M(v)} \sum_{h=1}^N x(i, h, k, v) = 1 \quad \forall i = 1..N, \quad \forall v = 1..V \quad (2.2)$$

$$\sum_{i=1}^N x(i, h, k, v) \leq 1 \quad \forall h = 1..N, \quad \forall k = 1..M(v), \quad \forall v = 1..V \quad (2.3)$$

$$c(i, v) \geq \sum_{r=1}^{h-1} \sum_{j=1}^N [x(j, r, k, v) \times p(j, k, v)] + p(i, k, v) + [x(i, h, k, v) - 1] \times HV \quad (2.4)$$

$$\forall i, h = 1..N, \quad \forall k = 1..M(v), \quad \forall v = 1..V$$

$$c(i, v) \geq c(i, v - 1) + \sum_{k=1}^{M(v)} \sum_{h=1}^N x(i, h, k, v) \times p(i, k, v) \quad \forall i = 1..N, \quad \forall v = 2..V \quad (2.5)$$

$$C_{\max} \geq c(i, V) \quad \forall i = 1..N \quad (2.6)$$

$$L_{\max} \geq c(i, V) - d(i) \quad \forall i = 1..N \quad (2.7)$$

$$x(i, h, k, v) \in \{0, 1\} \quad \forall i, h = 1..N, \quad \forall k = 1..M(v), \quad \forall v = 1..V \quad (2.8)$$

$$c(i, v) \geq 0 \quad \forall i = 1..N, \quad \forall v = 1..V \quad (2.9)$$

The objective function (2.1) is a statement of the minimization criteria that the schedule seeks to satisfy. Constraints (2.2) ensure that each job is processed once and once only in each workshop. Constraints (2.3) specify that each resource must be assigned to one job at most. A resource represents a possible machine loading in a workshop; it is defined by workshop and machine index and by a loading rank. Constraints (2.4) allow job completion times to be calculated; they depend on job processing times and on the order of jobs assigned to the machine which is indicated by the rank. Constraints (2.5) specify that each job must be processed in series in the workshops. Constraints (2.6), and (2.7) allow us to evaluate the minimization criteria. Finally, constraints (2.8) and (2.9) indicate that variables $x(i, h, k, v)$ are binary, while variables $c(i, v)$ are linear.

2.3.1.4. Comments on problem complexity

This model illustrates that scheduling non-preemptive jobs in serial workshops with parallel machines may be described as an assignment problem (constraints 2.2 and 2.3), and a sequencing problem with operation-precedence constraints (constraints 2.4 and 2.5). These two aspects will be exploited by our solution methodologies.

In regards to makespan criterion, for V equal to 1 and $M(v)$ greater than 1, this problem is NP-complete [French; 1982], [Cheng & Sin; 1990]. Therefore, for V and $M(v)$ greater than 1, the problem is NP-hard [Gupta; 1988].

In regards to maximum lateness criterion, for V equal to 2 and $M(v)$ equal to 1, the problem is NP-complete [MacCarthy & Liu; 1993]. Therefore, for V and $M(v)$ greater than 1, the problem is at least NP-complete.

2.3.2. EXTENSION TO COMPLEMENTARY CHARACTERISTICS

With no additional characteristics, the problem above is already too complex to solve. We will therefore work on a relaxed problem, and relaxed constraints will be integrated into the calculation of job completion times, and consequently into the calculation of the objective function. However, some characteristics will be modeled explicitly in the MILP formulation. First of all, we will define further parameters.

2.3.2.1. Complementary parameters

- $r(i)$ *release date* of job i ;
- $a(i,v)$ *minimum job time lag*: minimum time between the end of job i in workshop v and its beginning in workshop $v+1$;
- $A(i,v)$ *maximum job time lag*: maximum time between the end of job i in workshop v and its beginning in workshop $v+1$;
- $S_{nsd}(i,k,v)$ *nsd⁽¹⁾ machine setup time* on machine k in workshop v , before the beginning of job i ; this setup time is not sequence dependent, and immobilizes only the machine;
- $S_{sd}(i,j,k,v)$ *sd⁽²⁾ machine setup time* on machine k in workshop v , before the beginning of job j , when job i is processed just before job j on the same machine; this setup time is sequence dependent and immobilizes only the machine;
- $SI_{sd}(i,j,k,v)$ *sd job setup time* of job j , when job i is processed just before job j on machine k in workshop v ; this setup time is sequence dependent and immobilizes only the job;
- $S2_{sd}(i,j,k,v)$ *sd machine & job setup time* of job j on machine k in workshop v , when job i is processed just before job j on the same machine; this setup time is sequence dependent and immobilizes both the machine and the job;

⁽¹⁾ nsd: non-sequence dependent

- $R_{nsd}(i,k,v)$ *nsd machine removal time* on machine k in workshop v , after the end of job i ; this time is not sequence dependent and immobilizes only the machine, after the processing of the job;
- $b(v,v+1)$ *limited buffer storage capacity* between the output of workshop v and the input of workshop $v+1$, i.e. the number of jobs that can be stored between the two workshops, based on the fact that each job requires the same space in the buffer;
- $B(v,v+\beta)$ *limited number of auxiliary resources* available between the input of workshop v and the output of workshop $v+\beta$, based on the fact that each job requires one and only one auxiliary resource during the $\beta+1$ operations;
- $H(v)$ *number of working hours* per period on the scheduling horizon of all the machines in workshop v ;
- $prec(i)$ *job predecessor* of job i ; we are in the case of out-tree precedence constraints.

If the machines are identical in each workshop, all previous expressions are written without the index k .

2.3.2.2. Modeling of additional characteristics

We will first show how to turn MILP constraints into completion time calculation by transforming constraints (2.4) and (2.5):

$$c(j, k, v) = \text{MAX} \left[\begin{array}{l} c(j, k', v-1), \\ c(i, k, v) \end{array} \right] + p(j, k, v)$$

where i is the job processed just before j on machine k in workshop v .

In the following, we consider a hybrid flow shop environment with identical parallel machines. The MILP constraints and job completion time formulae are given for each new characteristic. Some of MILP constraints are adapted from previous ones, others are new. Job processing times which depend only on jobs and workshops, will be noted $p(i,v)$. We introduce $c(i,k,v)$ the completion time of job i on machine k in workshop v .

⁽²⁾ sd: sequence dependent

Various calendars:

This constraint can easily be taken into account by changing the processing time of each job in each workshop, as follows:

$$p'(i,v) = p(i,v) \times \frac{\text{MAX}_{v=1..V}[H(v)]}{H(v)} \quad (2.10)$$

Doing so is, of course, more restrictive than the reality (in reality, a job could be finished more quickly); but this complementary hypothesis will ensure the feasibility of the schedule obtained when faced with this constraint.

Release dates:

This characteristic concerns only the first workshop. The completion time of a job depends on its release date and on machine availability.

MILP additional constraints:

$$c(j,1) \geq r(j) + p(j,1) \quad \forall j = 1..N \quad (2.11)$$

The job completion time calculation, from constraints (2.4) and (2.11), where i is the job processed just before j on machine k in workshop v is:

$$c(j,k,1) = \text{MAX} \left[\begin{array}{l} r(j), \\ c(i,k,1) \end{array} \right] + p(j,1) \quad (2.12)$$

Precedence constraints:

If $prec(j)$ is the job predecessor of job j , it must end in workshop V before job j begins in workshop 1.

MILP additional constraints:

$$c(j,1) \geq c(prec(j),V) + p(j,1) \quad \forall j = 1..N \quad (2.13)$$

The job completion time calculation, from constraints (2.4) and (2.13), where i is the job processed just before j on machine k in workshop v is:

$$c(j,k,1) = \text{MAX} \left[\begin{array}{l} c(i,k,1), \\ c(prec(j),k',V) \end{array} \right] + p(j,1) \quad (2.14)$$

Minimum time lags:

The completion time of a job depends on the machine availability, the time of completion of its latter operation, and the time lag.

The MILP constraints (2.5) become:

$$c(j, v) \geq c(j, v-1) + a(j, v-1) + p(j, v) \quad \forall j = 1..N, \forall v = 2..V \quad (2.15)$$

The job completion time calculation, from constraints (2.4) and (2.15), where i is the job processed just before j on machine k in workshop v is:

$$c(j, k, v) = \text{MAX} \left[\begin{array}{l} c(i, k, v), \\ c(j, k', v-1) + a(j, v-1) \end{array} \right] + p(j, v) \quad (2.16)$$

Maximum time lags:

The completion time of a job also depends on the completion time of job l on the next workshop, when l is the first job, among all jobs in last machine position, completed in the next workshop.

The MILP constraints (2.5) become:

$$c(j, v) \geq c(j, v+1) - p(j, v+1) - A(j, v) \quad \forall j = 1..N, \forall v = 1..V-1 \quad (2.17)$$

$$c(j, v+1) \geq c(j, v) + p(j, v+1) \quad \forall j = 1..N, \forall v = 1..V-1 \quad (2.18)$$

Job completion time calculation:

If we define l as the earliest job completed in workshop $v+1$ which allows the processing of j , such as $c(l, v+1) = c(j, v+1) - p(j, v+1)$. Then, constraints (2.17) can be rewritten as:

$$c(j, v) \geq c(l, v+1) - A(j, v) \quad \forall j = 1..N, \forall v = 1..V-1 \quad (2.19)$$

From constraints (2.4), (2.18) and (2.19), we can calculate the job completion time, with l being the job previously defined and i , the job processed just before j on machine k in workshop v :

$$c(j, k, v) = \text{MAX} \left[\begin{array}{l} c(i, k, v), \\ c(j, k', v-1), \\ c(l, k'', v+1) - A(j, v) - p(j, v) \end{array} \right] + p(j, v) \quad (2.20)$$

Non sequence dependent machine setup and removal times:

The completion time of a job depends on the time of completion of its latter operation, its own setup time and the machine availability increased by the removal time of the job that precedes it on the machine.

The MILP constraints (2.4) become:

$$c(j, v) \geq \sum_{r=1}^{h-1} \sum_{i=1}^N x(i, r, k, v) \times [S_{nsd}(i, v) + p(i, v) + R_{nsd}(i, v)] + S_{nsd}(j, v) + p(j, v) \quad (2.21)$$

$$+ [x(j, h, k, v) - 1] \times HV \quad \forall j, h = 1..N, \forall k = 1..M(v), \forall v = 1..V$$

The job completion time calculation, from constraints (2.5), and (2.21), where i is the job processed just before j on machine k in workshop v is:

$$c(j, k, v) = MAX \left[\begin{array}{l} c(i, k, v) + R_{nsd}(i, v), \\ c(j, k', v-1) \end{array} \right] + S_{nsd}(j, v) + p(j, v) \quad (2.22)$$

Sequence dependent job setup times:

The completion time of a job depends on the completion time of its previous operation increased by its setup time, and the machine availability.

The MILP constraints (2.5) become:

$$c(j, v) \geq c(j, v-1) + \sum_{i=1}^N [x(i, h-1, k, v) \times S1_{sd}(i, j, v)] + p(j, v) \quad (2.23)$$

$$+ [x(j, h, k, v) - 1] \times HV \quad \forall j = 1..N, \forall v = 2..V$$

The job completion time calculation, from constraints (2.4) and (2.23), where i is the job processed just before j on machine k in workshop v is:

$$c(j, k, v) = MAX \left[\begin{array}{l} c(i, k, v), \\ c(j, k', v-1) + S1_{sd}(i, j, v) \end{array} \right] + p(j, v) \quad (2.24)$$

In order to treat the four other characteristics: S_{sd} , $S2_{sd}$, $b(v, v+1)$ and $B(v, v+\beta)$, the MILP formulation forces us to introduce complementary decision variables; for instance, Srikar & Ghosh [1986] introduced $\delta_{i,j}$, which is equal to 1 if job i is scheduled anytime before job j , to model the hybrid flow shop with S_{sd} . Nevertheless, we present their expression in terms of the calculation of job completion times.

Sequence dependent machine setup times:

The completion time of a job depends on the completion time of its previous operation, and the machine availability increased by the job setup time. If i is the job processed just before j on machine k in workshop v , the completion date of j on machine k in workshop v is:

$$c(j, k, v) = MAX \left[\begin{array}{l} c(i, k, v) + S_{sd}(i, j, v), \\ c(j, k', v-1) \end{array} \right] + p(j, v) \quad (2.25)$$

Sequence dependent machine & job setup times:

The completion time of a job depends on the completion time of its previous operation, its setup time, and the machine availability. If i is the job processed just before j on machine k in workshop v , the completion date of j on machine k in workshop v is:

$$c(j, k, v) = MAX \left[\begin{array}{l} c(i, k, v), \\ c(j, k', v-1) \end{array} \right] + S2_{sd}(i, j, v) + p(j, v) \quad (2.26)$$

Limited buffer storage capacity:

If we sort the jobs in descending order of their starting time in next workshop, the completion time of a job depends on the completion time of its previous operation, on the machine availability, and the beginning date of the job l in the next workshop, where l is the $b(v, v+1)$ th job. If i is the job processed just before j on machine k in workshop v , the completion date of j on machine k in workshop v is:

$$c(j, k, v) = MAX \left[\begin{array}{l} c(i, k, v), \\ c(j, k', v-1), \\ c(l, k'', v+1) - p(l, v+1) - p(j, v) \end{array} \right] + p(j, v) \quad (2.27)$$

Limited number of auxiliary resources:

If we sort the jobs in descending order of their completion date in workshop $v+\beta$, the completion time of a job in workshop v depends on the completion time of its previous operation, the machine availability, and the completion date of the job l on the workshop $v+\beta$, where l is the $B(v, v+\beta)$ th job. If i is the job processed just before j on machine k in workshop v , the completion date of j on machine k in workshop v is:

$$c(j, k, v) = MAX \left[\begin{array}{l} c(i, k, v), \\ c(j, k', v-1), \\ c(l, k'', v+\beta) \end{array} \right] + p(j, v) \quad (2.28)$$

Some refinements were given by Hentous & Guinet [1996], in the modeling of the two last characteristics together.

2.4. SUMMARY AND CONCLUSION

A hybrid flow shop consists of a series of production workshops, each of which has several facilities in parallel. Such shops are also called «flow shop with multiple processors», «multi-stage system» and when the machines in each stage are identical, «flexible flow shop». General characteristics of hybrid flow shops were established in §2.1.2, and many restrictions encountered in process industries have been identified: release dates, setup and removal times, time lags restrictions, storage constraints, and precedence constraints.

Hybrid flow shops have been placed in the production system hierarchy; they are a generalization of flow shop and parallel machine environments. A classification scheme was proposed in §2.2.3, taking into account the «hybrid» dimension, inspired by the well-known classification system of Conway, Maxwell & Miller. It will be a key, firstly, to judge the complexity of a problem, and secondly, to find an existing tool.

We have suggested a mixed integer linear programming (MILP) formulation for a general hybrid flow shop scheduling problem with makespan or maximum lateness criteria. It can be noticed from the general model that it is easy to formulate a hybrid flow shop problem with another performance measure as defined in §2.2.2 because all these measures depend only on a single variable. Additional characteristics were considered: release dates, job-precedence constraints, minimum and maximum time lags, non-sequence dependent machine-setup and removal times, sequence dependent machine-setup, job-setup, and machine & job-setup times, limited buffer storage capacity, and a limited number of auxiliary resources. We have shown how they can be taken into account in the MILP modeling and in the calculation of job completion times. With the inclusion of all these characteristics into the model for the job completion time calculation, every permutation of a set of unscheduled jobs, calculated with previous formulae gives an admissible solution of the scheduling problem. An evaluation of these partial or complete solutions according to the given formulae satisfies the constraints, and avoids a return to previous decisions [Botta & Guinet; 1995].

CHAPTER 3

SCHEDULING PROBLEMS: A CONCISE SURVEY

3.1. TYPOLOGY OF SOLUTION METHODS	39
3.2. PARALLEL MACHINES	40
3.3. FLOW SHOPS	43
3.4. HYBRID FLOW SHOPS	45
3.4.1. 2-stage Hybrid Flow Shops	45
3.4.2. V-stage Hybrid Flow Shops	47
3.4.3. Main Results	49
3.5. SUMMARY AND CONCLUSION	50

3. SCHEDULING PROBLEMS: A CONCISE SURVEY

The objectives identified by companies faced with scheduling problems vary according to their strategy of management, and each objective can be characterized by the minimization of one or several criteria (see Table 3-A). On the other hand, most resolution methods are based on the optimization of a single criterion.

management strategy	objectives	criteria
make-to-stock management	optimal utilization of facilities:	C_{max}
	work in process minimum:	\bar{C} , $\overline{C\omega}$
make-to-order management	respect of the due-dates:	L_{max} , T_{max} , \bar{T} , \overline{ET} , N_t or N_e
	fast answer to the demands:	\bar{C} , \bar{F} , or \bar{W}

Table 3-A: Optimization criteria according to management fashion.

Most scheduling problems are optimization problems whose complexity is either Polynomial, NP-complete, NP-hard or Open. In the first case, some optimal and efficient resolution methods exist; they appeal to classical techniques of Operational Research. In the other cases, resolution tools are either enumeration methods which give an optimal solution but which are expensive in computation time (possibly even too long to be practical methods), or heuristics which provide a good solution in a reasonable time even for big size problems.

We have mentioned in Chapter 2, that scheduling a V-stage hybrid flow shop involves two problems: sequencing jobs on V stages and assigning them to machines in each stage. These two problems can be treated sequentially or concurrently, and methods for scheduling hybrid flow shop may be based on flow shop and/or parallel machine scheduling techniques. We therefore concentrated our bibliographic research on scheduling problems in parallel machines, flow shops, and also in hybrid flow shop environments. Sections 3.2, 3.3 and 3.4 present the main results in each of these three environments respectively. But firstly, we introduce a classification of solution methods for scheduling problems which will be a guideline to characterize the scheduling tools.

3.1. TYPOLOGY OF SOLUTION METHODS

Portmann in [Artiba & Elmaghraby; 1993] proposes at least five families of algorithmic design: constructive (or progressive building) methods, neighborhood (or local search) methods, decomposition (or splitting-up) methods, model changing methods (generally relaxation methods), and miscellaneous methods including Artificial Intelligence approaches. It is obvious that in order to efficiently solve NP-hard problems, several approximation schemes must be used either independently, or concurrently in mixed approaches.

Constructive methods are iterative methods which begin with an empty set of decisions concerning the schedule and which, at each step, add one or several decisions to the decision set. The process ends when a complete schedule is obtained. At each step, a partial schedule may be completed by a new job in the sequence, by the calculation of the beginning time of a single operation or of the whole set of operation of a job. They are generally « greedy » algorithms, i.e. the partial schedule obtained at any step is not modified by the steps that follow. Constructive methods can be optimal; for example, an optimal solution to the problem $N/1, m(1) = 1 // \bar{F}$ is to sort the jobs in non-descending order of processing times [Smith; 1956]; other problems are solved by optimal constructive methods: $N/1, m(1) = 1 // N_i$ [Moore; 1968], $N/1, m(1) = 1 // T_{\max}$ [Jackson; 1955], $N/2, m(v) = 1 // F / C_{\max}$ [Johnson; 1954].

Neighborhood methods work on a complete schedule and try to improve it at each step. To design a neighborhood method, three procedures have to be defined: one which provides an initial schedule, one which provides a neighbor schedule, and one which provides the value of the criterion. Among neighborhood methods, we can mention three local search methods used in scheduling: the simplex method [Dantzig; 1962], simulated annealing [Echalier; 1991], [Fleury; 1994], and the tabu search [Widmer & Hertz; 1989].

Decomposition methods are the most natural approaches. Splitting up a problem creates sub-problems that are either smaller or simpler. Generally, even when the sub-problems are optimally solved, the global solution which is obtained is not optimal but can be better (or worse) than solutions obtained by a global approximation scheme. There are many kinds of decomposition methods, depending on the way the problem is split, and the way the sub-problems are solved. Hierarchical methods split the problem into several levels. Higher levels work on aggregate data, and their decisions become constraints for the lower levels. Branch-and-bound methods split the set of solutions; they are heuristics if the branch-and-bound

mechanism is truncated before having proved that the best schedule obtained is optimal. Splitting methods can also use temporal or spatial decomposition [Portmann 1988].

The family of model changing methods concerns methods that change the model of the problem in order to simplify it. Most relaxation methods insert the simplified model into a branch-and-bound method in order to find an optimal schedule. The linearization of the objective function or the addition of facultative constraints allows a restricted set of schedules to be selected. This restricted set of possible schedules is easier to explore; for example, the addition of the no-passing constraint in a N-jobs, V-machines flow shop scheduling problem, makes the problem easier to study: with additional constraints, there are $N!$ possible solutions to explore instead of $(N!)^V$; however, the obtained solution may be not optimal.

Miscellaneous methods are methods, including Artificial Intelligence approaches, that cannot be classified in the previous families. For instance, constraints programming [Esquirol; 1994], [Baptiste & al.; 1994], neural networks [Herault; 1994], expert systems, knowledge-based systems... Among the methods that take inspiration from natural phenomena, we can also mention genetic algorithms; like simulated annealing and tabu search, they are random algorithms but instead of working on a single solution, they work on a population of solutions which becomes globally better by selection, crossover, and mutation techniques [Portmann; 1996].

Faced with a complex scheduling problem, we will be able to mix several of the previous approaches to find a « good » solution.

3.2. PARALLEL MACHINES

This section gives an overview of the major results in parallel machine scheduling research. Main results in scheduling parallel machines can be found in books by Bedworth & Bailey [1982], Carlier & Chretienne [1988], French [1982], Lawler & al. [1989], Pinedo [1995] or in review articles by Graves [1981], Cheng & Sin [1990], Echalier [1991], Guinet & al. [1992], and GOTHA [1993].

The criteria most studied in the literature are based on job completion times: C_{max} , ΣC and $\Sigma C\omega$. There are nevertheless some results regarding other criteria like L_{max} , ΣT or $\Sigma T\omega$. Some problems have been studied in the case of job preemption, with sequence dependent changeover times, or with precedence constraints. Methods for parallel machine scheduling

are mainly list algorithms based on priority rules (SPT, LPT, EDD,...); but we have more recently found other techniques like simulated annealing and genetic algorithms.

In the following tables, for each problem we give its complexity (P or NP), and for each reviewed resolution method, its author(s), its efficiency (Optimal or Heuristic), and its type according to the classification presented in section 3.1 (Constructive, Neighborhood, Decomposition, Model changing, or Other miscellaneous methods). Scheduling problems are presented according to the performance measure they minimize: makespan in Table 3-B, sum or weighted sum of completion times in Table 3-C, maximum lateness in Table 3-D, and sum or weighted sum of tardiness in Table 3-E.

problem		reference		type
$N/I, m_1 \geq 1, I // C_{max}$	NP	Graham; 1969	H	C (list algorithm)
		Frenk & Rinnooy Kan; 1987	H	C (list algorithm)
$N/I, m_1 \geq 1, U // C_{max}$	NP	Cho & Sahni; 1980	H	C (list algorithm)
		Morrison; 1988	H	C (list algorithm)
$N/I, m_1 \geq 1, R // C_{max}$	NP	Horowitz & Sahni; 1976	H	N (linear prog.)
		Potts; 1985	H	N (linear prog.)
		Lenstra & <i>al.</i> ; 1989	H	N (linear prog.)
$N/I, m_1 \geq 1, I/pmt/C_{max}$	P	McNaughton; 1959	O	C (list algorithm)
$N/I, m_1 \geq 1, U/pmt/C_{max}$	P	Sahni & Cho; 1979	O	C (list algorithm)
$N/I, m_1 \geq 1, R/pmt/C_{max}$	P	Lawler & Labetoulle; 1978	O	N (linear prog.)
$N/I, m_1 \geq 1, I/S_{sd} / C_{max}$	NP	Guinet; 1993	H	M (routing pb)
$N/I, m_1 \geq 1, R/S_{sd} / C_{max}$	NP	Elmaghraby & Guinet; 1992	H	M (routing pb)
$N/I, m_1 \geq 1, I/r(i) \geq 0 / C_{max}$	NP	Carlier; 1987	O	D (branch&bound)
$N/I, m_1 \geq 1, I/in-tree, p(i) = 1 / C_{max}$	P	Hu; 1961	O	C (list algorithm)

Table 3-B: Scheduling methods in parallel machine environment (makespan criterion).

problem		reference		type
$N/I, m_1 \geq 1, I // \Sigma C$	P	Baker; 1974	O	C (list algorithm)
$N/I, m_1 \geq 1, U // \Sigma C$	P	Horowitz & Sahni; 1976	O	C (list algorithm)
$N/I, m_1 \geq 1, R // \Sigma C$	P	Bruno & <i>al.</i> ; 1974	O	N (hungarian meth.)
		Alidaee & Ahmadian; 1993	O	M(transportation pb)
$N/I, m_1 \geq 1, U/S_{sd} / \Sigma C$	NP	Guinet; 1991	H	M (routing pb)

problem	reference	type
$N/1, m_1 \geq 1, I // \Sigma C \omega$	NP Baker & Merten; 1973	H C (list algorithm)
	Barnes & Brenan; 1977	H D (branch&bound)
	Sarin & al.; 1988	O D (branch&bound)
	Webster; 1993	H C (list algorithm)
$N/1, m_1 \geq 1, R // \Sigma C \omega$	NP Bruno & al.; 1974	H N (hungarian meth.)
	Elmaghraby & Park; 1974	O D (branch&bound)
$N/1, m_1 \geq 1, I / S_{sd} / \Sigma C \omega$	NP So; 1990	H D (dynamic prog.)

Table 3-C: Scheduling methods in parallel machine environment (completion times criterion).

problem	reference	type
$N/1, m_1 \geq 1, I / r(i) \geq 0 / L_{max}$	NP Gusfield; 1984	H C (list algorithm)
	Carlier; 1987	O D (branch&bound)
$N/1, m_1 \geq 1, I / r(i) \geq 0, p(i) = p / L_{max}$	P Simons; 1983	O C (list algorithm)
$N/1, m_1 \geq 1, I / in-tree, p(i) = p / L_{max}$	P Brucker & al.; 1977	O C (list algorithm)
	Monma; 1982	O C (list algorithm)
$N/1, m_1 \geq 1, U / pmt / L_{max}$	P Sahni & Cho; 1980	O C (list algorithm)
$N/1, m_1 \geq 1, U / r(i) \geq 0, pmt / L_{max}$	P Sahni & Cho; 1979	O C (list algorithm)
	Federgruen & Groenevelt; 1986	O M (transportation pb)
$N/1, m_1 \geq 1, R / pmt / L_{max}$	P Lawler & Labetoulle; 1978	O N (linear prog.)

Table 3-D: Scheduling methods in parallel machine environment (maximum lateness criterion).

problem	reference	type
$N/1, m_1 \geq 1, I // \Sigma T$	NP Dogramaci & Surkis; 1979	H C (list algorithm)
$N/1, m_1 \geq 1, R / S_{sd} / \Sigma T$	NP Echaliier; 1991	H N (sim. annealing)
$N/1, m_1 \geq 1, I // T \omega$	NP Dogramaci; 1984	O M (dynamic prog.)
	Lash; 1993	H O (genetic)
$N/1, m_1 \geq 1, I / \omega(i) = a. p(i) / T \omega$	NP Arkin & Roundy; 1991	H C (list algorithm)
$N/1, m_1 \geq 1, U / p(i) = p / T \omega$	P Tremoliere; 1978	O N (permutation)
$N/1, m_1 \geq 1, U // T \omega$	NP Guinet; 1995	H N (sim. annealing)
$N/1, m_1 \geq 1, I / d(i) = d / ET \omega_{max}$	NP Li & Cheng; 1994	H C (list algorithm)
$N/1, m_1 \geq 1, R, d(i) = d / \Sigma(ET)$	NP Alidaee & Panwalkar; 1993a	H M (transportation pb)

Table 3-E: Scheduling methods in parallel machine environment (tardiness criterion).

3.3. FLOW SHOPS

This section reviews the main results in flow shop scheduling. In scheduling literature, many books or articles deal with general flow shop scheduling; Among publications during the last decade, we can mention Carlier & Chretienne [1988], Lawler & *al.* [1989], Dudek & *al.* [1992], Proust [1992], GOTHA [1993], and Pinedo [1995]. Flow shop scheduling complexity results can be found in Garey & Johnson [1979].

The criterion most studied in flow shop scheduling is the makespan. S. M. Johnson [1954] published the first result for the optimal resolution of the problem $N/2, m_1=m_2=1/F/C_{max}$. His work is the basis for numerous resolution heuristics for more complex problems, among which we can mention the problem $N/V, m_v=1/F/C_{max}$ studied by many researchers. Tardiness criteria have also been the object of some work, but there are few results concerning other criteria. The possible restrictions that have been studied in the V-machine flow shop scheduling problem generally concern non-sequence dependent changeover times, and operation-precedence constraints; we found only a few works about time lags (minimum or maximum), sequence dependent changeover times, limited storage constraints, and no-wait or no-idle constraints.

In general manner, resolution methods for flow shop scheduling problems are based on the Johnson algorithm, except for neighborhood methods which are inspired by single machine problem solutions, and some branch-and-bound methods.

In the following tables, for each problem, we give its complexity, and for each reviewed resolution method, its author(s), its efficiency, and its type according to the classification presented in section 3.1. Table 3-F presents 2- or 3-machine flow shop problems, while general V-machine flow shop problems are exposed in Table 3-G for makespan criterion, and in Table 3-H for other criteria.

problem	reference	type
$N/2, m_v=1/F/C_{max}$	P Johnson; 1954	O C (list algorithm)
$N/2, m_v=1/F, S_{nsd}, R_{nsd}/C_{max}$	P Sule & Huang; 1983	O C (list algorithm)
$N/2, m_v=1/F, S_{nsd}, R_{nsd}, a_{i,v}/C_{max}$	NP Nabeshima & Maruyama; 1983	H M + C (Johnson)
$N/2, m_v=1/F, r(i) \geq 0/L_{max}$	NP Grabowski; 1980	H D (branch&bound)

problem	reference	type
$N/2, m_v=1/F, b_{1,2}/C_{max}$	NP Papadimitriou & Kanellakis; 1980 Leisten; 1990	H M ($b_{1,2} \rightarrow$ no-wait) H C (list algorithm)
$N/2, m_v=1/F, no-idle/\Sigma C$	NP Adiri & Pohoryles; 1982	H C (list algorithm)
$N/2, m_v=1/F, no-wait/\Sigma C$	NP Adiri & Pohoryles; 1982	H C (list algorithm)
$N/2, m_v=1/F, o-prec, tree/C_{max}$	P Sidney; 1981	O N (permutation)
$N/2, m_v=1/F, o-prec/C_{max}$	NP Mc Mahon & Lim; 1993	H D (branch&bound)
$N/3, m_v=1/F, dominance/C_{max}$	P Johnson; 1954	O C (list algorithm)

Table 3-F: Scheduling methods in 2- or 3-machine flow shop environments.

problem	reference	type
$N/V, m_v=1/F/C_{max}$	NP Page; 1961 Palmer; 1965 Hundal & Rajgopal; 1965 Campbell & <i>al.</i> ; 1970 Dannenbring; 1977 King & Spachis; 1980 Nawaz & <i>al.</i> ; 1983 Widmer & Hertz; 1989 Taillard; 1990 Ho & Chang; 1991 Moccellin; 1995	H C (list algorithm) H C (list algorithm) H C (list algorithm) H M(V \rightarrow 2)+C(Johnson) H N (permutation) H C (list algorithm) H C (permutation) H N (tabu) H N (tabu) H N (permutation) H N (tabu)
$N/V, m_v=1/F, S_{nsd}, R_{nsd}/C'_{max}$	NP Szwarc; 1983 Proust & <i>al.</i> ; 1991	H M(V \rightarrow 2)+C(Johnson) H M(V \rightarrow 2)+C(Johnson)
$N/V, m_v=1/F, a_{i,v}/C_{max}$	NP Szwarc; 1983	H M(V \rightarrow 2)+C(Johnson)
$N/V, m_v=1/F, j-prec, out-tree, S_{nsd}, R_{nsd}, a_{i,v}/C_{max}$	NP Botta & Guinet; 1996	H M(V \rightarrow 2)+C(Johnson), C(permutation, or list)
$N/V, m_v=1/F, A_{i,v}/C_{max}$	NP Liou & Smith; 1992	H N(tabu, sim. annealing)
$N/V, m_v=1/F, S_{sd}/C_{max}$	NP Srikar & Ghosh; 1986 Szwarc & Gupta; 1987 Das & <i>al.</i> ; 1995	O D (branch & bound) H M(V \rightarrow 2)+C(Johnson) H C (list algorithm)
$N/V, m_v=1/F, B_{v,v+l}/C_{max}$	NP Daniels & Mazzola; 1994 Daniels & Mazzola; 1994	H N (permutation) O D (branch & bound)
$N/V, m_v=1/F, no-wait/C_{max}$	NP King & Spachis; 1980 Rajendran; 1994	H C (list algorithm) H C (permutation)

Table 3-G: Scheduling methods for V-machine flow shop environments with makespan criterion.

problem	reference	type
$N/V, m_v=1/F / \Sigma C$	NP Krone & Steiglitz; 1974 Bansal; 1977	H N (perm) O D (branch&bound)
$N/V, m_v=1/F, no-idle/\Sigma C$	O Adiri & Pohoryles; 1982	H C (list algorithm)
$N/V, m_v=1/F, no-wait/\Sigma C$	O Adiri & Pohoryles; 1982	H C (list algorithm)
$N/V, m_v=1/F / T_{max}$	NP Townsend; 1977 Guinet & Solomon; 1996	H D (branch&bound) H M ($T_{max} \rightarrow C_{max}$)
$N/V, m_v=1/F, r(i) \geq 0 / L_{max}$	NP Grabowski & <i>al.</i> ; 1983	H D (branch&bound)
$N/V, m_v=1/F / \Sigma T$	NP Kim; 1993	H C (list algorithm)
$N/V, m_v=1/F / \Sigma(T\omega + C\omega)$	NP Gelders & Sambandam; 1978	H C + N (permutation)

Table 3-H: Scheduling methods for V-machine flow shop environments with other criteria.

3.4. HYBRID FLOW SHOPS

The interest in hybrid flow shop scheduling problems is relatively recent. In the literature nearly all the studies regarding these problems, deal with the minimization of makespan criterion, and are often limited to two stages. Papers dealing with possible restrictions date from 1988, except for the no-wait restriction, which was studied by Salvador [1973]. Scheduling methods for hybrid flow shops are based on flow shop scheduling methods, (on Johnson's rule in particular) and on parallel machine scheduling methods (principally priority rules). Interesting detailed reviews can be found in Lee & Vairaktarakis [1994], and in Vignier & *al.* [1995].

In §3.4.1 and §3.4.2, we present some methods and results for 2-stage and V-stage hybrid flow shop scheduling problems respectively. A summary is given §3.4.3.

3.4.1. 2-STAGE HYBRID FLOW SHOPS

In this section, we present a selection of interesting methods to solve the 2-stage hybrid flow shop scheduling problems. A complete list of all the results or references relating to this topic that we are aware of are presented in §3.4.3. We note that many special cases, in which one of the two stages had only one machine have been studied. This situation can be encountered in industry: while working on a scheduling problem in a parallel machine environment, it can often be useful to take the previous or next workshop into account, if this is a bottleneck machine.

One of the first results concerning 2-stage hybrid flow shops was published by Shen & Chen, who in 1972 suggested an heuristic for the problem $N/2, m_1 > 1, I/F/C_{max}$. They sequenced the jobs according to a strategy called the *ME strategy* (More Earlier time) so that the job i would be scheduled before the job j if

$$\begin{cases} p(i, m_1) + p(i, m_2) \geq p(j, m_1) + p(j, m_2) \\ p(i, m_1) \leq p(j, m_1) \end{cases}$$

In 1988, Gupta proposed another constructive method for C_{max} criterion when the number of machines in the second stage is limited to one: he sequenced the jobs according to Page's index, and assigned jobs to the last busy machine (LBM rule) in the first stage. In 1991, Gupta & Tunc solved the opposite problem ($m_1=1$, and $m_2 > 1$) with a similar approach.

In 1989, Sriskandarajah & Sethi solved the problem when $m_1=1$, and $m_2 > 1$ by transforming it into $N/2, m_1=m_2=1/F/C_{max}$ and applying Johnson's rule for sequencing and a list algorithm for assignment in the second stage. They also proposed a three-step heuristic for the problem $N/2, m_1=m_2=m > 1, I/F/C_{max}$:

1. define m problems $N/2, m'_1=m'_2=1/F/C_{max}$,
2. solve the affectation problem $N/1, m > 1, I/C_{max}$ where the processing time of a job i on the machine is $p(i, m_1) + p(i, m_2)$ by using a list algorithm,
3. solve the m flow shop problems by applying Johnson's rule.

In 1993, Sriskandarajah studied the case of 2-stage hybrid flow shops with no-wait constraint and makespan criterion. When $m_1=1$, and $m_2 > 1$, he proposed to sort the jobs in descending order of their processing times (LPT rule) in stage 2. He also worked on a more general case where $m_1=m_2=m > 1$, and a no-wait constraint: for this case, he proposed either to use the previous algorithm or to apply a method similar to [Sriskandarajah & Sethi; 1989]: partitioning the problem into m sub-problems $N/2, m_1=m_2=1/F/C_{max}$, assigning the jobs on m fictitious machines (flow shop), and solving the m flow shop problems with two machines. When m_1 and m_2 are independent and both greater than one, he suggested decomposing the problem into m_1 sub-problems, each having only one machine in the first stage and a subset of the m_2 parallel machines in the second stage, and applying previous approaches.

In 1994, Gupta & Tunc studied the problem $N/2, m_1=1, m_2 > 1, I/F, S_{nsd}, R_{nsd}/C_{max}$. Their approach consisted of two steps: sequencing the jobs with Sule & Huang's algorithm [1983]

on a 2-machine flow shop problem, and assigning jobs in the second stage to the machine that minimizes the waiting time of the jobs (LBM rule).

Vignier & *al.* [1996-a] proposed a heuristic for the specific problem $N/2, m_1 > 1, I, m_2 = 1/F, pmt/T_{max}$ which uses the EDD rule for the sequencing problem and the algorithms from Horn and Mac Naughton for the assignment problem.

In 1994 Lee & Vairaktarakis proposed a two-step heuristic for solving the most general problem $N/2, m_v > 1, I/F/C_{max}$: sequencing the jobs with the Johnson rule on a fictitious 2-machine flow shop problem, where the processing times in each stage are divided by the number of machines in the considered stage, and assigning jobs with the FAM (first available machine) rule in the first workshop, and with the LBM (last busy machine) rule in the second. Guinet & *al.* [1996] solved the same problem with a similar method where only the assignment step differs: the jobs are assigned in workshop v to the machine which minimize their completion date in v , i.e. the FAM rule. M'Hallah & Haouari [1995] proposed a neighborhood method for the same problem: they construct an initial feasible solution by a constructive method, and then apply tabu search or simulating annealing approaches.

All these authors have only considered the case of identical machines in each stage. Narasimhan & Mangiameli [1987] studied the case of unrelated machines at the second stage and proposed several priority rules to solve the problem. When $m_1 = 1$, and $m_2 = 2$, R, Riane & *al.* [1996] suggested three approaches: an optimal dynamic programming approach, a branch-and-bound method, and a heuristic; this last approach considers two separated 2-machine flow shops by considering 2 machines in stage one whose capacity sum is equal to the capacity of the real machine. The jobs are sequenced applying Johnson's algorithm and the resulting sequences are merged to produce a single sequence of N jobs.

3.4.2. V-STAGE HYBRID FLOW SHOPS

A variety of methods for solving general V-stage hybrid flow shop scheduling problems are exposed here, and all results or references that we encountered in the literature, are given in §3.4.3.

Some branch-and-bound algorithms have been developed for the general V-stage hybrid flow shop scheduling problem. The first was proposed by Salvador [1973] for the problem with no-wait constraint and identical machines in each stage. Other optimal branch-and-bound algorithms were presented by Brah & Hunsucker [1991] and Vignier & *al.* [1996-c]

for C_{max} criterion, or by Vignier & *al.* [1996-b] for ΣC criterion. Rajendran & Chaudhuri [1992-a, -b] have also suggested branch-and-bound approaches to solve the general problem with C_{max} or ΣC criteria.

In 1994, Hunsucker & Shah studied the problem in a simulation environment in which the number of current jobs was constrained to be less than N^* ; they compared several priority rules as: FIFO, LIFO, SPT, LPT, MWkR, and LWkR, for the V-stage problem ($V \geq 2$) with makespan, sum of flow times or maximum flow time, when jobs are subject to release dates.

In 1994, Lee & Vairaktarakis proposed an approach similar to that of Campbell & *al.* for the M-machine flow shop: reducing the V-stages problem to $V/2$ 2-stage problems.

Guinet [1996] compared two approaches for makespan and mean completion time criteria to solve the general problem where machines at each stage are either identical or unrelated: an integrated approach in which the assignment and sequencing problems are performed at the same time, and a sequential approach in which the assignment is carried out after the sequencing. He concluded that the second approach gives better results when the machines are identical.

Artiba [1994] studied the special configuration of parallel multi-product manufacturing lines which can be called parallel flow shops: a product, once started in a line (flow shop) must follow that line until the end of the process. The author proposed an expert system approach to solve the problem.

Aggezzaf & *al.* [1995] analyzed a hybrid flow shop scheduling problem with sequence dependent setup times in the carpet industry: they decomposed the problem into two sub-problems: a multi-item, multi-level, capacitated lot sizing problem on parallel groups of processors with setup times, and several scheduling problems on parallel processors with sequence dependent setup times for each group. They suggested a linear programming approach to solve the problem.

in 1996, Artiba & Aggezzaf proposed an architecture of multi-model system for the general planning & scheduling problems. The system integrates expert system techniques, discrete event simulation, and optimization algorithms to support decision making for hybrid flow shops.

3.4.3. MAIN RESULTS

We identify in the following tables, the references of the author(s) who have proposed scheduling methods for each problem. Table 3-I presents 2-stage hybrid flow shop problems, and Table 3-J, general V-stage hybrid flow shop problems.

problem	reference
$N/2, m_1=1, m_2>1, I/F/C_{max}$	Narasimhan & Panwalkar; 1984 Sriskandarajah & Sethi; 1989 Gupta & Tunc; 1991 Chen; 1995
$N/2, m_1=1, m_2>1, I/F, S_{nsd}, R_{nsd}/C_{max}$	Gupta & Tunc; 1994
$N/2, m_1=1, m_2>1, I/F, no-wait/C_{max}$	Sriskandarajah; 1993
$N/2, m_1=1, m_2=2, R/F/C_{max}$	Riane, Artiba, Elmaghraby; 1996
$N/2, m_1>1, I, m_2=1/F/C_{max}$	Arthanari; 1974 see [Sriskandarajah; 1993] Gupta; 1988 Blazewicz, Dror, Pawlak, Stecke; 1994 Chen; 1995
$N/2, m_1>1, I, m_2=1/F, pmt_1/T_{max}$	Billaut, Hougbossa, Veillat, T'Kindt; 1996 Vignier, Aucanot, Sangouard, Billaut; 1996
$N/2, m_1>1, m_2=1, I/F, no-wait/C_{max}$	Sriskandarajah; 1993
$N/2, m_1=m_2>1, I/F/C_{max}$	Sriskandarajah & Sethi; 1989 Deal & Hunsucker; 1991
$N/2, m_1=m_2>1, I/F, no-wait/C_{max}$	Sriskandarajah; 1993
$N/2, m_1=m_2>1, I/F, a_{i,v}/C_{max}$	Langston; 1987
$N/2, m_v>1, I/F/C_{max}$	Shen & Chen; 1972 Buten & Shen; 1973 Guinet, Solomon, Kedia, Dussauchoy; 1996 Lee & Vairaktarakis; 1994 M'Hallah & Haouari; 1995
$N/2, m_v>1, I/F, no-wait/C_{max}$	Sriskandarajah; 1993
$N/2, m_1>1, I, m_2>1, R/F/C_{max}$	Narasimhan & Mangiameli; 1987

Table 3-I: Scheduling methods in 2-stage hybrid flow shop environments.

problem	reference
$N/V, m_v=m>1, I/F/C_{max} (+\Sigma C)$	Wittrock; 1988
$N/V, m_v>1, I/F/C_{max}$	Brah & Hunsucker; 1991 Rajendran & Chaudhuri; 1992-a Artiba 1994 Lee & Vairaktarakis; 1994 Vandeveld; 1994 see [Vignier & <i>al.</i> ; 1995] Vignier, Billaut, Proust; 1996 Guinet & Solomon; 1996 Guinet; 1996
$N/V, m_v>1, I/F/C_{max}, \Sigma F, F_{max}$	Hunsucker & Shah; 1994
$N/V, m_v>1, I/F, no-wait/C_{max}$	Salvador; 1973
$N/V, m_v>1, I/F, b_{v,v+1}/C_{max}$	Wittrock; 1988
$N/V, m_v>1, I/F, b_{v,v+1}, B_{v,v+\beta}/C_{max}$	Hentous & Guinet; 1996
$N/V, m_v>1, I/F, S_{sd}/C_{max}$	Elmaghraby & Karnoub; 1995 Aghezzaf, Artiba, Elmaghraby; 1995
$N/V, m_v>1, U/F, S_{nsd}, a_{i,v}/C_{max}$	Egbelu; 1991
$N/V, m_v>1, R/F/C_{max}$	Guinet; 1996
$N/V, m_v>1, I/F/\Sigma C$	Rajendran & Chaudhuri; 1992-b Guinet; 1996 Vignier, Dardilhac, Delalay, Proust; 1996
$N/V, m_v>1, R/F/\Sigma C$	Guinet; 1996
$N/V, m_v>1, I/F/T_{max}$	Guinet & Solomon; 1996
$N/V, m_v>1, I/F/\Sigma T$ or N_t	Hunsucker & Shah; 1992

Table 3-J: Scheduling methods in V -stage hybrid flow shop environments.

3.5. SUMMARY AND CONCLUSION

In parallel machine scheduling literature, the criteria most studied are C_{max} , ΣC , and $\Sigma C\omega$. There has been nevertheless some results concerning other criteria such as L_{max} , ΣT , or $\Sigma T\omega$. Among the possible restrictions, job preemption, sequence dependent changeover times, and precedence constraints have been studied. Scheduling methods are mainly list algorithms based on priority rules; these ideas are useful for solving the assignment phase in hybrid flow shop scheduling problems.

For flow shop scheduling, the criterion most studied is the makespan, and the main results are based on Johnson's algorithm. There are very few results concerning other criteria.

The restrictions studied are especially non-sequence dependent changeover times and operation-precedence constraints; we found very few concerning time lags, sequence dependent changeover times, limited storage constraints, and no-wait or no-idle constraints. In general, resolution algorithms are model changing methods which use Johnson's rule, neighborhood methods that take inspiration from results of single machine problems, or some branch-and-bound methods.

Studies on hybrid flow shop scheduling problems are relatively recent. In the literature, nearly all the studies regarding these problems deal with the minimization of makespan criterion, and are often limited to two stages; few restrictions have been considered. Resolution methods are based on flow shop scheduling methods, especially Johnson's rule, and on parallel machine scheduling methods, principally with priority rules. When the number of workshops is greater than two, decomposition methods have been also designed (branch-and-bound methods).

Based on all these studies, we can conclude that each problem is a special case, and the use of an algorithm that has been developed to solve the problem P1, may be inefficient for the problem P2, even if P2 is quite similar to P1. Due to the diversity of production system organizations in process industries, the scheduling problems are very varied. It is therefore advisable to pay a particular attention to the analysis of the production system in order to correctly identify the right problem(s). Only afterwards can the search for appropriate scheduling tools begin.

CHAPTER 4

A SUITABLE METHOD FOR PRODUCTION SYSTEM ANALYSIS & MODELING

4.1. ANALYSIS AND DESIGN METHODOLOGIES: A SURVEY	54
4.1.1. The Structured Approach	54
4.1.2. The Systemic Approach	55
4.1.3. The Object-oriented Approach	55
4.1.4. Production Engineering-oriented Methodologies	56
4.2. A NEW METHODOLOGY: OASISS	56
4.2.1. Top-down Analysis of Resources	58
4.2.1.1. A resource approach	58
4.2.1.2. A graphic language support	59
4.2.2. Bottom-up Recognition of Resources	61
4.2.2.1. Object classes	61
4.2.2.2. The recognition process	63
4.2.3. Instantiating and Integration of Objects	63
4.2.3.1. Specification tools for object behavior	63
4.2.3.2. Dynamic modeling	67
4.2.3.3. Selection of scheduling tools	69
4.2.4. Implementation of Objects	70
4.3. SUMMARY AND CONCLUSION	70

4. A SUITABLE METHOD FOR PRODUCTION SYSTEM ANALYSIS AND MODELING

As it has been emphasized in previous chapters, when we speak about scheduling, it is important to select the right technique for the right problem. The main characteristics of the production system must be identified in order to recognize the production activity control and production scheduling problems, and also to specify solution tools. These tools generally take into account only a subset of these characteristics; and the assessment of these tools, which requires the modeling of the production system, often takes place after the implementation of the scheduling or production activity control system.

In this chapter, we propose a new approach to production system analysis and modeling, whose aim is the design of a complete production activity control system. This approach enables us to identify scheduling problems, and design dynamic models to evaluate and validate scheduling decisions. The goal of this identification is to select pre-existing solution tools. The dynamic model will allow us to assess the feasibility and the robustness of the scheduling tools in virtual situations. This approach associates discrete event simulation with classical scheduling theory.

Before presenting this new methodology in section 4.2, we expose a brief review of analysis and design methodologies, which are already in used since the 70s in section 4.1.

4.1. ANALYSIS AND DESIGN METHODOLOGIES: A SURVEY

This section surveys three major approaches of analysis and design, that have been discussed and used in system development for a decade or more: the structured approach in §4.1.1, the systemic approach in §4.1.2 and the object-oriented approach in §4.1.3. In §4.1.4, we present some methodologies specially designed for production system analysis.

4.1.1. THE STRUCTURED APPROACH

Among the many existing structured analysis methods we note SADT [Ross; 1977, 1985], SASS [DeMarco; 1978, 1982], [Gane & Sarson; 1979], [Yourdon; 1989], and SA/RT [Ward; 1986], [Ward & Mellor; 1987], all of which are functional and hierarchical top-down

decomposition methods. Some computer aided software engineering packages have been developed for these methods: ASA [1] and TEAMWORK [2]⁽¹⁾.

Using Data Flow Diagram graphical language (DFD), these methods are user-friendly. Moreover, they match well to real time and automatic aspects (functions, events, and data), particularly SA/RT which models the system behavior. On the other hand, there is some redundancy in the data structure, and resource and organization aspects are not well detailed, except in SA/RT, where the implementation architecture is studied. Generally speaking, however decomposition levels are too detailed to study a system in its entirety.

4.1.2. THE SYSTEMIC APPROACH

Systemic methods like AMS [Melese; 1972, 1982], MERISE [Tardieu & *al.*; 1983, 1985] supported by MEGA [3], or AXIAL from IBM [Pellaumail; 1986] constitute an other class of methods which have been developed for a decade or more.

This approach allows a perfect structure and optimization of data (Merise or Axial), and the different abstraction levels (conceptual, organizational and physical) are well separated. On the other hand, these methods are not very user-friendly, and the integration between the data model and the functional model is poor.

4.1.3. THE OBJECT-ORIENTED APPROACH

More recently, object-oriented methodologies have attempted to regroup data and functions in the same independent entity: the object. Since 1980 with OOD [Abbott; 1986], [Cox; 1986], [Booch; 1991] many object-oriented methodologies have been developed: HOOD in 1986 [Delatte & *al.*; 1993], OOSA in 1988 [Shlaer & Mellor; 1988, 1992], and over the past five years: SYS-P-O [Jalent; 1991], OOA [Coad & Yourdon; 1991], OMT [Rumbaugh; 1991] (apparently appreciated by industry), MERISE-Object [Bouzeghoub; 1994], and M*-OBJECT [Berio & *al.*; 1995]. Recently, object-oriented software engineering packages have been developed to support some of these methodologies: TEAMWORK [2], OBJECTTEAM [4], and SYS-P-O [5].

Due to the independent nature of objects, these methodologies favor knowledge specification and facilitate its coherence, reuse and evolution; they enable us to model

⁽¹⁾ in Commercial Software References List

complex systems. However, it is relatively difficult to identify objects and this approach is slightly too abstract to meet user requirements.

4.1.4. PRODUCTION ENGINEERING-ORIENTED METHODOLOGIES

The increasing need for reliable techniques for production automated system modeling, design and implementation have induced analysts to define some production engineering-oriented methodologies. For example, GRAI method [Doumeingst; 1984, 1990] is a structured method allowing the functional and organizational aspects of decision systems to be modeled.

The intrinsic complexity of the CIM concept (Computer Integrated Manufacturing) leads to the definition of a reference model. The IDEF method is an expansion of SADT associated with the ICAM model (Integrated Computer Aided Manufacturing) for the study of manufacturing system. This method allows us to model the functional structure of an organization with IDEF0, its information structure with IDEF1 and its dynamic structure with IDEF2 [Bravoco & Yadan; 1985-a, -b, -c]. Another methodology, M*-OBJECT [Vernadat & al.; 1989], [Berio & al.; 1995] has been developed for analysis, design, and implementation of information systems in production.

Finally, we cite CIM-OSA [Gaches & al.; 1990], [Jorysz & Vernadat; 1991-a, -b, -c], an architectural framework which provides principles, tools, guidelines for enterprise-wide modeling. This methodology takes into account three axes: instantiating with different levels of genericity, derivation with different modeling areas (requirements, design, and implementation), and generation with different views (function, information, resource, and organization).

4.2. A NEW METHODOLOGY: OASISS

Generally, the production system studied is complex and knowledge (information) is in the hands of system actors. Our objective is to analyze and to model the physical production system in order to control its resource and organization aspects. Control tools must be reusable. In this chapter, we develop a new methodology, called OASISS (Object-oriented Analysis with Structured and Integrated Specifications and Solutions) for production activity control.

Currently, no specific engineering software or methodology is adapted to such study of production system. Structured methods employed in the past are efficient communication tools between system actors [Guinet; 1990]. But these methods are difficult to use, they do not easily allow the reuse of generic knowledge items, and the system evolution call into question modeled knowledge items. Systemic methods are more complete, but the communication tools they define ate too complex, and not adapted to information reuse. Object-oriented methods are well adapted to complex system analysis. They are based on the concept of reuse and take into account the evolution of knowledge. On the other hand, they are too abstract and define a bad communication tool for the users. No object-oriented method is really adapted for our purposes because they do not provide an executable implementation model with resource and organization views.

As shown in Figure 4-A, the overall architecture of our methodology follows four major steps:

- 1) Top-down analysis of resources, which produces the DFD requirement made up of processes (functions), control flows and data flows.

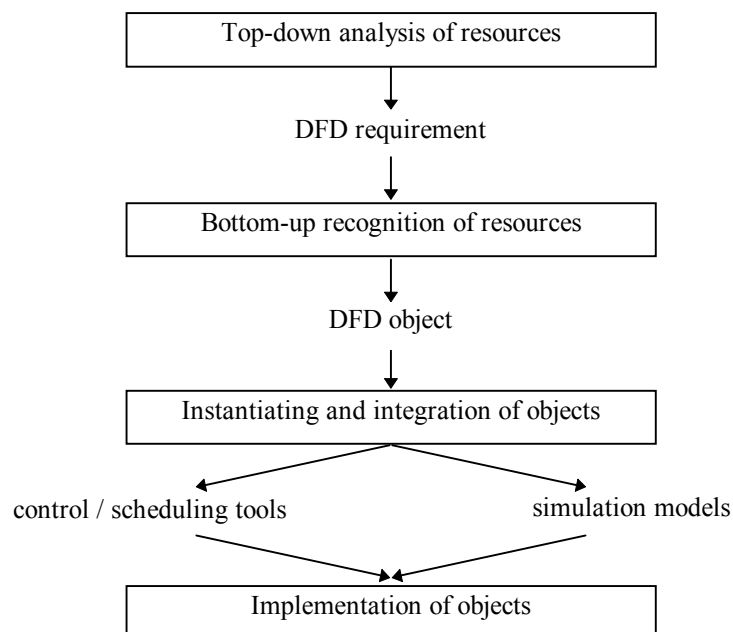


Figure 4-A: Overall architecture of OASISS methodology.

- 2) Bottom-up recognition of resources, which produces the DFD object containing objects and messages.

- 3) Instantiating and integration of objects which lead to the conceptual production system model, specified with simulation tools like Petri nets or ARENA, and to the identification of control /scheduling tools.
- 4) Implementation of objects, i.e. integration of selected heuristics into the simulation model.

Next paragraphs present the four steps of OASISS framework.

4.2.1. TOP-DOWN ANALYSIS OF RESOURCES

This analysis step is achieved using an object-oriented approach based on the identification of production facilities. The nature and the complexity of the production system components are identified; a control problem may be associated with each element. The facilities thus identified are treated as ‘objects’ and the production organization is translated by the layout of facilities. The communication language used is based on the graphical language component of the SADT methodology: SA.

4.2.1.1. A resource approach

The purpose of production control is « to utilize limited resources in the production of goods to satisfy customer demands and create a profit for investors » [Bedworth & Bailey; 1987]. Resources include production facilities, workers, and materials. Minimizing work and the use of materials are mainly taken into account during product and manufacturing design. Production control is constrained by the technical data (bill of materials, routings,...) that result from product and manufacturing design. Production control can only optimize the use of production facilities such as employees, machines, stock, conveyors, tools, ...

According to our objectives:

- The production system is considered as a set of resources which model the production facilities in general and the physical facilities in particular;
- links between the resources are the result of production organization and process design; they model routings, delivery event, orders, order releases, ...
- the resource layout is also a result of production organization; resources represent factories, workshops, centers, lines, cells, ...

In regards to production control problems, an analysis of the production system must show the production facilities, use a resource approach and model physical and organization

features. A black box decomposition has been retained to discover the system. This hierarchical top-down analysis allows us to gradually specify the production facilities, modeling several levels of resources which depend on the organization.

The organization defines the different use of resources. The complexity of the production system cannot be set before the use of resources at the different organization levels has been identified. These levels represent different views of the production system, and will be very useful later to select the appropriate detail of resource specifications according to the horizon of the production control problem.

This hierarchical decomposition based on resources seems to offer the most promise regarding to complexity classification. It ends when all the system characteristics are discovered, and provide a requirement scheme supported by a graphic language.

4.2.1.2. A graphic language support

Knowledge specification requires the definition of communication tools between system actors and the analyst. These tools must first of all be universal and easy to understand for people from different cultures; secondly, they must provide a communication support which allows us to model, critique and assess knowledge [Guinet, 1990], [Kusiak & *al.*; 1994]. We took our inspiration from SADT structured approach in using its graphical language DFD to model the production system for the above reasons and additionally for its simplicity and its user-friendliness.

Ross [1977] introduced the term of structured analysis, as well as a set of symbols and a methodology for creating data flow models. The **data flow diagram** (DFD) models information and transformation flows on varying detail levels. At level 0, the DFD represents the entire system; additional information can be incorporated on level 1, level 2, and so on, as sub-functions of the overall system. From all the variations and the extensions of the structured analysis, the Structured Analysis and Design Technique (SADT) was developed. Its notation, the SA graphic language, consists of boxes and arrows which represent system components and interfaces respectively. Since the 1970s, numerous applications of SADT have been made and its power as a communication and analysis tool was recognized in 1978 by the United States Air Force who selected it as the language used to support the Integrated Computer Aided Manufacturing (ICAM) program. SADT activity modeling was adopted by the ICAM program to develop the ICAM Definition Methodology (IDEF0).

An IDEF0 model consists of three components, diagrams, text, and a glossary, all cross-referenced with one another. The major components of the model are the box and arrow diagrams. A box is assigned to an active verb phrase to represent function, and an arrow, which represents the interfaces (input, control, output, mechanism), is assigned to a descriptive noun phrase (see Figure 4-B). Inputs enter the box from the left, are transformed by the function, and exit to the right as outputs; a control enters the top of the box, and influences or determines the function performed. A mechanism is a tool or resource which performs the function.

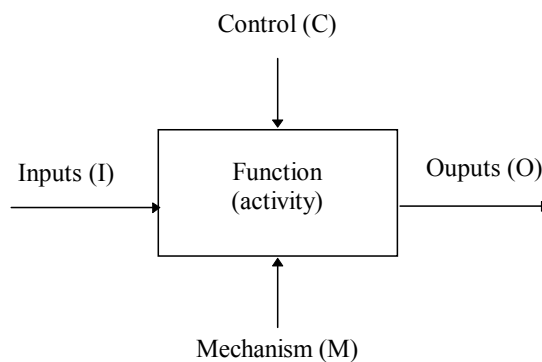


Figure 4-B: IDEF0 function box and interface arrows.

The decomposition principle of an IDEF0 activity model is illustrated in Figure 4-C.

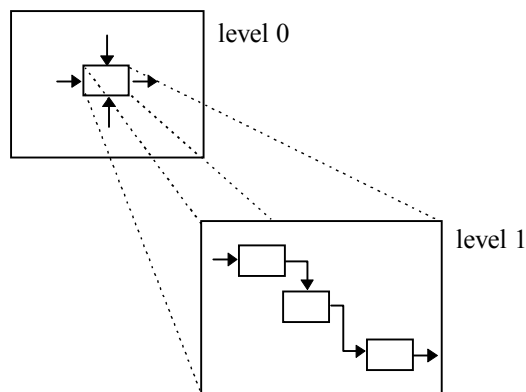


Figure 4-C: IDEF0 model decomposition.

In our approach, we have used the box and arrow diagrams from SA graphic language, and a decomposition principle which is similar to IDEF0, but the meaning of box and arrows has been somewhat modified: An objet will model a resource or a set of resources and will be represented by a function (the box), and the identification of the resource (or set of resources) is modeled by mechanism arrows (M). An organization or process link between two resources will be modeled by a control flow (Control and Output arrows), because it determines the

object performed; Data flows (Input arrows) will only be used for secondary organization links, that do not influence the object such as auxiliary resources or tools. Secondary flow is information that is less useful for complexity analysis and simulation model design but useful for communication between analysts and readers.

Whereas SADT or IDEF0 are based on activity decomposition, in our approach, the top down analysis guideline is the resource. The mechanisms that appear on the different boxes on a single diagram are disjointed; for instance, if a machine may be used for two different operations, these two operations will model a single object. We will refer to these diagrams as **resourcegrams**.

4.2.2. BOTTOM-UP RECOGNITION OF RESOURCES

Object classes have been defined according to the classification presented in section 2.2 and take into account hybrid flow shop environments with identical machines in each stage. Each object class is characterized by its attributes and services. A message will be the request for execution of an object service, and message parameters will be the values of the attributes.

4.2.2.1. Object classes

Some of the main object classes are the following:

- *Identical parallel resource*: This class has six attributes: resource number, job list, release date list, processing time list, completion time list, due date list. A single resource will be an object of this class when the attribute resource number valued to one.
- *Identical parallel resource with nsd⁽¹⁾ machine setup & removal times*: This class is a specialization of the previous object class. Its additional attributes are setup time list and removal time list. A single resource with setup and removal times will be an object of this class when the attribute resource number valued to one.
- *Identical parallel resource with nsd machine setup times,*
- *Identical parallel resource with sd⁽²⁾ machine setup times,*
- *Identical parallel resource with sd machine & job setup times, ...*

⁽¹⁾ nsd : non-sequence dependent

⁽²⁾ sd : sequence dependent

- *V-hybrid flow shop resource*: This class is an aggregation of the first object class, and models an hybrid flow shop organization with V workshops in series, each consisting of identical parallel resources. It has one attribute, the workshop number (i.e. the value of V). A flow shop resource will be an object of this class with the attribute machine number of each identical parallel resource valued to one.
- *V-hybrid flow shop resource with limited storage*: This class is a specialization of the V -hybrid flow shop resource class. It has one additional attribute: buffer capacity list.
- *V-hybrid flow shop resource with minimum time lags*: This class is also a specialization of the V -hybrid flow shop resource class. Its additional attribute is the time lag list.
- *V-hybrid flow shop resource with no storage,*
- *V-hybrid flow shop resource with nsd machine setup & removal times,*

An extract of the specialization and aggregation tree is depicted in Figure 4-D.

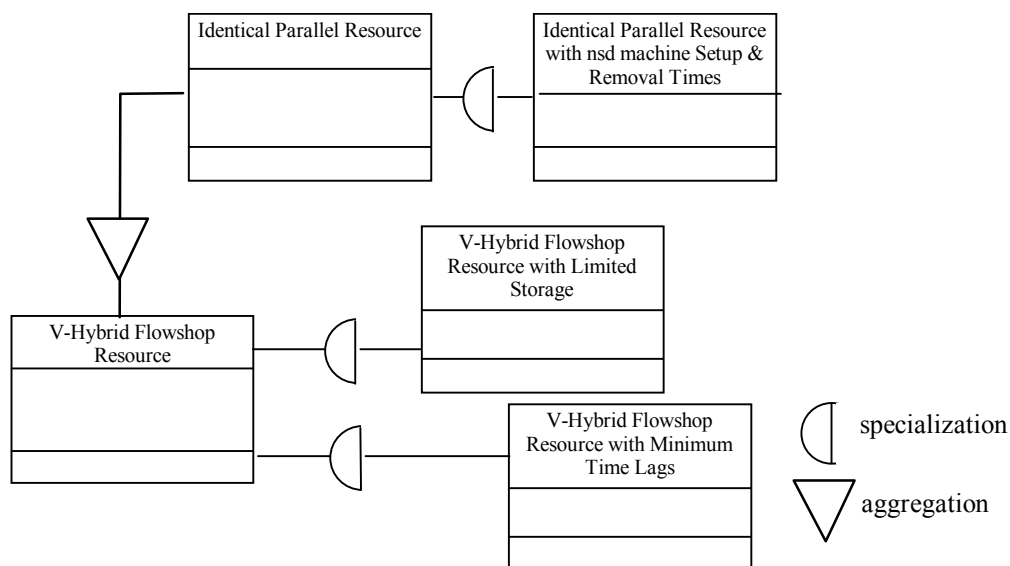


Figure 4-D: Specialization and aggregation tree.

The class « Identical parallel resource » and its specialization's are simple object classes. The others are made by the aggregation of these simple object classes and are called complex object classes.

A set of algorithms and heuristics is associated with each object class, as services, in order to solve its scheduling problems. The object activation with the service « schedule » will allow the definition of the order in which jobs must be scheduled, i.e. to determine the attribute job list. Object classes have another kind of service: execute. A second activation of

the object with the service « execute » allows us to simulate the object working and will give the completion time list.

4.2.2.2. The recognition process

The object identification takes place in a bottom-up fashion after the top-down system characteristic discovery. Every elementary resource identified during the analysis phase of the production system is acknowledged as belonging to a pre-defined simple object class. The hierarchical ascending procedure of resource recognition allows the analyst to identify scheduling problems according to our classification scheme (section 2.2). The general scheduling problem might belong to a complex object class.

4.2.3. INSTANTIATING AND INTEGRATION OF OBJECTS

This phase allows us, on the one hand, to obtain a dynamic model associated with the production system under study, and on the other hand, to select scheduling tools depending on the management strategy (make-to-stock or make-to-order management).

The dynamic model of the whole production system will be obtained from simulation models of the predetermined objects. The structured specification of objects have been done according to the scheduling problem classification presented in section 2.2.

4.2.3.1. Specification tools for object behavior

We firstly propose a structured specification of objects, based on Petri Nets [Silva & Valette; 1989], which allows us to simulate the processing of production orders called jobs, according to a production control plan. The production orders introduce different characteristics: processing times, setup times, removal times, and so on. The production control plan assigns jobs to resources and orders job processing. We use timed and colored Petri nets in order to represent such aspects.

A Petri net is a directed bipartite graph $G(P, T, U)$. P is a set of vertices called places. T is a set of vertices called transitions. U is a set of directed arcs connecting a vertex $p \in P$ and a vertex $t \in T$. The definition of rules that value the P vertices (token) allows us to give the graph a dynamic aspect. Petri nets show two particular aspects of the system: events and event conditions. Transitions model events (the beginning of job processing, the end of job processing,...), places represent event conditions (the job, the state of a resource, the state of

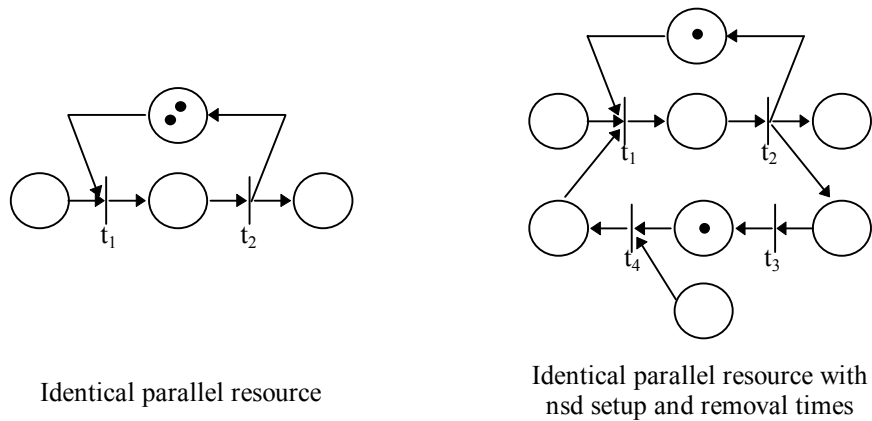
a tool,...). Arcs connect the conditions required by the events to the events, and the events to the conditions produced by the events. Place values allow us to simulate event succession and to study the different system states. Petri nets can be hierarchically modeled: a transition (or a place) can define a sub-network which represents a sub-system.

A colored Petri net [Proth & Xie; 1994] is an extension of a Petri net in order to reduce the graphic size of the network. The specification of several place valuation rules which depend on a place attribute (the token color) allows us to represent different places with one place i.e. several event conditions with one place. We use colored Petri nets in order to specify job identity. A job is identified by a color and have its own place valuation rules in order to introduce its processing characteristics. We have defined a priority for each color (i.e. for each job) in order to avoid rule conflicts and represent a control plan (i.e. a job schedule).

A timed Petri net [Carrier & Chretienne; 1988] is an extension of a Petri net in order to model time characteristics. During simulation, the valuation of a timed place is suspended during its waiting time after its valuation. We have assigned a waiting time to each place and each color. This waiting time can represent job processing time, setup time, or removal time.

Figure 4-E illustrates the specification of two simple object classes using Petri nets:

- An *identical parallel resource* is defined by four places and two transitions. The transition t_1 models the beginning of resource use. The two input places of this transition represent the presence or absence of a job and the state of resources. Token model respectively the job and the free resources. The transition t_2 represents the end of a resource employment. Its input place models the job in process and its output places model the job completion and the resource state.
- An *identical parallel resource with nsd machine setup and removal times* is an identical parallel resource with two more transitions: t_3 models the end of the removal operation, its input place represents the removal operation; t_4 models the beginning of the setup operation, its input places represent the bare machine and the setup requirement, and its output place represents the setup operation.



Identical parallel resource

Identical parallel resource with
nsd setup and removal times

Figure 4-E: Specification of simple object classes using Petri nets

Petri nets are universal and commonly used modeling tools; they have been used for object specification and design of simulation sub-model in order to emphasize the feasibility of our approach. Thereafter, we have selected a more powerful simulation tool: the ARENA simulation environment [6].

ARENA is an integrated framework for discrete and/or continuous system modeling, allowing the construction of SIMAN simulation models for various application fields. It has a graphic user interface and includes all the functions also bound to simulation such as animation, input data analysis, model entry support, scenario management, output result analysis, in the same environment [Banks; 1994].

The engine of ARENA for the model construction is SIMAN language [Collins & Watson; 1993], [Banks & al.; 1995]. It is particularly adapted to the modeling of manufacturing systems, thanks to its ability to describe the components of the environment and the logic of their changes of state. Jobs are represented by entities which are characterized by attributes (kind of product, availability date,...), and their movements through the system produce some changes in the state of the system. The working of the system is defined by processes that indicate the sequences of operation followed by the entities. These processes are described by pre-defined functional blocks; each block, which generally corresponds to an operation in the real system, is characterized by a 'change of state' algorithm and can make calls to a library of pre-defined procedures or a user procedure that describes the decision rules. The state of the system is characterized by the global variables (SIMAN and user) and the attributes of the entities.

SIMAN distinguishes:

- The *model*, which is constructed by the assembly of functional blocks like: creation of entities, assignment of variables and attributes, queue management, resource management, branching, delays, synchronization, ...
- The *experiment frame*, a set of experiment parameters, called elements, that describe the conditions under which the simulation must be conducted: facilities (machines, operators), and their states (capacity, speed, ...), jobs characteristics (attributes, routings), queues, variables, and statistics to be collected.

The main functional blocks (or modules) used to define the simulation sub-models are the following:

- **STATION** defines a station corresponding to a physical (for instance a workshop) or logical location;
- **QUEUE** depicts the queue where entities (for example jobs to be processed) are waiting for resources and specifies a ranking rule (FIFO, LIFO, SPT,...) for the queue;
- **SEIZE** allocates units of one or more resources (for example a machine) to an entity when they are available simultaneously;
- **DELAY** delays an entity for a specified amount of time (for example the processing time);
- **RELEASE** is used to release units of resources that have previously been seized by an entity;
- **ROUTE** transfers an entity to a specified station (for example the following workshop);
- **ASSIGN** module allows the assignment of a value to a user-defined variable, an entity attribute, or a resource state; for example the sequence dependent setup-time may be assigned to an entity attribute).

An illustration of the specification of two simple object classes specified with ARENA is given in Figure 4-F:

- An *identical parallel resource* has been modeled by six blocks and the arcs that connect the blocks together. An entity (a job) which arrives from a previous object, enters the block **STATION**. The entity is then placed in a queue (**QUEUE**) where it waits for a resource (a machine). When the resource is available, the entity seizes it (**SEIZE**) and keeps it for an amount of time corresponding to the processing time (**DELAY**) before making it available again (**RELEASE**) for an other entity. The block **ROUTE** allows the entity to leave this object and to be routed to another object.

- An *identical parallel resource with non-sequence dependent machine removal time* is modeled likewise, but its particularity is found after the processing operation (*DELAY*). At that point, a block *DUPLICATE* creates a replica of the current entity which allows the release of the machine after a time corresponding to the removal time; the replica is finally destroyed by a block *DISPOSE*. The real entity is routed to the next object.

Figure 4-F: Specification of simple object classes using ARENA.

With Petri net specifications, object attributes are modeled by tokens in places, by transition firing rules (colored Petri nets), by times associated with places (timed Petri nets), by colored priorities associated with transitions, and by a number of component objects that make up the compound objects. The execution of a service is equal to the execution of the Petri net. With ARENA specifications, attributes are modeled in the experiment frame by the elements *ATTRIBUTES*, *QUEUES*, *RESOURCES*, *VARIABLES*, *EXPRESSIONS*, ... and the services in the system are modeled by the blocks and their interconnections.

Both structured specifications verify aggregation and inheritance principles: for example, a V-hybrid flow shop resource will be defined by the aggregation of several identical parallel resources; an identical parallel resource with non-sequence dependent machine removal time inherits the properties of an identical parallel resource.

4.2.3.2. Dynamic modeling

We have associated a simulation sub-model to each simple object class; the global dynamic model can then be obtained by the bottom-up integration of sub-models, following the hierarchical data flow diagrams, level by level. It unfolds in three steps:

1. design of simulation models of non recognized resources, and design of the associated heuristics;

2. instantiating of each object, i.e. valuation of the object attributes;
3. unification of simulation models of a same parent object, each child object being represented by a simulation model (message parameters are represented by attributes which are common to the client and server object).

The unification of two child objects is carried out using of a special kind of object, the **synchronization object**. The choice of the synchronization object is done according to the two child objects and to the link(s) that exist between them. The secondary flows of the DFD are also modeled in these synchronization objects.

With Petri net object specifications, a synchronization object is made of one or more transitions and may also include places. The simplest synchronization object is made of a transition to call a server object, which has as inputs, the output places of the client object and as outputs, the input places of the server object. The firing rules of this transition will be defined according to object attributes. Figure 4-G illustrates the unification of two objects belonging to two different classes: an identical parallel resource and an identical parallel resource with non-sequence dependent machine setup & removal times.

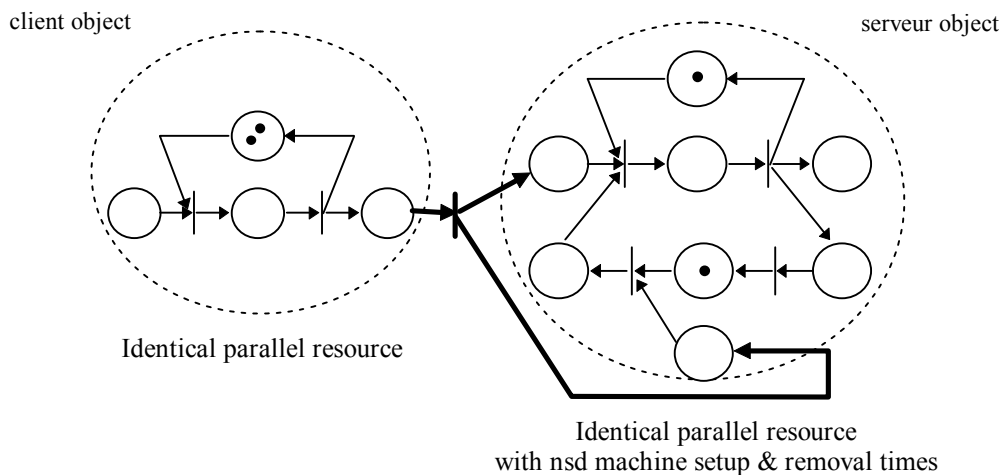


Figure 4-G: Object unification with Petri nets.

Synchronization objects have also been modeled with ARENA. The simplest synchronization object consists of a link between the block *ROUTE* of the first object and the block *STATION* of the second; the modeling of more complex synchronization objects involves one or more blocks, depending on the secondary flow and the characteristics of the object links, like auxiliary resources or minimum time lag. Figure 4-H illustrates the unification of two objects belonging to two different classes: an identical parallel resource and an identical

parallel resource with non-sequence dependent machine removal times, when these objects are linked by minimum time lag constraints.

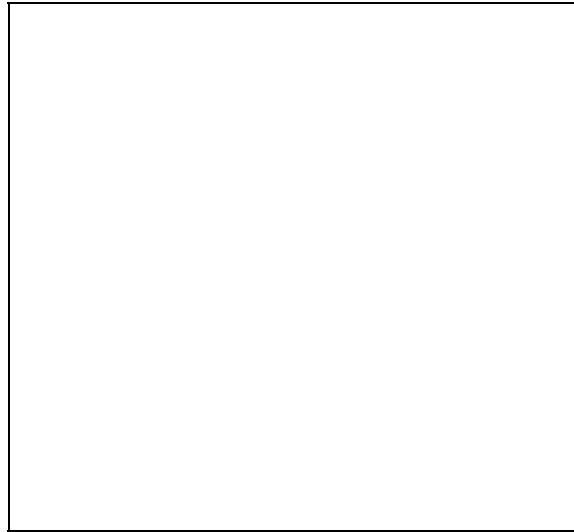


Figure 4-H: Object unification with ARENA.

The non-recognition of certain resources may lead the analyst to create new object classes, according to the level of genericity of the resource characteristics. Model integration can be partial or complete in accordance to simulation interests. Moreover, the simulation model obtained can be enriched by incorporating the features that are not taken into account in the definition of objects (other conditions of synchronization, intervention of operators) and the possible existence of unforeseen events (rate of breakdown, and so on).

4.2.3.3. Selection of scheduling tools

In a similar manner, a set of algorithms has been associated with each object class (simple and complex) in the form of « schedule » services, in order to solve the classes' scheduling problems; for example:

- Several algorithms can be associated with an object of the parallel identical resource class, according to the value of its attribute « number of resources » and to the criterion to minimize:

$$N/I, m(I)=I//Nt: \text{ [Moore; 1968],}$$

$$N/I, m(I) \geq I, I//C: \text{ [Bruno \& al.; 1974],}$$

$$N/I, m(I) \geq I, I//T: \text{ [Dogramaci \& Surkis; 1984].}$$

- Consider an object of the hybrid flow shop resource class characterized by the attribute « number of workshops » equal to V , and the attributes « number of resources » of every parallel identical resource objects that compose it equal to one; different algorithms for the makespan criterion can be associated with this object:

$N/V, m(v)=1/F/C_{max}$: algorithms of Campbell & al. [1970], Nawaz & al. [1983], ...

The bibliographic survey presented in Chapter 3 allows us to associate some resolution tools with the main simple object classes like the parallel identical resource, the parallel identical resource with nsd machine setup and removal time (see section 3.2 for both) or the hybrid flow shop resource with only one resource per workshop (see section 3.3). On the other hand, for complex object classes like hybrid flow shop resource in the general case or with additive characteristics, only a few scheduling tools exist (see section 3.4).

4.2.4. IMPLEMENTATION OF OBJECTS

This last step corresponds to the integration of the selected scheduling tools with the dynamic model; the simulation process will allow us:

1. to validate the scheduling decisions (sequencing and assignment) generated by the heuristics, and
2. to evaluate the consistency, the efficiency and the robustness of the various scheduling tools faced with unexpected events (breakdowns, rupture of raw materials, ...) that can be modeled in the simulation framework.

4.3. SUMMARY AND CONCLUSION

None of the analysis and design methodologies, in use since the 70s is really appropriate for the analysis of a production system in order to identify and specify its scheduling problems, and for the design of its production activity control tools.

In this chapter, we have proposed a new method for production system analysis in order to identify and specify scheduling problems: OASISS. This method is based on the identification of the resources and the organization of the system (see §4.2.1), and follows an object-oriented approach (see §4.2.2). The object classes have been defined according to the classification scheme of scheduling problems. This method allows the selection of appropriate scheduling tools, and the generation of simulation models of the production system (see §4.2.3) through the association of a set of scheduling tools and a simulation model to each

object class. The implementation of scheduling tools in the simulation model enables material managers to validate tool robustness in realistic situations, and to verify the quality of the solutions (§4.2.4). More information can be found in [Guinet & Botta; 1995-a], and in [Botta & *al.*; 1996] for a general presentation of the methodology, and in [Levecq & *al.*; 1997] for the object specification and unification process with ARENA.

This methodology was first applied in textile industry in the INOSETA case study [Botta; 1994, 1995-a, -b]. It was then validated by three other real case studies from textile and carpet industries: an application in a DMC printing plant [Boffet; 1996], in the LIMONY finishing plant [Arod; 1996], and in the LOUIS DE POORTERE carpet manufacturing plant [Dubois & Michels; 1996].

Our approach is not dedicated to a specific kind of scheduling problem, nor to a special type of manufacturing systems. Moreover, the process diversity of textile industry leads us to believe that this methodology could be easily applied to other process industries. We hope that it will be helpful for analysts when designing production control tools and also for material managers in their decision making.

The next chapter presents the application of this method to the INOSETA case study.

CHAPTER 5

APPLICATION IN THE TEXTILE INDUSTRY

5.1. INTRODUCTION	74
5.2. DESCRIPTION & ANALYSIS OF THE PRODUCTION SYSTEM	76
5.2.1. The Grey Manufacturing Process (P1)	76
5.2.1.1. The weft twisting process	78
5.2.1.2. The warping process	79
5.2.1.3. The weaving process	80
5.2.1.4. Grey manufacturing process characteristics and constraints	80
5.2.2. The Finishing Process (P2)	82
5.3. IDENTIFICATION OF SCHEDULING PROBLEMS	83
5.4. SIMULATION MODELS	86
5.5. CONCLUSION	87

5. APPLICATION IN THE TEXTILE INDUSTRY⁽¹⁾

This chapter presents an application of the OASISS methodology to the INOSETA textile company. We have analyzed and modeled its production system in order to identify the correct scheduling problems. The description of the system and the top-down analysis process are shown in section 5.2; the bottom-up recognition of resources and the problems identified are described in section 5.3. Following the previous methodology, dynamic models of the production systems were designed; they are presented in section 5.4. They will enable the material manager to assess his/her scheduling decisions, and to appreciate the consequences of a planning oversight. First of all, we introduce the reader in section 5.1 to INOSETA and its environment.

5.1. INTRODUCTION

Created in 1991, INOSETA produces and merchandises 100% polyester fabrics for blouses, lingerie, trousers, dresses, and jackets on the European market. Its present turnover is of the order of one hundred million francs. This company supplies polyester silk for the fashion market, a market nowadays served mostly by the imports from Japan, Asian South-East, but also from Europe. This type of product which seeks to imitate the texture of a natural fabric like silk, is destined for the feminine clothing and high-quality lingerie markets. The first objective of INOSETA is to respond to customer needs: reliability, short delivery delays and optimal service quality.

INOSETA's production plant, located in La Mure (Isère) employs 180 persons and produces 500,000 meters of grey textiles and 400,000 meters of finished textiles per month. The factory has four shops (twisting, warping/sizing, weaving and finishing) and a laboratory for coloration studies. Its raw materials (yarns) come principally from Japan.

Production control problems principally concern detailed scheduling: finished goods are to be delivered to customers within two or three weeks, even though manufacturing lead times range from one to two months and supplying lead times vary from three to five months. The

⁽¹⁾ an English - French glossary of textile terms is provided in appendix A.3.

material manager is interested in optimizing the plant performance and customer service. To achieve these goals, he/she needs not only suitable scheduling tools, but also a way to assess his/her scheduling decisions, and to appreciate the consequences of a planning oversight.

From this overview, we have retained that manufacturing and supplying lead times are very long, that manufacturing workshops are very dependent on one another and that the optimization of any one of them cannot be achieved without optimizing the rest. Due to these characteristics, classic approaches like MRP (Figure 5-A) are not suitable.

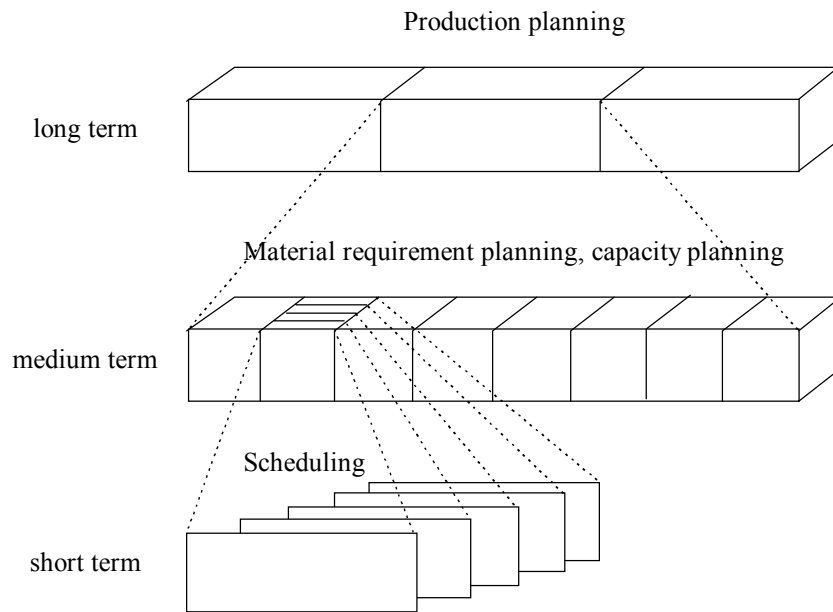


Figure 5-A : 3-level production management approach.

We propose a two-level approach (Figure 5-B) consisting of production planning and multi-workshop scheduling. The material requirement and capacity planning characteristics will be included in the scheduling level (bill of materials, economical order quantity, ...). This 2-level approach allows us to make up for two disadvantages of classic approaches like MRP.

- Decisions that are made at a level k (for instance medium term) involve constraints on lower level decision making (short term), but they are also constrained by incompatibilities from these lower levels. the smaller the level number is, the fewer the impossibilities between decision makings.
- MRP-like approaches work on a different spatial horizons: global problems are studied in the long term while local problems are treated in the short term. Our 2-level approach has the same spatial horizon for long and medium & short terms.

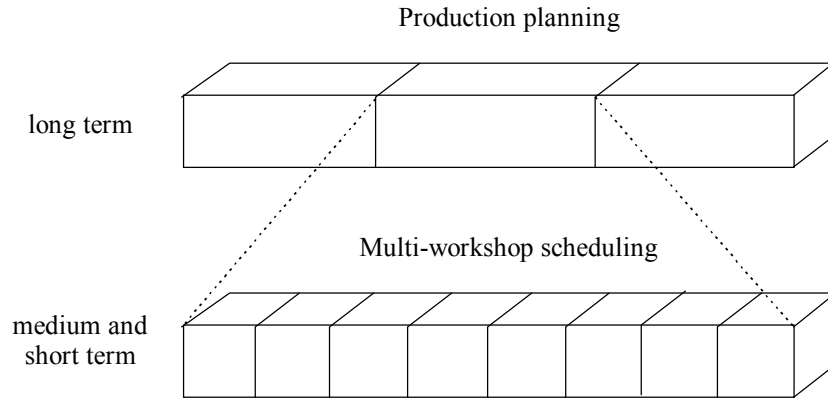


Figure 5-B: 2-level production management approach.

Given this environment, manufacturing has been divided into two processes: grey manufacturing (twisting, warping/sizing, and weaving) and finishing, respectively P1 and P2; the former is stock-management oriented while the latter is order-management oriented. They both consist of several continuous or discontinuous processes in series, with different processing units.

5.2. DESCRIPTION & ANALYSIS OF THE PRODUCTION SYSTEM

This section describes the manufacturing processes P1 and P2, and details the top-down analysis of their resources. This work was carried out during the first year of this thesis; more details can be found in earlier technical reports [Botta; 1994-a, 1995-a].

5.2.1. THE GREY MANUFACTURING PROCESS (P1)

Due to its duration (from three to five weeks) and to supply constraints (at least three months), the workshops planning of the grey manufacturing process is made from a production plan based on medium- and long-term sale forecasts. The planning/scheduling horizon is about six months.

Three elements define the environment of the grey production system: a stock of flat yarns and a planning department as origins, and a stock of grey fabrics as destination. The physical production system is made up of three different processes: weft twisting, warping, and weaving which are detailed in the following sub-sections. The series of the different operations is given in Figure 5-C, and the ‘resourcegram’ which results from the top-down analysis of this process is depicted in Figure 5-D.

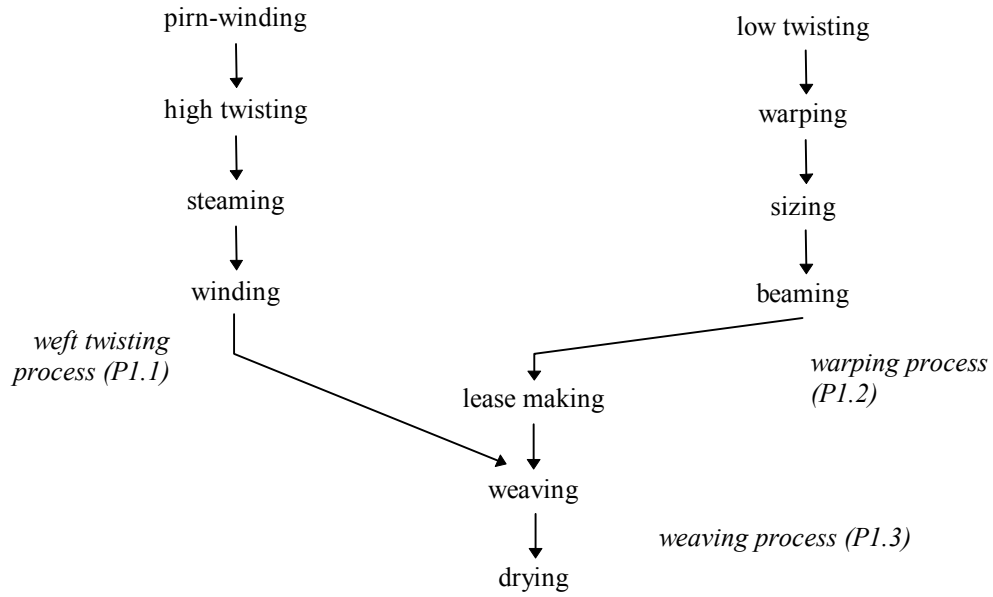


Figure 5-C: Grey manufacturing process (P1).

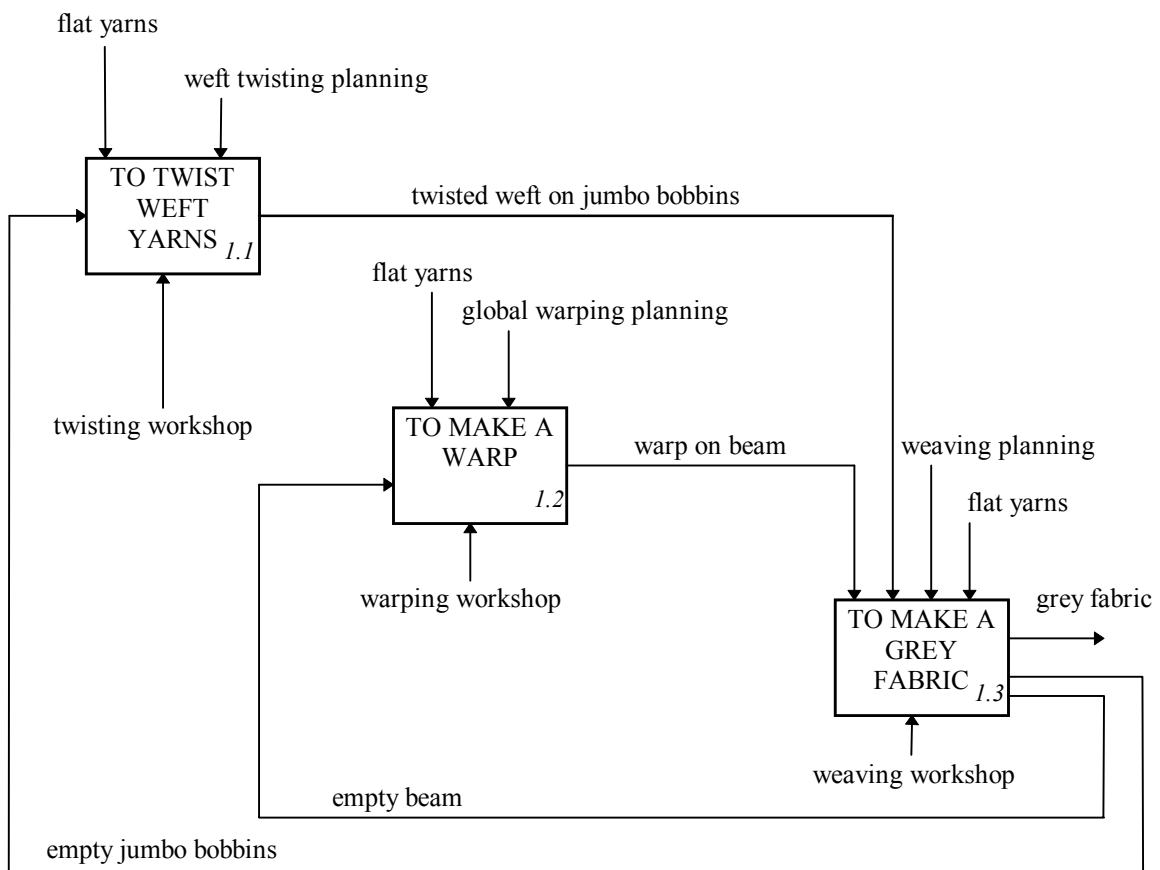


Figure 5-D: Resourcegram of the grey manufacturing process (P1).

5.2.1.1. The weft twisting process

A majority of grey materials are woven with high twisted weft yarns. The weft twisting process (P1.1) consists of four operations: pirn-winding, high twisting, steaming, and winding. The ‘resourcegram’ obtained is yielded in Figure 5-E.

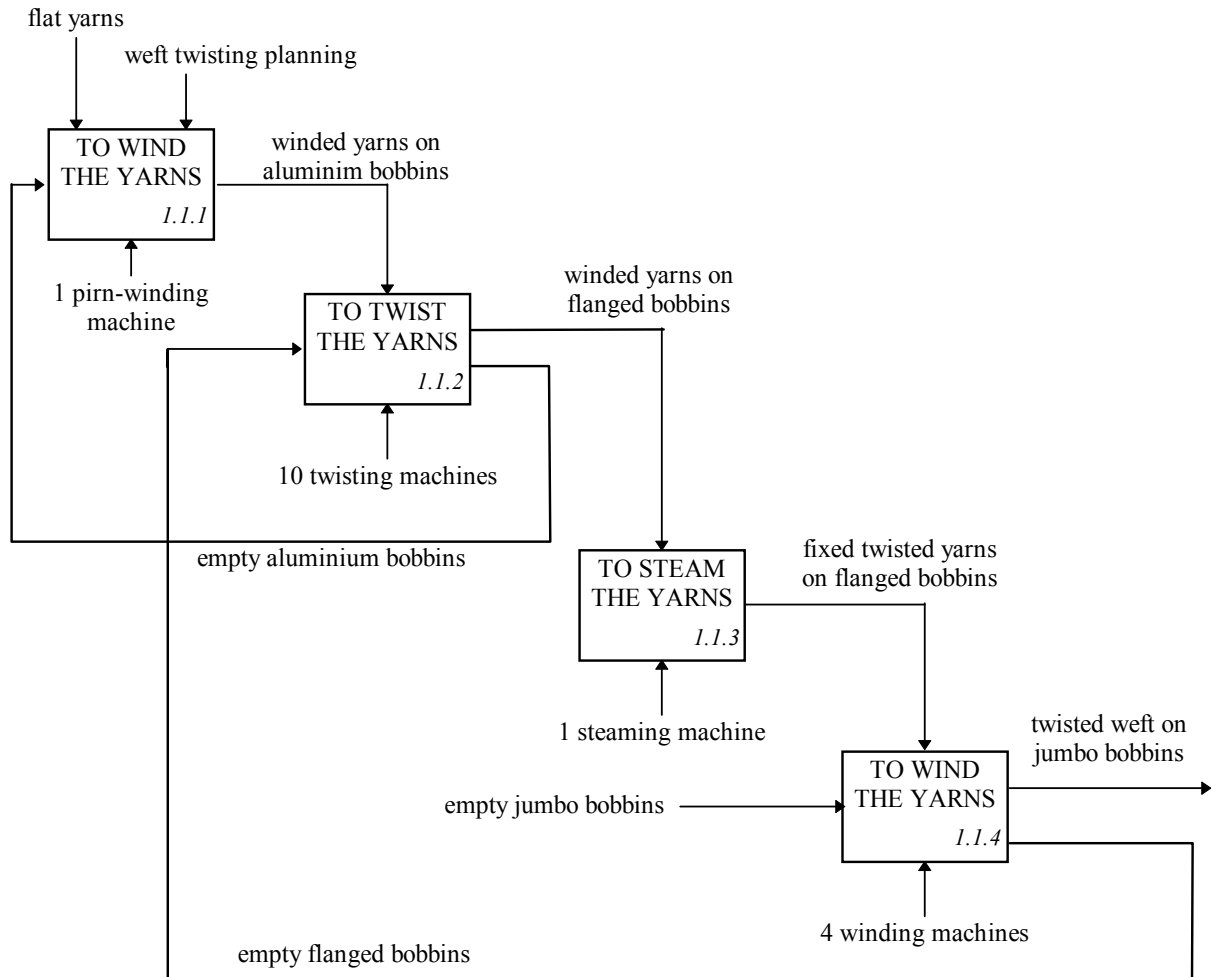


Figure 5-E: Resourcegram of weft twisting process (P1.1).

The pirn-winding operation which takes place on a single machine, transfer yarns from the bobbins on which they are supplied to aluminum bobbins, each weighted around one kilogram (the input bobbins weigh between two and four kilograms). This machine can wind up to 160 aluminum bobbins in parallel, two, three or four times per batch.

The high twisting operation consists of twisting the yarn up to 2000 to 2800 rounds per meter. This can be done on one of the ten identical high twisting machines; each machine can twist up to 256 flanged bobbins in parallel.

Flanged bobbins of twisted yarns are set in an autoclave (up to 256 per batch) to fix the twisting; this is called the steaming operation.

Before they can be used on weaving looms, yarns must be wound again on jumbo bobbins (5 flanged bobbins make 1 jumbo bobbin). This operation is completed on one of the four identical winding machines; each machine can wind twelve jumbo bobbins in parallel.

5.2.1.2. The warping process

For the warping process (P1.2), four operations are performed in sequence: low twisting, warping, sizing, and beaming. The 'resourcegram' associated with this process is given in Figure 5-F.

The low twisting operation consists in twisting the yarn up to 800 rounds per meter. This can be done on one of the twenty four identical low twisting machines; each can twist up to 160 bobbins in parallel.

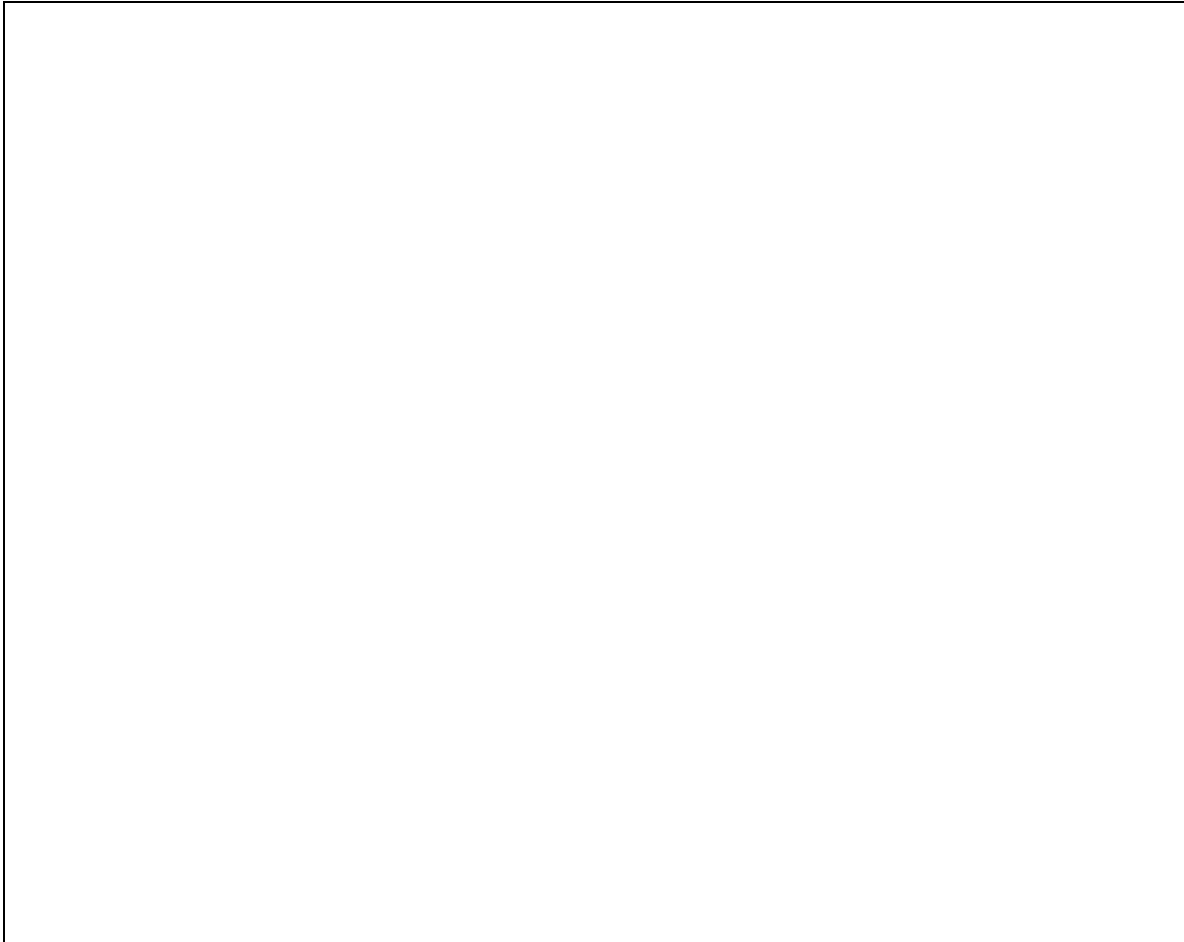


Figure 5-F: Resourcegram of the warping process (P1.2).

The warping operation consists of the parallel rolling up of thousands of flat or low twisted yarns which are set out on a creel, on a beam called primary beam. This is done on the single warping machine, and one creel unit (up to 1200 bobbins) gives rise to between two and eight primary beams.

During the sizing operation chemical coatings are applied to yarns to give them consistency and added strength. This can be done on one of the two identical sizing machines. The input for a sizing machine is a primary beam, and the output is from five to fourteen sectional beams.

The beaming operation consists of beaming together the five to fourteen sectional beams to make from two to eight warps, on loom beams. This is done on the single beaming machine.

5.2.1.3. The weaving process

Leases are made on warps and a drawing in / sleying operation may be performed to prepare warps for the weaving machines. The weaving operation takes place on one of the one hundred and twelve water jet looms; it consists in perpendicularly interlacing weft and warp yarns. Grey pieces are dried and kept in stock.

5.2.1.4. Grey manufacturing process characteristics and constraints

The resourcegram only depicts the organization and process flows (arrows C and O) and secondary flows like auxiliary resources (arrows I). Other information concerning resource characteristics (setup, removal,...) or process characteristics between resources (transportation time, limited buffer storage, maximum time lags,...) are listed in the following sub-section.

A particularity of the grey manufacturing process is the numerous size of lots (job units). They are specified during the design of the products. Job units and the average duration of each operation are given in Table 5-A. This characteristic has been modeled with job-precedence constraints (see section 5.3).

Some operations are subject to setup or removal times which can be sequence-dependent: for low twisting and high twisting operations, there is a setup time on the machine (around eight hours) between two successive batches when the yarns are different; for the sizing operation, the machine must be cleaned after each job has been processed; for the

weaving operation, a sequence dependent machine & job setup time is required on the looms (from two to ten hours), and the drawing in / sleying operation can be seen as a sequence-dependent job-setup time.

operation	job unit	average duration
pirn winding	≤ 256 bobbins	11 hours
high twisting	-	192 hours
steaming	-	2 hours
winding	5 flanged bobbins	4 hours
low twisting	160 bobbins	96 hours
warping	1 creel unit	38 hours
sizing	1 primary beam	21 hours
beaming	-	9 hours
lease making	1 warp	2 hours
drawing in/sleying	-	23 hours
weaving	-	468 hours

Table 5-A: Job units and average duration for grey manufacturing process.

Minimum waiting times may exist between some operations: after steaming, jobs must wait for at least one hour for relaxing before processing the next operation; similarly after low twisting, jobs must wait at least forty-eight hours, and after sizing jobs must wait at least twenty-four hours. The waiting time required after twisting takes place in a hot room where the bobbins are stored to fix the twist; this room can only store two creel units.

The grey manufacturing process is also characterized by many limited buffer storage capacities: the storage capacities between pirn-winding and high twisting, high twisting and steaming, steaming and winding are limited to four, four and twenty-five job units respectively.

There are other storage constraints due to the limited number of auxiliary resources which are needed for several consecutive operations: flanged bobbins from the beginning of the high twisting operation to the end of the winding operation, jumbo bobbins from the beginning of winding to the end of weaving, primary beams from the beginning of warping to the end of sizing, and warp beams from the beginning of beaming to the end of weaving.

The grey manufacturing process scheduling objectives are to optimize resource utilization and to minimize intermediate inventories. The former objective squares with makespan criterion, the latter with the average completion time criterion.

5.2.2. THE FINISHING PROCESS (P2)

The finishing process consists of two processes, preparation and dyeing, which share common resources. Workshop planning of the preparation process is derived from medium-dated sale forecasts; dyeing process planning is derived from customer orders. The planning horizon is around two months. The series of the different operations is given in Figure 5-G (dotted arrows represent preparation process flows, full arrows represent dyeing process flows).

In the beginning of both processes, pieces are cut or sewn in order to make fabric pieces of a desired length, determined by the relaxing and dyeing machine capacities which are respectively around 1000 meters and 400 meters. In the dyeing process, this is done on a single join-clothing machine. In the preparation process, both join-clothing and desizing operations are performed on one of the two desizing machines; pieces are desized to be able to absorb dyestuffs and chemicals during next operations.

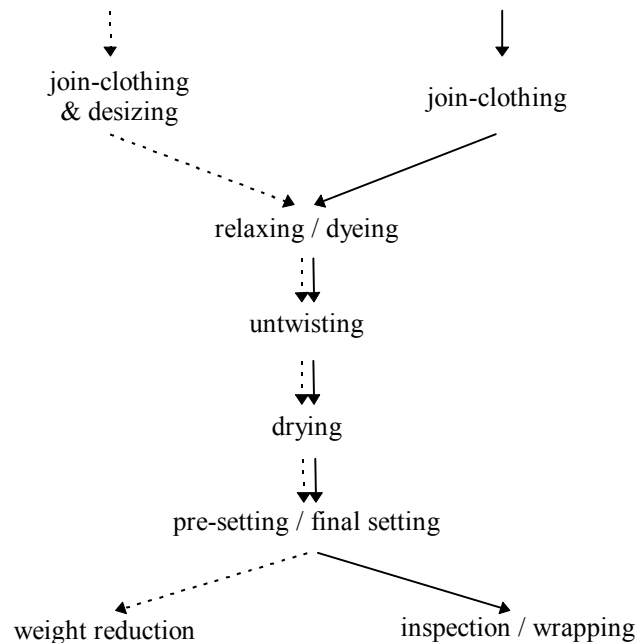


Figure 5-G: Finishing manufacturing process (P2).

Depending on the process followed, pieces are relaxed or dyed in one of the fourteen jets, each of which have different capacities. Then, fabrics are untwisted on the untwisting machine, and dried without tension, on the drying machine. The next operation, pre-setting, or final setting, consists of stabilizing the warp and weft perpendicularly; this is done on one of two identical machines.

The last operation of the preparation process consists of reducing the weight of the fabric. Prepared fabrics are kept in stock until they are loaded in dyeing section. About dyeing process, the last operation consists of inspecting and wrapping the fabrics.

Some operations are subject to setup or cleaning times depending on the sequence of fabrics: for relaxing / dyeing operations, a two hours of machine cleaning time is required to process a light-colored fabric after a darker one. In the same way, sequence-dependent setup times exist for the drying operation, the pre-setting / final setting operation and the weight reduction operation.

The processing of fabrics is subject to maximum waiting times between some consecutive operations: after desizing or untwisting, wet fabrics must not wait more than twelve hours before processing the next operation, otherwise defaults may occur.

The minimum time required between two consecutive operations can be important: after continuous processes like desizing or weight reduction, a job (a fabric) must wait until the other jobs from the same roller are completed. This duration can be equal to three times the operation processing time.

The finishing process scheduling objective is to minimize maximum and average tardiness of jobs.

5.3. IDENTIFICATION OF SCHEDULING PROBLEMS

The criterion which leads the decomposition of the system is the resource. As a matter of fact, a machine or a set of machines is fitted to an object class and after the object instantiates, the links between objects model the organizational characteristics. The bottom-up recognition process begins by acknowledgment of each lowest-level resource as belonging to a pre-defined object class.

For instance, given the resourcegrams of the grey manufacturing process and the previous specifications for the sub-system P1.1 (Figure 5-E), we recognize identical parallel resources objects (1.1.1 pirn-winding, 1.1.3 steaming, 1.1.4 winding) and identical parallel resources with sd machine setup times objects (1.1.2 high twisting); for the sub-system P1.2 (Figure 5-F): identical parallel resources objects (1.2.2 warping, 1.2.4 beaming), identical parallel resource with sd machine setup times objects (1.2.1 low twisting) and identical parallel resource with nsd machine removal times objects (1.2.3 sizing).

When each lowest-level resource is recognized, the recognition process allows to identify the upper-level resources according to the child diagram and complementary specifications. At the end of this recognition step, we recognize the grey manufacturing process as a linear organization of objects, each object belonging to one of the classes defined in §4.2.2.

For this problem (P1), precedence constraints have been used in order to model the different manufacturing units: a batch of 256 flanged bobbins, jumbo bobbin, a batch of 160 bobbins, the creel unit, the primary beam, the warp... A manufacturing order i will be decomposed into six kinds of jobs: $i1$ for the first three weft twisting operations, $i2$ for the winding operation, $i3$ for the low twisting operation, $i4$ for the warping operation, $i5$ for the sizing and beaming operations, and $i6$ for the weaving operations. Each job will be processed by one machine in each workshop, and some processing times will be equal to zero. The precedence constraints are: $i1 \ll i2$, $i3 \ll i4$, $i4 \ll i5$, $i5 \ll i6$, and $i2 \ll i6$, i.e. four plus three precedence levels, and arbitrary precedence constraints (combining of in-tree and out-tree). Moreover, on the scheduling horizon, it involves around 100.000 jobs to schedule.

The combinational characteristic of this scheduling problem allows us to break it down into two sub-problems:

- the first, called WWP, consists of scheduling warping (P1.2) and weaving (P1.3) processes with makespan criterion and mean completion time as a second performance measure according to the company strategy. This modeling involves the scheduling of around 2000 jobs on the six-month horizon. The beginning date of the weaving operation will give the corresponding weft twisting due-date.
- The second, called WTP, consists of scheduling weft twisting process (P1.1) with maximum tardiness and maximum earliness criteria, i.e. maximum absolute lateness (see §2.2.2) to respect prescribed deadlines, and in order to limit the number of jumbo bobbin supports used. This model involves the scheduling of around 1000 jobs on the six-month horizon.

These two dependent problems will be solved sequentially: first WWP then WTP. The results of the first problem (WWP) are taken into account for the definition of due-dates for the second problem (WTP). If one of the due dates is not respected, the results of the second problem WTP (the calculated completion times) are used to define release dates for the first problem (WWP), which is re-scheduled.

The finishing scheduling problem, called FP, concerns the whole finishing process (P2), and precedence constraints have been introduced in order to reduce the job shop organization into a flow shop organization of resources (machine or set of machines): the succession of the preparation and dyeing processes will be modeled by two kinds of job i and j , the former for the preparation process and the latter for the dyeing process, with $i \ll j$. Due to the process constraints, the length of a fabric that can be processed on a relaxing/dyeing machine depends on the operation: around 1000 meters for relaxing and around 400 meters for dyeing. According to our model, a «preparation» fabric will be divided into two to five «dyeing» fabrics.

Avoiding splitting the finishing process into two different sub-processes (preparation and dyeing) enables us to gain more flexibility in regards to the common resource assignment (no common resource will be dedicated). This problem encompasses around 1500 jobs to be schedule on a 1- or 2- month planning horizon.

According to this analysis, we have a set of workshops, each workshop including either one machine or a set of identical parallel machines. The general objects WWP, WTP and FP belong to an object class from the family « hybrid flow shop resource ». The incompatibilities between jobs and machines are taken into account by setting the machine setup times at their maximum value; an example of these incompatibilities is that a 400 meter length grey fabric can not be processed on a machine whose capacity is 50 meters.

In summary, in the three cases , the problem is to schedule N jobs (from 1000 to 3000) on V serial workshops (between 4 to 9) consisting of identical parallel machines $M(v)$. Each job has its own due-date (except for WWP) and its own release date. Job processing are subject to machine setup and/or removal times which are either sequence dependent, or non sequence dependent. These times may immobilize only the machine, only the job, or both. Job processing can also depend on the buffer limited storage capacity, or on a limited number of supports. Some operations depend on minimum and/or maximum time lags.

According to the classification scheme presented in §2.2.3, the three previously identified scheduling problems can be written as follows:

WWP: $N/V, m(v) \geq 1, I/F, H_v, r(i) \geq 0, a(i,v), R_{nsd}, S_{sd}, SI_{sd}, B_{v,v+\beta}, j\text{-prec}, out\text{-tree}/C_{max} (+ \bar{C})$

WTP: $N/V, m(v) \geq 1, I/F, H_v, r(i) \geq 0, R_{nsd}, S_{sd}, b_{w,w+l}, B_{v,v+\beta}/ET_{max}$

FP: $N/V, m(v) \geq 1, I/F, H_v, r(i) \geq 0, a(i,v), A(i,v), S_{sd}, j\text{-prec}, out\text{-tree}/L_{max} (+ \bar{T})$

The identification and the modeling of these problems are detailed in [Botta; 1995-b].

5.4. SIMULATION MODELS

We have specified object classes with the aim of obtaining a dynamic model of the production system. To illustrate the use of these object classes, the structured specification of the object P1.2 (warping process) based on Petri nets, is given in Figure 5-H.

Each child object from object P1.2 has been identified as belonging to a pre-designed class. The integration step has been done by the unification of dynamic sub-models associated with each object. In Figure 5-H, the three main child objects that make P1.2 are identified by their transitions : P1.2.2 (t_1 and t_2), P1.2.3 (t_4 , t_5 , and t_6) and P1.2.4 (t_8 and t_9). Transitions t_3 and t_7 identify the two synchronization objects used for unification. The limited storage constraint due to the number of primary beams is modeled by the arcs between t_5 and t_1 .

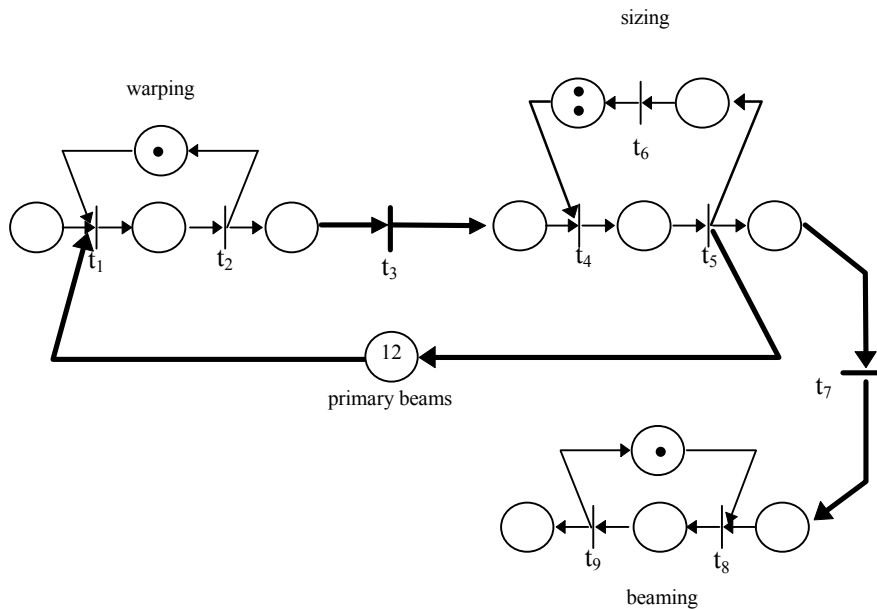


Figure 5-H: Specification of the warping process with Petri nets.

To complete this example, an aggregate structured specification of the warping process using ARENA is given in Figure 5-I.

Figure 5-1: Specification of the warping process using ARENA.

5.5. CONCLUSION

The three problems WWP, WTP and FP belong to «complex» object classes; as a matter of fact, our research focused principally on the design of scheduling tools which take the constraints identified in the production system of INOSETA company in account. This work is presented in the subsequent chapter. Dynamic models have been completely designed for the weft twisting process (P1.1) and for the finishing process (P2) [Gregor; 1996]. They will enable us to perform the fourth step of the OASISS methodology, i.e. to validate the scheduling decisions.

CHAPTER 6

NEW DEVELOPMENTS IN HYBRID FLOW SHOP SCHEDULING

6.1. PRINCIPLES	90
6.2. THE SEQUENCING STEP: PREVIOUS RESULTS	91
6.2.1. $N/2, m_1=m_2=1/F/C_{max}$: Johnson [1954]	92
6.2.2. $N/V, m_v=1/F/C_{max}$: Campbell, Dudek & Smith [1970]	92
6.2.3. $N/V, m_v=1/F, a_{i,v}/C_{max}$: Szwarc [1983]	92
6.2.4. $N/V, m_v=1/F/C_{max}$: Nawaz, Ensore & Ham [1983]	93
6.2.5. $N/V, m_v=1/F/L_{max}$: Townsend [1977]	93
6.3. THE SEQUENCING STEP: NEW DEVELOPMENTS	94
6.3.1. Integration of various workshop calendars	94
6.3.2. Integration of non-sequence dependent setup and removal times	95
6.3.3. Integration of job-precedence constraints and minimum time lags	95
6.3.4. From makespan to maximum lateness	97
6.3.5. A fast heuristic for numerous job-precedence constraints: VBA	98
6.4. THE ASSIGNMENT STEP	99
6.5. SUMMARY: FIVE HEURISTICS	100
6.6. LOWER BOUNDS AND ANOTHER HEURISTIC: VBB	103
6.7. CONCLUSION	104

6. NEW DEVELOPMENTS IN HYBRID FLOW SHOP SCHEDULING

Following the methodology presented in Chapter 4, a multi-model approach has been used to solve scheduling problems in hybrid flow shop environments: it associates discrete event simulation with classic scheduling theory to evaluate and validate production schedules.

In this chapter we focus on the scheduling aspects. We present heuristic algorithms to schedule hybrid flow shops with identical machines under the following constraints: job-precedence constraints, minimum time lags, various workshop calendars, release dates, and non-sequence dependent setup and removal times.

$$N/V, m_v \geq 1, I/F, j\text{-prec}, \text{out-tree}, a_{i,v}, H_v, r(i) \geq 1, S_{nsd}, R_{nsd} / C_{max} \text{ or } L_{max}$$

Other previously specified constraints can also be taken into account in the calculation of job completion times. These algorithms have been designed to minimize either makespan or maximum lateness.

The approach adopted in order to design our resolution tools for the hybrid flow shop scheduling problem of interest is presented in section 6.1. These tools solve the sequencing and the assignment problems in sequence. For the sequencing problem, previous studies have been adapted and improved; methods from which we took inspiration are presented in section 6.2, and new developments are given in section 6.3. The assignment problem is studied in section 6.4. A summary of the global scheduling scheme is presented in section 6.5. Lower bounds on the optimal solution have been designed to assess the quality of these algorithms for both C_{max} and L_{max} criteria; they are presented in section 6.6. Design of lower bounds for maximum lateness lead us to develop another heuristic for this criterion.

6.1. PRINCIPLES

Given the complexity of the problem and the computational time requirements, a set of heuristic algorithms have been developed to solve it; as we have described [Guinet & al.; 1995] this approach seems to offer the most promising results when faced with real size problems.

As we have noticed before, scheduling in hybrid flow shop environments involves solving two problems: a sequencing problem and an assignment problem. Billaut & al.;

[1995] outlines the importance of the assignment problem on scheduling in industrial environments.

We have chosen to consider the sequencing and assignment characteristics in sequence, because this approach made it easier to take additional restrictions into account. Moreover, Guinet [1996] has shown that the sequential approach gives better results when the machines are identical.

For the given problem: N/V , $m_v \geq 1$, I/F , j -prec, out -tree, $a_{i,v}$, H_v , $r(i) \geq 1$, S_{nsd} , R_{nsd}/C_{max} or L_{max} , our approach consists of two steps:

- 1) Sequencing jobs on V machines, applying flow shop heuristics for the problem:

$$N/V, m_v = I/F, H_v, a_{i,v}, S_{nsd}, R_{nsd}, j\text{-prec}, out\text{-tree}/C_{max} \text{ or } L_{max}$$

in which the processing times have been divided by the number of machines in each workshop, and

- 2) Assigning jobs to the machines in each stage, using a list algorithm.

For the sequencing step, we have considerably adapted and improved previous research to take into account both kinds of objectives and additional constraints. We have also developed other fast algorithms especially suitable for problems subject to job-precedence constraints. These last constraints have not really been studied in the literature.

For the assignment step, we apply simple priority rules such as: assigning jobs to the machine in a workshop in order to minimize their completion times in the workshop; because the machines are identical, this rule is equivalent to assigning the jobs to the first available machine in each workshop (FAM rule). This step can take on all constraints that have been modeled in the calculation of job completion times in §2.3.2, and especially job-precedence constraints, minimum time lags and non-sequence dependent setup and removal times.

6.2. THE SEQUENCING STEP: PREVIOUS RESULTS

After many experiments, four algorithms have been selected according to their ability for solving flow shop problems minimizing makespan or maximum lateness: The Campbell, Dudek & Smith [1970], and the Nawaz, Ensore & Ham [1983] algorithms for the problem N/V , $m_v = I/F/C_{max}$, the Szwarc method [1983] for the problem N/V , $m_v = I/F$, $a_{i,v}/C_{max}$, and Townsend's algorithm [1977] for N/V , $m_v = I/F/L_{max}$. They are very efficient, and flexible enough to take into account many additional constraints [Taillard; 1990], [Leisten; 1990],

[Proust & *al.*; 1991], [Liou & Smith; 1992]. These four methods are presented in the subsequent paragraphs. As Campbell and Szwarc use Johnson's algorithm, we first of all introduce the principle of this algorithm.

6.2.1. $N/2, m_1=m_2=1/F/C_{max}$: JOHNSON [1954]

In 1954, Johnson proposed an optimal algorithm to schedule jobs on two serial machines A and B in order to minimize the makespan. He proved that a job i must be schedule before a job j if $MIN[p(i, A), p(j, B)] \leq MIN[p(j, A), p(i, B)]$.

An algorithm, whose complexity is $o(N.logN)$ may be the following:

1. Define two sets U and V so that

$$U = \{i / p(i,A) < p(i,B)\}, \text{ and } V = \{i / p(i,A) \geq p(i,B)\}$$

2. Sort U jobs in non-descending order of their processing time on machine A , into the sub-sequence U_s .
3. Sort V jobs in descending order of their processing time on machine B , into the sub-sequence V_s .
4. Chain the two sub-sequences $U_s \oplus V_s$ in order to obtain the optimal schedule.

6.2.2. $N/V, m_v=1/F/C_{max}$: CAMPBELL, DUDEK & SMITH [1970]

Campbell's heuristic, which is known to be very robust, is based on Johnson's rule; it solves $V-1$ two-fictitious machine problems, and computes the makespan for each schedule obtained; it retains the schedule among the $V-1$ solutions, with the smallest makespan to solve the V -machine problem. The processing times of jobs for the k^{th} problem on the two fictitious machines A and B are:

$$p(i, A) = \sum_{v=1}^k p(i, v) \quad p(i, B) = \sum_{v=V-k+1}^V p(i, v) \quad \forall i = 1..N \quad (6.1)$$

The complexity of Campbell's heuristic is within $o(V.N.logN)$.

6.2.3. $N/V, m_v=1/F, a_{i,v}/C_{max}$: SZWARC [1983]

Szwarc uses a similar approach with another definition of the two-fictitious machine processing times to solve the V -machine flow shop problem with minimum time lags. He defines the processing times of jobs for the k^{th} problem on the two fictitious A and B as follows:

$$p(i, A) = \sum_{v=k}^{V-1} [p(i, v) + a(i, v)] \quad p(i, B) = \sum_{v=k+1}^V [p(i, v) + a(i, v-1)] \quad \forall i = 1..N \quad (6.2)$$

The complexity of Szwarc's heuristic is also within $o(V.N.\log N)$.

6.2.4. $N/V, m_v=1/F/C_{max}$: NAWAZ, ENSCORE & HAM [1983]

Nawaz's heuristic progressively constructs the solution to the flow shop problem for makespan criterion. It is based on the hypothesis that greater attention must be given to jobs with large processing times. His algorithm is the following:

- 1) Sort the jobs in descending order of $ind(i)$, with:

$$ind(i) = \sum_{v=1}^V p(i, v) \quad (6.3)$$

- 2) Place the first job in the schedule.
- 3) Repeat: try all possible positions of the following jobs in the partial sequence, and retain the sequence which gives the smallest C_{max} , until all jobs are scheduled.

The complexity of Nawaz's heuristic is within $o(V.N^2)$ [Taillard; 1990].

6.2.5. $N/V, m_v=1/F/L_{max}$: TOWNSEND [1977]

Townsend suggests a branch and bound algorithm to solve the V -machine flow shop problem with the maximum lateness criterion.

At each node, the algorithm explores all the possible branches by adding a job to the partial schedule (called pre-sequence \bar{A}) in the last position and chooses the one that will minimize the maximum lateness of the complete schedule: the choice is made by computing the lower bound $LB(\bar{A})$ of the maximum lateness of \bar{A} as follows:

$$LB(\bar{A}) = \text{MAX}_{s=1..V} [LB_s(\bar{A})] \quad (6.4)$$

$$LB_s(\bar{A}) = \text{MAX}_{j \in A} \left[\text{MAX}_{j_r \notin A} [c(j, V) - d(j)]; \text{MAX}_{r=1..N} \left[c(\bar{A}, s) + \sum_{\substack{q=1 \\ j_q \notin A}}^r p(j_q, s) - d_s(j_r) \right] \right]$$

The first term of $\text{MAX}_{j \in A} [\dots]$ represents the maximum lateness of \square on machine s . The second term corresponds to the greatest lateness on machine s of jobs that are not in \square where:

- $c(\bar{A}, s)$ is the availability date of the machine s , i.e. the completion time of the last job of \bar{A} scheduled on machine s .
- $\sum_{\substack{q=1 \\ j_q \notin \bar{A}}}^r p(j_q, s)$ is the time required to process the jobs j_1, j_2, \dots, j_r on machine s , when these jobs are sorted in non-descending order of $d_s(j_q)$, according to the fact that the EDD rule minimizes the $N/I, m(I)=I/L_{max}$ problem and that machine s defines a single machine.
- $d_s(j_r)$ is the adjusted due-date of j_r on machine s :

$$\begin{aligned} d_s(j_r) &= d(j_r) - \sum_{v=s+1}^V p(j_r, v) \quad \forall s = 1, \dots, V-1 \\ d_V(j_r) &= d(j_r) \end{aligned} \quad (6.5)$$

The complexity of Townsend's heuristic is within $o(V.N^3)$.

6.3. THE SEQUENCING STEP: NEW DEVELOPMENTS

We present here the adaptation of the algorithms listed above to take into account five additional characteristics: different workshop calendars, out-tree job-precedence constraints, minimum time lags, and non-sequence dependent setup and removal times.

For workshop calendar constraints and non-sequence dependent setup and removal times, we used model-changing approaches, respectively defined in §6.3.1 and §6.3.2, whereas job-precedence constraints and minimum time lag characteristics have been integrated in the combinational exploration scheme of previous heuristics, explained in §6.3.3. In §6.3.4, we show how heuristics that minimize makespan can be use to minimize maximum lateness or other lateness criteria, and we propose an similar method for integrating release date constraints. We have also developed a fast heuristic for the case in which job-precedence constraints are numerous (more than two levels of precedence or more than 50% of jobs having a predecessor); this heuristic, which is suitable for both criteria, is presented in §6.3.5.

6.3.1. INTEGRATION OF VARIOUS WORKSHOP CALENDARS

If γ represents the criterion to minimize , the problem $N/V, m_v > 1, I/F, H_v/\gamma$ can be transformed into $N/V, m_v = I/F/\gamma$

For the sequencing step, we have modified job processing times $p(i,v)$ into $p'(i,v)$ to include the various workshop calendars and the number of machines in each workshop:

$$p'(i,v) = p(i,v) \times \frac{H^*}{H(v)} \times \frac{1}{m(v)} \quad \text{where} \quad H^* = \text{MAX}_{v=1..V} [H(v)] \quad (6.6)$$

6.3.2. INTEGRATION OF NON-SEQUENCE DEPENDENT SETUP AND REMOVAL TIMES

The problem $N/V, m_v=1/F, a_{i,v}, S_{nsd}, R_{nsd}/\gamma$ can be transformed into $N/V, m_v=1/F, a_{i,v}/\gamma$.

This transformation can be carried out using a new definition of processing times and minimum time lags. If $p(i,v), a(i,v), S(i,v)$ and $R(i,v)$ are the parameters of a job i on machine v in the first problem, we can define $p'(i,v)$ and $a'(i,v)$ for the second problem as follows:

$$\begin{aligned} p'(i,v) &= S(i,v) + p(i,v) + R(i,v) & \forall i = 1..N, \forall v = 1..V \\ a'(i,v) &= -R(i,v) + a(i,v) - S(i,v+1) & \forall i = 1..N, \forall v = 1..V-1 \\ a'(i,V) &= -R(i,V) + a(i,V) & \forall i = 1..N \end{aligned} \quad (6.7)$$

We observe that the value of the new parameter minimum time lag can be negative, depending on the initial setup and removal time values. An illustration of this transformed scheme is shown in Figure 6-A.

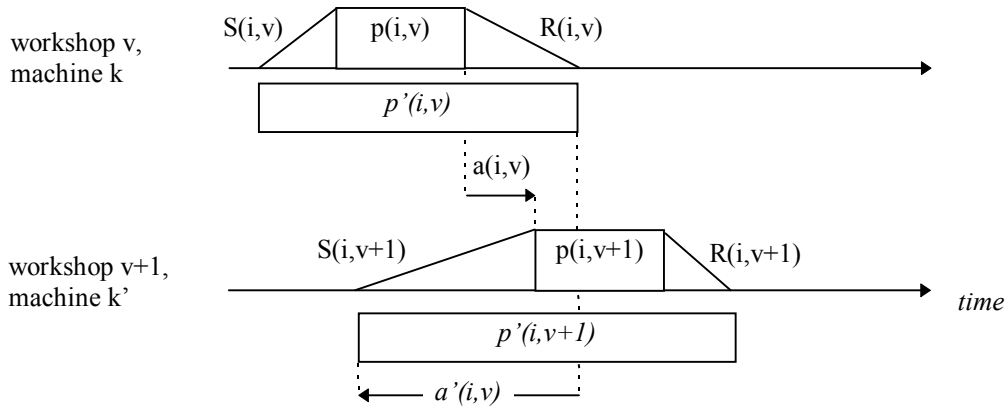


Figure 6-A: Nsd setup and removal times changed into minimum time lags.

6.3.3. INTEGRATION OF JOB-PRECEDENCE CONSTRAINTS AND MINIMUM TIME LAGS

We have applied Szwarc's idea to adapt Campbell's algorithm. Time lags are included in the definition of the two-fictitious machine processing times $p(i,A)$ and $p(i,B)$. For the k^{th} problem, their calculation (6.1) changes into (6.8) as follows:

$$\begin{aligned}
 p(i, A) &= \sum_{v=1}^k [p(i, v) + a(i, v)] \quad \forall i = 1..N \\
 p(i, B) &= \sum_{v=v'-k+1}^{v'} [p(i, v) + a(i, v-1)] \quad \forall i = 1..N
 \end{aligned} \tag{6.8}$$

We propose an adaptation of Johnson's algorithm to include out-tree job-precedence constraints.

$N/2, m_1=m_2=1/F, j\text{-prec}, out\text{-tree}/C_{max}$ can be change into $N/2, m_1=m_2=1/F/C_{max}$.

The transformation is carried out using a new definition of the processing times; $p(i, A)$ and $p(i, B)$ are modified into $p'(i, A)$ and $p'(i, B)$ as shown in (6.9). We have proved that this new definition of processing times ensures that the schedule obtained applying Johnson's algorithm respects the job-precedence constraints (see Appendix A.1). A symmetric definition could allow us to take in-tree job-precedence constraints into account.

$$\begin{aligned}
 p'(i, A) &= p(i, A) + p'(pred(i), A) + p(pred(i), B) \quad \forall i = 1..N \\
 p'(i, B) &= p(i, B) + \sum_{j \in succ(i)} [p(j, A) + p'(j, B)] \quad \forall i = 1..N
 \end{aligned} \tag{6.9}$$

with $succ(i)$ = set of immediate successors of i .

In Nawaz's algorithm, both characteristics have been incorporated into the definition of the saving index $ind(i)$, for which the calculation formulae (6.3) become (6.10); the idea here is to pay greater attention to jobs that have many successors. The sequence obtained by sorting jobs in descending order of $ind(i)$ respects precedence constraints, and the study of job insertions in the partial solution is limited by these constraints.

$$\begin{aligned}
 ind(i) &= \sum_{v=1}^{v'} [p(i, v) + a(i, v)] + \sum_{j \in succ(i)} \sum_{v=1}^{v'} [p(j, v) + a(j, v)] \quad \forall i = 1..N \\
 &\text{with } succ(i) = \text{set of immediate successors of } i.
 \end{aligned} \tag{6.10}$$

In Townsend's approach, time lags and job-precedence constraints are taken into account naturally in the calculation of the lateness bound $LB_s(\bar{A})$. First of all, job completion times are calculated depending on these two characteristics, by applying formulae (2.14) and (2.16), as defined in Chapter 2. Secondly, the second term of $LB_s(\bar{A})$, which corresponds to the greatest lateness on machine s of jobs that are not in \bar{A} , is calculated as follows:

$$\underset{\substack{j_r \notin A \\ r=1..N}}{\text{MAX}} \left[c(\bar{A}, s) + \sum_{\substack{q=1 \\ j_q \notin A}}^r [S(j_q, s) + p(j_q, s) + R(j_q, s)] - R(j_r, s) - d_s(j_r) \right] \quad (6.11)$$

where

- $\sum_{\substack{q=1 \\ j_q \notin A}}^r [S(j_q, s) + p(j_q, s) + R(j_q, s)] - R(j_r, s)$ is the time required to process the jobs j_1, \dots, j_r on machine s , when these jobs are sorted in non-descending order of $d_s(j_q)$, while respecting precedence constraints (see Appendix A.2).
- $d_s(j_r)$ is the adjusted due-date of j_r on machine s :

$$\begin{aligned} d_s(j_r) &= d(j_r) - \sum_{v=s+1}^V [a(j_r, v-1) + p(j_r, v)] \quad \forall s = 1, \dots, V-1 \\ d_V(j_r) &= d(j_r) \end{aligned} \quad (6.12)$$

Moreover, job-precedence constraints limit the exploration in the branching scheme: at each node, the branches that can be explored agree only with jobs whose the predecessors are already scheduled.

6.3.4. FROM MAKESPAN TO MAXIMUM LATENESS

The problem $N/V, m_v=1/F/L_{max}$ is transformed into $N/V+1, m_v=1/F/C_{max}$

It is important to consider the first problem. When we study maximum lateness criterion, we calculate $L(i)$ the lateness of job i as $L(i) = c(i, V) - d(i)$. If we add a fictitious workshop $V+1$ in which the processing time of job i is calculate using (6.13), we can use scheduling tools that minimize C_{max} on $V+1$ workshops to minimize L_{max} on V workshops:

$$p(i, V+1) = \underset{j=1..N}{\text{MAX}} [d(j)] - d(i) \quad \forall i = 1..N \quad (6.13)$$

$$\begin{aligned}
 \text{Minimize } L_{\max} &\Leftrightarrow \text{Minimize } \left(L_{\max} + \text{MAX}_{j=1..N}[d(j)] \right) \text{ because this last term is a constant} \\
 &\Leftrightarrow \text{Minimize } \left(\text{MAX}_{i=1..N}[L(i)] + \text{MAX}_{j=1..N}[d(j)] \right) \\
 &\Leftrightarrow \text{Minimize } \left(\text{MAX}_{i=1..N}[C(i, V) - d(i)] + \text{MAX}_{j=1..N}[d(j)] \right) \\
 &\Leftrightarrow \text{Minimize } \left(\text{MAX}_{i=1..N} \left[C(i, V) + \text{MAX}_{j=1..N}[d(j)] - d(i) \right] \right) \\
 &\Leftrightarrow \text{Minimize } \left(\text{MAX}_{i=1..N} [C(i, V) + p(i, V + 1)] \right) \\
 &\Leftrightarrow \text{Minimize } \left(\text{MAX}_{i=1..N} [C(i, V + 1)] \right) \\
 &\Leftrightarrow \text{Minimize } C'_{\max}
 \end{aligned}$$

This property allows us to use makespan criterion heuristics to minimize maximum lateness. A similar method could be used to take into account release date constraints, by introducing a fictitious workshop θ , in which the processing time of a job i is equal to $r(i)$.

6.3.5. A FAST HEURISTIC FOR NUMEROUS JOB-PRECEDENCE CONSTRAINTS: VBA

Precedence constraints may be a very important part of a scheduling problem: for instance when more than 50% of jobs have predecessors, or when there are more than two precedence levels; these two aspects were encountered in the INOSETA case study.

These numerous precedence constraints, lead us to develop another heuristic, called VBA, which gives more attention to jobs with many successors. For each job i , we define an index $NS(i)$ by the recursive formula (6.14).

$$\begin{aligned}
 NS(i) &= 1 + \sum_{j \in \text{succ}(i)} [1 + NS(j)] \quad \forall i = 1..N \\
 &\text{with } \text{succ}(i) = \text{set of immediate successors of } i.
 \end{aligned} \tag{6.14}$$

For makespan criterion, a schedule will be obtained by sorting the jobs in descending order of NS , i.e according to the GNS rule (Greatest Number of Successors). This rule is an extension of the NOR rule (Number of Operations Remaining) which is known to be very efficient for job shop scheduling problems [Baker; 1974], [French; 1982]. For maximum lateness criterion, a schedule will be obtained by sorting the jobs according to the GNS-EDD rule, i.e. in descending order of NS , and for jobs that have the same number of successors, in non-descending order of job due-date (EDD rule).

6.4. THE ASSIGNMENT STEP

This step consists of assigning jobs in the order given by the sequence (obtained in the previous step) to the machines in each of the V workshops. First of all three principles were considered:

- Assign the job to the first available machine in each workshop (FAM rule).
- Assign the job to the last busy machine (LBM rule) which enables the completion of the job at the earliest time in the next workshop, and repeat this operation for each workshop. This rule minimizes the jobs waiting time between two workshops.
- Assign the job to the machine series which minimizes first its final completion time and then the machine idle time, i.e. assign the job to the succession of the last busy machines minimizing its final completion time. This rule is a generalization of the LBM rule which attempts to minimize the total jobs waiting time (GLBM rule).

The FAM and LBM rules are simple rules that have often been employed for hybrid flow shop scheduling (see Chapter 3, section 3.4). LBM is known to be very efficient [Gupta; 1988], [Gupta & Tunc; 1994], and better than GLBM for hybrid flow shop problems with identical parallel machines [Guinet & Solomon; 1996].

Consequently, we have worked on the FAM and LBM rules and we have adapted them to precedence constraints, nsd setup and removal times and minimum time lags. Two rules have been designed first R1 and R2, and improved to minimize machine idle time. When using these two rules, we assume that the release date of a job in a workshop depends either on the completion time of its preceding operation and on the time lag, or on the completion time of its predecessor, and that the availability of a machine depends on the removal time of the last processed operation.

Assignment rule R1:

For each workshop v and each job i ,

- if ($v=1$) and (i has a predecessor), the release date T of i is the completion time of its predecessor on the last workshop;
 - assign i in order to minimize machine idle time in workshop 1, i.e.:
 - if one or more machines in v are available at time T , assign i to the last busy machine (LBM rule) and i begins at time T ;
 - else, assign i to the first available machine (FAM rule);
- else assign i to the first available machine (FAM rule).

Assignment rule R2:

For each workshop v and each job i ,

- if ($v=1$) and (i has a predecessor), and if (the release date T of i is greater than the maximum machine availability in workshop 1),
 - assign i in order to minimize machine idle time in workshop 1, i.e.:
 - if one or more machines in v are available at time T , assign i to the last busy machine (LBM rule) and i begins at time T ;
 - else, assign i to the first available machine (FAM rule);
- else
 - assign i in workshop v , to the machine which minimize the waiting time of i in **workshop $v+1$** (LBM rule).

R1 and R2 have been improved by a neighborhood method, in order to minimize machine idle time in the first workshop when assigning a job i which has a predecessor: if one or more machines are available in the first workshop, we try to process other jobs before i , without however delaying the scheduled starting time of i . The other jobs are chosen among those that have no predecessor and that have not yet been assigned.

For the assignment step, we use the real values of the parameters $p(i,v)$, $a(i,v)$, $S(i,v)$ and $R(i,v)$.

6.5. SUMMARY: FIVE HEURISTICS

The problem N/V , $m_v \geq 1$, I/F , j -prec, out -tree, $a_{i,v}$, H_v , S_{nsd} , R_{nsd} / C_{max} or L_{max} , is solved in two steps:

1. Jobs are sequenced on V (or $V+1$) machines, applying flow shop heuristics for the problem N/V , $m_v=1/F$, j -prec, out -tree, $a_{i,v}$, H_v , S_{nsd} , R_{nsd} / C_{max} or L_{max} . Five heuristics, whose algorithms are presented below, have been defined: CDSB (from CDS), SZWB, NEHB, TOWB, and VBA that we have defined in §6.3.5; their complexities being $o(V.N^2)$, $o(V.N^2)$, $o(\max[V.N^3; N^{pl+2}])$, $o(V.N^4)$ and $o(N^2)$ respectively, where pl is precedence level number.
2. Jobs are assigned by applying one of the improved rules R1 or R2; this step is performed at each criterion calculation.

CdsB, inspired by the Campbell, Dudek & Smith heuristic:

Algorithm for C_{max} or L_{max} criterion:

- 1) modify $p(i, v)$ with formulae (6.6),
- 2) compute $p'(i, v)$ and $a'(i, v)$ from $p(i, v)$, $a(i, v)$, $S(i, v)$ and $R(i, v)$ with formulae (6.7),
- 3) if criterion= $\ll L_{max} \gg$, compute $p'(i, V+1)$ with formulae (6.13), and $a'(i, V)=-R(i, V)$,
- 4) Set $k=0$ and $\gamma^*=\infty$,⁽¹⁾
- 5) if $k=0$, $p(i, A)=p(i, B)$ =number of non-zero operations plus non-zero time lags, else compute $p(i, A)$ and $p(i, B)$ with formulae (6.8) for the problem $k>0$,
- 6) modify these processing times into $p'(i, A)$ and $p'(i, B)$ with formulae (6.9),
- 7) obtain a sequence S by applying Johnson's algorithm with $p'(i, A)$ and $p'(i, B)$,
- 8) compute γ for the sequence S on V machines, applying formulae (2.14) and (2.16),
- 9) if $\gamma < \gamma^*$ then ($\gamma^*=\gamma$ and $S^*=S$),
- 10) $k=k+1$, if [$k \leq V$ and criterion= $\ll C_{max} \gg$] or [$k \leq V+1$ and criterion= $\ll L_{max} \gg$] go to step 5, else S^* is the final sequence, and γ^* the best criterion value.

In this algorithm, $V+1$ sub-problems are studied instead of the $V-1$ problems in Campbell's initial approach.

For $k=0$, the application of Johnson's rule with $p(i, A)=p(i, B)$ =number of non-zero operations plus non-zero time lags is an extension of the NOR rule (Number of Operations Remaining). For $k=V$, the application of Johnson's rule with $p(i, A)=p(i, B)$ =total sum of processing and time lags is an extension of the MWkR rule (Most Work Remaining).

SzWB, inspired by the Szwarc heuristic:

The algorithm is the same as the previous one, with two exceptions: only $V-1$ sub-problems are studied ($k=1$ to $V-1$) and for step 5, we use the formulae (6.2) instead of (6.8).

NEHB, inspired by the Nawaz, Ensore & Ham heuristic:

Algorithm for C_{max} or L_{max} criterion:

- 1) modify $p(i, v)$ with formulae (6.6),
- 2) compute $p'(i, v)$ and $a'(i, v)$ from $p(i, v)$, $a(i, v)$, $S(i, v)$ and $R(i, v)$ with formulae (6.7),
- 3) if criterion= $\ll L_{max} \gg$, compute $p'(i, V+1)$ with formulae (6.13), and $a'(i, V)=-R(i, V)$,
- 4) compute $ind(i)$ with formulae (6.10) on V machines if criterion= $\ll C_{max} \gg$, or on $V+1$ machines if criterion= $\ll L_{max} \gg$

- 5) sort the jobs in descending order of $ind(i)$, and obtain a sequence S ,
- 6) $S^* = \{S(1)\}$, $j=2$,
- 7) for $j=2$ to N do
 insert $S(j)$ in every position in S^* and for each partial schedule compute γ , applying formulae (2.14) and (2.16), and retain $S(j)$ in the position which minimize γ ,
- 8) S^* is the final sequence, and compute γ^* applying formulae (2.14) and (2.16).

TowB, inspired by the Townsend heuristic:

Algorithm for C_{max} (with $d(i)=0, \forall i$) or L_{max} criterion:

- 1) modify $p(i, v)$ with formulae (6.6),
- 2) $\bar{A} = \emptyset$,
- 3) for $k=1$ to $N-1$ (nodes) do
 $\gamma^* = \infty$;
 for $i=1$ to N (branches) do
 if $(i \notin \bar{A})$ and $\neg(\exists pred(i) / pred(i) \notin \bar{A})$ then
 $\bar{A} = \bar{A} \oplus \{i\}$,
 compute $LB(\bar{A})$ with formulae (6.4), (6.11), and (6.12),
 if $LB(\bar{A}) \leq \gamma^*$, then $(\gamma^* = LB(\bar{A}))$ and retain i in position k in \bar{A}
- 4) chain \bar{A} to the last job to make the final sequence S^* , and compute γ^* applying formulae (2.14) and (2.16).

VBA:

Algorithm for C_{max} or L_{max} criterion:

- 1) modify $p(i, v)$ with formulae (6.6),
- 2) compute $p'(i, v)$ and $a'(i, v)$ from $p(i, v)$, $a(i, v)$, $S(i, v)$ and $R(i, v)$ with formulae (6.7),
- 3) compute $NS(i)$ with formulae (6.14),
- 4) if criterion = « C_{max} », sort the jobs with GNS rule, and obtain the sequence S^* ,
- 5) if criterion = « L_{max} », sort the jobs with GNS-EDD rule, and obtain the sequence S^*
- 6) compute γ^* applying formulae (2.14) and (2.16).

⁽¹⁾ γ^* represents the best value of the criterion studied, while γ represents an intermediate value for the criterion.

In every heuristic, the calculation of γ^* is carried out according to one of the assignment rules R1 or R2; so that the decision to retain a partial or complete sequence is taken according to the assignment scheme (i.e. using the assignment step).

6.6. LOWER BOUNDS AND ANOTHER HEURISTIC: VBB

To evaluate the performance of the heuristics, several lower bounds on the optimal solution have been defined for maximum lateness criterion.

The first lower bound for the hybrid flow shop, $LB1$, is an adaptation of the bound proposed by Guinet & Solomon [1996]: a set of lower bounds $LB(v)$ has been derived by reducing the V serial workshop problem with parallel machines to V single workshop problems with parallel machines, including minimum time lags, setup and removal times, and precedence constraints.

$$LB1 = \underset{s=1..V}{MAX}[LB1(s)]$$

$$LB1(s) = \underset{i=1..N}{MIN} \left[S(i,1) + \sum_{v=1}^{s-1} [p(i,v) + a(i,v)] + \sum_{j \in pred(i)} \sum_{v=1}^V [p(j,v) + a(j,v)] \right]$$

$$+ \underset{i=1..N}{MAX} \left[\frac{1}{M(s)} \sum_{i=1}^N [S(i,s) + p(i,s) + R(i,s)] - \underset{i=1..N}{MAX}[S(i,s)] - \underset{i=1..N}{MAX}[R(i,s)]; \right]$$

$$- \underset{i=1..N}{MAX} \left[d(i) - \sum_{v=s+1}^V [p(i,v) + a(i,v-1)] - \sum_{j \in succ(i)} \sum_{v=1}^V [p(j,v) + a(j,v)] \right]$$

The first term of $LB1(s)$ is the minimum waiting time of workshop s . The second term is the time required to process all jobs in workshop s in the preemptive case, adapted from McNaughton [1959]. The third term is the greatest adjusted due-date on workshop s ; it enables us to calculate the least maximum lateness on each workshop s . The overall lower bound $LB1$ is the largest lower bound among all workshops; it is calculated with the initial parameter values.

We were inspired by Townsend to design a second lower bound of maximum lateness for a pure flow shop: $LB2$.

$$LB2 = \underset{s=1..V}{MAX}[LB2(s)]$$

$$LB2(s) = \underset{j_r, r=1..N}{MAX} [c(j_r, s) - d'_s(j_r)]$$

with the jobs $j_r, r=1..N$ sorted in non-descending order of $[d'_s(j_r) + R(j_r, s)]$; remember that s represents a single machine. We have proved that the optimal solution to the problem $N/I//S_{nsd}, R_{nsd}/L_{max}$ is to sort the jobs in non-descending order of $d(i) + R(i)$ (see Appendix A.2).

$$c(j_1, s) = \underset{w \min(s); S(j_1, s)}{MAX} + p(j_1, s)$$

$$c(j_r, s) = c(j_{r-1}, s) + R(j_{r-1}, s) + S(j_r, s) + p(j_r, s) \quad \text{for } r > 1$$

$$w \min(s) = \underset{i=1..N}{MIN} \left[\underset{S(i, s); S(i, 1) + \sum_{v=1}^{s-1} [p(i, v) + a(i, v)] + \sum_{v=1}^V [p(pred(i), v) + a(pred(i), v)]}{MAX} \right]$$

$$d'_s(i) = d(i) - \sum_{v=s+1}^V [p(i, v) + a(i, v)] \quad \forall s = 1..V-1, \quad \forall i = 1..N$$

$$\text{and } d'_V(i) = d(i) \quad \forall i = 1..N$$

It is important to note that $w \min(s)$ is the minimum waiting time for the workshop s , $c(i, s)$ is the completion time of job i in workshop s , and d'_s is the adjusted due-date of i in s .

$LB2$ was used to compare heuristics on pure flow shop configurations, but it does not represent a lower bound of maximum lateness for hybrid flow shops. Nevertheless, we have transformed it into another heuristic **VBB** for hybrid flow shop scheduling.

For makespan criterion, we used the lower bound $LB1$, which is also a lower bound for makespan criterion when due-dates are not present. In that case, the third term of $LB1(s)$ represents the minimum time required to process all the jobs from workshop s to the last workshop; it is equivalent to:

$$+ \underset{i=1..N}{MIN} \left[\sum_{v=s+1}^V [p(i, v) + a(i, v-1)] + \sum_{j \in succ(i)} \sum_{v=1}^V [p(j, v) + a(j, v)] \right]$$

6.7. CONCLUSION

Previous research has been adapted to take makespan and maximum lateness, using the same heuristics into account. Four algorithms have been selected according to their capacity for solving pure flow shop problems minimizing such criteria; they have been improved and adapted to three types of constraint (job-precedence, minimum time lags, different workshop calendars, non-sequence dependent setup and removal times): CDSB, SZWB, NEHB, TOWB.

Moreover, two new heuristics were designed: VBA which is especially suited for numerous precedence constraints and VBB which can be used for maximum lateness criterion.

The experimented framework presented in the next chapter will allow us to assess the performance and robustness of these new heuristics when faced with these constraints.

CHAPTER 7

EXPERIMENTS & RESULTS

7.1. GENERAL CASE STUDY	108
7.1.1. Data	109
7.1.2. Results for C_{max} Criterion	110
7.1.2.1. Flow shop configurations	110
7.1.2.2. Hybrid flow shop configurations	113
7.1.3. Results for L_{max} Criterion	115
7.1.3.1. Flow shop configurations	116
7.1.3.2. Hybrid flow shop configurations	117
7.1.4. Conclusions	121
7.2. INOSETA'S CASE STUDY	122
7.2.1. Data	122
7.2.2. The Warping / Weaving Stage (WWP)	124
7.2.3. The Finishing Stage (FP)	125
7.3. CONCLUSIONS & RECOMMENDATIONS	127

7. EXPERIMENTS & RESULTS

Numerous experiments have been carried out to assess the quality and robustness of the six proposed heuristics. These methods have been applied on a large range of both theoretical and real data. Their performance was examined by solving 17600 different problems, involving 40 different problem configurations, up to 24 problem sizes. Two different criteria C_{max} and L_{max} were considered separately. For each problem size, 32 tests were generated. Computation experiments have been done with Locho software [Proust & *al.*; 1991-b], on two micro-computers: 486 - 66 MHz and Pentium 75 MHz.

We use the average deviation of the criterion value from the lower bound of the criterion (LB) to evaluate the quality of each heuristic (here H is the criterion value calculated by a given heuristic):

$$\frac{H - LB}{LB}$$

To evaluate their robustness, we began with makespan experiments on pure flow shop without constraints. We then have added job-precedence constraints, minimum time lags and finally, setup and removal times. Afterwards, we have experimented the hybrid dimension and maximum lateness criterion. Other performance measures have been also calculated: mean completion time, mean tardiness, absolute lateness, and the number of tardy jobs.

We have studied three cases: a general case with and without the hybrid dimension, and two INOSETA cases: warping/ weaving stage (WWP) and finishing stage (FP).

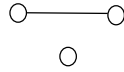
The experiment results are presented in section 7.1 for the general case, and in section 7.2 for INOSETA cases. Section 7.3 concludes this chapter and attempts to propose a user-guide for scheduling tools.

7.1. GENERAL CASE STUDY

This section presents a synthesis of our experiments with general case. These experiments enabled us to assess the robustness of heuristics with respect to various constraints. The characteristics of the generated data are given in §7.1.1, and the results for makespan criterion and maximum lateness criterion are given in §7.1.2 in §7.1.3 respectively. In both cases, pure and hybrid flow shops have been studied. Conclusion from these experiments is given in §7.1.4.

7.1.1. DATA

We have tested the heuristics on a general case where 33% of jobs have one predecessor and no successor, 33% of jobs have one successor and no predecessor and 33% of jobs have no predecessor or successor.



The number of jobs (N) to be processed were: 39, 51, 63, 75, 87 and 99, depending on precedence constraint generation.

For pure flow shop, the number of machines (V) considered were: 5, 6, 7 and 8; for hybrid flow shops, the number of workshops (V) were 4 and 5, with 1 or 3 identical machines per workshop:

V = 4				V = 5				
m(1)	m(2)	m(3)	m(4)	m(1)	m(2)	m(3)	m(4)	m(5)
3	3	3	3	3	3	3	3	3
1	1	3	3	1	1	1	3	3
3	3	1	1	3	3	3	1	1
3	1	1	3	3	1	1	1	3

Table 7-a: Problem dimension (number of machines per workshop)

Processing times were integer and randomly generated, with a log-normal distribution of mean 10 and standard deviation 3. These values were adjusted to take into account the different number of machines in the workshops, because we observe that in process industries, a bottleneck workshop is not necessarily the one which consists of a minimum number of machines.

Minimum times lags were also generated with a log-normal distribution, and we have considered three cases: small with regard to processing times (mean 1.5 and standard deviation 0.5), equivalent to processing times (mean 10 and standard deviation 3), or large compared with processing times (mean 20 and standard deviation 3).

We tested two configurations of setup times: small compared with processing times (10% of processing times) or large (40%). In all cases, removal times were the half of setup times.

For maximum lateness experiments the due-date of a job that have no successor was generated with a uniform distribution between A and $A+B$, defined using the following formulae, where A represents the time required to process the shortest job, and B the mean processing duration of a workshop:

$$A = \underset{i=1..N}{MIN} \left[\sum_{v=1}^V (p(i,v) + a(i,v)) + \sum_{j \in pred(i)} \sum_{v=1}^V (p(j,v) + a(j,v)) \right]$$

$$B = \frac{1}{V} \sum_{v=1}^V \sum_{i=1}^N [S(i,v) + p(i,v) + R(i,v)]$$

For a job i that have one or more successors, the due-date $d(i)$ is calculate as follows:

$$d(i) = \underset{j \in succ(i)}{MIN} \left[d(j) - \sum_{v=1}^V (p(j,v) + a(j,v)) \right]$$

7.1.2. RESULTS FOR C_{max} CRITERION

We have tested the five algorithms CDSB, SZWB, NEHB, TOWB and VBA on both flow shop and hybrid flow shop configurations; this approach enables us to assess their robustness.

7.1.2.1. Flow shop configurations

The performance of the five heuristics with respect to makespan criterion, was examined by solving 9984 different problems, involving 13 different configurations and 24 different problem sizes. For each problem size, 32 tests were generated.

Two other heuristics have been designed from CDSB and NEHB :

- CDSB2, which is obtained by the introduction of the Dannenbring factor when computing the two-fictitious machine processing times for the k^{th} problem; the formulae (6.8) become (7.1):

$$p(i,A) = \sum_{v=1}^k [p(i,v) + a(i,v)] \times (V - v + 1)$$

$$p(i,B) = \sum_{v=V-k+1}^V [p(i,v) + a(i,v)] \times v$$
(7.1)

- NEHB2, which is obtained by the introduction of the Dannenbring factor in the calculation of the sorting index; the formulae (6.10) become (7.2):

$$ind(i) = \sum_{v=1}^V [p(i,v) + a(i,v)] \times (V - v + 1) + \sum_{j \in succ(i)} \sum_{v=1}^V [p(j,v) + a(j,v)] \times (V - v + 1) \quad (7.2)$$

with $succ(i)$ = set of immediate successors of i .

Table 7-B shows the average deviation of each heuristic value from the lower bound of makespan criterion, in the 13 configurations:

- without any constraints,
- with precedence constraints,
- with precedence constraints, setup times (either 40% or 10% of processing times) and removal times (the half of the setup times) labeled S and R,
- with minimum time lags (small, medium, or large compared with processing times), labeled a(s), a(m) and a(l),
- and a mix of the last two configurations.

	CDSB	CDSB2	SzWB	NEHB	NEHB2	TowB	VBA
no constraints	3,3%	2,5%	6,0%	9,4%	13,7%	4,6%	9,9%
precedence	6,3%	5,9%	8,6%	9,4%	11,7%	10,5%	9,8%
prec. + S(10), R(5)	6,7%	6,6%	8,7%	9,9%	12,4%	9,5%	10,2%
prec. + S(40), R(20)	4,9%	4,6%	7,0%	8,4%	10,8%	5,2%	8,7%
prec. + a(s)	7,3%	7,2%	9,4%	10,5%	12,5%	12,3%	10,6%
prec. + a(m)	13,2%	16,6%	17,3%	16,8%	18,3%	28,4%	14,6%
prec. + a(l)	14,2%	22,1%	22,1%	19,7%	21,4%	32,3%	14,6%
prec. + S(10), R(5) + a(s)	7,3%	7,3%	9,5%	11,0%	13,2%	11,1%	11,0%
prec. + S(10), R(5) + a(m)	13,1%	16,2%	17,2%	16,7%	18,6%	26,3%	14,8%
prec. + S(10), R(5) + a(l)	14,0%	21,3%	21,4%	19,3%	21,1%	29,6%	14,5%
prec. + S(40), R(20) + a(s)	5,1%	4,8%	7,4%	8,9%	11,3%	6,0%	9,1%
prec. + S(40), R(20) + a(m)	8,6%	9,6%	11,4%	11,6%	13,8%	13,6%	11,5%
prec. + S(40), R(20) + a(l)	9,5%	13,1%	13,7%	12,3%	14,4%	16,4%	10,9%

Table 7-B: Average deviation from C_{max} lower bound in pure flow shops (C_{max} criterion).

To assess the robustness of the different heuristics, we calculated a robustness measure (RM) as:

$$RM = (D_{max} - D_{mean})/D_{mean}$$

with D_{max} : the maximum of the average deviation with respect to all the experiments,

Dmean : average deviation without constraints

RM ratio has been computed for each heuristic.

From this ratio and the heuristic grading (according to Table 7-B), we obtain following results:

	grading	RM value
CDSB	1	3,3%
CDSB2	2	7,8%
SzWB	3	2,7%
VBA	4	0,5%
NEHB	5	1,1%
TowB	6	6,0%
NEHB2	7	0,6%

From this ranking, we observe that VBA and NEHB2 are the most robust. From Table 7-B we see that CDSB remains the best heuristic whatever the configuration with the exception of the first four cases, where CDSB2 is better. The constraints that most affect heuristics are the medium or large time lags. In such cases, only CDSB and VBA have to be retained. We also note that SzWB does not perform well with setup and removal time constraints.

In Table 7-B, if we look at the 11 latter configurations in more detail, we observe for instance, that the two configurations « *prec.* + $S(10)$, $R(5)$ + $a(s)$ » and « *prec.* + $S(40)$, $R(20)$ + $a(m)$ » give equivalent results. We can also identify three representative configurations, depending on the relative value of the setup (S), removal (R) and time lag (a) parameters compared with processing time:

- A : problems introducing negative time lags,
- B : problems introducing time lags closed to zero,
- C : problems introducing positive time lags.

Examples of these three configurations are illustrated in the figures below. Consequently, we decided to limit our experiments on hybrid flow shops to the three representative constraint configurations A, B and C.

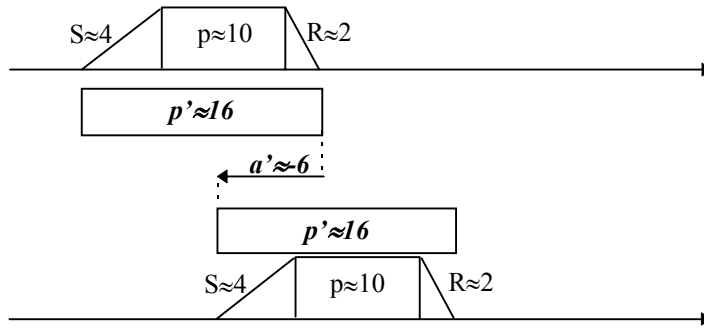


Figure 7-A: Configuration A: problems introducing negative time lags.

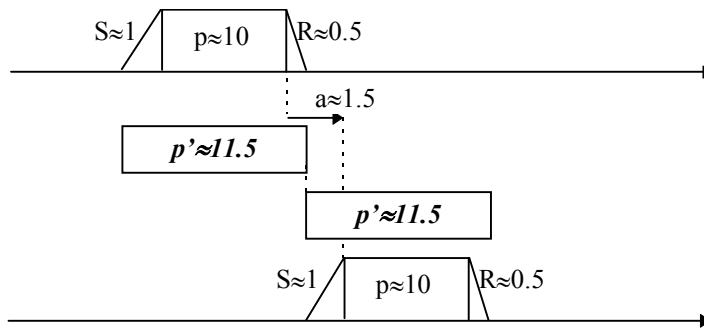


Figure 7-B: Configuration B: problems introducing time lags closed to zero.

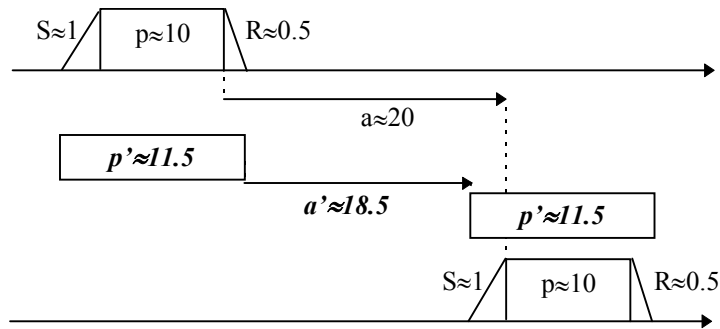


Figure 7-C: Configuration C: problems introducing positive time lag.

7.1.2.2. Hybrid flow shop configurations

The performance of the heuristics was tested by solving 2816 different problems, involving 10 different cases (3 dimensions and the 3 configurations A, B, and C) and 8 different problem sizes. For each problem size, 32 tests were generated.

We excluded the TowB heuristic from our experiment framework in flow shop environments for three reasons: it gives the worst results, it is not robust and it is the most expensive in computing requirements.

Table 7-C presents the average deviation of each heuristic value from the lower bound of makespan criterion in the 3 configurations A, B and C, with the two improved assignment rules R1 and R2, in three cases: the same machine number per workshop (3 3 3 3 3), single machine workshops upstream (1 1 1 3 3), and single machine workshops downstream (3 3 1 1 1).

	CdsB		CdsB2		SzwB		NEHB		NEHB2		VBA	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
3 3 3 3 3												
no constraints	5,3%	6,5%	4,0%	5,1%	7,0%	7,8%	11,9%	12,4%	16,6%	16,7%	13,2%	14,2%
configuration A	6,1%	7,0%	5,1%	5,9%	7,6%	8,2%	9,7%	9,8%	10,0%	10,3%	11,3%	12,0%
configuration B	8,7%	10,2%	8,3%	9,8%	12,0%	13,1%	13,4%	14,4%	14,1%	15,1%	14,7%	16,2%
configuration C	18,2%	72,0%	18,3%	72,8%	24,0%	74,0%	28,6%	80,5%	28,5%	80,4%	19,6%	79,0%
1 1 1 3 3												
configuration A	2,2%	2,2%	1,8%	1,8%	2,4%	2,4%	4,1%	4,1%	5,5%	5,5%	4,0%	4,0%
configuration B	1,9%	1,9%	1,7%	1,7%	2,5%	2,5%	3,4%	3,4%	4,7%	4,7%	4,3%	4,3%
configuration C	2,5%	2,5%	2,3%	2,3%	3,5%	3,5%	4,4%	4,4%	5,4%	5,5%	6,0%	6,0%
3 3 1 1 1												
configuration A	1,0%	1,0%	0,9%	0,9%	1,0%	1,0%	1,8%	1,8%	2,4%	2,4%	2,1%	2,1%
configuration B	0,9%	0,9%	0,8%	0,8%	0,8%	0,8%	2,2%	2,2%	2,4%	2,4%	2,2%	2,2%
configuration C	1,6%	1,6%	1,7%	1,7%	2,4%	2,4%	5,6%	5,6%	5,8%	5,8%	3,6%	3,6%

Table 7-C: Average deviation from C_{max} lower bound in hybrid flow shops (C_{max} criterion).

The heuristic grading is the following (similar to pure flow shop environments):

CdsB2	1
CdsB	2
SzwB	3
NEHB	4
VBA	5
NEHB2	6

We firstly note that assignment rules R1 and R2 are equivalent, except when the machine number is constant in each workshop: in this case, R1 is better than R2. Large time lags have a negative influence only when the machine number is constant in each workshop; this phenomena already exists in pure flow shop environments.

As in pure flow shops, when the machine number is constant in each workshop, results of configuration A are better than those of B, which are themselves better than those of C; on the other hand, the heuristics perform better in configuration A, than in B and C.

For each problem studied we also calculated another performance measure, the mean completion time, which is a good indicator of the work in-process value. The average deviations for each heuristic value from the best obtained value are presented in Table 7-D.

The heuristic grading for this measure is similar to the one obtained for makespan criterion, except for VBA which is in the third position instead of the fifth. We also note that all the heuristic values fall within a small interval; it means that they have similar performances with regards to the mean completion time criterion.

	CdsB		CdsB2		SZWB		NEHB		NEHB2		VBA	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
3 3 3 3 3												
no constraints	2,2%	2,5%	0,3%	0,8%	3,1%	3,7%	13,1%	13,1%	15,1%	15,0%	5,1%	5,7%
configuration A	2,1%	2,1%	2,0%	2,0%	3,1%	3,1%	5,5%	5,5%	6,1%	6,1%	0,8%	0,8%
configuration B	1,2%	2,0%	0,4%	1,4%	3,0%	3,6%	5,1%	5,8%	5,5%	6,0%	4,9%	5,8%
configuration C	1,0%	32,4%	0,8%	32,9%	2,1%	33,4%	6,0%	41,6%	6,1%	41,2%	2,4%	39,4%
1 1 1 3 3												
configuration A	2,1%	2,1%	2,0%	2,0%	3,1%	3,1%	5,5%	5,5%	6,1%	6,1%	0,8%	0,8%
configuration B	1,4%	1,4%	1,5%	1,5%	3,7%	3,7%	5,5%	5,5%	6,0%	6,0%	1,2%	1,2%
configuration C	1,1%	1,1%	0,7%	0,7%	3,4%	3,4%	4,9%	5,0%	5,4%	5,5%	1,6%	1,7%
3 3 1 1 1												
configuration A	2,3%	2,3%	1,9%	1,9%	4,5%	4,5%	4,0%	4,0%	3,2%	3,2%	1,2%	1,2%
configuration B	1,5%	1,5%	1,0%	1,0%	2,5%	2,5%	4,9%	4,9%	3,9%	3,9%	2,0%	2,0%
configuration C	0,8%	0,8%	0,9%	0,9%	1,5%	1,5%	6,7%	6,8%	6,0%	6,1%	3,5%	3,5%

Table 7-D: Average deviation from the best \bar{C} in hybrid flow shops (C_{max} criterion).

7.1.3. RESULTS FOR L_{MAX} CRITERION

In the same way, we have tested the heuristics for the maximum lateness criterion in pure and hybrid flow shop environments. For this criterion, we have used the lower bound $LB2$ for pure flow shop experiments because it was better than $LB1$, and $LB1$ the only available lower bound for hybrid flow shop experiments (see section 6.6). We also calculated four other performance measures in order to decide between heuristics that gave similar

results: maximum absolute lateness (ET_{max}), number of tardy jobs (Nt), mean tardiness (\bar{T}), and mean completion time (\bar{C}).

7.1.3.1. Flow shop configurations

The performance of eight heuristics was examined by solving 1536 different problems, involving 24 different problem sizes, and for the two opposite constraint configurations A and C. For each problem size, 32 tests were generated.

The lower bound LBI was very good for makespan criterion, but it performed very badly for maximum lateness. We therefore calculated the average deviation from both the lower bound and the best obtained value. Results are presented in Table 7-E and Table 7-F respectively

	CdsB	CdsB2	SzWB	NEHB	NEHB2	TowB	VBA	VBB
configuration A	283%	481%	517%	99%	164%	96%	207%	182%
configuration C	1152%	1421%	1512%	620%	944%	675%	900%	835%

Table 7-E: Average deviation from L_{max} lower bound in pure flow shops (L_{max} criterion).

	CdsB	CdsB2	SzWB	NEHB	NEHB2	TowB	VBA	VBB
configuration A	101%	203%	220%	7%	43%	6%	61%	48%
configuration C	85%	124%	137%	6%	54%	14%	47%	37%

Table 7-F: Average deviation from the best L_{max} in pure flow shops (L_{max} criterion).

The best obtained values are the most often obtained by NEHB and TowB; the general heuristic grading is presented below:

NEHB	1
TowB	2
VBB	3
NEHB2	4
VBA	5
CdsB	6
CdsB2	7
SzWB	8

The results for the other performance measures are presented in Table 7-G.

For ET_{max} , the heuristic VBB consistently gives the best value, and is closely followed by NEHB and VBA. This means that the best heuristics for L_{max} criterion minimize ET_{max} as well. On the other hand, heuristics that perform well for L_{max} , perform badly for Nt and \bar{C} , and vice versa. Results for \bar{T} performance measures confirm the conclusions we drew from the L_{max} analysis: NEHB minimizes in the same time maximum lateness and mean tardiness.

	CdsB	CdsB2	SzWB	NEHB	NEHB2	TowB	VBA	VBB
configuration A	249%	222%	227%	38%	40%	105%	47%	35%
configuration C	118%	120%	128%	35%	43%	66%	33%	24%

ET_{max} : average deviation from the best value

	CdsB	CdsB2	SzWB	NEHB	NEHB2	TowB	VBA	VBB
configuration A	12,3%	0,5%	1,9%	8,6%	7,7%	5,6%	20,9%	18,4%
configuration C	16,8%	0,8%	2,6%	8,3%	6,3%	4,2%	22,8%	20,7%

\bar{C} : average deviation from the best value

	CdsB	CdsB2	SzWB	NEHB	NEHB2	TowB	VBA	VBB
configuration A	78%	61%	61%	93%	75%	84%	98%	97%
configuration C	80%	67%	67%	96%	79%	84%	100%	100%

Nt : average percentage of tardy jobs

	CdsB	CdsB2	SzWB	NEHB	NEHB2	TowB	VBA	VBB
configuration A	141%	156%	181%	1%	46%	26%	53%	42%
configuration C	106%	83%	96%	2%	41%	22%	50%	42%

\bar{T} : average deviation from the best value

Table 7-G: Average deviation from the best value for other performance measures in pure flow shops (L_{max} criterion).

7.1.3.2. Hybrid flow shop configurations

The performance of the heuristics was examined by solving 2560 different problems, involving 8 different problem sizes, in each constraint configuration A, B and C. For each problem size, 32 tests were generated.

The average deviation for each heuristic value from the lower bound LBI is given in Table 7-H. We have already stated that this bound was bad in pure flow shop environments, and this remark remains true in hybrid flow shop environments when the machine number in

each workshop is constant, although it does perform better in other cases. The average deviations from the best obtained value are presented in Table 7-I.

We have not experimented with SZWB because it gave the worst results in pure flow shops, nor with CDSB2 and NEHB2 because they were worse than CDSB and NEHB.

	CDSB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
3 3 3 3 3										
no constraints	4483%	4537%	4738%	4780%	4069%	4096%	5075%	5133%	4980%	5029%
configuration A	4774%	4818%	4936%	5030%	4176%	4199%	5289%	5309%	5099%	5099%
configuration B	5089%	5158%	5228%	5339%	4813%	4868%	5751%	5807%	5523%	5590%
configuration C	3494%	5592%	4085%	6036%	3857%	5734%	4438%	6370%	4130%	6210%
1 1 1 3 3										
configuration A	106%	106%	112%	112%	113%	113%	119%	119%	117%	117%
configuration B	107%	107%	115%	115%	117%	117%	124%	124%	121%	121%
configuration C	120%	120%	128%	128%	133%	133%	137%	137%	134%	134%
3 3 1 1 1										
configuration A	49%	49%	51%	51%	55%	55%	57%	57%	56%	56%
configuration B	51%	51%	55%	55%	58%	58%	60%	60%	60%	60%
configuration C	60%	60%	69%	69%	64%	64%	72%	72%	71%	71%

Table 7-H: Average deviation from L_{max} lower bound in hybrid flow shops (L_{max} criterion).

	CDSB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
3 3 3 3 3										
no constraints	13,6%	15,0%	20,2%	21,3%	2,8%	3,5%	29,4%	30,7%	26,7%	27,8%
configuration A	1,3%	2,0%	1,6%	2,1%	4,9%	5,2%	3,7%	4,1%	2,2%	2,8%
configuration B	10,6%	11,8%	12,8%	14,5%	5,5%	6,4%	23,9%	25,0%	19,0%	20,4%
configuration C	3,6%	62,1%	20,2%	74,4%	11,6%	66,4%	29,9%	85,1%	20,9%	80,2%
1 1 1 3 3										
configuration A	4,4%	4,4%	7,4%	7,4%	7,9%	7,9%	11,0%	11,0%	9,8%	9,8%
configuration B	3,4%	3,4%	7,1%	7,1%	8,3%	8,3%	11,8%	11,8%	10,2%	10,2%
configuration C	3,7%	3,7%	7,9%	8,0%	9,9%	10,0%	12,2%	12,2%	10,5%	10,5%
3 3 1 1 1										
configuration A	2,7%	2,7%	4,5%	4,5%	7,0%	7,1%	8,3%	8,4%	7,9%	7,8%
configuration B	2,8%	2,8%	5,2%	5,2%	7,1%	7,1%	9,0%	9,1%	8,5%	8,5%
configuration C	3,0%	3,1%	8,7%	8,7%	5,6%	5,7%	11,0%	11,0%	10,2%	10,2%

Table 7-I: Average deviation from the best L_{max} in hybrid flow shops (L_{max} criterion).

As for makespan criterion, the assignment rule R1 is once again better than R2, when the machine number in each workshop is constant.

From these two tables, we can conclude that CdsB, which gave unsatisfactory results in pure flow shops, becomes the best heuristic in hybrid flow shops; TowB and NEHB gave results that were quite similar. However, all the heuristic values in Table 7-I fall within a small interval, so we can conclude that their performances are equivalent.

The heuristic grading for the maximum lateness criterion is the following:

CdsB	1
TowB	2
NEHB	3
VBB	4
VBA	5

Results on the other performance measures are presented in Table 7-J for ET_{max} , in Table 7-K for Nt , in Table 7-L for \bar{C} and in Table 7-M for \bar{T} .

	CdsB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
3 3 3 3 3										
no constraints	8,0%	8,8%	12,5%	13,5%	3,3%	3,7%	20,6%	21,9%	18,3%	19,3%
configuration A	13,5%	14,2%	16,5%	17,7%	4,6%	3,8%	25,4%	25,9%	20,9%	21,3%
configuration B	10,0%	11,3%	12,0%	13,7%	5,5%	6,5%	22,9%	24,0%	18,1%	19,4%
configuration C	3,6%	62,1%	20,2%	74,4%	11,6%	66,4%	29,9%	85,1%	20,9%	80,2%
1 1 1 3 3										
configuration A	4,4%	4,4%	7,4%	7,4%	7,9%	7,9%	11,0%	11,0%	9,8%	9,8%
configuration B	3,4%	3,4%	7,1%	7,1%	8,3%	8,3%	11,8%	11,8%	10,2%	10,2%
configuration C	3,7%	3,7%	7,9%	8,0%	9,9%	10,0%	12,2%	12,2%	10,5%	10,5%
3 3 1 1 1										
configuration A	2,7%	2,7%	4,5%	4,5%	7,0%	7,1%	8,3%	8,4%	7,9%	7,8%
configuration B	2,8%	2,8%	5,2%	5,2%	7,1%	7,1%	9,0%	9,1%	8,5%	8,5%
configuration C	3,0%	3,1%	8,7%	8,7%	5,6%	5,7%	11,0%	11,0%	10,2%	10,2%

Table 7-J: Average deviation from the best ET_{max} in hybrid flow shops (L_{max} criterion).

	CdsB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
3 3 3 3 3										
no constraints	56%	56%	61%	61%	61%	61%	60%	60%	60%	60%
configuration A	61%	61%	62%	62%	65%	66%	63%	63%	62%	62%
configuration B	63%	64%	66%	66%	69%	70%	66%	67%	65%	65%
configuration C	78%	85%	75%	84%	76%	86%	78%	85%	75%	85%
1 1 1 3 3										
configuration A	72%	72%	74%	74%	75%	75%	74%	74%	74%	74%
configuration B	73%	73%	75%	75%	76%	76%	74%	74%	74%	74%
configuration C	77%	77%	79%	79%	80%	80%	79%	79%	79%	79%
3 3 1 1 1										
configuration A	82%	82%	82%	82%	83%	83%	82%	81%	82%	82%
configuration B	81%	81%	82%	82%	83%	83%	82%	82%	82%	82%
configuration C	84%	84%	86%	86%	87%	87%	86%	86%	86%	86%

Table 7-K: Average percentage of tardy jobs (Nt) in hybrid flow shops (L_{max} criterion).

	CdsB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
3 3 3 3 3										
no constraints	0,6%	1,3%	7,9%	8,3%	5,7%	6,2%	7,7%	8,2%	7,7%	8,2%
configuration A	1,4%	1,8%	3,3%	3,7%	4,0%	4,2%	5,1%	5,4%	3,5%	3,9%
configuration B	0,8%	1,5%	3,5%	4,3%	5,0%	5,5%	6,2%	6,9%	4,0%	4,9%
configuration C	1,6%	37,6%	3,7%	39,6%	1,1%	36,8%	7,9%	40,1%	2,1%	38,3%
1 1 1 3 3										
configuration A	1,7%	1,7%	4,8%	4,8%	1,7%	1,7%	4,9%	4,9%	4,7%	4,7%
configuration B	1,6%	1,6%	5,1%	5,1%	1,8%	1,8%	5,1%	5,1%	5,2%	5,2%
configuration C	2,3%	2,3%	4,8%	4,7%	1,0%	1,0%	4,9%	5,0%	5,1%	5,1%
3 3 1 1 1										
configuration A	1,7%	1,7%	2,5%	2,5%	1,2%	1,2%	2,4%	2,4%	2,4%	2,4%
configuration B	1,4%	1,4%	3,6%	3,6%	1,0%	1,0%	3,2%	3,2%	3,4%	3,4%
configuration C	1,9%	1,9%	5,7%	5,8%	0,4%	0,5%	4,8%	4,8%	5,0%	5,0%

Table 7-L: Average deviation from the best \bar{C} in hybrid flow shops (L_{max} criterion).

L_{max} grading is preserved for ET_{max} , and \bar{C} . It may therefore be deduced that heuristics that minimize L_{max} , minimize ET_{max} , and \bar{C} as well. This property was not true for pure flow shop experiments.

	CdsB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
3 3 3 3 3										
no constraints	9,3%	11,1%	14,3%	15,3%	4,3%	5,3%	18,0%	19,3%	17,5%	18,6%
configuration A	15,3%	16,2%	16,1%	17,4%	5,0%	4,8%	19,3%	20,0%	17,2%	18,0%
configuration B	12,1%	13,2%	12,5%	13,8%	4,7%	5,8%	17,8%	19,0%	14,4%	16,6%
configuration C	3,5%	79,2%	15,8%	86,2%	6,8%	75,5%	21,0%	85,6%	10,7%	81,5%
1 1 1 3 3										
configuration A	9,3%	9,3%	11,3%	11,3%	1,3%	1,3%	11,8%	11,8%	11,5%	11,5%
configuration B	9,7%	9,7%	11,9%	11,9%	1,0%	1,0%	12,7%	12,7%	12,7%	12,7%
configuration C	10,9%	10,9%	11,1%	11,2%	0,9%	0,9%	13,1%	13,2%	12,9%	12,9%
3 3 1 1 1										
configuration A	4,8%	4,8%	6,0%	6,0%	2,1%	2,0%	6,2%	6,3%	6,1%	6,1%
configuration B	5,9%	6,0%	7,8%	7,7%	1,8%	1,7%	7,7%	7,8%	7,5%	7,6%
configuration C	7,4%	7,4%	10,7%	10,8%	0,5%	0,5%	9,6%	9,6%	9,7%	9,7%

Table 7-M: Average deviation from the best \bar{T} in hybrid flow shops (L_{max} criterion).

Contrary to results obtained for pure flow shops, in hybrid flow shop experiments, all the heuristics gave similar results for Nt . With regard to \bar{T} , the grading obtained for the L_{max} criterion is preserved, except for TowB which shows a much better performance then previously.

7.1.4. CONCLUSIONS

This experimented framework allows us to assess the quality and robustness of the six proposed heuristics in both pure and hybrid flow shop environments, for makespan and maximum lateness criteria, and with regards to several constraints: job-precedence constraints, different workshop calendars, minimum time lags and non-sequence dependent setup and removal times. Concerning the combination of the last two constraints, we have identified three representative kinds of configuration: equivalent to negative, zero, or positive time lags.

Based on the criterion to minimize and to the workshop layout (single machine or multi-machines), we may formulate the following conclusions:

The makespan criterion:

- For pure flow shop configurations, we observe that CdsB and CdsB2 are the best whatever the constraints, while VBA and NEHB2 are the most robust.

- For hybrid flow shop configurations, the heuristic grading is similar to that obtained for pure flow shops; when the number of machines per workshop is constant, the assignment rule R1 is better than R2, otherwise, R1 and R2 are equivalent.
- The mean completion time analysis leads to the same conclusions.

The maximum lateness criterion:

- For pure flow shop configurations, the best heuristics are NEHB and TOWB; they are also the best for mean tardiness measure. From maximum absolute lateness measure analysis, we can conclude that VBB, NEHB and VBA provide the best results.
- For hybrid flow shop configurations CDSB outperforms other heuristics. This heuristic also minimizes other measures like mean tardiness and maximum absolute lateness.
- Due-date based criteria leads to the same assignment rule conclusions as processing time based criteria.

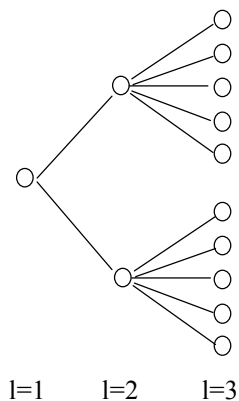
7.2. INOSETA'S CASE STUDY

7.2.1. DATA

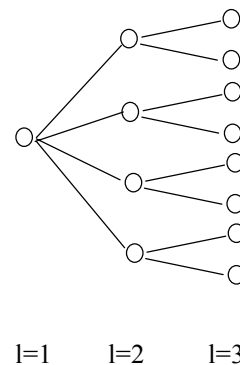
- The warping/ weaving stage (WWP):

In this case, the number of workshops $V=7$, and an analysis of real data allows us to approximate processing times using a log-normal distribution of mean 4, and standard deviation 1.5; they have been adjusted according to the real data. We tested our heuristics on two extreme cases of precedence constraints encountered in the company:

1 creel unit => 2 primary beams
1 primary beam => 5 warps



1 creel unit => 4 primary beams
1 primary beam => 2 warps



In both cases, there were three precedence levels. The number of jobs N to be processed were 39, 52, 65, 78, 91. The number of machines $m(v)$ per workshop, and other constraints are given in the following table.

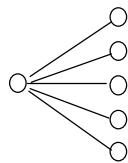
workshop v	$m(v)$	non-zero operation per level		
		$l=1$	$l=2$	$l=3$
1 low twisting	3	x		
2 fixing	2	x		
3 warping	1		x	
4 sizing	2		x	
5 beaming	1		x	
6 lease making	1			x
7 weaving	112			x
% of jobs per precedence level		8%	15%	77%
		8%	31%	61%

We considered real minimum time lags and removal times for the sizing operation, and real setup times for the low twisting and weaving operations. This problem was studied for makespan criterion.

- The finishing stage (FP):

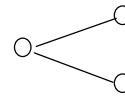
For this problem, the number of workshops $V=7$, and processing times were approximated using a log-normal distribution with mean 8, and standard deviation 3; they were adjusted according to the real data. We tested our heuristics on two extreme cases of precedence constraints encountered in the plant:

83% of jobs have a predecessor



$l=1$ $l=2$

67% of jobs have a predecessor



$l=1$ $l=2$

In both cases, there were two precedence levels. The number of jobs N to be processed were 44, 55, 66, 77, 88, 99. The number of machines $m(v)$ per workshop, and other constraints are exposed in the following table.

workshop v	m(v)	non-zero operation per level	
		l=1	l=2
1 join-clothing	1		x
2 desizing	2	x	
3 relaxing/dyeing	14	x	x
4 untwisting	1	x	x
5 drying	1	x	x
6 pre/final setting	2	x	x
7 weight reduction	1	x	
% of jobs per precedence level		17%	83%
		33%	67%

We considered real minimum time lags for each operations, and average setup times⁽¹⁾ for relaxing/dyeing, drying, pre/final setting and weight reduction. This problem was studied for maximum lateness criterion.

7.2.2. THE WARPING/WEAVING STAGE (WWP)

The performance of the heuristics on makespan criterion was examined by solving 320 different problems, involving the two precedence configurations and 5 different problem sizes. For each problem size, 32 tests were generated.

The average deviation of each heuristic value from the lower bound LBI for C_{max} criterion are given in Table 7-N; and the average deviation from the best obtained value of \bar{C} are presented in Table 7-O.

	CDSB		SZWB		NEHB		TOWB		VBA	
	R1	R2	R2	R1	R1	R2	R2	R1	R1	R2
1 creel => 2 beams 1 beam => 5 warps	21,7%	21,8%	30,2%	30,3%	28,0%	28,2%	32,8%	33,0%	33,5%	33,7%
1 creel => 4 beams 1 beam => 2 warps	26,3%	26,4%	42,2%	42,4%	39,3%	39,5%	52,5%	52,7%	53,9%	54,0%

Table 7-N: Average deviation from C_{max} lower bound for INOSETA warping/weaving stage.

⁽¹⁾ they were in reality sequence dependent.

	CDSB		SZWB		NEHB		TowB		VBA	
	R1	R2	R2	R1	R1	R2	R2	R1	R1	R2
1 creel => 2 beams 1 beam => 5 warps	13,5%	14,0%	21,4%	21,6%	20,0%	20,3%	0,4%	0,6%	0,4%	0,6%
1 creel => 4 beams 1 beam => 2 warps	15,0%	15,2%	21,8%	22,0%	20,5%	20,8%	0,3%	0,5%	0,3%	0,5%

Table 7-O: Average deviation from the best \bar{C} for INOSETA warping/weaving stage.

The heuristic grading for makespan criterion is the following:

CDSB	1
NEHB	2
SZWB	3
TowB	4
VBA	5

It is interesting to see that, in opposition to the general case study for makespan criterion, heuristics that minimize makespan criterion do not minimize the mean completion time in this specific case.

7.2.3. THE FINISHING STAGE (FP)

The performance of the heuristics for maximum lateness criterion was tested by solving 384 different problems, involving the two precedence configurations and 6 different problem sizes. For each problem size, 32 tests were generated.

The average deviations of each heuristic value from the lower bound LBI for the L_{max} criterion are given in Table 7-P, and the average deviations from the best obtained value in Table 7-Q.

	CDSB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
17% of jobs have a successor	52%	52%	90%	90%	47%	47%	77%	77%	75%	75%
33% of jobs have a successor	53%	54%	86%	86%	52%	53%	65%	65%	64%	64%

Table 7-P: Average deviation from L_{max} lower bound for INOSETA finishing stage.

	CDSB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
17% of jobs have a successor	6%	6%	44%	44%	3%	3%	33%	33%	30%	30%
33% of jobs have a successor	5%	6%	46%	46%	3%	4%	20%	20%	18%	18%

Table 7-Q: Average deviation from the best L_{max} value for INOSETA finishing stage.

The heuristic grading for maximum lateness criterion is the following:

TowB	1
CdsB	2
VBB	3
VBA	4
NEHB	5

In this particular case, and in opposition to the general case study, TowB gives better results than NEHB; nevertheless its complexity leads us to rather consider CdsB as the best solution. Moreover, CDSB and TowB give often the best obtained values. We note also that VBA and VBB give better results when the number of jobs that have a successor increases

The results for the four other performance measures are presented in Table 7-R. For all the other performance measures, this same grading is preserved.

	CDSB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
17% of jobs have a successor	6%	6%	35%	35%	2%	2%	26%	26%	24%	24%
33% of jobs have a successor	3%	4%	31%	31%	3%	3%	13%	13%	12%	12%

ET_{max} : average deviation of each heuristic from the best value

	CDSB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
17% of jobs have a successor	2%	4%	39%	39%	6%	8%	23%	24%	22%	23%
33% of jobs have a successor	4%	9%	34%	34%	1%	6%	9%	14%	9%	13%

\bar{C} : average deviation of each heuristic from the best value

	CDSB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
17% of jobs have a successor	89%	90%	100%	100%	91%	93%	96%	97%	95%	96%
33% of jobs have a successor	96%	98%	98%	98%	95%	97%	96%	98%	96%	98%

Nt : average percentage of tardy jobs

	CDSB		NEHB		TowB		VBA		VBB	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
17% of jobs have a successor	5%	6%	64%	64%	6%	7%	38%	38%	35%	35%
33% of jobs have a successor	6%	15%	64%	65%	3%	10%	18%	24%	16%	23%

\bar{T} : average deviation of each heuristic from the best value

Table 7-R: Results for other performance measures for INOSETA finishing stage.

7.3. CONCLUSIONS & RECOMMENDATIONS

This experimented framework allows us to assess the quality and the robustness of the six heuristics proposed in pure and hybrid flow shop environments, for makespan and maximum lateness criteria, and with respect to several constraints.

A comment on computational times: we noticed that CDSB, SzWB, VBA and VBB calculations are completed in less than one second for every problem studied, whatever the size; their algorithms are in $o(N^2)$ or $o(V.N^2)$. On the other hand, NEHB and TowB require more computational time:

- for a pure flow shop problem with 99 jobs and 8 machines, NEHB takes about 20 seconds on a 486-66Mhz, while TowB takes about 8 minutes.
- for a hybrid flow shop with 99 jobs, 5 workshops, and between 1 and 3 machines per workshops, NEHB takes about 20 seconds with R1 assignment rule and 35 seconds with the R2 rule, while TowB takes around 4 minutes whatever the assignment rule.

In conclusion, we can formulate some recommendations with respect to the minimization criterion and the workshop layout (single machine or multi-machines) for scheduling problems that are similar to the configurations studied:

	C_{\max} criterion	L_{\max} criterion
pure flow shop	CdsB, CdsB2	NEHB, TOWB
hybrid flow shop	CdsB2, CdsB with R1	CdsB, then NEHB, TOWB with R1

For the makespan criterion, we observe that CDS-based heuristics are the most robust when faced with constraint integration for both pure and hybrid flow shop configurations.

For maximum lateness criterion, we note that NEHB is the best heuristic, significantly better than TOWB for the constrained problems; This is very interesting because it is a quicker heuristic. With the introduction of the hybrid dimension, this result do not remain true: CdsB performs the best.

For INOSETA case study, we advise the use of CdsB in both cases : warping/weaving stage with makespan criterion, and finishing stage with maximum lateness criterion.

The results obtained in all these case studies show that one must be careful when generalizing theoretical conclusions; in reality, heuristic grading based on theoretical case studies may differ from the performances obtained in specific real cases.

CHAPTER 8

CONCLUSIONS & PERSPECTIVES

8.1. CONCLUSIONS	130
8.2. PERSPECTIVES	131
8.3. TOWARDS A DECISION ENGINEERING SYSTEM FOR PRODUCTION PLANNING & PRODUCTION ACTIVITY CONTROL	131

8. CONCLUSIONS & PERSPECTIVES

8.1. CONCLUSIONS

We have designed a suitable method for production system analysis in order to identify and specify scheduling problems. This method allows selection of appropriate scheduling tools, and design of simulation models for the production system; this multi-model approach helps to assess the feasibility and robustness of the scheduling tool in virtual situations, and to verify the quality of the solution(s). Our methodology is based on a top-down analysis of resources which produces the requirement model, a bottom-up recognition of resources which produces the object model, the instantiating and integration of objects which lead to the conceptual model, and implementation of objects i.e. the heuristics selection and model simulation.

This approach has been successfully used to analyze and model a production system for a textile company, and enables us to identify its complex scheduling problems.

We have studied scheduling problems in hybrid flow shop environments, which are representative of process industries. General characteristics of hybrid flow shops were modeled and many restrictions encountered in process industries were identified such as release dates, setup and removal times, time lags restrictions, storage constraints, and bill of materials constraints (precedence constraints). Makespan and maximum lateness criteria were investigated in order to take into account make-to-stock and make-to-order management environments respectively. Twelve scheduling heuristics were developed or adapted from previous research, and improved to solve previous constrained problems. They have been tested with success on 17 600 different problems.

We have proposed a scheduling framework to help production planners and material managers in their planning and scheduling decision-making and we applied our investigations to a textile company. This industrial application consisted of identifying the production system, proposing a formal model, designing planning and scheduling tools, testing the tools in both theoretical and industrial contexts, designing simulation models, and applying the whole scheduling framework to the firm.

8.2. PERSPECTIVES

Concerning the INOSETA case study, simulation models and scheduling tools were designed. The robustness of these tools when faced with setup and removal times, different workshop calendars, minimum time lags and precedence constraints, was assessed by numerous experiments. The next steps will be:

- The implementation of the scheduling tools in the simulation model to evaluate the robustness of these same tools when faced with characteristics that have not previously been taken into account and unforeseen events. This stage will allow an industrial validation of our approach.
- The industrial implementation of the proposed scheduling framework in the information system of the company.

In regards to our research work, this thesis opens several research tracks in the field of Operational Research:

- Extension of assignment rules to all the constraints that have been modeled. This will allow us to use designed heuristics more efficiently to solve real problems.
- Extension of the designed scheduling tools to hybrid flow shops with uniform or unrelated machines.
- Extension of the experiments to other kinds of precedence constraints (in-tree, arbitrary network), to other criteria (for instance, those that have been just computed), with the aim of determining the discriminating characteristics of a scheduling problem. This could enable the research community to make a user-guide for scheduling tools.

8.3. TOWARDS A DECISION ENGINEERING SYSTEM FOR PRODUCTION PLANNING & PRODUCTION ACTIVITY CONTROL

We have generalized and formalized our approach to design a decision engineering system for production planning & production activity control [Artiba & *al.*; 1996-b]. This framework, called GEPETO (GENie Productique Emulant des Techniques d'Ordonnancement), will implement OASISS methodology. It will enable the identification of planning/scheduling problems, and will propose an automatic system to generate scheduling tools and simulation models of a system, i.e. a production activity control tool.

The two first steps of the OASISS methodology will be implemented with modules Geppetto and Blue Fairy, used for the top-down analysis and bottom-up recognition of object classes respectively; and the instantiating and integration of objects will be done using two libraries: Pinocchio et Jiminy Cricket, used for simulation models and scheduling tools respectively. An architecture of the system is proposed in Figure 8-A.

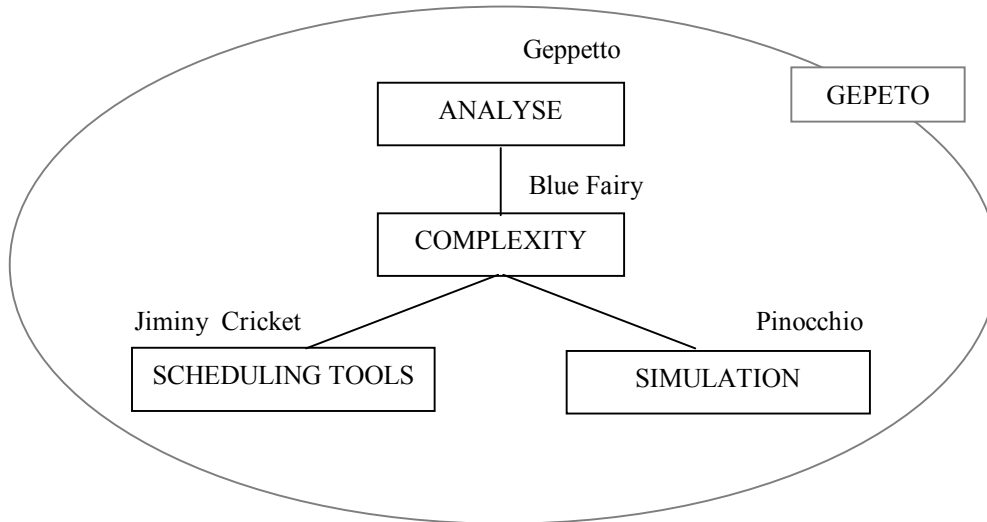


Figure 8-A : Architecture of GEPETO.

This project is as part of the INSA-Lyon and Claude Bernard University PRISMa laboratory (France) and the FUCaM CREGI research center (Belgium).

BIBLIOGRAPHIC REFERENCES

9.1. BIBLIOGRAPHIC REFERENCES	134
9.2. COMMERCIAL SOFTWARE REFERENCES	153

9.1. BIBLIOGRAPHIC REFERENCES

[Abbott; 1986]

ABBOTT R.J., *An integrated approach to software development*; Wiley & Sons (1986).

[Adiri & Pohoryles; 1982]

ADIRI I. and D. POHORYLES, flowshop/no-idle or no-wait scheduling to minimize the sum of completion times, *Naval Research Logistics Quarterly*, vol.29, N°3, 495-504, (1982).

[Aghezzaf & al.; 1995]

AGHEZZAF E., A. ARTIBA and S.E. ELMAGHRABY, Hybrid flowshops : an LP based heuristic for the planning level problems, ETFA, Paris (France), Proceedings 551-559, october 1995.

[Alidaee & Ahmadian; 1993]

ALIDAE B. and A. AHMADIAN, Two parallel machine sequencing problems involving controllable job processing times, *European Journal of Operational Research*, vol.70, 335-341 (1993).

[Alidaee & Panwalkar; 1993]

ALIDAE B. and S.S. PANWALKAR, Single stage minimum absolute lateness problem with a common due date on non-identical machines, *Journal of the Operational Research Society*, vol.44, N°1, 29-36 (1993).

[Allweyer & al.; 1996]

ALLWEYER T., P. LOOS, A.-W. SCHEER, Requirements and new concepts for production planning and scheduling in the process industries, I-CIMPRO'96, Eindhoven, The Netherland, Proceedings J.C.Fransoo, W.G.M.M.Rutten Eds, 4-17, june 1996.

[Anthony; 1965]

ANTHONY R.N., *Planning and Control Systems: A framework for analysis*, Harward University, Graduate School of Business Administration, Boston (1965).

[Arkin & Roundy; 1991]

ARKIN E.M. and R.O. ROUNDY, Weighted-tardiness scheduling on parallel machines with proportional weights, *Operations Research*, vol.39, N°1, 64-81 (1991).

[Arod; 1996]

AROD M., Mise en place d'un suivi de fabrication par code à barre, Projet de fin d'études, INSA, juin 1996, Confidentiel Industrie.

[Arthanary & Ramaswamy; 1971]

ARTHANARY T.S. and K.G. RAMASWAMY, An extension of two machine sequencing problems, *Opsearch*, vol.8, 10-22 (1971).

[Artiba & Aghezzaf; 1996]

ARTIBA A. and E. AGHEZZAF, An architecture of a multi-model system for planning and scheduling, *Int. J. Computer Integrated Manufacturing*, to appear (1996).

9. BIBLIOGRAPHIC REFERENCES

- [Artiba & al.; 1996-a]
ARTIBA A., E. AGHEZZAF, O. KAUFMANN and P. LEVECQ, An application of planning and scheduling multi-model approach in chemical industry, I-CIMPRO'96, Eindhoven, The Netherland, Proceedings J.C.Fransoo, W.G.M.M.Rutten Eds, 33-46, june 1996.
- [Artiba & al.; 1996-b]
ARTIBA A., V. BOTTA, A. GUINET, P. LEVECQ, N MEBARKI et F. RIANE, Conception d'un Atelier de Génie Décisionnel pour le Pilotage des Systèmes de Production, Projet Tournesol N° 96087, Rapport d'Etape, 11p, Août 1996.
- [Artiba & al.; 1997]
ARTIBA A., V. BOTTA, A. GUINET, H. HENTOUS, P. LEVECQ and F. RIANE, A unified framework and a classification scheme to model production systems, ICOQM, Jaipur (India), January 1997 (proceedings to appear)
- [Artiba & Elmaghraby; 1996]
ARTIBA A. and S.E. ELMAGHRABY, *The planning and the scheduling of production systems: Methodologies and Applications*, Chapman & Hall, to appear (1996).
- [Artiba; 1994]
ARTIBA, A., A rule-based planning system for parallel multiproduct manufacturing lines, *Production Planning & Control*, vol.5, N°4, 349-359 (1994).
- [Artigues & al.; 1996]
ARTIGUES C., F. ROUBELLAT and J.-C. BILLAUT, Characterization of a set of schedules in resource-constrained multi-project scheduling problem with multiple modes, Workshop on Production Planning and Control, Mons (Belgium), Proceedings FUCaM Eds, 251-254, September 1996.
- [Baker & Merten; 1973]
BAKER K.R. and A.G. MERTEN, Scheduling with parallel processors and linear delay costs, *Naval Research Logistics Quarterly*, vol.20, N°4, 193-204 (1973).
- [Baker; 1974]
BAKER R.B., *Introduction to Sequencing and Scheduling*, John Wiley & Sons Inc, New York (1974).
- [Banks & al.; 1995]
BANKS J., B. BURNETT, J.D. ROSE, and H. KOZLOSKI, *SIMAN V and CINEMA V*; New York, John Wiley (1995).
- [Banks; 1994]
BANKS J., Software for Simulation, Winter Conference, Proceeding T.D.Tew, S.Manivannan, D.A.Sadowski, A.F.Seila Eds., 26-33, 1994.
- [Bansal; 1977]
BANSAL S.P., Minimizing the sum of completion times of n jobs over m machines in a flowshop - a branch and bound approach, *AIIE Transactions*, vol.9, N°3, 306-311 (1977).

9. BIBLIOGRAPHIC REFERENCES

[Baptiste & al.; 1994]

BAPTISTE P., B. LEGEARD, M.A. MANIER and C. VARNIER, A scheduling problem optimisation solved with constraint logic programming, 2nd International Conference on Pratical Application of Prolog, London, April 1994.

[Barnes & Brenan; 1977]

BARNES J.W. and J.J. BRENNAN, An improved algorithm for scheduling jobs in identical machines, *AIIE Transactions*, vol.9, N°1, 25-31 (1977).

[Bedworth & Bailey, 1982]

BEDWORTH D.D. and J.E. BAILEY, *Integrated Production Control Systems*, John Wiley & Sons Inc, New York (1982).

[Bedworth & Bailey; 1987]

BEDWORTH D.D. and J.E. BAILEY, *Integrated Production Control Systems : Management, Analysis, Design 2/E*, John Wiley & Sons Inc, New York (1987).

[Berio & al.; 1995]

BERIO G., A. DI LEVA and F. VERNADAT, The M*-Object Methodology for Information System Design in CIM environments, *IEEE Transactions on Systems, Man and Cybernetics*, vol.25, N°1, 68-85 (1995).

[Bertrand & al.; 1990]

BERTRAND J.W.M, J.C. WORTMANN and J. WIJNGAARD, *Production Control: a structural and design oriented approach*, Elsevier, Amsterdam (1990).

[Billaut & al.; 1995]

BILLAUT J.-C., F. ROUBELLAT and M. VILLAUMIE, Prise en compte de l'affectation dans le logiciel d'ordonnancement temps réel ORDO. Journées d'Etude "Affectation et Ordonnancement", GR Automatique du CNRS, Tours (France), Actes 113-131, Septembre 1995.

[Billaut & al.; 1996]

BILLAUT J.-C., A. HOUNGBOSSA, D. VEILLAT and V. T'KINDT, An efficient algorithm to minimize the maximum tardiness in a 2-stage hybrid flowshop scheduling problem, Workshop on Production Planning and Control, Mons (Belgium), Proceedings 307-310, September 1996.

[Bitran & Tirupati; 1988]

BITRAN G.R. and D. TIRUPATI, Planning and scheduling for epitaxial wafer production facilities, *Operations Research*, vol.36, N°1, 34-49 (1988).

[Blazewicz & al.; 1994-a]

BLAZEWICZ J.K. ECKER, G. SCHMIDT and J. WEGLARZ, *Scheduling in Computer an Manufacturing Systems*, Springer-Verlag Eds (1994).

[Blazewicz & al.; 1994-b]

BLAZEWICZ J., M. DROR, G. PAWLAK, and K.E. STECKE, A note on Flexible Flowshop Scheduling with two-stages, Internal Report, Technical University of Poznan, Institute of Computing Science, Poznan, (Poland), 1994.

9. BIBLIOGRAPHIC REFERENCES

[Boffet; 1996]

BOFFET L., 1996, Aide à l'analyse et à la conception de systèmes de gestion de production, Projet de fin d'études, INSA, juin 1996, Confidentiel Industrie.

[Booch; 1991]

BOOCH G., *Object oriented design with applications*, Benjamin Cumming (1991).

[Bossink; 1992]

BOSSINK G., *Planning and Scheduling for flexible discrete parts manufacturing*, PhD. Thesis, University of Twente, Enschede, The Netherlands (1992).

[Botta & al.; 1996]

BOTTA V., A. GUINET and D. BOULLE, Object-oriented analysis with structured and integrated specifications and solutions for production system control, *Journal of Intelligent Manufacturing*, to appear (1996).

[Botta & Guinet; 1995]

BOTTA V. and A. GUINET, Production scheduling problems modelling and heuristics in textile industry, 3rd IFAC/IFIP/IFORS Workshop on Intelligent Manufacturing systems, IMS'95, Bucharest (Romania), Proceedings 281-286, October 1995.

[Botta & Guinet; 1996]

BOTTA V. and A. GUINET, Scheduling flowshops with precedence constraints and time lags. Workshop on Production Planning and Control, Mons (Belgique), Proceedings 16-19, September 1996.

[Botta; 1994-a]

BOTTA V., INOSETA: description du système de production, Rapport Technique, Novembre 1994, 40p., Confidentiel Industrie.

[Botta; 1994-b]

BOTTA V., Rapport de Recherche pour la Région Rhône-Alpes, 1ère année de thèse, 10p., mai 1994.

[Botta; 1995-a]

BOTTA V., INOSETA: analyse et cahier des charges, Rapport Technique, mars 1995, 36p., Confidentiel Industrie.

[Botta; 1995-b]

BOTTA V., INOSETA: Identification et modélisation des problèmes d'ordonnancement, Rapport Technique, juin 1995, 21p., Confidentiel Industrie.

[Botta; 1995-c]

BOTTA V., Rapport de Recherche pour la Région Rhône-Alpes, 2ème année de thèse, 10p., mai 1995.

[Botta; 1996]

BOTTA V., Rapport de Recherche pour la Région Rhône-Alpes, 3ème année de thèse, 10p., mai 1996.

[Bouzeghoub; 1994]

BOUZEGHOUB M., Merise Object : les modèles statique et dynamique revisités, Rapport Technique, Projet Kheops, laboratoire PRISM (1994).

9. BIBLIOGRAPHIC REFERENCES

[Brah & Hunsucker; 1991]

BRAH S.A. and J.L. HUNSUCKER, Branch and Bound algorithm for the flow shop with multiple processors, *European Journal of Operational Research*, vol. 51, 88-99 (1991).

[Bravoco & Yadan; 1985-a]

BRAVOCO R.R. and S.B. YADAN, A methodology to model the functional structure of an organization. *Computers in Industry*, vol.6, 345-361 (1985).

[Bravoco & Yadan; 1985-b]

BRAVOCO R.R. and S.B. YADAN, A methodology to model the information structure of an organization, *Journal of Systems and Software*, vol.5, N°1, 59-71 (1985).

[Bravoco & Yadan; 1985-c]

BRAVOCO R.R. and S.B. YADAN, A methodology to model the dynamic structure of an organization, *Information Systems*, vol.10, N°3, 299-317 (1985).

[Brucker & al.; 1977]

BRUCKER P., M.R. GAREY and D.S. JOHNSON, Scheduling equal-length tasks under treelike precedence constraints to minimize lateness, *Mathematics of Operations research*, vol.2, N°3, 275-284 (1977).

[Bruno & al.; 1974]

BRUNO J., J.R. COFFMAN and R. SETHI, Scheduling independent tasks to reduce mean finishing time, *Communications of the ACM*, vol.17, N°7, 382-387 (1974).

[Buten & Shen; 1973]

BUTEN R.E. and V.Y. SHEN, A scheduling model for computer systems with two classes of processors, Sagamore Computer Conference on Parallel Processing, Proceedings 130-138 (1973).

[Campbell & al.; 1970]

CAMPBELL H.G., R.A. DUDEK and M.L. SMITH, A heuristic algorithm for the n job, m machine sequencing problem, *Management Science*, vol.16, N°10, B630-B637 (1970).

[Carlier & Chretienne; 1988]

CARLIER J. et P. CHRETIENNE, *Problèmes d'ordonnancement : modélisation / complexité / algorithmes*, Editions Masson, Paris (1988).

[Carlier; 1987]

CARLIER J., Scheduling jobs with release dates and tails on identical machines to minimize makespan, *European Journal of Operational Research*, vol.29, 298-306 (1987).

[Carvalho & al.; 1990]

CARVALHO L., R. ALMEIDA, C. BISPO and J. SENTIEIRO, An integrated environment for planning and scheduling in flow-shop manufacturing plants, IEEE Internatinal Conference on Computer Integrated Manufacturing, Rensselaer, Troy, New York, proceedings 55-62, May 1990.

9. BIBLIOGRAPHIC REFERENCES

[Chen; 1995]

CHEN B., Analysis of classes of heuristics for scheduling a two-stage flowshop with parallel machines at one stage, *Journal of the Operational Research Society*, vol.46, 234-244 (1995).

[Cheng & Sin; 1990]

CHENG T.C.E. and C.C.S. SIN, A state of art review of parallel machine scheduling research, *European Journal of Operation Research*, vol.47, 271-292 (1990).

[Cho & Sahni; 1980]

SO Y. and S. SAHNI, Bounds for list schedules on uniform processors, *SIAM J. Compt.*, vol.9, 91-103 (1980).

[Coad & Yourdon; 1991]

COAD P. and E. YOURDON, *Object-Oriented Analysis*, Second Edition, Yourdon Press, Prentice Hall Building Englewood Cliffs (1991).

[Collins & Watson; 1993]

COLLINS N. and C.M. WATSON, Introduction to ARENA, 1993 Winter Simulation Conference, Proceedings, G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles Eds, 205-212, Association for Computer Machinery, New York (1993).

[Conway & al.; 1967]

CONWAY R.W., W.L. MAXWELL and L.W. MILLER, *Theory of scheduling*, Reading, Massachusetts, Addison-Wesley Publishing Company (1967).

[Cox; 1986]

COX B.S., *Object oriented programming : a evolutionary approach*, Addison Wesley (1986).

[Daniels & Mazzola; 1994]

DANIELS R.L. and J.B. MAZZOLA, Flow shop scheduling with resource flexibility, *Operations Research*, vol.42, N°3, 504-522 (1994).

[Dannenbring; 1977]

DANNENBRING D.G., An evaluation of flowshop sequencing heuristics, *Management Science*, vol.23, N°11, 1174-1182 (1977).

[Dantzig; 1962]

DANTZIG G.B., *Linear Programming and Extentions*, Princeton University Press, Princeton, New Jersey (1962).

[Das & al.; 1995]

DAS S.R., J.N.D. GUPTA and B.M. KHUMAWALA, A savings index heuristic algorithm for flowshop scheduling with sequence dependent setup times, *Journal of the Operational Research Society*, vol.46, 1365-1373 (1995).

[Deal & Hunsucker; 1991]

DEAL D.E. and J.L. HUNSUCKER, The two stage flowshop scheduling problem with M machines at each stage, *Journal of Information and Optimization Sciences*, vol.12, 407-471 (1991).

9. BIBLIOGRAPHIC REFERENCES

- [Delatte & al.; 1993]
DELATTE B., M. HEITZ and J.F. MULLER, *Hood Reference Manual 3.1*, Masson, Paris (1993).
- [DeMarco; 1978]
DE MARCO T., *Structured Analysis and System specification*, Prentice-Hall, New York (1978).
- [DeMarco; 1982]
DE MARCO T., *Structured Analysis and Design*, Prentice-Hall, New York (1982).
- [Dogramaci & Surkis; 1979]
DOGRAMACI A. and J. SURKIS, Evaluation of a heuristic for scheduling independent jobs on parallel identical processors, *Management Science*, vol.25, N°12, 1208-1216 (1979).
- [Dogramaci; 1984]
DOGRAMACI A., Production scheduling of independent jobs on parallel identical processors, *International Journal of Production Research*, vol.22, N°4, 535-548 (1984).
- [Doumeigst; 1984]
DOUMEINGST G., *Méthode de conception des systèmes en productique*, thèse d'état, Bordeaux (1984).
- [Doumeigst; 1990]
DOUMEINGST G., Méthodes pour concevoir et spécifier les systèmes de production. CIM'90, Bordeaux, Proceedings 89-103 (1990)
- [Dubois & Michels; 1996]
DUBOIS D. and F. MICHELS, Le cas Louis de Poortere, Projet de fin d'études, FUCaM, année académique 1995-1996, Confidentiel Industrie.
- [Dudeck & al.; 1992]
DUDEK R.A., S.S. PANWALKAR and M.L. SMITH, The lessons of flowshop scheduling research, *Operations Research*, vol.40, N°1, 7-13 (1992).
- [Echalier; 1991]
ECHALIER F., Problèmes d'ordonnancement sur machines parallèles : apport du recuit simulé, Thèse de Doctorat, Université Claude Bernard - Lyon I, (1991).
- [Egbelu; 1991]
EGBELU P.J., Batch production time in a multi-stage system with material-handling consideration, *International Journal of Production Research*, vol.29, N°4, 695-712 (1991).
- [Elmaghraby & Guinet; 1992]
ELMAGHRABY S.E. and A. GUINET, Scheduling on parallel machines: an alternative approach, Third International Workshop on Project Management and Scheduling, Como, (Italy) 1992.
- [Elmaghraby & Karnoub; 1995]
ELMAGHRABY S.E. and R.E. KARNOUB, Production control in flexible flowshops : an example from textile manufacturing, OR Report N°305 (1995).

9. BIBLIOGRAPHIC REFERENCES

[Elmaghraby & Park; 1974]

ELMAGHRABY S.E. and S.H. PARK, Scheduling jobs on a number of identical machines, *AIIE Transactions*, vol.6, 1-13 (1974)

[Esquirol; 1994]

ESQUIROL P., Programmation logique sous contraintes et Ordonnancement, Journées d'étude « Ordonnancement et Entreprise » du GR Automatique du CNRS, Toulouse (France), Actes 127-165, juin 1994.

[Federgruen & Groenenvelt; 1986]

FEDERGRUEN A. and H. GROENENVELT, Preemptive scheduling on uniform machines by ordinary network flow technics, *Management Science*, vol.32, N°3, 341-349 (1986).

[Feo & al.; 1995]

FEO T.A., J.F. BARD and S.D. HOLLAND, Facility-wide planning and scheduling of printed wiring board assembly, *Operations Research*, vol.43, N°2, 219-230 (1995).

[Fleury; 1994]

FLEURY G., Application du recuit simulé et de ses variantes à des problèmes d'ordonnancement, Journées d'étude « Ordonnancement et Entreprise » du GR Automatique du CNRS, Toulouse (France), Actes 69-95, juin 1994.

[Fransoo; 1996]

FRANSOO J.C., Medium-term planning in batch process industries, I-CIMPRO'96, Eindhoven, The Netherland, Proceedings J.C.Fransoo, W.G.M.M.Rutten Eds, 244-254, june 3-4, 1996.

[French; 1982]

FRENCH S., *Sequencing and Scheduling : An introduction to the Mathematics of the Job-Shop*, Horwood, Chichester (1982).

[Frenk & Rinnooy Kan; 1987]

FRENK J.B.G. and A.H.G. RINNOOY KAN, The asymptotic optimality of the LPT rule, *Mathematics of Operations Research*, vol.12, 241-254 (1987).

[Gaches & al.; 1990]

GACHES R., B. QUERENET, P. VIOLLET and F. VERNADAT, Une architecture ouverte pour la productique, CIM'90, Bordeaux, Proceedings 227-234, 1990.

[Gane & Sarson; 1979]

GANE C. and T. SARSON, *Structured Systems Analysis*, Prentice-Hall, New York (1979).

[Garey & Johnson; 1979]

GAREY M.R. and D.S. JOHNSON, *Computers and Intractability, A guide to the theory of NP-Completness*, Bell Telephone Laboratories, Inc, USA (1979).

[Gelders & Sambandam; 1978]

GELDERS L.F. and N. SAMBANDAM, Four simple heuristics for scheduling a flow-shop, *International Journal of Production Research*, vol.16, N°3, 221-231 (1978).

9. BIBLIOGRAPHIC REFERENCES

[GOTHA; 1993]

GOTHA, Les problèmes d'ordonnancement, *R.A.I.R.O. Recherche Opérationnelle / Operations Research*, vol.27, N°1, 77-150 (1993).

[Grabowski & al.; 1983]

GRABOWSKI J., E. SKUBALSKA and C. SMUTNICKI, On flowshop scheduling with release and due dates to minimize maximum lateness, *Journal of the Operational Research Society*, vol.34, 615-620 (1983).

[Grabowski; 1980]

GRABOWSKI J., On two-machine scheduling with release and due dates to minimize maximum lateness, *Opsearch*, vol.17, N°4, 133-154 (1980).

[Graham & al.; 1979]

GRAHAM R.L., E.L. LAWLER, J.K. LENSTRA and A.H.K. RINNOOY KAN, Optimization and approximation in deterministic sequencing and scheduling: a survey, *Annals of Discrete Mathematics*, vol.5, 287-326 (1979).

[Graham; 1969]

GRAHAM R.L., Bounds on multiprocessing timing anomalies, *SIAM Journal of Applied Mathematics*, vol.17, 416-429 (1969).

[Graves; 1981]

GRAVES S.C., A review of production scheduling, *Operations Research*, vol.29, N°4, 646-675 (1981).

[Gregor; 1996]

GREGOR S., La simulation d'un système de production « ennoblissement » du secteur textile, Projet de fin d'études, INSA, juin 1996, Confidentiel Industrie.

[Guinet & al.; 1992]

GUINET A., F. ECHALIER and A. DUSSAUCHOY, Scheduling jobs on parallel machines : a survey, EURO 12, TIMS XXXI, Helsinki, (Finland), 1992.

[Guinet & al.; 1995]

GUINET A., V. BOTTA et M. SOLOMON, Minimisation du plus grand retard ou de la plus grande date d'achèvement dans les problèmes d'ordonnancement de type flowshop hybride, Journées d'Etude « Affectation et Ordonnancement », GR Automatique du CNRS, Tours (France), Actes 95 - 111, Septembre 1995.

[Guinet & al.; 1996]

GUINET A., M. SOLOMON, P. KEDIA and A. DUSSAUCHOY, Scheduling jobs on two serial workshops with parallel processors, *International Journal of Production Research*, vol.34, N°5, 1399-1415 (1996).

[Guinet & Botta; 1995-a]

GUINET A. and V. BOTTA, Object oriented Analysis with Structured and Integrated Specifications and Solutions for production system control, International Conference on Industrial Engineering and Production Management, IEPM'95, Marrakech (Maroc), Proceedings I 171-182, avril 1995.

9. BIBLIOGRAPHIC REFERENCES

[Guinet & Botta; 1995-b]

GUINET A. et V. BOTTA, Réduction de Problèmes et Pilotage des Systèmes de Production, Journées Francophones de la Recherche Opérationnelle / FRANCORO, Mons (Belgique), juin 1995.

[Guinet & Botta; 1995-c]

GUINET A. et V. BOTTA, Changement de modèles et pilotage des systèmes de production, Congrès International de Génie Industriel de Montréal "La productivité dans un monde sans frontières", Montréal (Canada), Actes 2019-2028, Editions de l'Ecole Polytechnique de Montréal, octobre 1995.

[Guinet & Solomon; 1996]

GUINET A. and M. SOLOMON, Scheduling hybrid flowshop to minimize maximum tardiness or maximum completion time, *International Journal of Production Research*, vol.34, N°6, 1643-1654 (1996).

[Guinet; 1990]

GUINET A., Knowledge Acquisition and Assessment about Production Management Systems, *European Journal of Operational Research*, vol.45, n° 2-3, 265-274 (1990).

[Guinet; 1991]

GUINET A., Textile Production Systems : a Succession of Non-Identical Parallel Processor Shops, *Journal of the Operational Research Society*, vol.42, 655-671 (1991).

[Guinet; 1993]

GUINET A., Scheduling sequence-dependent jobs on identical parallel machines to minimize completion time criteria, *International Journal of Production Research*, vol.31, N°7, 1579-1594 (1993).

[Guinet; 1995]

GUINET A., Scheduling independent jobs on uniform parallel machines to minimize tardiness criteria, *Journal of Intelligent Manufacturing*, vol.6, 95-103 (1995).

[Guinet; 1996]

GUINET A., Integrated and sequential approaches to schedule hybrid flowshop in make-to-stock environment, Workshop on Production Planning and Control, Mons, Belgium, Proceedings 299-302, september 1996.

[Gupta & Tunc; 1991]

GUPTA J.N.D. and E.A. TUNC, Schedules for a two-stage hybrid flowshop with parallel machines at the second stage, *International Journal of Production Research*, vol.29, N°7, 1489-1502 (1991).

[Gupta & Tunc; 1994]

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

[Gupta; 1988]

GUPTA J.N.D., Two stage hybrid flowshop scheduling problem, *Journal of the Operational Research Society*, vol.39, N°4, 359-364 (1988).

9. BIBLIOGRAPHIC REFERENCES

[Gusfield; 1984]

GUSFIELD D., Bounds for naive multiple machine scheduling with release times and deadlines, *J. Algorithms*, vol.5, 1-6 (1984).

[Harrison; 1996]

HARRISON M.C., The selection of planning and scheduling systems in the food processing industries, I-CIMPRO'96, Eindhoven, The Netherlands, Proceedings J.C.Fransoo, W.G.M.M.Rutten Eds, 298-312, June 1996.

[He & al.; 1996]

HE D.W., A. KUSIAK and A. ARTIBA, A scheduling problem in glass manufacturing, *IIE Transactions*, vol.28, 129-139 (1996).

[He & Kusiak; 1992]

HE W. and A. KUSIAK, Scheduling manufacturing systems, *Computers in Industry*, vol.20, 163-157 (1992).

[Hendry & al.; 1996]

HENDRY L.C., K.K. FOK and K.W. SHEK, A cutting stock and scheduling problem in the copper industry, *Journal of the Operational Research Society*, vol.47, 38-47 (1996).

[Hentous & Guinet; 1996]

HENTOUS H. and A. GUINET, A constraint hybrid flowshop problem, Workshop on Production Planning and Control, Mons (Belgium), Proceedings 303-306, september 1996.

[Herault; 1994]

HERAULT L., Réseaux neuromimétiques pour les problèmes d'ordonnancement - application à l'allocation de ressources, Journées d'étude « Ordonnancement et Entreprise » du GR Automatique du CNRS, Toulouse (France), Actes 167-212, Juin 1994.

[Ho & Chang; 1991]

HO J.C. and Y.-L. CHANG, A new heuristic for the n-job, M-machine flowshop problem, *European Journal of Operational Research*, vol.52, 194-202 (1991).

[Horowitz & Sahni; 1976]

HOROWITZ E. and S. SAHNI, Exact and approximate algorithms for scheduling non-identical processors, *Journal of the Association for Computing Machinery*, vol.23, 317-327 (1976).

[Hu; 1961]

HU T.C., Parallel sequencing and assembly line problems, *Operations Research*, vol.9, 841-848 (1961).

[Hundal & Rajgopal; 1988]

HUNDAL T.S. and L.J. RAJGOPAL, An extension of Palmer's heuristic for the flow shop scheduling problem, *International Journal of Production Research*, vol.26, N°6, 1119-1124 (1988).

9. BIBLIOGRAPHIC REFERENCES

[Hunsucker & Shah; 1992]

HUNSUCKER J.L. and J.R. SHAH, Performance of priority rules in a due date flowshop, *OMEGA Int. J. of Mgmt Sci.*, vol.20, N°1, 73-89 (1992).

[Hunsucker & Shah; 1994]

HUNSUCKER J.L. and J.R. SHAH, Comparative performance analysis of priority rules in a constrained flow shop with multiple processors environment, *European Journal of Operational Research*, vol.72, 102-114 (1994).

[Jackson; 1955]

JACKSON J.R., Scheduling a production line to minimize maximum tardiness, Research report 43, Management Science Research Project, University of California, Los Angeles, USA (1955).

[Jaulent; 1991]

JAULENT P., *Manuel SYS-P-O version 2.0*, Microprocess, Courbevoie (1991).

[Johnson; 1954]

JOHNSON S.M., Optimal two- and three-stage production schedules with set-up times included, *Naval Research Logistics Quarterly*, vol.1, 61-68 (1954).

[Jorysz & Vernadat; 1991-a]

JORYSZ H.R. and F. VERNADAT, Defining CIM Enterprise Requirements using CIM-OSA, *IFIP*, Elsevier Science Publishers B.V., North Holland (1991).

[Jorysz & Vernadat; 1991-b]

JORYSZ H.R. and F. VERNADAT, CIM-OSA Part 1 : Total enterprise modelling and function views, *International Journal of Computer Integrated Manufacturing*, vol.3, 144-156 (1991).

[Jorysz & Vernadat; 1991-c]

JORYSZ H.R. and F. VERNADAT, CIM-OSA Part 2 : Information view, *International Journal of Computer Integrated Manufacturing*, vol.3, 157-167 (1991).

[Kim; 1993]

KIM Y.-D., Heuristics for flowshop scheduling problems minimizing mean tardiness, *Journal of the Operational Research Society*, vol. 44., N°1, 19-28 (1993).

[King & Spachis; 1980]

KING J.R. and A.S. SPACHIS., Heuristics for flow-shop scheduling, *International Journal of Production Research*, vol.18, N°3, 345-357 (1980).

[Krone & Steiglitz; 1974]

KRONE M.J. and K. STEIGLITZ, Heuristic-programming solution of a flowshop-scheduling problem, *Operations Research*, vol.22, 629-638 (1974).

[Kusiak & al.; 1994]

KUSIAK A., T.N. LARSON and J.R. WANG, Reengineering of design and manufacturing processes, *Computers Ind. Engng.*, vol.26, N°3, 521-536 (1994).

[Langston; 1987]

LANGSTON M.A., Interstage transportation planning in the deterministic flowshop environment, *Operations Research*, vol.35, N°4, 556-564 (1987).

9. BIBLIOGRAPHIC REFERENCES

- [Lash; 1993]
LASH S., Genetic algorithms for weighted tardiness scheduling on parallel machines, Technical Report 93-01, Northwestern University, Evanston, USA (1993).
- [Lawler & al.; 1989]
LAWLER E.L., J.K. LENSTRA, A.H.G. RINNOOY KAN and D.B. SHMOYS, Sequencing and Scheduling : algorithms and complexity, Report BS-R8909, Center for Mathematics and Computer Science, Amsterdam, The Netherland (1989).
- [Lawler & Labetoulle; 1978]
LAWLER E.L., and J. LABETOULLE, On preemptive scheduling of Unrelated parallel processors, *Journal of the Association for Computing Machinery*, vol.25, N°4, 612-619 (1978).
- [Lee & Vairaktarakis; 1994]
LEE C.Y. and G.L. VAIRAKTARAKIS, Minimizing makespan in hybrid flowshops, *Operations Research Letters*, vol.16, N°3, 149-158 (1994).
- [Leisten; 1990]
LEISTEN R., Flowshop sequencing problems with limited buffer storage, *International Journal of Production Research*, vol.28, N°11, 2085-2100 (1990).
- [Lenstra & al.; 1989]
LENSTRA J.K., D.B. SHMOYS and E. TARDOS, Approximation algorithms for scheduling unrelated parallel machines, *Mathematics Programming*, (1989).
- [Lenstra; 1976]
LENSTRA J.K., *Sequencing by enumerative methods*, Mathematisch Centrum, Amsterdam (1976).
- [Levecq & al.; 1997]
LEVECQ L., V. BOTTA, A. GUINET and A. ARTIBA, Object oriented design of generic simulation models in hybrid flow-shop organisations, ICOQM, Jaipur (India), January 1997 (proceedings to appear).
- [Li & Cheng; 1994]
LI C.-L. and T.C.E. CHENG, The parallel machine Min-Max weighted absolute lateness scheduling problem, *Naval Research Logistics*, vol.41, 33-46 (1994).
- [Liou & Smith; 1992]
LIOU J., and M.L. SMITH, Scheduling flowshop with limited in-process wait, Research report, Texas Technology University, Lubbock, Texas (1992).
- [M'Hallah & Haouari; 1995]
M'HALLAH R. and M. HAOUARI, A heuristic algorithm for two stage hybrid flowshop problem, Research Report, Institut de Recherche en Sciences Informatique et Télécommunications, Tunis (1995).
- [MacCarthy & Liu; 1993]
MACCARTHY B.L., and J. LIU, Addressing the gap in scheduling research : a review of optimization and heuristic methods in production scheduling, *International Journal of Production Research*, vol.31, N°1, 59-79 (1993).

9. BIBLIOGRAPHIC REFERENCES

[Mc Mahon & Lim; 1993]

MAC MAHON G.B. and C.-J. LIM, The two-machine flow shop problem with arbitrary precedence relations, *European Journal of Operational Research*, vol.64, 249-257 (1993).

[Mc Naughton; 1959]

MAC NAUGHTON R., Scheduling with deadlines and loss functions, *Management Science*, N°6, 1-12 (1959).

[Melese; 1972]

MELESE J., *L'analyse modulaire des systèmes de gestion*, Ed. Hommes et Techniques, Boulogne Billancourt (1972).

[Melese; 1982]

MELESE J., Quelques apports des théories de l'organisation et de la pratique de l'AMS à la conception des systèmes d'information. Actes des journées d'études : théories des organisations et nouvelles approches d'informatisation des entreprises, *AFCEP*, Paris, vol.39, 1-7 (1982).

[Mellor; 1966]

MELLOR P., A review of job shop scheduling, *Operational Research Quarterly*, vol.17, 161-171 (1966).

[Moccelin; 1995]

MOCCELLIN J.V., A new heuristic method for the permutation flow shop scheduling problem, *Journal of the Operational Research Society*, vol.46, 883-886 (1995).

[Monma & Sidney; 1979]

MONMA C.L. and J.B. SIDNEY, Sequencing with series-parallel precedence constraints, *Mathematics of Operations Research*, vol.4, N°3, 215-224 (1979).

[Monma; 1982]

MONMA C.L., Linear-time algorithms for scheduling on parallel processors, *Operations Research*, vol.30, N°1, 116-124 (1982).

[Moore; 1968]

MOORE J.M., An n-job, one-machine sequencing algorithm for minimizing the number of late jobs, *Management Science*, vol.15, 102-109 (1968).

[Morrison; 1988]

MORRISSON J.F., A note on LPT scheduling, *Operations Research Letters*, vol.7, 77-79 (1988).

[Nabeshima & Maruyama; 1983]

NABESHIMA I. and S. MARUYAMA, note on the two-machine flowshop scheduling problem with separated setup and clean-up times, times lags, and transportation times, *Rep. Univ. Electr-Comm.*, vol.34, N°1, 29-31 (1983).

[Narasimhan & Mangiameli; 1987]

NARASIMHAN S.L. and P.M. MANGIAMELI, A comparison of sequencing rules for a two-stage hybrid flow shop, *Decision Sciences*, vol.18, 250-265 (1987).

9. BIBLIOGRAPHIC REFERENCES

[Narasimhan & Panwalkar; 1984]

NARASIMHAN S.L. and S.S. PANWALKAR, Scheduling in a two-stage manufacturing process, *International Journal of Production Research*, vol.22, 555-564 (1984).

[Nawaz & al.; 1983]

NAWAZ M., E. ENSCORE and I. HAM, A heuristic algorithm for the m machine, n job flowshop sequencing problem, *OMEGA*, vol.11, N°1, 91-95 (1983).

[Page; 1961]

PAGE E.S., An approach to the sequencing of jobs on machines, *J. Roy. Statistic Society*, vol.23, 484-492 (1961).

[Palmer; 1965]

PALMER D.S., Sequencing jobs through a multi-stage process in the minimum total time. A quick method of obtaining a near optimum, *Operational Research Quarterly*, vol.16, N°1, 101-107 (1965).

[Papadimitriou & Kanellakis; 1980]

PAPADIMITRIOU C.H. and P.C. KANELLAKIS, Flowshop scheduling with limited temporary storage, *Journal of the Association for Computing Machinery*, vol.27, N°3, 533-549 (1980).

[Pellaumail; 1986]

PELLAUMAIL P., *La méthode Axial, conception d'un système d'information*, Ed. d'Organisation, tomes 1 et 2, Paris (1986).

[Pinedo; 1995]

PINEDO M., *Scheduling: Theory, Algorithms and Systems*, Prentice Hall, Englewood Cliffs, New Jersey (1995).

[Portmann; 1988]

PORTMANN M.C., Méthodes de décomposition spatiales et temporelles en ordonnancement de la production, *RAIRO-APII*, vol.22, N°, 439-451 (1988).

[Portmann; 1996]

PORTMANN M.C., Genetic algorithms and scheduling : a state of art and some propositions, Workshop on Production Planning and Control, Mons, Belgium, Proceedings i-xxiv, September 1996.

[Potts; 1985]

POOTS C.N., Analysis of a linear programming heuristic for scheduling unrelated parallel machines, *Discret Applied Mathematics*, vol.10, 155-164 (1985).

[Proth & Xie; 1994]

PROTH J.M. and X. XIE, *Les réseaux de Petri pour la conception et la gestion des systèmes de production*, Eds Masson, Paris (1994).

[Proust & al.; 1991-a]

PROUST C., J.N.D. GUPTA and V. DESCHAMPS, Flowshop scheduling with setup, processing and removal times separated, *International Journal of Production Research*, vol.29, N°3, 479-493 (1991).

9. BIBLIOGRAPHIC REFERENCES

- [Proust & al.; 1991-b]
PROUST C., A. FERREIRA and J. BIJAOU, The interprogramming paradox: the diversity of softwares for an effective integration, International Congress of Industrial Engineering, Tours (France), Proceedings 1157-1166, 1991.
- [Proust; 1992]
PROUST C., Influence des idées de S.M. JOHNSON sur la résolution de problèmes d'ordonnement de type n/m/F, contraintes diverses/ C_{max} , Rapport interne, Laboratoire d'Informatique / E3i, Université François Rabelais, Tours (France), (1992).
- [Rajendran & Chaudhuri; 1992-a]
RAJENDRAN C. and D. CHAUDHURI, Scheduling in n jobs, m stage flowshop with parallel-processors to minimize makespan, *Int. J. of Production Economics*, vol.27, 137-143 (1992).
- [Rajendran & Chaudhuri; 1992-b]
RAJENDRAN C. and D. CHAUDHURI, A multi-stage parallel-processor flowshop problem with minimum flowtime, *European Journal of Operational Research*, vol.57, 111-122 (1992).
- [Rajendran; 1994]
RAJENDRAN C., A no-wait flowshop scheduling heuristic to minimize makespan, *J. Opl Res. Soc.*, vol.45, N°4, 472-478 (1994).
- [Riane & al.; 1996]
RIANE F., A. ARTIBA and S.E. ELMAGHRABY, Sequencing on hybrid two-stages flowshop to minimize makespan, Workshop on Production Planning and Control, Mons (Belgium), Proceedings 93.1-93.8, September 1996.
- [Rinnooy Kan; 1976]
RINNOOY KAN A.H.G., *Machine scheduling problems : Classification, complexity and computations*, Nijhoff, The Hague (1976).
- [Ross; 1977]
ROSS D.T., Structured Analysis (S.A) : a language for communicating ideas, *IEEE Transactions on Software Engineering*, vol.SE-3, N°1, 16-34 (1977).
- [Ross; 1985]
ROSS T.D., Applications and extensions of SADT, *Computer*, 25-34 (1985).
- [Rumbaugh; 1991]
RUMBAUGH E., *OMT - Object Oriented Modeling and Design*, Prentice Hall, New York (1991).
- [Sahni & Cho; 1979]
SAHNI S. and Y. CHO, Nearly on line scheduling of a uniform processor system with release times, *SIAM J. Compt.*, vol.8, 275-285 (1979).
- [Sahni & Cho; 1980]
SAHNI S. and Y. CHO, Scheduling independent tasks with due times on a uniform processor system, *Journal of Assoc. Compt. Machine*, vol.27, 550-563 (1980).

9. BIBLIOGRAPHIC REFERENCES

[Salvador; 1973]

SALVADOR M.S., A solution to a special case of flow shop scheduling problems, in S.E.Elmaghraby (Ed.), *Symposium of the Theory of Scheduling and Applications*, Springer-Verlag, New York (1973).

[Sarin & al.; 1988]

SARIN S.C., S. AHN and A.B. BISHOP, An improved branching scheme for the branch and bound procedure of scheduling n jobs on m parallel machines to minimize total weighted flowtime, *International Journal of Production Research*, vol.26, N°7, 1183-1191 (1988).

[Serafini & Sperenza; 1990]

SERAFINI P. and M.G. SPERANZA, Production scheduling problems in a textile industry, Internal report, Universita degli Studi di Udine, Udine, Italy (1990).

[Shen & Chen; 1972]

SHEN V.Y. and Y.E. CHEN, A scheduling strategy for the flowshop problem in a system with two classes of processors, Conference on Information and Systems Science, Proceeding 645-649, 1972.

[Shlaer & Mellor; 1988]

SHLAER S. and S.J. MELLOR, *Object Oriented Systems Analysis - Modeling the world in data*, Yourdon Press, Prentice Hall (1988).

[Shlaer & Mellor; 1992]

SHLAER S. and S.J. MELLOR, *Object Lifecycles - Modeling the world in states*, Yourdon Press, Prentice Hall (1992).

[Sidney; 1981]

SIDNEY J.B., A decomposition algorithm for sequencing with general precedence constraints, *Mathematics of Operations research*, vol.6, N°2, 190-204 (1981).

[Silva & Valette; 1989]

SILVA M. and R. VALETTE, Petri Nets and Flexible Manufacturing, *Advances in Petri Net*, G. Rozenberg, Springer Verlag, 374-417 (1989).

[Simons; 1983]

SIMONS B., Multiprocessor scheduling of unit time jobs with arbitrary release times and deadlines, *SIAM J. Computer*, vol.12, 294-299 (1983).

[Slomp; 1993]

SLOMP J., Production control for flexible manufacturing systems : An application-oriented approach, PhD. Thesis, University of Twente, Enschede, Netherlands (1993).

[Smith; 1956]

SMITH W.E., Various optimizers for single-stage production, *Naval Research Logistics Quarterly*, vol.3, 56-66 (1956).

[So; 1990]

SO K.C., Some heuristics for scheduling jobs on parallel machines with setups, *Management Science*, vol.36, N°4, 467-475 (1990).

9. BIBLIOGRAPHIC REFERENCES

[Srikar & Ghosh; 1986]

SRIKAR B.N. and S. GHOSH, A MILD model for the n-job, M stage flowshop with sequence dependent set-up times, *International Journal of Production Research*, vol.24, N°6, 1459-1474 (1986).

[Sriskandarajah & Sethi; 1989]

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

[Sriskandarajah; 1993]

SRISKANDARAJAH C., Performance of scheduling algorithms for no-wait flowshops with parallel machines, *European Journal of Operational Research*, vol.70, 365-378 (1993).

[Stevenson; 1993]

STEVENSON J., *Production / Operation management*, Fourth Edition, R.D.IRWIN Eds (1993).

[Sule & Huang; 1983]

SULE D.R. and K.Y. HUANG, Sequency on two and tree machines with setup, processing and removal time separated, *Naval Research Logistics Quarterly*, vol.29, 517-519 (1983).

[Szwarc & Gupta; 1987]

SZWARC W. and J.N.D. GUPTA, A flowshop problem with sequence-dependent additive setup times, *Naval Research Logistics Quarterly*, vol.34, 619-627 (1987).

[Szwarc; 1983]

SZWARC W., Flowshop problems with time lags, *Management Science*, vol.29, N°4, 477-481 (1983).

[Tahmassebi & Hindi; 1996]

TAHMASSEBI T. and K.S. HINDI, Detailed scheduling of packing system with sequence dependent changeovers, I-CIMPRO'96, Eindhoven, The Netherlands, Proceedings J.C.Fransoo, W.G.M.M.Rutten Eds, 513-525, June 1996.

[Taillard; 1990]

TAILLARD E., Some efficient heuristic methods for the flowshop sequencing problem, *European Journal of Operational Research*, vol.47, 65-74 (1990).

[Tardieu & al.; 1983]

TARDIEU H., O. ROCHFELD et R. COLLETTI, *La méthode Merise - tome 1, principes et outils*, Ed. d'Organisation, Paris (1983).

[Tardieu & al.; 1985]

TARDIEU H., G. PANET et G. VAHEE, *La méthode Merise - tome 2, démarche et pratique*, Ed. d'Organisation, Paris (1985).

[Townsend; 1977]

TOWNSEND W., Sequencing n jobs on m machines to minimize maximum tardiness : a branch-and-bound solution, *Management Science*, vol.23, N°9, 1016-1019 (1977).

9. BIBLIOGRAPHIC REFERENCES

[Tremoliere; 1978]

TREMOLIERE R., Scheduling jobs of equal length with tardiness costs and resource limitations, *Journal of Operational Research Society*, vol.29, N°3, 229-233 (1978).

[Van Dam; 1995]

VAN DAM P., Scheduling packaging lines in the process industry, PhD. Thesis, Groningen University, The Netherlands (1995).

[Vernadat & al.; 1989]

VERNADAT F., A. DI LEVA and P. GIOLITO, Organization and Information System Design of Manufacturing Environments : the new M* approach. *Journal of Computer Integrated Manufacturing System*, vol.2, N°2, (1989).

[Vignier & al.; 1995]

VIGNIER A., J.-C. BILLAUT and C. PROUST, Les problèmes d'ordonnement de type flowshop hybride : Etat de l'art. Journées d'Etude « Affectation et Ordonnement », GR Automatique du CNRS, Tours (France), Actes 7-47, Septembre 1995.

[Vignier & al.; 1996-a]

VIGNIER A., O. AUCANOT, A. SANGOUARD and J.-C. BILLAUT, Minimisation du plus grand retard dans un flow-shop hybride à deux étages, 5th International Congress of Industrial Engineering, Grenoble (France), Proceedings 117-126, April 1996.

[Vignier & al.; 1996-b]

VIGNIER A., D. DARDILHAC, C. DEZALAY and C. PROUST, An optimal approach to solve a k-stage hybrid flowshop, Workshop on Production Planning and Control, Mons (Belgium), Proceedings 315-319, September 1996.

[Vignier & al.; 1996-c]

VIGNIER A., J.-C. BILLAUT and C. PROUST, Solving k-stage hybrid flowshop scheduling problems, CESA'96 Computational Engineering in Systems Applications, Ecole Centrale Lille (France), juillet 1996.

[Wallace; 1984]

WALLACE T.F., *APICS Dictionary*, 5th Edition, Falls Church, American Production and Inventory Control (1984).

[Ward & Mellor; 1987]

WARD P.T. and S.J. MELLOR *Structured Development for Real Time systems*, Yourdon Press (1985, 1987).

[Ward; 1986]

WARD P.T., The transformation Schema : An extension of Data Flow Diagrams to Represent Control and Timing. *IEEE Transactions on Software Engineering*, SE-12, N°2, 198-210 (1986).

[Webster; 1993]

WEBSTER S.T., A priority rule for minimizing weighted flow time in a class of parallel machine scheduling problems, *European Journal of Operational Research*, vol.70, 327-334 (1993).

9. BIBLIOGRAPHIC REFERENCES

[Widmer & Hertz; 1989]

WIDMER M. and A. HERTZ, a new heuristic method for flow shop sequencing problem, *European Journal of Operational Research*, vol.41, 186-193 (1989).

[Wittrock; 1988]

WITTROCK R.J., An adaptable scheduling algorithm for flexible flow lines, *Operations Research*, vol.36, N°3, 445-453 (1988).

[Yourdon. 1989]

YOURDON E., *Modern Structured Analysis*, Yourdon Press, Prentice-Hall (1989).

9.2. COMMERCIAL SOFTWARE REFERENCES

[1] ASA, Verilog S.A., 52 rue Aristide Brian 92220 Bagnieux.

[2] TEAMWORK, Cadre Technologies Inc., 222 Richmond Street, Providence Rhode Island 02903

[3] MEGA, Gamma International S.A., 3 place de Valois 75001 Paris.

[4] OBJECTTEAM, Cadre Technologies Inc., 222 Richmond Street, Providence Rhode Island 02903

[5] SYS-P-O, Microtools, Microprocess, 19 rue Pierre Curie 92405 Courbevoie.

[6] ARENA, Systems Modeling Corporation, The Park Building, 504 Beaver Street, Sewickley, PA 15143.

9. BIBLIOGRAPHIC REFERENCES

APPENDIX

<u>10.1</u> - N/2/F, J-PREC, OUT-TREE/ C_{MAX}	156
10.2 - N/1// S_{NSD} , R_{NSD} / L_{MAX}	158
10.3 - ENGLISH-FRENCH GLOSSARY OF TEXTILE TERMS	160
10.4 - ABBREVIATIONS & NOTATIONS	161

10.1. N/2/F, J-PREC, OUT-TREE/C_{MAX}

Johnson in 1954 gave an optimal method to sequence N jobs on 2 serial machines in order to minimize the makespan (see §6.1.2.1). We have used his works to solve the problem $N/2/F, j\text{-prec}, \text{out-tree}/C_{max}$, where jobs are subject to out-tree job-precedence constraints.

For a job i , we note $t_1(i)$ and $t_2(i)$, its processing times respectively on machine 1 and 2, $pred(i)$ designates the job that is predecessor of i , and $succ(i)$ the set of jobs which are successors of i .

Algorithm:

1. calculate $t'_1(i)$ and $t'_2(i)$ such as:

$$\begin{aligned} t'_1(i) &= t_1(i) && \text{if } i \text{ do not have any predecessors} \\ &= t_1(i) + t'_1(pred(i)) + t_2(pred(i)) \\ t'_2(i) &= t_2(i) && \text{if } i \text{ do not have any successors} \\ &= t_2(i) + \sum_{j \in succ(i)} [t'_2(j) + t_1(j)] \end{aligned}$$

2. for $i=1$ to N , if $t'_1(i) < t'_2(i)$ then put job i in sub-sequence SS1, else in sub-sequence SS2;
3. sort the jobs of SS1 in non-descending order of t'_1 ,
4. sort the jobs of SS2 in descending order of t'_2 ,

(in each sub-sequence, two jobs which have identical processing times t'_1 or t'_2 , are sequenced according to the precedence constraints)

5. the final schedule is: $SS1 \oplus SS2$ and it respects the precedence constraints.

We prove that

- 1) if $(i \in SS1)$ and $(\exists pred(i))$, then $pred(i) \in SS1$
- 2) if $(j \in SS2)$ and $(succ(j) \neq \emptyset)$, then $succ(j) \subseteq SS2$

Proof of 1):

Let a and b denote two jobs such as $b \ll a$:

$$pred(a) = b \quad \text{and} \quad a \in succ(b) \tag{A1}$$

We can calculate t'_1 and t'_2 , for both jobs:

$$\begin{aligned} t'_1(b) &= t_1(b) \\ t'_1(a) &= t_1(a) + t'_1(b) + t_2(b) = t_1(a) + t_1(b) + t_2(b) \\ t'_2(a) &= t_2(a) \\ t'_2(b) &= t_2(b) + t'_2(a) + t_1(a) = t_2(b) + t_2(a) + t_1(a) \end{aligned} \tag{A2}$$

Suppose that $a \in SS1$ and $b \in SS2$, then:

$$\begin{aligned} \begin{cases} a \in SS1 \\ b \in SS2 \end{cases} &\Leftrightarrow \begin{cases} t'_1(a) < t'_2(a) \\ t'_1(b) \geq t'_2(b) \end{cases} \\ &\Leftrightarrow \begin{cases} t_1(a) + t_1(b) + t_2(b) < t_2(a) \\ t_1(b) \geq t_2(b) + t_2(a) + t_1(a) \end{cases} \\ &\Leftrightarrow \begin{cases} t_1(a) + t_2(b) < t_2(a) - t_1(b) \\ t_1(a) + t_2(b) \leq -[t_2(a) - t_1(b)] \end{cases} \text{ which is impossible} \end{aligned}$$

So if $(i \in SS1)$ and $(\exists pred(i))$, then $pred(i) \in SS1$

note that according to formulae (A2), $\forall i = 1..N$, $t'_1(i) \geq t'_1(pred(i))$ ■

We prove with an alike demonstration that if $(j \in SS2)$ and $(succ(j) \neq \emptyset)$, then $succ(j) \subseteq SS2$.

10.2. $N/1/S_{NSD}, R_{NSD}/L_{MAX}$

We consider the problem of scheduling N jobs on a single machine, when each job is subject to non-sequence dependent setup and removal times (which do immobilize only the machine) and also to due-date.

For a job i , we note respectively $S(i)$, $R(i)$, $d(i)$, and $p(i)$ its setup time, its removal time, its due-date and its processing time; all this parameters are positive or nul.

Theorem:

An optimal solution to the problem $N/1/S_{nsd}, R_{nsd}/L_{max}$ is to sort the jobs in non-descending order of $d(i)+R(i)$; we call this rule ESDRD (Earliest Sum of Due and Release Date).

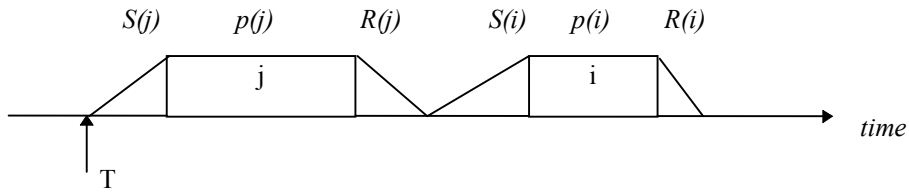
Proof:

Let i and j denote two jobs such as

$$d(i) + R(i) < d(j) + R(j) \quad (B1)$$

and we analyse the two possible situations to process these two jobs from a point of time T .

1) j is processed before i



The completion times of i and j are:

$$\begin{aligned} c(j) &= T + S(j) + p(j) \\ c(i) &= T + S(j) + p(j) + R(j) + S(i) + p(i) \end{aligned}$$

and their lateness:

$$\begin{aligned} l(j) &= c(j) - d(j) \\ &= T + S(j) + p(j) - d(j) \\ l(i) &= c(i) - d(i) \\ &= T + S(j) + p(j) + R(j) + S(i) + p(i) - d(i) \end{aligned} \quad (B2)$$

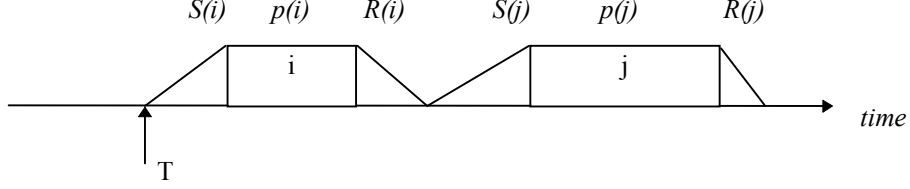
From formulae (B2), we can write:

$$\begin{aligned} l(j) - l(i) &= T + S(j) + p(j) - d(j) - (T + S(j) + p(j) + R(j) + S(i) + p(i) - d(i)) \\ &= -d(j) - R(j) - S(i) - p(i) + d(i) \\ &= [d(i) - d(j) - R(j)] - S(i) - p(i) \\ &< -R(i) - S(i) - p(i) \quad \text{according to (B1)} \\ &< 0 \end{aligned}$$

Then, in that case, the maximum lateness between i and j is given by:

$$L_{\max} = \max(l(i), l(j)) = l(i) \quad (B3)$$

1) i is processed before j



The completion times of i and j are:

$$\begin{aligned} c'(i) &= T + S(i) + p(i) \\ c'(j) &= T + S(i) + p(i) + R(i) + S(j) + p(j) \end{aligned}$$

and their lateness:

$$\begin{aligned} l'(i) &= c'(i) - d(i) \\ &= T + S(i) + p(i) - d(i) \\ l'(j) &= c'(j) - d(j) \\ &= T + S(i) + p(i) + R(i) + S(j) + p(j) - d(j) \end{aligned} \quad (B4)$$

From (B4) and (B2), we can write:

$$\begin{aligned} l'(i) - l(i) &= T + S(i) + p(i) - d(i) - (T + S(j) + p(j) + R(j) + S(i) + p(i) - d(i)) \\ &= -S(j) - p(j) - R(j) \\ &\leq 0 \\ l'(j) - l(i) &= T + S(i) + p(i) + R(i) + S(j) + p(j) - d(j) \\ &\quad - (T + S(j) + p(j) + R(j) + S(i) + p(i) - d(i)) \\ &= R(i) - d(j) - R(j) + d(i) \\ &< 0 \quad \text{according to (B1)} \end{aligned}$$

$$\text{Then} \quad l'(i) \leq l(i) \quad \text{and} \quad l'(j) < l(i) \quad (B5)$$

From (B3) and (B5), we can conclude that

$$L'_{\max} = \max(l'(i), l'(j)) \leq l(i) = L_{\max}$$

So when we have two jobs i and j such as (B1), processing i before j will minimize the maximum tardiness.

10.3. ENGLISH-FRENCH GLOSSARY OF TEXTILE TERMS

beam (primary, sectional)	ensouple (primaire, sectionelle)
beaming	reunissage
bobbin	bobine
aluminium bobbin	bobine aluminium
flanged bobbin	bobine à joues
jumbo bobbin	bobine jumbo
creel	cantre
creel unit	encantrée
desizing	desencollage
drawing in	rentrage
drying	séchage
dyeing	teinture
finishing	finissage (ennoblissement)
grey fabric	tissu écu
grey manufacturing	fabrication des écus
inspection	visite
join-clothing	misage
layer	nappe de fils
lease making	enverjage
pirn-winding	copsage
pre- or final setting	pré-fixation / fixation finale
relaxing	relaxation
sizing	encollage
sleying	piquage
steaming	vaporisage
twisting	retordage
untwisting	détorsionnage
warp	chaîne
warping	ourdissage
water jet loom	métier à tisser jet d'eau
weaving	tissage
weft	trame
weight reduction	réduction de poids
winding	bobinage
wrapping	emballage

10.4. ABBREVIATIONS & NOTATIONS***Abbreviations***

CDS	Heuristic of Campbell, Dudek and Smith [1970]
CdsB	One of our heuristic inspired by Campbell, Dudek and Smith
DFD	Data Flow Diagram
EDD	Earliest Due Date
F	Flow shop
FAM	First Available Machine
G	Job shop
GNS	Greatest Number of Successors
I	Identical parallel machines
JIT	Just In Time
LBM	Last Busy Machine
MILP	Mixed Integer Linear Programming
MRP I	Material Requirement Planning
MRP II	Manufacturing Resource Planning
MWkR	Most Work Remaining
NEH	Heuristic of Nawaz, Ensore and Ham [1983]
NEHB	One of our heuristic inspired by Nawaz, Ensore and Ham
NOR	Number of Operation Remaining
O	Open shop
P	Permutation flow shop
R	unRelated parallel machines
SA	Structured Analysis
SADT	Structured Analysis and Design Technique
SPT	Shortest Processing Time
SZW	Heuristic of Szwarc [1983]
SzWB	One of our heuristic inspired by Szwarc
TOW	Heuristic of Townsend [1977]
TowB	One of our heuristic inspired by Townsend
TQC	Total Quality Control
U	Uniform parallel machines

VBA	One of our heuristic (dedicated to important precedence constraints)
VBB	One of our heuristic (dedicated to maximum lateness criterion)

Main notations

$a(i,v)$	Minimum time lag of job i
$A(i,v)$	Maximum time lag of job i
$B(v,v+\beta)$	Number of auxiliary resources available between the input of workshop v and the output of workshop $v+\beta$
$b(v,v+1)$	Capacity (number of jobs) between the output of workshop v and the input of workshop $v+1$
$c(i,k,v)$	Completion time of job i on machine k in workshop v
$c(i,v)$	Completion time of job i in workshop v
C_{\max}	Makespan, maximum of job completion time
$d(i)$	Due date of job i
$E(i)$	Earliness of job i
$F(i)$	Flow time of job i
$H(v)$	Number of working hours of workshop v on the scheduling horizon
$L(i)$	Lateness of job i
L_{\max}	Maximum of the job lateness
$M(v)$	Number of machines in the workshop v
N	Number of jobs
N_e	Number of early jobs
N_t	Number of tardy jobs
$p(i,k,v)$	Processing time of the v^{th} operation of job i processed on machine k in workshop v
$p(i,v)$	Processing time of the v^{th} operation of job i processed in workshop v when machines are identical
$\text{prec}(i)$	Job predecessor of job i (out-tree precedence constraints)
$r(i)$	Release date of job i
$R_{\text{nsd}}(i,k,v)$	Non-sequence dependent machine removal time on machine k in workshop v , after the end of job i
$S1_{\text{sd}}(i,j,k,v)$	Sequence dependent job setup time of job j , when i is processed just before j on machine k in workshop v
$S2_{\text{sd}}(i,j,k,v)$	Sequence dependent machine & job setup time of job j on machine k in workshop v , when i is processed just before j on the same machine

$S_{nsd}(i,k,v)$	Non-sequence dependent machine setup time on machine k in workshop v , before the beginning of job i
$S_{sd}(i,j,k,v)$	Sequence dependent machine setup time on machine k in workshop v , before the beginning of job j , when i is processed just before j on the same machine
$T(i)$	Tardiness of job i
$U_e(i)$	Earliness unit penalty of job i
$U_t(i)$	Tardiness unit penalty of job i
V	Number of workshops
$W(i)$	Waiting time of job i

Scheduling problem notation

A / B, B' / C, D / E

A	number of jobs to schedule;
B	number of workshops;
B'	machine layout in the workshops (the number of machines of each workshop and the type of the machines I, U, or R);
C	workshop layout (F, G, or O);
D	possible restrictions:
$r(i) \geq 0$	job release dates exist,
job-spl	job splitting is allowed,
pmt	job preemption is allowed,
$a_{i,v}$	minimum time lag,
$A_{i,v}$	maximum time lag,
S_{sd}, S_{nsd}	sequence-dependent or non-sequence-dependent job setup times,
R_{nsd}	non-sequence-dependent job removal times,
$b_{v,v+1}$	limited buffer storage constraint (between v and $v+1$),
$B_{v,v+\beta}$	limited number of auxiliary resources (from v to $v+\beta$),
no-wait	no waiting time is allowed between two successive operations of a job,
H_v	various workshop calendars,
j-prec	general job-precedence constraints,
o-prec	general operation-precedence constraints,
in-tree	in-tree precedence constraints,

out-tree out-tree precedence constraints,...
E the criterion to minimize.

RESUME

Pour être compétitives dans le contexte économique actuel, les entreprises doivent évoluer sur des marchés internationaux, et être capables de faire face à la concurrence accrue des pays où la main d'oeuvre est bon marché. Durant ces dernières années, sous la pression du marché, le nombre de produits des industries manufacturières a fortement augmenté, les volumes de commande ont été réduits, et les délais de livraison ont été raccourcis. Les entreprises reconnaissent donc qu'il devient très difficile de réaliser de bons plannings en un temps raisonnable. De plus, il est pratiquement impossible d'apprécier les conséquences d'un écart de planning. Ces entreprises sont donc constamment à la recherche d'outils de planification et d'ordonnancement fiables et efficaces.

Dans cette thèse, nous étudions les problèmes d'ordonnancement des organisations linéaires d'ateliers des industries manufacturières, et proposons un système d'aide à la décision pour le pilotage de la production. Nous illustrons nos principaux résultats dans le cadre d'une entreprise pilote du secteur textile.

Nous proposons une nouvelle approche des systèmes de production pour identifier les problèmes d'ordonnancement, et spécifier des modèles dynamiques permettant d'évaluer et valider les décisions d'ordonnancement. Cette approche associe la simulation à événements discrets aux techniques classiques de l'optimisation. Les problèmes d'ordonnancement ont été identifiés et modélisés dans le cadre d'organisations linéaires d'ateliers (flow shop hybride), et intègrent des contraintes technologiques entre travaux, des contraintes de ressources et prennent en compte différents objectifs. Ces problèmes étant NP-difficiles, plusieurs heuristiques ont été développées ou adaptées de précédents travaux pour prendre en compte les diverses contraintes et les différents objectifs.

De nombreuses expérimentations ont été effectuées pour juger de la qualité et de la robustesse de ces algorithmes dans des cas généraux. La validation industrielle de cette approche de résolution a été réalisée en expérimentant les algorithmes sur des données industrielles; elle sera poursuivie par la simulation des résultats obtenus sur un modèle dynamique du système de production.

Enfin, nous proposons de formaliser et de généraliser notre démarche dans le cadre de la conception d'un atelier de génie décisionnel pour le pilotage des systèmes de production.

MOTS-CLES: Modélisation, Ordonnancement, Ligne de fabrication (organisation linéaire), Génie logiciel, Heuristiques, Simulation.

ABSTRACT

Competitive firms are operating nowadays in global and worldwide markets. In recent years, under market pressure, the number of products in the process industry has been increased, order sizes have been reduced, and delivery times have been shortened. As a result, companies recognize that it is difficult for material managers to make good production schedules within a reasonable time delay, and that it is practically impossible to appreciate the consequences of a planning error. Therefore, there is an increasing demand by these companies for improved scheduling tools.

In this thesis, we study the scheduling of flow shop organizations in the process industry, and we propose a scheduling framework to help production planners and material managers in their decision making. We illustrate our main results using a company from the textile industry.

We propose a new approach to production systems in order to identify the scheduling problems, and to specify dynamic models for the evaluation and validation of scheduling decisions. This approach associates discrete event simulation with classical scheduling theory. Scheduling problems have been identified and modeled in a hybrid flow shop pattern, taking into account technological constraints on jobs, resource constraints and various scheduling objectives. Given that these problems are NP-hard, several heuristics have been developed or adapted from previous works to take into account additional constraints and different objectives.

The quality and the robustness of these algorithms are illustrated by experimental results on a large range of theoretical data, and the industrial validation of the proposed framework is made by experimenting designed heuristics on industrial cases; simulation of the obtained results on a dynamic model of the production system will continue this process.

As a conclusion, we propose to formalize and generalize our approach in the designing of a decision engineering system for production planning and production activity control.

KEY-WORDS: Modeling, Scheduling, Flow Shop, Software engineering, Heuristics, Simulation.
