



Etude de l'application du raisonnement à partir de cas pour des problèmes d'ordonnancement

Tibor Kocsis

► To cite this version:

Tibor Kocsis. Etude de l'application du raisonnement à partir de cas pour des problèmes d'ordonnancement. Chemical and Process Engineering. Institut National Polytechnique de Toulouse - INPT; Budapesti műszaki és gazdaságtudományi egyetem (Budapest), 2011. English. NNT : 2011INPT0119 . tel-04240412

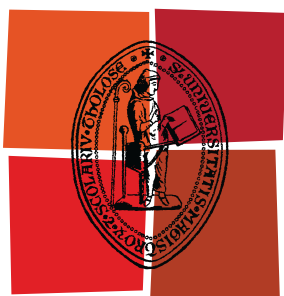
HAL Id: tel-04240412

<https://theses.hal.science/tel-04240412>

Submitted on 13 Oct 2023

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.



Université
de Toulouse

THÈSE

En vue de l'obtention du DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Génie des Procédés et de l'Environnement (GPE)

Présentée et soutenue par :

Tibor KOCSIS

le : lundi 12 décembre 2011

Titre :

Study on application possibilities of Case-Based Reasoning on the domain
of scheduling problems

Ecole doctorale :

Mécanique, Energétique, Génie civil et Procédés (MEGeP)

Unité de recherche :

Laboratoire de Génie Chimique

Directeur(s) de Thèse :

Prof. Xuân-Mi MEYER, INP-ENSIACET, Toulouse

Prof. Endre REV, USTEB, Budapest

Rapporteurs :

Prof. Jean-Michel RENEAUME, ENSGTI, Pau

Prof. Francis COURTOIS, AgroParisTech, Paris

Membre(s) du jury :

Prof. Pascal FLOQUET, INP-ENSIACET, Toulouse, Président

Silvère MASSEBEUF, Ingénieur, ProSIM SA, Toulouse, Membre

Stéphane NEGNY, Maître de conférences, INP-ENSIACET, Toulouse, Invité

Zoltan LELKES, Ingénieur, OptaSoft SARL, Budapest, Invité

ABSTRACT

The purpose of this study is to work out the foundations of a decision-support system in order to advise efficient resolution strategies for scheduling problems in process engineering. This decision-support system is based on Case-Based Reasoning.

A bibliographic study based on co-citation analysis has been performed in order to extract knowledge from the literature and obtain a landscape about scheduling research, its intensity and evolution.

An open classification scheme has been proposed to scheduling problems, mathematical models and solving methods. A notation scheme corresponding to the classification has been elaborated based on the nomenclature proposed by Blazewicz et al. (2007).

The difficulties arising during the adaptation of a mathematical model to different problems is discussed, and the performances of four literature mathematical models have been compared on three flow-shop examples. A resolution strategy is proposed based on the characteristics of the scheduling problem.

KEYWORDS:

Scheduling, Case-Based Reasoning, Decision-support system

RESUME

Ces travaux concernent la mise en place d'un système d'aide à la décision, s'appuyant sur le raisonnement à partir de cas, pour la modélisation et la résolution des problèmes d'ordonnancement en génie des procédés.

Une analyse de co-citation a été exécutée afin d'extraire de la littérature la connaissance nécessaire à la construction de la stratégie d'aide à la décision et d'obtenir une image de la situation, de l'évolution et de l'intensité de la recherche du domaine des problèmes d'ordonnancement.

Un système de classification a été proposée, et la nomenclature proposée par Blazewicz et al. (2007) a été étendue de manière à pouvoir caractériser de manière complète les problèmes d'ordonnancement et leur mode de résolution.

Les difficultés d'adaptation du modèle ont été discutées, et l'efficacité des quatre modèles de littérature a été comparée sur trois exemples de flow-shop. Une stratégie de résolution est proposée en fonction des caractéristiques du problème mathématique.

MOTS CLES :

Ordonnancement d'atelier, Raisonnement à partir de cas, Système d'aide à la décision

ACKNOWLEDGEMENT

This research work has been performed in the context of a co-tutelle agreement between the Institut National Polytechnique de Toulouse and the Budapest University of Techniques and Economics. Therefore I express my gratitude to the Laboratoire de Génie Chimique who received and welcomed me and to the French people, granted me a scholarship in order to realize my studies in France. Also I am grateful to the Hungarian people, affording me the state-financed education during my studies.

I would like to express my deep gratitude to Professor Xuân Meyer first of all for her patience, help and supervision. My deepest appreciation for Dr Stéphane Negny and Professor Pascal Floquet for their advices, help, and serious efforts without that this work would have never been made. From the Hungarian side I am thankful to Dr Zoltán Lelkes and Professor Endre Rév for all their help, and for starting me on the way that ended up to this thesis.

I express my hearties thanks to Barbara Czuczai, for her useful advices, to Timea Lukács and Csaba Stéger, for helping my insertion when I arrived to Toulouse the first time. I am also thankful to Péter Huszár for his corrections and remarks on the manuscript, and to Nayiri Khatcheressian for her knowledge in word processing that she shared with me.

Finally, I wish to thank all their help and efforts to my parents, who helped my every step from the first one, and to Yulia who was always with me and encouraged me when I needed that.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	3
INTRODUCTION	10
SCHEDULING PROBLEMS IN INDUSTRIAL PROCESSES	13
1.1. Introduction	14
1.2. Definitions and terminology	14
1.3. Examples of process scheduling problems	16
1.3.1. Example 1-1 – A batch scheduling example	17
1.3.3. Example 1-3 – a semi-continuous industrial process	21
1.4. Resolution methods for scheduling problems	23
1.4.1. Approaches based on Artificial Intelligence techniques	23
1.4.1.1. Constraint programming	24
1.4.1.2. Expert systems	24
1.4.1.3. Fuzzy logic	24
1.4.1.4. Case-Based Reasoning	25
1.4.1.5. Neural networks	25
1.4.1.6. Multi Agent Systems	26
1.4.2. Simulation based approaches	26
1.4.3. Mathematical approaches	26
1.4.4. Example – solving a scheduling problem with mathematical approach	28
1.4.5. Conclusion	30
1.5. Case-Based Reasoning	31
1.5.1. Main steps of Case-Based Reasoning	31
1.5.1.1. Case representation	32
1.5.1.2. Case retrieval	33
1.5.1.3. Case adaptation	33
1.5.2. Models of knowledge in Case-Based Reasoning systems	34
1.5.3. Advantages of Case-Based Reasoning	34
1.5.4. Conclusion	35
1.6. Conclusion	35
1.7. References of Chapter One	36
BIBLIOGRAPHICAL ANALYSIS	39
2.1. Methods of bibliographic analysis	40
2.1.1. Analysis of connection between papers	40
2.1.2. Clustering techniques	41
2.1.2.1. Traditional methods	43
2.1.2.2. Matrix decomposition techniques	44
2.1.2.3. Combinations	44
2.1.3. Ranking papers	45

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

2.1.4.	Labelling clusters	45
2.1.5.	Conclusion	45
2.2.	The Ensemble NMF algorithm	46
2.2.1.	Application of the method to scheduling literature	47
2.2.1.1.	Construction of the database of papers	47
2.2.1.2.	Clustering	48
2.2.1.3.	Analysis and labelling of clusters	49
2.3.	Discussion of results	49
2.3.1.	General remarks	50
2.3.2.	Finding the most appropriate meta-cluster to a keyword	50
2.3.3.	Cluster analysis based on the full text of papers	52
2.4.	Conclusion	53
2.5.	References of Chapter Two	54
2.6.	Appendix	55

CLASSIFICATION OF SCHEDULING PROBLEMS, MATHEMATICAL MODELS AND SOLVING METHODS 61

3.1.	Introduction	62
3.2.	Aspects of classification of a scheduling problem	63
3.2.1.	Machines	65
3.2.2.	Resources and constraints	67
3.2.2.1.	Functionality constraints	68
3.2.2.2.	Secondary resources (β_2)	76
3.2.2.3.	Other constraints (β_{14})	78
3.2.3.	Objective function	78
3.2.4.	Application to illustrative examples	80
3.3.	Mathematical models	81
3.3.1.	Time representation	81
3.3.2.	Material balance handling	82
3.4.	Solving methods	84
3.5.	Conclusion	86
3.6.	References of Chapter Three	87
3.7.	Appendix	89

ASSOCIATION RULES BETWEEN FLOW-SHOP SCHEDULING PROBLEMS, CORRESPONDING MATHEMATICAL MODELS AND APPROPRIATE SOLVING METHODS 97

4.1.	Introduction	98
4.2.	Adaptation of mathematical models	98
4.2.1.	Adaptation of a mathematical model by minor modifications	98
4.2.1.1.	The mathematical model	99
4.2.1.2.	The applied solving methods	101
4.2.1.3.	Calculation results	103
4.2.2.	Application of different mathematical models	112
4.2.2.1.	The scheduling problems	112
4.2.2.2.	The mathematical models and the solving method	115
4.2.2.3.	Calculation results	115
4.3.	Rule based strategy to associate model/method to a scheduling problem	119
4.3.1.	Summary of the rules of association	119
4.3.2.	Example cases	119
4.4.	Case based reasoning based strategy to associate model/method to a scheduling problem	120
4.4.1.	Introduction	120
4.4.2.	Main steps of case-based reasoning and their realization	120
4.4.3.	Case retrieval – Similarity function	121
4.4.4.	Adaptation of a case	121
4.4.5.	Examples of application	121
4.5.	Conclusion	123
4.6.	References of Chapter Four	124
4.7.	Appendix	126
CONCLUSIONS		134
PERSPECTIVES		135

INTRODUCTION

INTRODUCTION

Markets for industries are becoming more and more competitive, and the quality requirements on a product are at higher and higher levels. Today, in order to fulfill these requirements, industries frequently use techniques of project management, industrial engineering, and operations research. However, quality of the product and capacity of production are not the only constraints: markets impose new industrial strategies, based on more and more flexible production processes, requiring a greater reactivity. In this context and in order to remain competitive, the operating conditions have to be optimized.

Flux production mode is often applied by industries, both in continuous and discontinuous modes. For a long period process engineering was necessarily focused on the conception of processes characterized by high production capacity and continuous production mode (e.g. petrol chemistry). From this historical heritage, continuous production mode became dominant in process engineering. Nevertheless, all production systems of process engineering are not limited to this production mode. Although the installations of food-, biotechnological and pharmaceutical industries may be described and analyzed with using the terms of unit operations, balance equations, operating conditions, the production process is rather only continuous. The opening of process engineering to the domains of applications aroused new problem types. The interest of the scientific community to the discontinuous production mode today is already undeniable.

Discontinuous modes are applied e.g. because of an important rotation of several new products, a high quality or cost of products to produce, a seasonality of production or a lack of time to create a continuous factory. In discontinuous production mode the material is transferred through the system in finite quantities: the lots. Each material lot undergoes a set of transformation operations on specific units during its path. The path sequence of material lots in different equipments has a significant impact on the performance of the production process. In order to use the units rationally the temporal and volume synchronization of the lots becomes necessary. The scenarii of the realization have to be found for a production plan, where among others the number, volume, starting times and the corresponding machine of each lot are defined, minimizing thus the inactivity time of the machines. Therefore, the role of scheduling problems in industry is very important. Consequently, lot of research efforts has been investigated in the area of modeling and solving this type of problems.

Our research is focusing on how to work out an assignment strategy between scheduling problems, mathematical models and appropriate solving methods. The objective is to develop a decision-support system based on a database of previous experiences, gained from previously successfully solved cases. The system should not only answer the question about which model and which method should be applied to solve new problem in the most convenient way, but also should be able to learn from the

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

new case, from the new adapted solution. The decision-support system is to be constructed on the foundations of Case Based Reasoning (CBR) approach. The remainder of the manuscript is organized into four chapters.

In the first chapter, the role and importance of scheduling problems are presented, and the basic definitions and terminologies of this area are introduced. Through some illustrative examples the complexity of resolution process is highlighted, and the resolution methods of scheduling problems are briefly discussed. As the resolution strategy is complex, and the expert has to make several choices, the attempt of constructing a decision-support system will be justified. This decision-support system is based on methods of Artificial Intelligence. Different methods are presented and discussed. Among them, the advantages of CBR (flexibility in knowledge modeling, learning over time, ability to start in a domain with a relatively small body of knowledge...) make this approach an excellent candidate. CBR is a knowledge-based method, inspired by human thinking (see Section 1.5) which can be illustrated by Figure I-1:

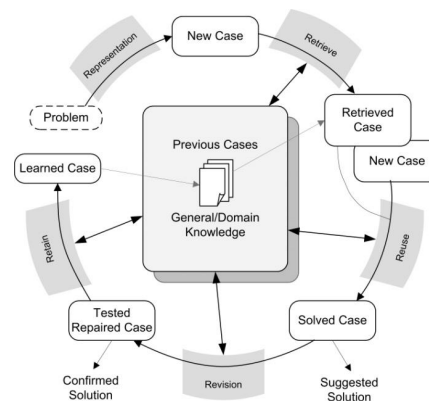


Figure I-1

The problem solving life cycle in a CBR system consists essentially of the following five parts: representing, retrieving, reusing, revising and retaining. As the core of a CBR application is the use of previously solved cases in order to propose solutions to a new problem, a case database has to be established. To be effective, CBR system must encompass a large number of cases in order to cover the whole problem space. Therefore it is necessary to extract the existing knowledge on this domain. In order to collect and extract the existing expert knowledge and to get a view of the evolution of scheduling research, a bibliographical analysis has been performed. The results presented in the second chapter of this document, using the co-citation analysis based Ensemble NMF algorithm of Greene et al. (2008).

Based on the previous analysis, in the third chapter a classification and notation scheme is proposed for scheduling problems, mathematical models and solving methods. The purpose of this classification and notation scheme is to obtain an efficient scheduling case representation in the CBR. The proposed scheme is illustrated on three examples. As this chapter deals with the case representation, it supports the first steps of the CBR system.

Before to draw a conclusion and give some perspectives, the fourth chapter deals with the two next steps of the CBR cycle, i.e. retrieve and reuse (Figure I-1). Because of the large number of scheduling problems, this fourth chapter is limited to the study of flow-shop problems solved with mathematical based approaches. The foundations of a CBR system and the application of the preliminary version of this decision-support software are shown on a flow-shop example of 5 machines.

CHAPTER ONE

Scheduling problems in industrial processes

1.1. Introduction

The application of process engineering towards new areas, such as food, biotechnological, electronic or pharmaceutical industries is generating new production problems with discontinuous modes. Based on competitive markets, these industries set up a reactive production mode, because of an important rotation of several new products, a high quality of products to produce, a seasonality of production (cosmetics) or the lack of time to create a continuous factory. During a long time, these processes have produced very high added value products and the profit was such important that it was not necessary to study the performances of these production processes. The quality of the product and the capacity of production were the two main constraints and there was no need to optimize the operating conditions.

However, the growing worldwide competition in chemical process imposes new industrial strategies based on more and more flexible processes affording a greater reactivity and flexibility to remain competitive in the global marketplace. Indeed, for the manufacture of such chemicals or materials, the production process or the demand pattern is likely to change. The inherent operational flexibility of these industrial plants provides the platform for great savings in good production schedules because it is the core of production management. Moreover, processes need reengineering to respect new constraints coming from the legislative world (environmental, security constraints) or from the enterprise itself (cost reduction, production centralization). These new constraints cannot be neglected to conceive the discontinuous processes and to achieve sufficient profit.

In the above context, each company must optimize its production management by creating multiproduct or multipurpose batch, semi-continuous or continuous plants where products are manufactured with the same or different sequence of operations by sharing production resources (equipments, intermediate materials...). Consequently, the path sequence of material lots in different equipment environments has an important impact to improve efficiently and reduce costs of the production process. This impact justifies the need for more and more sophisticated study on the solution possibilities of scheduling problems. The research area of process scheduling received great attention from both the industry and the academia world resulting in significant advances in relevant modeling and solution techniques. Numerous research studies have been made of this area, e.g. (Esquirol and Lopez, 1999, Pinedo, 2008, Blazewicz et al., 2007). Due to the huge number of possible scenarii and the need to consider several production constraints these problems are highly combinatorial and hence very challenging from the computational complexity point of view.

1.2. Definitions and terminology

The presence of a complex terminology justifies – even makes it necessary – to precise some definitions of the scheduling.

Scheduling is a decision making process to determine when, where and how to produce a set of products given requirements to achieve certain objectives. A **scheduling problem** consists in planning on a time horizon the realization of a set of different product orders, subject to temporary constraints (due dates, processing recipes, time horizon...) and the constraints on a set of limited resources (material availability, equipment assignment...) which are necessary to the realization of the operations. It defines a **schedule** of the organization of the work by resources and/or group of resources of the factory. It describes also the execution of operations (starting times,

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

finishing times) but also the assignment of resources to activities according to the time. When only the relative order has to be fixed between operations, which can be in conflict by the use of the resources, we speak about **sequencing**. Thus, for a schedule, the execution time, the unit used to realize each operation and sometimes the number and volume are to be defined.

After the previous definition it is necessary to precise the different elements who constitute a schedule.

The **recipe** gives the description of the decomposition of the production of a product into elementary tasks. In process engineering, the recipe is usually known in advance. It encompasses also the minimal set of data to define the prescriptions of fabrication of a product, i.e. the relations and connections between tasks. The recipe allows describing a product and the ways it is produced. Frequently it is composed of five principal elements: the *header* (regrouping the administrative information), the *formulation* (indicating the list of raw materials, intermediate products, their proportions and the operating conditions), the *needs* (equipments or equipment types), the *procedure* (defining the sequence of unit operations scheduled in time who describes the products fabrication), and *supplementary information* corresponding to quality and security constraints.

An **operation** or **task** is an elementary entity of activity which is localised in time by a starting and a finishing date (or by one of these dates and its duration). In order to be performed an operation uses one or more resources with an intensity which is usually considered to be invariable during the execution of the operation.

A **job** is the set of successive operations to perform in order to obtain the final **product** from **raw materials** (often but not necessarily via **intermediates**).

A **resource** is a technical or human mean which is required for the realization of an operation and is available in limited quantities. The duration of an operation is not always known à priori, but can be a function of the quantity and of the performance of the resource used. A **machine** is a specific technical resource. It corresponds to a unit, used to perform a group of operations. In process scheduling it is also called primary resource. Resources other than machines are often called secondary resources, e.g. utilities.

The set of machines and the including connections between them is called **workshop**. A **workstation** gathers a set of machines with the same function. In a workstation there can be one or more machines.

Constraints express restrictions on the values that the decision variables can jointly take. In scheduling there are two great categories of constraints: resource constraints and time constraints. For example a set of raw materials with release time and a limited amount data are constraints to satisfy product orders. Product deadlines or relations and connections between tasks are also constraints.

For highly constrained scheduling problems the solution technique can be limited to find a solution, complying with all the scheduling constraints. In this case we are in a feasibility strategy. But the determination of the locations, times, sequences of processing activities with finite units and resources can lead to several potential solutions. In order to classify them, one or more criteria are established to evaluate the quality of the solution (optimality strategy) such as maximization of profit or minimization of makespan. Sometimes it is difficult to model all the objectives with one or more criteria, in this case, some of them can be included as constraints in the model. Finally, the goal is to find either a solution optimizing an objective function or a feasible solution that satisfies the constraints.

Numerous aspects have to be taken into account during the study of a scheduling problem: plant purpose (multiproduct or multipurpose), production mode (batch, semi-continuous, continuous), production resources...

The following part details three illustrative examples which will highlight some aspects of our research study in the rest of the manuscript.

1.3. Examples of process scheduling problems

Any process can be viewed as a succession of unit operations, and can be characterized by the way that its operations are interconnected and by the way that the material circulates in the workshop. According to these conditions the production processes can be distinguished by:

- The production topology. Based on the complexity and features of batch processing, all processes are classified into two groups: sequential or network processes. In practice most of the batch processes are sequential, but as applications become more complex, networks can handle processes with arbitrary topology. Complex products recipes involving mixing and splitting operations and recycles can be considered.
- The production purpose of the unit: mono-product, multiproduct or multipurpose unit. A mono-product unit produces one product, which is invariable during the process. The sequential processes including single or multiple stages and units can be divided into two categories: multiproduct or multipurpose. In the former all jobs are processed in the same production paths and the processing sequences of batches in each unit are identical. While in the latter, the production paths of some jobs are different and may be in the opposite direction. This aspect will be detailed in chapter three.
- The production mode which depends on the nature of the flux flowing through the workshop: batch, semi-continuous and continuous. Usually the process industry operates rather in continuous mode of production. However, with the changing of the economic environment and the appearance of the interest for the new industrial applications, the batch production mode has been implanted, leading to the appearance of discontinuous workshops. This production mode comes historically from the manufacturing sector. It permits to create several different products using the same resources: equipment, storage units, utilities, operators... This production mode answers more and more demanding requirements in the terms of quality and specificity of products, constraints of security and reduction of delays. This flexible, dynamic and reactive mode of production is actually well established in pharmacy, cosmetics, food, and polymer industries.

The following three examples illustrate the main features discussed above and start to introduce some plant specifications and technological constraints.

1.3.1. Example 1-1 – A batch scheduling example

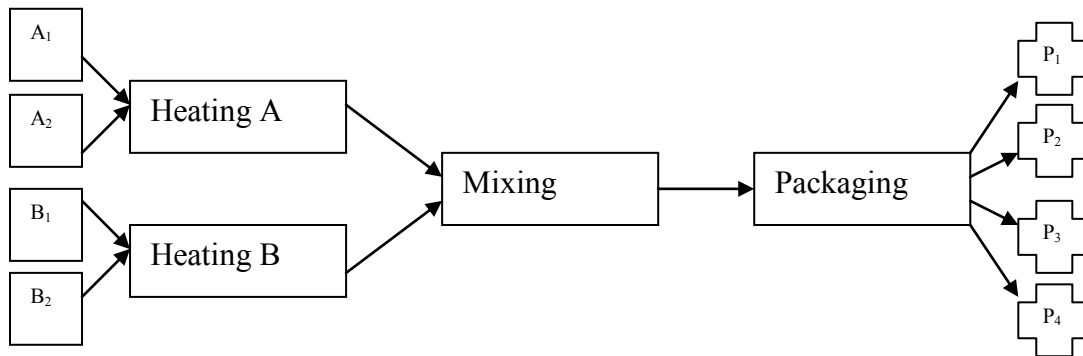


Figure 1-1 – Recipe of example 1-1

On Figure 1-1 a batch scheduling example is presented, delivering 4 products: P_1 to P_4 . Each product can be made by the following recipe: to produce P_i one has to preheat A_j and B_k independently, then mix them and finally package the product. P_1 , P_2 , P_3 and P_4 can be produced from A_1 and B_1 , A_2 and B_1 , A_1 and B_2 , A_2 and B_2 respectively.

To perform the operations 4 heaters (2 dedicated to A, 2 dedicated to B), 4 mixers and 2 packagers are available. The workshop of example 1-1 is illustrated on Figure 1-2. It consists either in 3 workstations (Heaters, Reactors and Packagers) or in 4 workstations if the Heaters dedicated to A and the Heaters dedicated to B are distinguished. In the former case, the restriction on the use of Heaters has to be taken into account by adding assignment constraints (i.e. two Heaters can only perform raw material A, and two Heaters can only perform raw material B), while there is no practical advantage. In the latter case, there is no such inconvenience, therefore in this work this point of view is preferred.

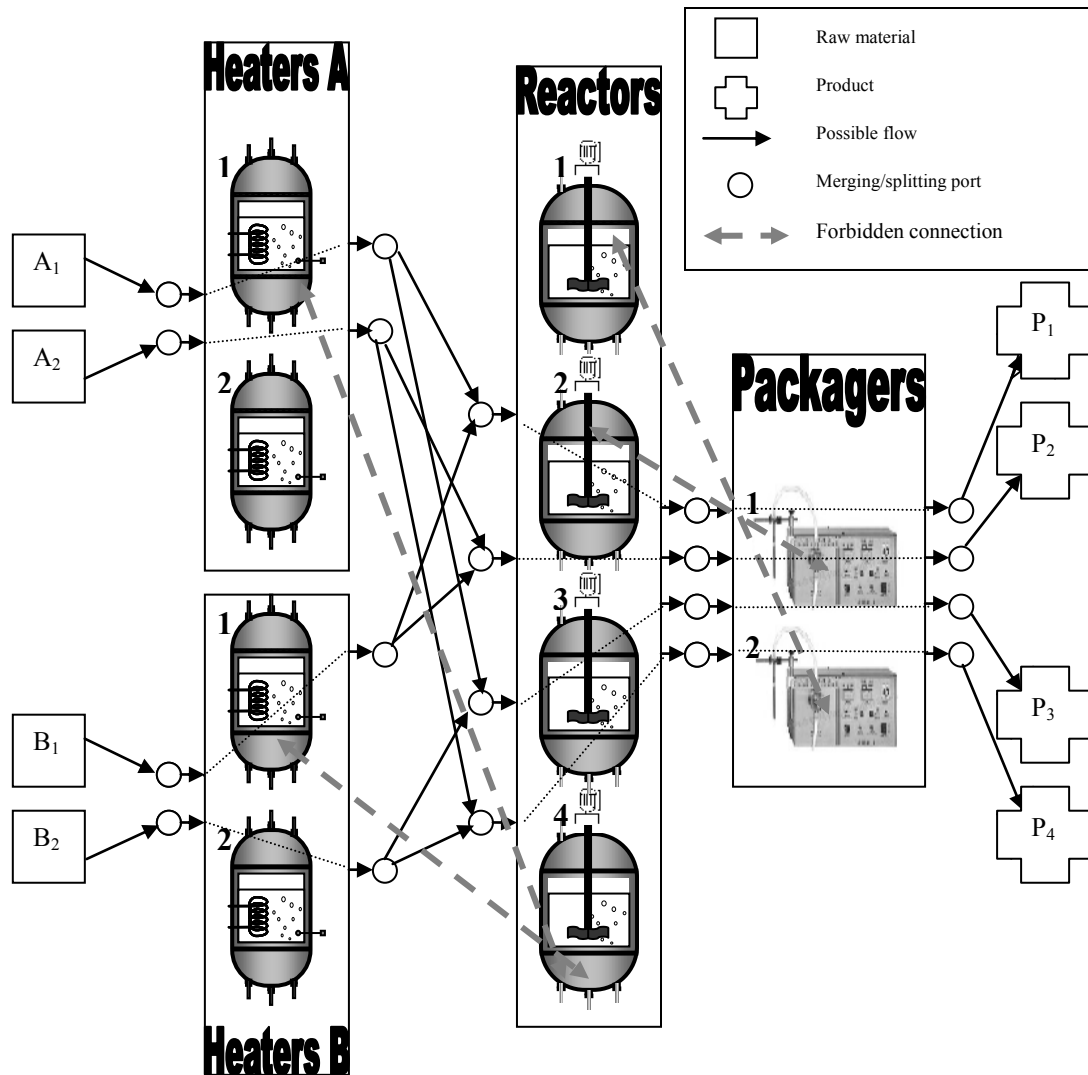


Figure 1-2 – Workshop of example 1-1

In a process, the connection possibilities of the units can be fully or partially allowed. For the former category, there is a possible connection between each pair of machines and the materials can be transferred from one to any particular machine. In the second category, some connections between the devices are not allowed. In the current example, the circles and arrows on Figure 1-2 make it possible to follow every possible production way in the workshop. The circles represent merging/splitting ports and inlet/outlet ports of workstations ($\rightarrow O$ means inlet and $O \rightarrow$ means outlet port), and the thin continuous arrows represent the possible flows between machines. The thick dashed arrows illustrate forbidden connections between the pointed two machines, e.g. the sequence Heater A 1 \rightarrow Reactor 4 is not allowed. Raw materials are represented with squares; the products are represented with crosses.

In order to illustrate usual complexities in scheduling area, several other restrictions have been introduced:

1. Heating operation of type A raw materials and heating operation of type B raw materials must be finished at the same time, in order to mix them while both streams are hot (decreasing energy consumption).

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

2. As type B raw materials are delivered from an external place, the corresponding heating operations can only start when the raw material is available.
3. After the end of mixing operation, a waiting time before packaging has to be respected.
4. After being used, each reactor has to be cleaned before its next operation. As the mixtures have different properties, the cleaning time depends both on the former and the current operation.
5. The operation processing times depend on the machine used.

The characteristics of example 1-1 are summarized in Table 1-1.

Table 1-1 – Problem characteristics of example 1-1

Process topology:	Sequential
Production purpose:	Multiproduct
Production mode:	Batch
Time constraints:	Double end-end relation Waiting times Changeover times Machine dependant process times
Resource constraints:	Resource availability

1.3.2. Example 1-2 – a network represented process

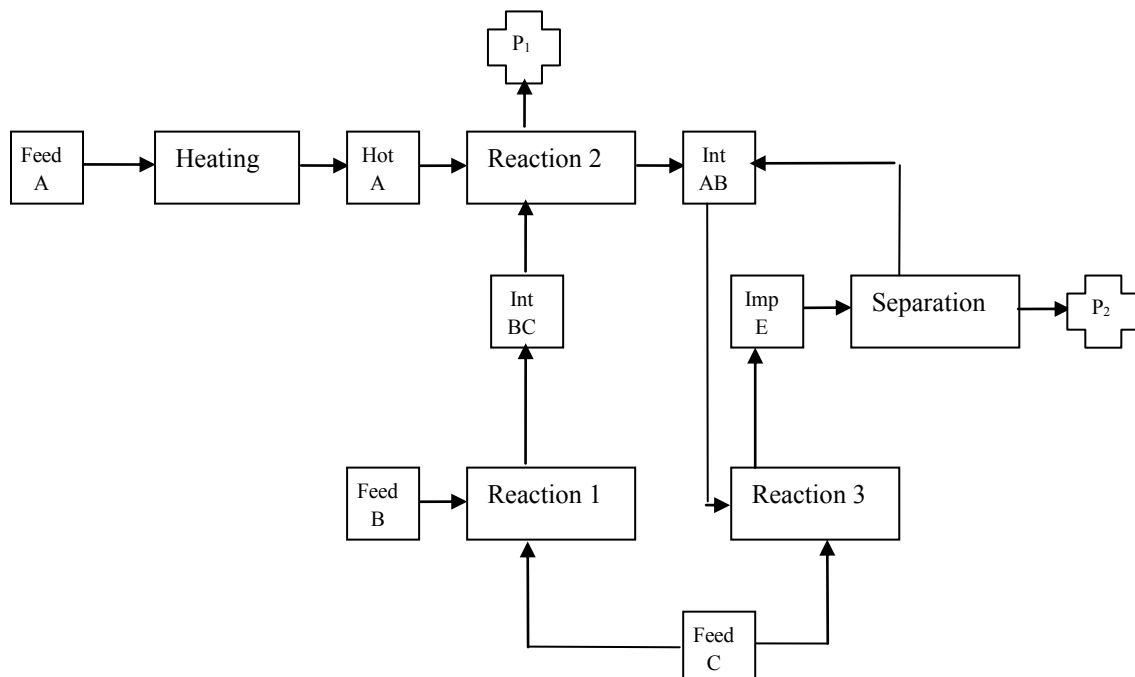


Figure 1-3 – Recipe of example 1-2 (from Kondili et al., 1993)

In this example proposed by Kondili et al. (1993), 2 products are delivered from 3 raw materials with the recipe detailed in Figure 1-3. The workshop of this problem (Figure 1-4) is composed of 3 workstations: a heater, a workstation with two reactors, and a distillation column. The production can be realized in batch or in continuous mode. Example 1-2 consists of two jobs with respect to the number of products: sequence to obtain P_1 (heating, reaction 1 and reaction 2), and sequence to obtain P_2 (reaction 3 and separation). The assigned inlet and outlet ports are denoted with the same colour, for example the IntBC intermediate (output from the gray outlet port of Reactors workstation) is the production result of B and C raw materials entering in the gray inlet port of Reactors workstation. Hot A has to be performed immediately after the Heating task in order to avoid increasing energy consumption.

Comparing to example 1-1, example 1-2 contains less products and raw materials, however, the recipe is more complicated.

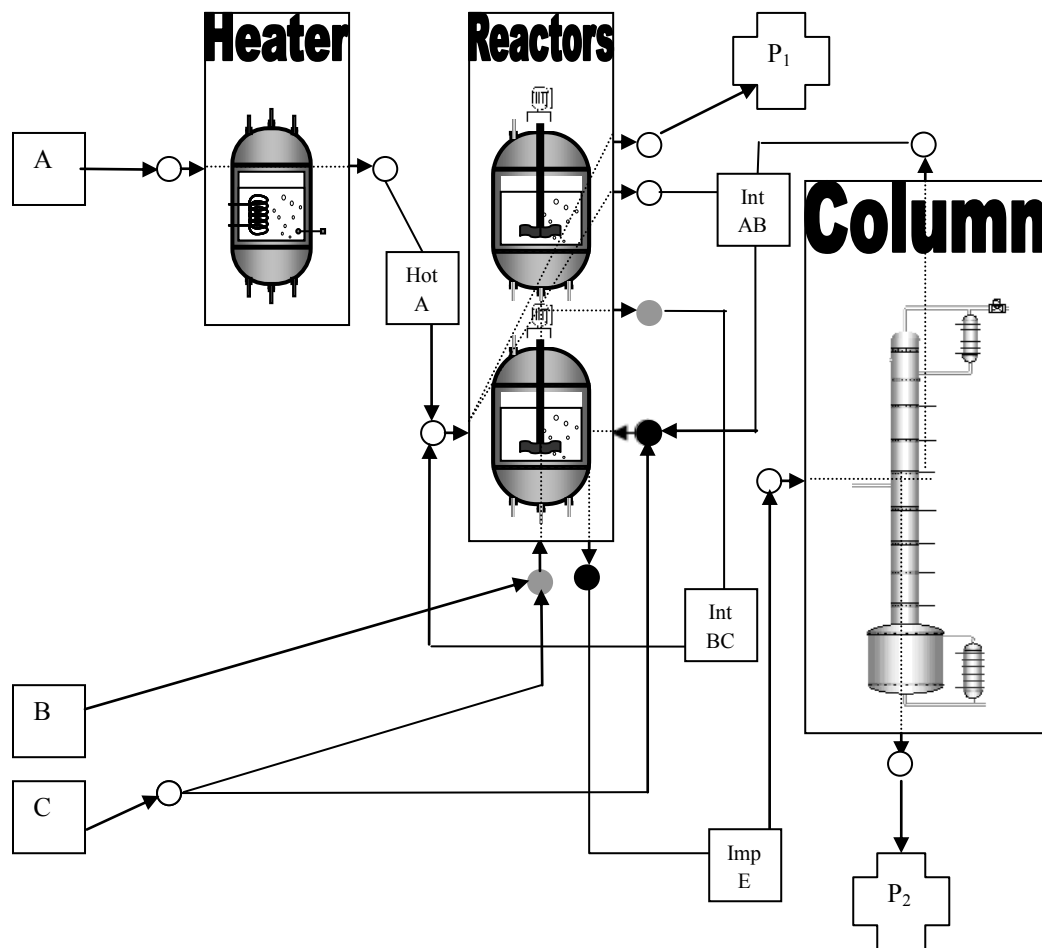


Figure 1-4 – Workshop of example 1-2

Figure 1-4 gives an illustration of the network topology. As network processes include merging, splitting and recycling flows, material balances are required to be taken explicitly. These processes are modelled using the

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

State Task Network (STN) or the Resource Task Network (RTN) representations (Schilling and Pantelides, 1996) detailed in Chapter Three.

The characteristics of example 1-2 are summarized in Table 1-2:

Table 1-2 – Problem characteristics of example 1-2

Process topology:	Network
Production purpose:	Multipurpose
Production mode:	Continuous
Time constraints:	End-start relation (zero wait)
Resource constraints:	Final intermediate storage

1.3.3. Example 1-3 – a semi-continuous industrial process

Apart from batch and continuous processes, there are also production units functioning in semi-continuous mode. They ally elementary operations corresponding to continuous production mode, with other elementary operations corresponding to batch production mode. One of the difficulties of this mode comes from the management of transition phase between the two previous production modes.

In the third example, 3 products (bottled, canned and barrelled beer) are delivered from one raw material (draft beer). However, there are several restrictions and constraints concerning the process (the recipe is shown on Figure 1-5):

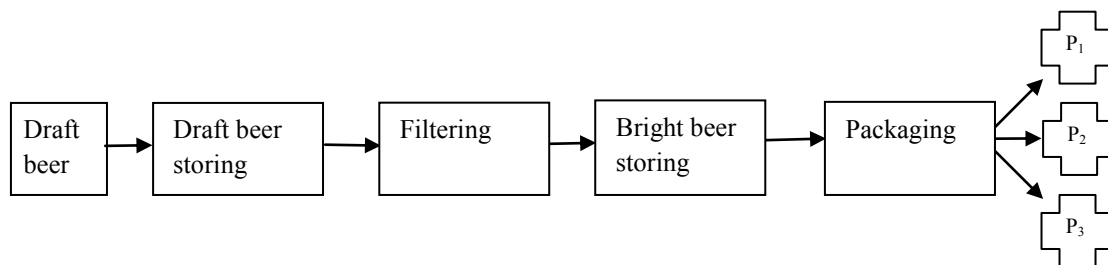


Figure 1-5 – Recipe of example 1-3

Example 1-3 has been published by Czuczai et al. (2009). The workshop of this example (Figure 1-6) is composed of 4 workstations: a draft beer tank, a filter, a bright beer tank, and a workstation with three package lines. The raw beer is stored in raw beer tanks (RBT) and is filtered by several alternative filters. Deadlines for utilizing the beer are specified for the raw beers stored in each raw beer tank. Any RBT can be connected to only one filter at a time and any filter can be connected to only one RBT at a time. During filtration, the beer is loaded to a bright beer tank (BBT). Any filter can be connected to only one BBT at a time and any BBT can be connected to only one filter at a time. Filtration is a continuous task, the beer flows continuously through the filter, and the RBT and the BBT must be available through the whole filtration process. The beer can be accumulated in the BBT from several filtration operations during a time period for which an upper bound is also specified.

After accumulation of beer, the bright beer storing task is performed, during which the beer has to spend a minimal waiting time in the BBT. The actual waiting time may be longer than the minimum. After that, the beer is loaded to packing lines. The load of a BBT can be packed in several different packing operations. For the

emptying phase, an upper bound to the time is also specified. Any BBT can be connected to only one packing line at a time, and any packing line can be connected to only one BBT at a time. Deadlines are also specified for satisfying product orders. Several orders may refer to each product.

Since filtration and packaging are continuous tasks, the starting and ending time must be synchronized with the corresponding operation of the actual BBT. The beer can be stored in the BBT before and after the BBT operation, only for a specified maximal time period.

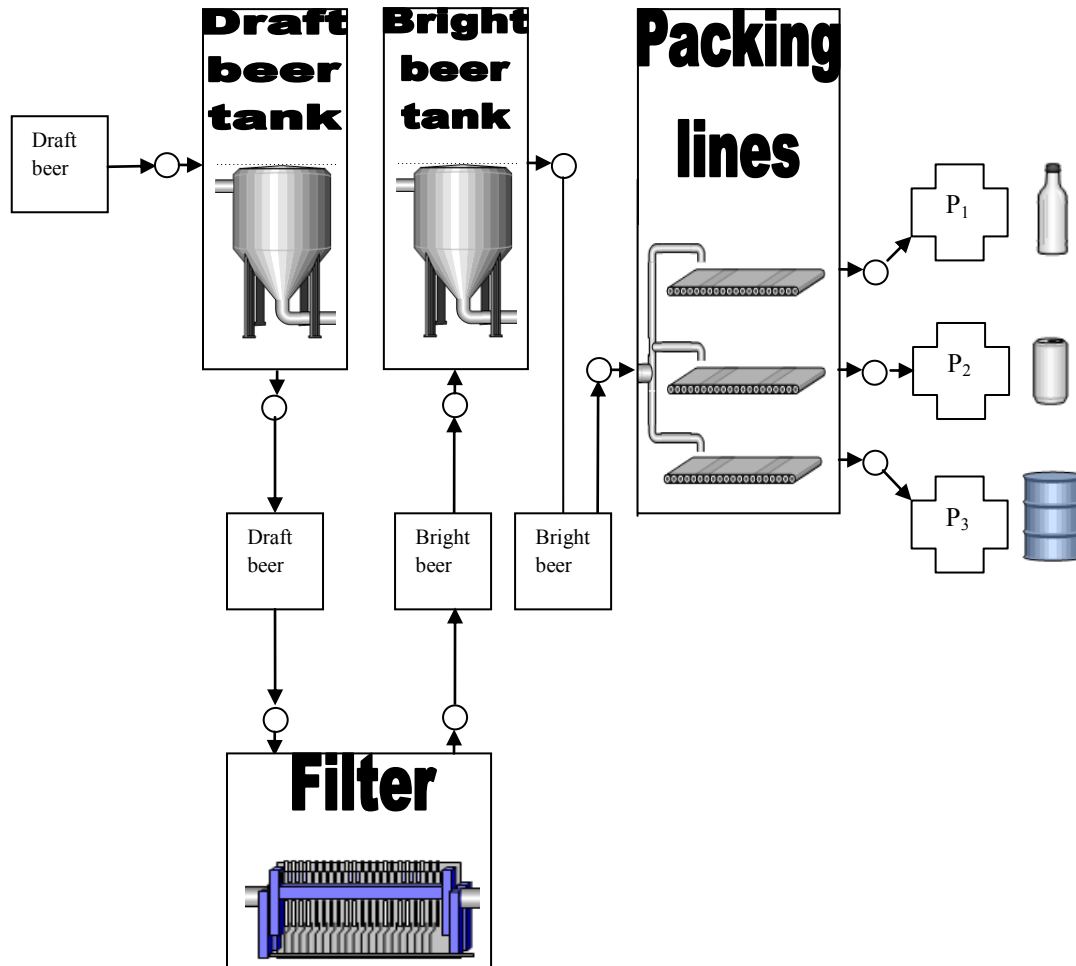


Figure 1-6 – Workshop of example 1-3

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

The characteristics of Example 1-3 are summarized in Table 1-3:

Table 1-3 – Problem characteristics of example 1-3

Process topology:	Sequential
Production purpose:	Multiproduct
Production mode:	Semi-continuous
Time constraints:	Waiting time limits Deadlines
Resource constraints:	Connectivity restrictions Required minimal quantities of products

1.4. Resolution methods for scheduling problems

Whatever the considered production type (manufactory or process) and the applied modeling are, the scheduling problem leads usually to the resolution of a combinatorial optimization problem. While in process engineering studies, papers has been focused on optimization approaches and related modeling aspects, it is important to note that there are other solution methods for dealing with scheduling problems. As we are going to explain, these methods can be used either as alternative methods, or as methods that can be combined with mixed integer linear programming. The different approaches proposed in the literature to treat this type of problems can be classified into three great categories:

- Methods coming from **Artificial Intelligence**
- **Simulation approaches** (continuous, hybrid, discrete events) using more or less sophisticated placing rules concerning the sequence of tasks and the priority of sharing of machines.
- **Mathematical approaches** apply the techniques of operational research, such as mathematical programming (linear, non-linear, integer and mixed-integer), heuristics (often specified to a problem type), meta-heuristics...

Each of these approaches can be divided into two groups: **Exact methods** ensure the achievement of the optimal solution but often need important computational efforts, depending on the complexity of problem. **Approximate methods** permit to find a solution more or less close to the optimum, in a reasonable time. For industrial applications, the choice of the solution methods is often a balance between computational effort, robustness and solution quality, even with the growing computational power.

It is difficult to be exhaustive on the different techniques applied, so the next part is limited to the approaches which permit to solve n-jobs, m-machines problems ($m > 2$) frequently met in engineering.

1.4.1. Approaches based on Artificial Intelligence techniques

Artificial Intelligence (AI) is the mimicking of human taught and cognitive processes to solve complex problems. AI uses techniques and builds tools to represent, capitalize, manipulate and reuse knowledge. Different approaches try to mimic the different ways that people think and reason. Grabot (2006) delivered a more detailed presentation about these methods on the domain of artificial intelligence. He distinguishes six

main categories: constraint-based approaches (constraint programming), expert systems, fuzzy logic, case-based reasoning, neural networks and multi-agent systems. Most of these approaches encapsulate knowledge gained from human experts and apply that knowledge automatically to make decisions. The process of acquiring expert knowledge and to manage it requires considerable skills to perform successfully.

1.4.1.1. Constraint programming

Constraint propagation techniques reduce progressively the solution space using actively the constraints restricting the order of selection of variables and the sequence of affectation of possible values. Constraint Programming is a relatively new modeling and solution paradigm that was originally developed to solve feasibility problems, but it has been extended to solve optimization problems, e.g. scheduling problems. The solution of Constraint Programming models is based on an active use of the constraints to perform constraint propagation at each node by a progressive reduction of the variable domain (Galipienso and Sanchis, 2001). If an empty domain is found, the node is pruned. These techniques are frequently combined with other methods in order to obtain a precise schedule, because, alone, the constraint propagation rarely leads to a unique solution. Constraint propagation is especially useful when the problem is very constrained and it is difficult to find a solution satisfying all the constraints (Baptiste et al., 2006).

Besides the use of Constraint Programming in combination with mixed integer linear programming techniques known as hybrid methods has received attention since they are complementary to each other (e.g. Roe et al., 2005).

1.4.1.2. Expert systems

Expert systems imitate human reasoning, considering it as being decomposable into elementary steps, often modeled by rules consisting of a condition and a consequence part (IF THEN rules). An expert system is made up of a base of rules (knowledge about the problem) and a base of facts regrouping the properties that are “true”. Then an inference engine permits to determine the condition parts of rules that are satisfied and the consequences that can be deduced.

Several attempts have been made in order to model the knowledge on the domain of scheduling or on a given workshop. These experiences have met two great difficulties: little general knowledge seems to exist about this area and the development of a base of knowledge needs important effort (especially during the maintenance of this base). Additionally, the knowledge applied to scheduling problem does not seem to really fit to a binary schema such as the “simple” production rules. Therefore, in the main applications, only local knowledge is modeled in order to manage some choices inside more performing methods (Bel et al., 1988).

1.4.1.3. Fuzzy logic

Fuzzy logic attempts to formalize the imprecision and uncertainty, belonging to the most of human knowledge. Instead of translating this knowledge into binary values, a proposition of “truth value” is affected, which is usually comprised between 0 and 1. Then, the evolution of this truth value will be interpreted as a membership function, (Kaufmann, 1992).

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

Two principal categories of use can be distinguished on scheduling domain: the use in order to integrate imprecise knowledge on expert systems (Kong et al., 2011) and the use in order to describe flexible constraints into constraint propagation (Lu and Yiu, 2011).

1.4.1.4. Case-Based Reasoning

Case-Based Reasoning (CBR) tries to find the solution of a given problem based on the solution of a similar problem, resolved in the past. In this approach the central element is a case, which represents a contextual experience composed of the description, the solution and the environment of a problem. Numerous cases are stored in a case memory. Then, when a new problem is met, the similarity between the stored cases and the new one is analyzed in order to extract a previous case which is considered to be close enough. The solution of the retrieved case is adapted in order to withdraw the discrepancies between the two problems and to match more precisely with the initial problem. The CBR assumes:

- To be able to formalize the knowledge by some parameters in order to describe a case.
- To determine a similarity function permitting to extract a relevant case in order to solve the faced problem.
- To be able to find the adaptation of the retrieved solution.
- To have enough cases stored in the memory in order to cover a maximal space of problems.

Encouraging applications of CBR have already been realized (e.g. Napoli, 1999; Cunningham and Smyth, 1995). As with the previous approaches, resolutions with CBR offer to formalize the knowledge of an expert during choices or to determine some general variables of the problem rather than managing completely the schedule.

1.4.1.5. Neural networks

Contrary to expert systems or case based reasoning, neural networks do not imitate the human reasoning but the structure of the human brain (Arbib, 1998). The central element of a network based on this approach is a neuron, emitting an output in function of its inputs. The neurons in the network are in connection with each other, and the information is stored by the weighting of the connection lines.

Before the use of a neural network, a learning phase has to be performed, when the network is taught by the representation of so-called “learning cases”, i.e. entering input values with desired outputs. The weights of the connections are systematically updated until a correct solution can be obtained for all of the learning cases.

Similarly to expert systems, neural networks are used to support or replace choices made by the expert inside more performing methods (Shiue and Su, 2002).

1.4.1.6. Multi Agent Systems

The main idea of multi-agent systems is the decomposition of the problem into sub-problems and next, to find the solutions of these sub-problems modeling each one with an entity (agent) which agents are in connection with each other. These systems belong to the category of model based reasoning, which concentrates on reasoning about a system behavior from an explicit model of mechanisms underlying that behavior.

In scheduling domain there are two kinds of agents; the machine whose goal is to find work and the job which tries to be made at lower cost and shorter time. The main features of multi agent system are its flexibility and its ability to deal with multi-objective optimization (Archimède and Coudert, 2001). Despite theoretical advantages, this approach has important weaknesses such as huge work of programming in order to implement agents, important processing time due to negotiation between agents, and problems due to local optimization (because of the decomposition in sub problems) while scheduling problems have global objectives.

1.4.2. Simulation based approaches

Simulation based approaches (or simulation of priority rules) belong to the methods which construct progressively a schedule. These methods build rapidly a schedule respecting the technical constraints. The method is based on the generalization of the priority choice method between operations being in conflict on a machine. Simulation based approaches can be described as a progressive elaboration of the schedule, constructing the priority rules of the conflicting operations:

1. Start with the workshop at $t = 0$
2. List the conflicting operations of the schedule
3. Apply a priority rule in order to manage the first conflict (in fact a hierarchical list of priority rules)
4. Plan the chosen operations
5. Increment the time and return to step 2 until the deadline.

The advantages of simulation based approaches are the great simplicity and reactivity. The main drawbacks are its instability (with the increase of the quantity of available resources, for example, a high makespan can be proposed). The performance of these approaches depends strongly on the rules and these rules are numerous. For more details on the application of this approach on scheduling problems see Fabre et al. (2011).

1.4.3. Mathematical approaches

From a mathematical point of view, a scheduling problem is a combinatorial optimization problem (Equation 1-1 and Table 1-4), where the optimal solution has to be found according to an evaluation criteria.

$$\begin{aligned} \min f(\mathbf{x}) \\ \text{s.t.} \\ \mathbf{g}(\mathbf{x}) &\leq 0 \\ \mathbf{h}(\mathbf{x}) &= 0 \\ \mathbf{x} &\in X \end{aligned}$$

Equation 1-1 – Constrained optimization problem

Table 1-4 – Nomenclature of Equation 1-1

$f(\mathbf{x})$	Objective function
$\mathbf{g}(\mathbf{x})$	Inequality type constraints
$\mathbf{h}(\mathbf{x})$	Equality type constraints
\mathbf{x}	Vector of operating conditions

A **feasible solution** of Equation 1-1 (also referred as mathematical model) is a pair of an $f(\mathbf{x}_0)$ value and $\mathbf{x}_0 \in X$ operating conditions satisfying $\mathbf{g}(\mathbf{x}_0) \leq 0$ and $\mathbf{h}(\mathbf{x}_0) = 0$ constraint equations. An **optimal solution** is a feasible solution which fulfils the following condition: the value $f(\mathbf{x}_0)$ is minimal in a subspace $X' \subseteq X$.

This means that the modification of any operating condition (in any direction) leads to a solution which is either not feasible, or worse than the former (with respect to the objective function value as an evaluation criterion).

If $X' = X$, we speak about a **global optimal solution**, or – in the domain of scheduling problems – about a **globally optimal schedule**.

According to the objective function and the constraint equations, the optimization problems can be distinguished into four categories: Linear Programming problems (LP), Nonlinear Programming problems (NLP), Mixed Integer Linear Programming problems (MILP) and Mixed Integer Nonlinear Programming problems (MINLP). If all the variables are continuous and the objective function and the constraints are linear, the problem is called a Linear Programming (LP) problem. If the variables are all continuous but the objective function or some constraints are non-linear, it is called a Nonlinear Programming (NLP) problem. If the model contains variables which are interpreted only on integer values, we speak about a Mixed Integer Programming (MIP) problem, which can be either linear (MILP) or nonlinear (MINLP).

There are several performance criteria of a scheduling production, which are not independent from each other. For example the satisfaction of an order is influenced by the respect of due-dates, the average waiting times for a demand, or the price of the service. Availability of resources, flexibility, quality requirements, productivity are very important aspects that have to be taken into account either implicitly (in the assumptions of the problem) or explicitly (in the objective function or the constraints). Due to the presence of several constraints and the high number of possible schedules, scheduling problems usually lead to difficult Mixed Integer Linear Programming problems.

The mathematical model has to fulfil two very important criteria: it needs to be detailed enough and correct to be able to describe sophisticatedly the reality; and it has to be solved by the available numerical methods. These

criteria are often contradictory to each other (a more detailed model is also more complicated, and thus more difficult to solve), therefore a practical compromise needs to be found.

However, in practice to find a solution to an industrial problem some simplifying strategies are applied, such as computational effort saving assumptions with decomposition of the problem and/or heuristic approaches. Several solvers have been elaborated to solve mathematical models, based on Branch and Bound or decomposition strategies: CPLEX (ILOG), XPRESS (Dash Optimization). Very efficient on small academic problems, these methods reach their limits when dealing with current real-world applications (hundreds of batches, long scheduling periods, numerous equipments...). Usually industrial problems are large and very hard constrained, which means that optimization solvers have to find the optimal (or non-optimal) solution in a huge search space with a relatively small feasible region. This may result in unstable and unpredictable computational performance of optimization models. A clear disadvantage of some of these techniques is that the optimality of the solution can no longer be guaranteed.

1.4.4. Example – solving a scheduling problem with mathematical approach

Due to the complexity of the production processes and the consideration of several, often contradictory aspects, it is usually hard to find even a feasible solution, not speaking about an optimal one. The purpose of the firm is to minimize its costs and the presence of several constraints justifies, and even makes it necessary to apply the strategies of optimization. During the optimization process there are two very important questions: which model should represent the problem? Which solving method should be used to find a solution? To find the answers we have to go further and examine the process of decision of the engineer.

The solution process of a scheduling problem contains three main phases (Figure 1-7): modeling of the problem, choice of a corresponding numeric method to apply, and resolution of the model. The evolution of the resolution process is not unidirectional: experiences from the resolution phase for example can force us to change the applied numeric method, or to modify the model.

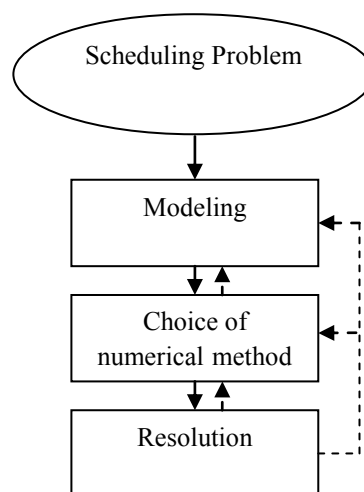


Figure 1-7 – Solution process of a scheduling problem

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

In order to illustrate the difficulties of the choice, let us solve a scheduling problem. For simplicity reason an easy problem with minimal complexities has been created: in Example 1-4 ten jobs are performed on ten machines (multiproduct workshop). Each job is decomposed into ten operations. The objective function to minimize is the total makespan. The problem fulfils the following criteria:

1. The machines are 100% reliable, no breakdown and no maintenance.
2. The workshop is available in 24/24 hours, 7/7 days.
3. There is no secondary resource needed to perform an operation.
4. The duration of each operation is deterministic, given in advance, and includes all necessary transfer times.
5. All job sequence follows the same path with respect to the machines and each operation has one and only one corresponding machine.

The recipe of Example 1-4 is shown on Figure 1-8.

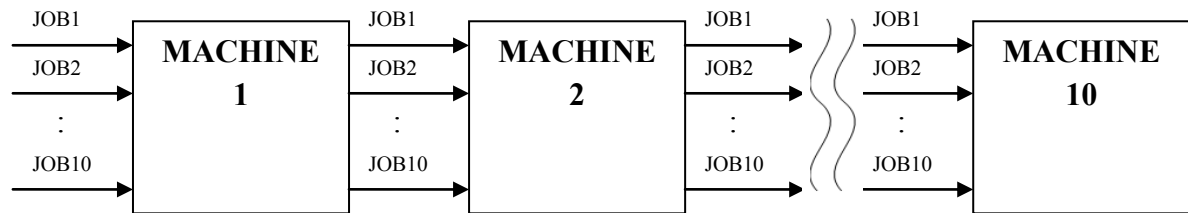


Figure 1-8 – Recipe of Example 1-4

10 scheduling problems have been generated of the above described kind. The problem duration data has been generated randomly as it is proposed by Taillard (1993): the duration of each operation is between 10 and 100 with uniform distribution. The ten problems have been modelled with the same model (disjunctive graph model, presented and detailed in chapter four), but solved with different solution techniques, namely Shortest Process Time First (SPT) and Longest Process Time First (LPT) heuristics, Ant Colony Optimization (ACO), Taboo Search (TS), Simulated Annealing (SA) and Genetic Algorithm (GA) meta-heuristics.

The diagram in Figure 1-9 shows the Box-Whisker plot of makespan values from calculations.

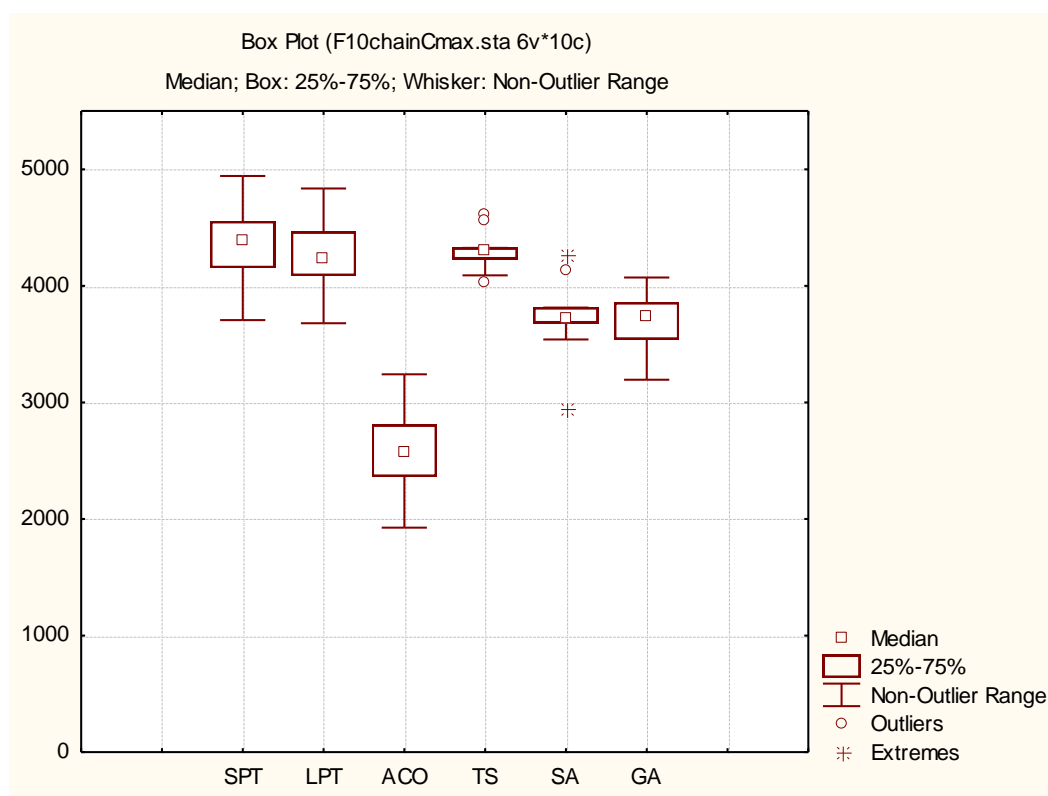


Figure 1-9 – Box-Whisker plots of makespan values provided for Example 1-4

In Figure 1-9 it is illustrated that different methods do not provide the same results, neither the same statistical properties. Without a sophisticated study, no conclusion can be made about the relative performance of these solving methods, but it can be deduced that the “optimal choice” of resolution strategy is not obvious at all.

In order to effectively solve scheduling problems, the elaboration of an association strategy is needed, which indicates a deeper, more sophisticated analysis of problems, models and methods. To the elaboration of association strategy, the development of a decision-aid system seems to be the ideal way.

1.4.5. Conclusion

Several resolution methods have been developed in order to find solutions to scheduling problems. These methods can be classified in three groups: simulation approaches, approaches based on artificial intelligence, and mathematical approaches.

Simulation approaches are mostly tending to determine priority rules on different machines. Although they can be very effective in special cases, they are not enough generic to be successfully applied to a wide variety of chemical engineering scheduling problems.

In the domain of artificial intelligence for scheduling issues, the most of applications attempt to support the human expert rather than to create a complete schedule.

Mathematical approaches, however, seem to be useful and successfully applicable to the concerned domain of scheduling problems. In chemical engineering, the literature in the scheduling area highlights the successful application of different optimization approaches to an extensive variety of challenging problems. More and more difficult and larger problems than those studied years ago can be now solved, sometimes even to optimality in a

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

reasonable time thanks to more efficient integrated mathematical frameworks. This important achievement comes mainly from huge advances in modelling techniques, algorithmic solutions and computational technologies that have been made in the last few years. As a conclusion, as the possible associations between scheduling problem, mathematical model and solution method are very numerous, there is a need for a decision support system. The goal of this decision-support system is to help user in choosing the modelling options and the associated solving methods thanks to a detailed description of the faced problem. But in front of the difficulty to build such a system and the huge interest of the chemical engineering community to mathematical approaches, in the rest of the study we voluntarily limit this work to a decision support dedicated to mathematical approaches. However, in order to keep generality for further evolutions of the decision support system, the Artificial Intelligence approaches will be discussed and included in the system.

In order to construct such a decision support system the existing knowledge has to be extracted, modelled, adapted, diffused, maintained and actualized. The methods of Artificial Intelligence presented in the paragraph 1.4.1 seem to be appropriate to construct our decision-aid system, which will be applied not to solve directly a scheduling problem, but to choose the best modelling options and resolution strategy to the problem.

There are several methods of Artificial Intelligence presented: expert systems, neural networks, constraint propagation, multi-agent systems, and case based reasoning. Regarding to the wide variety of scheduling area, the rule based binary scheme of expert systems does not seem appropriate. The complexity of the decision process makes it difficult to construct a sufficiently complex neural network to model the resolution strategy too. Constraint propagation and multi-agent systems are more applicable when the goal is to solve a scheduling problem, rather than to associate a good resolution strategy to a problem. The case based reasoning makes it possible to store the complex characteristics of the problems in the case memory, and in addition this approach possesses learning ability, facilitating to comply with the wideness and with the dynamic challenge of scheduling domain. Therefore, Case-Based Reasoning approach seems to be the best choice to elaborate our decision support system.

1.5. Case-Based Reasoning

Case-Based Reasoning solves new problems by adapting solutions coming from past-solved problems. The idea of Case-Based Reasoning is intuitively appealing because it is similar to human problem-solving behaviour. People draw on past experience while solving new problems and this approach is both convenient and effective, and it often relieves the burden of in-depth analysis of the problem domain.

The equivalent of human memory in a CBR system is the case library (case base). A case is composed of a problem faced in the past and its solution. To use the stored information a research engine is applied, whose role is to find the cases similar to the faced problem. After analysis, the engine will present an advised solution which has to be adapted and validated. Finally, the problem and its solution can be added to the case memory.

1.5.1. Main steps of Case-Based Reasoning

The problem solving life cycle in a CBR system consists essentially of the following five parts: representing, retrieving, reusing, revising and retaining. These parts are illustrated in Figure 1-10.

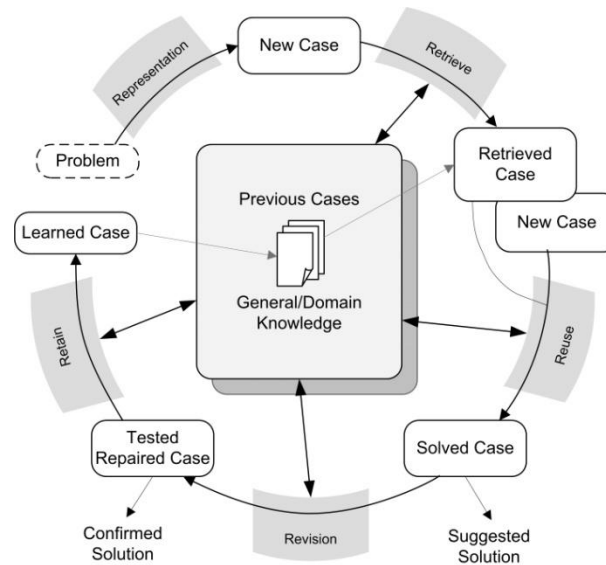


Figure 1-10 – Case-Based Reasoning cycle

The first task is the representation of a new problem. Then, the most similar case is selected from the case-memory applying a similarity function. Retrieving is the process of identifying previously experienced cases whose problem is judged to be similar. Reusing is the copy or integration of the solution from the retrieved case. Revising is the adaptation of the retrieved solution in an attempt to solve the new problem. Retaining is the saving of the new solution once it has been confirmed or validated (Pal and Shiu, 2004).

CBR thus needs to maintain a memory of experiences (case memory), the process of reminding (retrieval), the intelligence of experiences (adaptation) and the update of experiences (learning). To develop a CBR system it is necessary to find an efficient solution to the following problems: representation of a case, retrieval the most similar case, reusing and revision of an existing solution. Adaptation and storage are two other questions coming from the three above.

There are two types of maintenance tasks concerning the case-memory: qualitative and quantitative maintenance. Qualitative maintenance deals with assurance of the correctness, consistency and completeness of the CBR system, while quantitative maintenance is concerned with assurance of the problem-solving efficiency (e.g. the average problem-solving time), the practical limit of the size of the case base (e.g. storage limits), reorganization of case representation structures, etc.

Case representation, case retrieval and case adaptation are discussed in the following sub-sections.

1.5.1.1. Case representation

Cases, in general can be considered as contextualized experiences, and from this point of view, the case representation is the task of enabling the system to recognize, store, and process past contextualized experiences. Selection of an appropriate scheme for case representation is essential because it provides the basic structure for the next CBR tasks.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

Traditional case representation methods can be categorized into three groups (Pal and Shiu, 2004): the relational, the object oriented and the predicate based approaches.

The relational database model is simple and flexible and has been adopted widely. Each object (case) is represented by a row in a relational table where the columns are used to define the attributes (fields) of the objects. Hence, the relational table is divided into two parts: a problem description part and a solution part.

The object-oriented approach uses data structures (objects) consisting of data fields together with their interactions in order to represent a case. The advantage of this approach comes from its compact case representation ability, and the associated software reusability.

The predicate based approach interprets a case as a collection of facts and predicates (relationships between production rules and facts). One of the principal advantages of this approach is the possibility to form a hybrid rule/case-based reasoning system, incorporating many production rules in it, which may be very effective in some special application domain. However, the predicate based approach has a major drawback: retrieving data values from predicates for the purpose of comparing similarity among cases is more difficult than for other approaches.

1.5.1.2. Case retrieval

Case selection and retrieval is usually regarded as the most important step within the Case-Based Reasoning cycle. In this process the similarity measures adopted in a CBR system will greatly influence retrieval performance. The similarity assumption (i.e. similar experiences can guide future reasoning, problem solving and learning) is one of the most important assumptions in case-based reasoning. We remark here, that apart from the traditional similarity concept other, mostly Artificial Intelligence based techniques are applied successfully, e.g. fuzzy similarity, neural networks, and combined techniques (Pal and Shiu, 2004).

The computation of similarity thus becomes a very important issue in the case retrieval process. The effectiveness of a similarity measurement is determined by the usefulness of a retrieved case in solving a new problem. There are broadly two major retrieval approaches (Liao et al., 1998): distance-based approaches, based on the computation of distance between cases (where the most similar case is determined by evaluation of a similarity measure) and indexation based approaches, related more to the representational and indexing structures of the cases. The most frequently applied type of distance measure is the weighted Euclidean distance, but Hamming (number of bits that are different between two bit vectors) and Levenshtein distances (number of deletions, insertions, or substitutions required to transform a source string to the target string) are also popular (Pal and Shiu, 2004).

1.5.1.3. Case adaptation

Usually the past solution gained from the retrieved case needs adjustment to fit the new situation. The process of adjusting the old solution is called case adaptation. Two ways can be distinguished to acquire adaptation knowledge. The traditional approach is coding the task-specific adaptation knowledge manually into the CBR system by interviewing domain experts. This knowledge may be represented as a decision table, semantic tree or IF – THEN rules. However, acquiring adaptation knowledge through interviews with domain experts is both labor intensive and time consuming. The maintenance of the acquired knowledge rise difficulties, too.

Alternatively, the adaptation knowledge can be learned from the cases using machine-learning techniques. Recently, because of the availability of cases and the increase in computer processing power, many machine-learning approaches for deriving adaptation knowledge are developed (Pal and Shiu, 2004).

Traditional case adaptation strategies can be regrouped into three categories: reinstantiation, substitution and transformation. Reinstantiation is the simplest form of adaptation, in which the solution of the new problem is simply copied from the case retrieved and used directly, without modification. Substitution replaces parts of the old solution attributes considered to be invalid (due to conflicts and contradicts with the new problem requirements). Transformation is used when no appropriate substitute item is available. Based on the constraints describing or defining the properties of a solution component and the characteristics of the required solution (some predefined expert knowledge or heuristic must be available) a modified solution is proposed which must conform to these constraints (no contradiction or conflict is allowed).

For case adaptation through machine learning, several examples can be found in Pal and Shiu (2004), e.g. fuzzy decision tree, back-propagation neural network, or Bayesian model.

1.5.2. Models of knowledge in Case-Based Reasoning systems

Case-Based Reasoning systems exist in numerous domains of application, and can be classified into two main categories with respect to knowledge modelling approaches (Fuchs, 1997):

1. For problem resolution systems a problem is composed of a set of initial data, an objective (goal), and a solution able to reach this objective. The problems can be distinguished into two groups: planning/synthesis problems and diagnostic/decision-support problems.
 - a. Planning/synthesis problems can be divided further into planning problems, consisting in constructing a plan in the form of sequential actions to execute in order to reach the set of objectives, and conception problems consisting of constructing an object satisfying a given specification, expressed with constraints, characteristics, and desired functions.
 - b. The class of diagnostic/decision-support problems is composed of the diagnostic problems and the decision support problems. Diagnostic problems consist in looking for possible causes that could drive to the faced symptoms, and proposing a set of treating methods. Then, based on the objective, the problem is translated into a conception problem or a decision-support problem. Decision-support problems provide interactive support to the elaboration of a decision for the user.
2. For interpretation systems, the objective to be satisfied is not explicitly formalized, but has to be discovered by evaluation of situation.

1.5.3. Advantages of Case-Based Reasoning

In this paragraph, some of the advantages of CBR are summarized (Pal and Shiu, 2004). By eliminating the need to express the acquired knowledge into a model or a set of rules, as it is necessary in model/rule-based systems, the knowledge acquisition tasks of CBR consists primarily in the collection of relevant existing experiences/cases and their representation and storage. Thus reducing the knowledge acquisition task is the first advantage of CBR.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

CBR also avoids repeating mistakes made in the past, as these systems record failures as well as successes (and perhaps the reason for those failures). Thus, information about what caused failures in the past can be used to predict potential failures in the future.

Due to their rigidity in problem formulation and modelling, model-based systems sometimes cannot solve a problem that is on the boundary of their knowledge or when there is missing or incomplete data. In contrast, case-based reasoning approaches use past experience as the domain knowledge and can often provide a reasonable solution, through appropriate adaptation to these types of problems. The flexibility in knowledge modelling is thus an important advantage of case-based reasoning.

When information is stored regarding the level of success of past solutions, the case-based reasoning may be able to predict the success of the solution suggested for a current problem.

As CBR systems are used, they encounter more problem situations and create more solutions. If solution cases are tested subsequently in the real world and a level of success is determined for those solutions, these cases can be added to the case base and used to help in solving future problems. As cases are added, a CBR system should be able to reason in a wider variety of situations and with a higher degree of refinement and success – thus a CBR system learns over time.

Finally, case-based reasoning is possible in a domain with a small body of knowledge. While in a problem domain for which only a few cases are available, a case-based reasoning approach can start with these few known cases and build its knowledge incrementally as cases are added. The addition of new cases will cause the system to expand in directions that are determined by the cases encountered in its problem-solving endeavours.

1.5.4. Conclusion

Case based reasoning tries to model the acting by experience, using a case-memory (case-base) where previously faced problems and their solutions are stored. Some principal advantages of CBR are: avoiding to repeat past mistakes, flexibility in knowledge modelling, learning over time, and its ability to start in a domain with a relatively small body of knowledge. These advantages make case-based reasoning to be an excellent candidate for constructing a decision support system in order to help the expert to choose an appropriate resolution strategy for a scheduling problem.

1.6. Conclusion

In this chapter the importance and the role of scheduling problems in industrial processes has been presented. Basic definitions and terminology of this area have been discussed. The different process topologies, production modes, production purposes, and the most frequent time and resource constraints have been illustrated with three examples. Due to the wideness of the scheduling domain the research decided to be focused mostly on chemical engineering process scheduling problems.

Then the resolution methods wrought out to scheduling problems have been briefly reviewed. Artificial Intelligence based approaches usually attempts to aim the human expert on his decisions rather than to create a complete schedule. Simulation based approaches – as generalizations of priority rules – are useful when a

feasible schedule is needed to be rapidly constructed, but did not seem to be rich enough to fit to the complex and large domain of chemical engineering scheduling problems. In chemical engineering the literature in the scheduling area highlights the successful application of mathematical based approaches. Despite of the huge advances made in modelling techniques, algorithmic solutions and computational technologies, due to the complexity of the resolution process it is reasonable and necessary **to apply a decision support system**. Artificial Intelligence based approaches seem to be useful helping the expert to find a good resolution strategy. In front of the difficulty to build such a system and the interest of the chemical engineering community to mathematical approaches we voluntarily limited this work to a decision support dedicated to these approaches.

Among the most important Artificial Intelligence approaches expert systems have several drawbacks. The first of them is the time consuming aspect of the knowledge acquisition task especially in cases where few generic knowledge seem to exist. Besides, the binary scheme of the rules is not suited to the knowledge developed in scheduling problem. Upon the complexity of problem, simple rule-based systems do not seem to be efficient enough. The complexity of the decision process makes it difficult to construct a sufficiently complex neural network to model the resolution strategy too. Constraint propagation techniques and multi-agent systems are more applicable when the goal is to find a solution to a concrete problem rather than to look for a good resolution strategy. On the other hand, case based reasoning has numerous advantages. The reasoning can be started with relatively few initial knowledge. It is flexible and reactive, and the method is capable to learn in time which assures a continuous quality improvement. Therefore, among the possible candidates the **Case-Based Reasoning** (CBR) has been chosen.

The central element of a CBR system is a case. The case consists of a problem and its solution. The elaboration of a decision-aid system based on CBR needs:

1. to collect the existing knowledge on the domain from expert or literature,
2. to develop the representation of a case,
3. to obtain past experiences.

Therefore, in order to collect and extract the existing knowledge and study the evolution of scheduling research domain, a scientific bibliographical analysis has to be performed. For the reason of being able to store the associations proved to work successfully, a classification and notation system has to be constructed. In order to have past experiences, a case base has to be constructed from previously solved problems as well from the literature as from computational results.

1.7. References of Chapter One

- Arbib, M. A., *The Handbook of Brain Theory and Neural Network*; The MIT Press, 1998.
- Archimede, B.; Coudert, T. Reactive scheduling using a multi-agent system: the SCEP framework. *International Journal Engineering Application of Artificial Intelligence* **2001**, 14(5)
- Baptiste, P.; Le Pape, C.; Nuijten, W. *Constraint-based scheduling: applying constraint programming to scheduling problems*; Kluwer, 2003.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

Bel, G.; Bensana, E.; Dubois, D. Construction d'ordonnancement provisionnel: un compromis entre approches classiques et systèmes experts. *APII* **1988**, 22.

Blazewicz, J.; Ecker, H.; Pesch, E.; Schmidt, G.; Weglarz, J. *Handbook on Scheduling From Theory to Applications*; Springer-Verlag, 2007.

Cunningham – Smyth: CBR in Scheduling: Reusing Solution Components. Presented at the 14th Workshop of the UK Planning and Scheduling Special Interest Group, **1995**.

Czuczai, B.; Farkas, T.; Rev, E.; Lelkes, Z. New Mixed Integer Linear Programming Model for Solving Scheduling Problems with Special Characteristics. *Industrial Engineering and Chemistry Research* **2009**, 48, 5321.

Esquirol, P.; Lopez, P. *L'ordonnancement*; ECONOMICA: Paris, 1999.

Fabre, F.; Hetreux, G.; Le Lann, J. M.; Zarate, P. Dynamic hybrid simulation of batch processes driven by a scheduling module. *Computers & Chemical Engineering* **2011**, 35(10), 2098.

Fuchs, B. *Représentation des connaissances pour le raisonnement à partir de cas. Le système ROCADE*. PhD Thesis, Université Jean Monnet de Saint-Etienne, 1997.

Galipienso, M. I. A.; Sanchís, F.B. A mixed closure CSP method to solve scheduling problems; In: *IEA/AIE 2001, LNA/2070*; Springer: Heidelberg, 2001.

Grabot, B. Ordonnancement d'ateliers manufacturiers. *Techniques de l'ingénieur, AG3015* **2006**.

Kaufmann, A. Introduction à la logique floue. *Techniques de l'Ingénieur, A120* **1992**.

Kondili, E.; Pantelides, C.C.; Sargent, R. W. H. A general algorithm for short-term scheduling of batch operations. Part I: MILP formulation. *Computers & Chemical Engineering* **1993**, 17, 2211.

Kong, X. Z.; Lin, C.; Jiang, Y. X.; Yan, W.; Chu, X. W. Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction. *Journal of Network and Computer Applications* **2011**, 34(4), 1068.

Liao, T. W.; Zhang, Z.; Mount, C. R. Similarity measures for retrieval in case-based reasoning systems. *Applied Artificial Intelligence* **1998**, 12, 267.

Lu, M.S.; Liu, Y.J. Dynamic dispatching for a flexible manufacturing system based on fuzzy logic. *International Journal of Advanced Manufacturing Technology* **2011**, 54(9-12), 1057.

Napoli, A. Raisonnement à partir de cas. *Intelligence Artificielle* **1999**, 13(1), Hermès.

Pal, S. K.; Shiu, S. C. K. *Foundations of Soft Case-Based Reasoning*; Wiley, 2004.

Pinedo, M. *Scheduling: Theory, algorithms and systems*; Prentice Hall, 2008.

Roe, B.; Papageorgiou, L. G.; Shah, N. A hybrid MILP/CLP algorithm for multipurpose batch process scheduling. *Computers and Chemical Engineering* **2005**, 29, 1277.

Schilling, G.; Pantelides, C. C. A simple continuous-time process scheduling formulation and a novel solution algorithm. *Computers and Chemical Engineering* **1996**, *20*, S1221.

Shiue, Y. R.; Su, C. T. Attribute selection for neural network-based adaptive scheduling systems in flexible manufacturing systems. *International Journal of Advanced Manufacturing Technology* **2002**, *20*(7), 532.

Taillard, E. Benchmarks for basic scheduling problems. *European Journal of Operational Research* **1993**, *64*, 278.

CHAPTER TWO

Bibliographical analysis

In order to construct a decision support system on the domain of scheduling problems, it is important to extract the expert knowledge, to collect information and obtain a landscape on the situation of research in this area. Scientific methods of bibliography are useful to perform a sophisticated study, especially if a rich database of papers is available (Osareh, 1996). This is the case with scheduling problems, as the problematic of scheduling is very large and complex, there are several resolution strategies and approaches to map and the corresponding research area is important. Therefore the application of a scientific bibliographic method of analysis seems to be an adequate choice.

In this chapter, the methods of bibliographic analysis are firstly briefly summarized. Then, the selected method is presented, and finally the results of application are discussed.

2.1. Methods of bibliographic analysis

Methods of bibliographic analysis can be classified into two groups (Osareh, 1996): methods of **bibliometrics**, and **citation based approaches**. Bibliometrics (also called bibliometrical statistics) are usually applied to evaluate the scientific work of an author, of an institute, or to characterize research intensity of a specific area. Most frequently the number of publications is used in descriptive statistics, however, in order to perform sophisticated studies it becomes insufficient due to the important differences between e.g. conceptual planning and application studies. Consequently, this criterion alone does not allow to perform reasonable comparisons with respect to the research intensity between different areas.

There are on the other hand successful attempts to apply bibliometrics methods to analyze research tendency, e.g. Sitarz et al. (2010) used a method based on words co-occurrence in article abstracts, to identify thematic clusters in an important area of chemical engineering, the distillation research area. Applying financial analyzing techniques they have made interesting predictions of the development trends of research as well, and identified diversified patterns of development like stagnation, revival, slow development or intensive growth.

The central assumption of citation based approaches is that if a paper cites another one that indicates a relationship between them. Developed on the base of this assumption citation based approaches attempt to group the papers with respect to their mutual relation. Greene et al. (2008) applied successfully a co-citation based approach to identify thematic clusters in Case-Based Reasoning literature and showed the research evolution and intensity in several themes.

2.1.1. Analysis of connection between papers

In order to identify documents likely to be closely related, citation-based approaches are the most appropriate. Two approaches can be distinguished: bibliographic coupling and co-citation analysis (Smith, 1981). Two documents are **bibliographically coupled** if their reference lists share one or more of the same cited documents. Two documents are **co-cited** when they are jointly cited in one or more subsequently published documents. To illustrate the concept of the two methods a set of five fictive papers has been created, illustrated in Figure 2-1, where $P_1 \dots P_5$ denote papers, and arrows denote citations (i.e. P_1 cites P_3 is denoted as $P_1 \rightarrow P_3$).

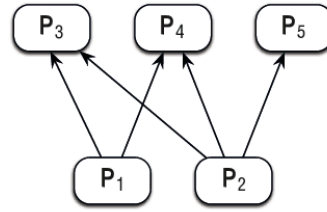


Figure 2-1 – Citation links between papers $P_1 \dots P_5$

Example 2-1

(from Greene et al., 2008)

The papers P_1 and P_2 are bibliographically coupled because they both cite articles P_3 and P_4 . From another point of view: P_3 and P_4 are co-cited by P_1 and P_2 . In Example 2-1 thus bibliographic coupling determines a relationship between P_1 and P_2 (based on the citations $(P_1, P_2) \rightarrow P_3$, and $(P_1, P_2) \rightarrow P_4$), while co-citation analysis suggests a relationship between P_3 and P_4 (based on the co-citations $P_1 \rightarrow (P_3, P_4)$ and $P_2 \rightarrow (P_3, P_4)$) and also a weaker relationship between P_4 and P_5 (based on the co-citation $P_2 \rightarrow (P_4, P_5)$).

An important difference between bibliographic coupling and co-citation analysis is that the former is an association intrinsic to the documents, thus the relationship remains static (once a paper is published, the reference list does not change any more). On the other hand the latter is a linkage extrinsic to the documents, and the connection is valid only so long as they continue to be co-cited. Therefore the relationship based on co-citations is dynamic. Because of the dynamic evolution of the scheduling literature the co-citation analysis based approach seems to be a more appropriate choice to study the bibliography of scheduling area.

2.1.2. Clustering techniques

Once the database of papers is constructed and a relation network is created, the next step is the regrouping of the papers. A group of papers with multiple connections to each other is called **cluster** and the regrouping process is called **clustering**. A cluster is determined by the connection rules of the network. The number, the forms and the interconnections of the generated clusters are then applied to analyze the results. In order to perform the clustering, i.e. to classify papers, a similarity measure between papers has to be firstly defined.

Gmür (2003) compared six widespread methods of similarity measure, and showed that the analysis based on Co-citation Score values (defined with Equation 2-1) is a particularly effective choice for clustering co-citation data. Compared to Co-citation Score, other approaches have several drawbacks, e. g.: overrating of most cited references (co-citation maximum based approach), overrating of co-citations between commonly cited references (citation mean based approach) or between less cited references (citation minimum based approach).

The calculation of Co-citation Score values is illustrated on Example 2-1. Suppose that the initial database of papers (seeded papers) is the set $\{P_3; P_4; P_5\}$ and we attempt to determine the Co-citation Score values based on the citing papers $\{P_1; P_2\}$.

Denote the co-citation count of two papers i and j with C_{ij} , and define this value as the number of papers that jointly cite papers i and j , for $i \neq j$. Diagonal elements C_{ii} are defined by convention as the total number of papers citing paper i . The co-citation counts of Example 2-1 are shown in Table 2-1:

Table 2-1 – Co-citation counts for Example 2-1

	P_3	P_4	P_5
P_3	2	2	1
P_4	2	2	1
P_5	1	1	1

The Co-citation Score of a pair of papers (P_i, P_j) is calculated on the base of their co-citation count value C_{ij} and the minimum and the mean of the respective citation counts C_{ii} and C_{jj} .

$$S_{ij} = \frac{C_{ij}^2}{\min(C_{ii}, C_{jj}) \cdot \text{mean}(C_{ii}, C_{jj})} \quad \forall i \neq j.$$

Equation 2-1 – Co-citation Score values

The Co-citation Score links two criteria (the minimum and the mean of the total citation counts of the two papers) for measuring the relative significance of a co-citation. To show the importance of this linking, let us suppose that paper i is cited much more than paper j (asymmetrical pairing). Then the normalization based on the mean of the citation counts C_{ii} and C_{jj} underestimates the importance of each co-citation for paper i . If the opposite is true, i.e. the citation counts are close to each other (symmetrical pairing) then the normalization based on the minimum leads to a distortion of the estimation. In Equation 2-1 symmetrical and asymmetrical co-citation pairings are taken into account with similar weighting, which is the main advantage of applying Co-citation Score values.

Each entry is now in the range $[0,1]$, where a larger value indicates a stronger association between the papers (Table 2-2).

Table 2-2 – Co-citation Score values for Example 2-1

	P ₃	P ₄	P ₅
P ₃	1	1	0.67
P ₄	1	1	0.67
P ₅	0.67	0.67	1

Once the measure of significance is decided, the next step is to bring together papers into clusters. Several clustering techniques have been developed. These techniques can be classified into the following three categories: traditional methods (hierarchical, agglomerative clustering), matrix decomposition techniques (e. g. Non-negative Matrix Factorization method) and combined techniques.

Whatever the clustering method is, the similarity measure has to be extended to define a distance between the different subsets of papers. While this distance function has to be explicitly defined for traditional methods, it is frequently encoded implicitly in matrix factorization techniques.

2.1.2.1. Traditional methods

Traditional methods of clustering use hierarchical approach. The hierarchical algorithms construct new clusters based on previously established clusters. Usually these algorithms are either agglomerative ("bottom-up") or divisive ("top-down"). Whatever the principle of the algorithm is, the result of a hierarchical algorithm is a tree of clusters. Except for the root node every cluster has exactly one parent and except for the leaf nodes every cluster has one or more child nodes.

Agglomerative algorithms begin with defining an initial cluster for each paper (i.e. each paper is considered as a set of one paper) and merge the initial clusters successively into new clusters. Agglomerative clustering is based on the following principle: find the two clusters with the smallest distance value, merge them into a single new (parent) cluster, and repeat this process until all objects and clusters are merged into a single one (root node). During the merging process, it is necessary to record the distances of the merged objects in order to construct a graph. Let us define the inverse of the co-citation count as a distance function for Example 2-1. Then from papers P₃ – P₅ firstly P₃ and P₄ are merged as their distance is the smallest ($1/1 = 1$). Then, as there is only one paper remained, the merging of [P₃, P₄] with [P₅] leads to the root cluster.

Divisive algorithms begin with an initial cluster containing the whole set of papers and proceed to divide this cluster successively into smaller clusters. Applying the same distance function, in the set of papers P₃ – P₅ the largest distance found is the distance between P₅ and the rest of the group ($1/0.67 = 1.5$). Thus [P₅] is separated from [P₃, P₄]. Finally, the set [P₃, P₄] is divided to [P₃] and [P₄] which leads to a tree where all final nodes are leaf nodes.

Both algorithm leads to the same scheme for Example 2-1 (Figure 2-2).

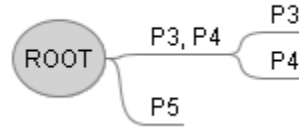


Figure 2-2 – Hierarchical clustering of Example 2-1

We remark that each paper can only reside in a single branch of the tree at a given level, and can only belong to a single leaf node.

2.1.2.2. Matrix decomposition techniques

In order to apply a matrix decomposition technique, firstly the information about the connections between the base articles has to be transformed into a matrix. The construction of this matrix depends on the applied technique, e.g. the matrix of Co-Citation Score values is applied in this work.

Then, this matrix is decomposed applying a non-negative matrix factorization approach. Non-negative matrix factorization (NMF) is a group of algorithms in multivariate analysis and linear algebra where a matrix \mathbf{X} is factorized into (usually) two matrices: $NMF(\mathbf{X}) \rightarrow \mathbf{W}, \mathbf{H}$ where the original matrix \mathbf{X} is the production of the factor matrixes \mathbf{W} and \mathbf{H} . Different matrix factorization methods have been developed, e. g. principal component analysis and singular value decomposition (Lee and Seung, 2001). Finally, based on the interpretation of the result matrices, membership values are associated to each paper – cluster pairing, indicating the weight of membership of the given paper to the corresponding cluster.

2.1.2.3. Combinations

A distinct drawback of the hierarchical techniques lies in the fact that each paper can only reside in a single branch of the tree at a given level, and can only belong to a single leaf node. On the other hand, drawbacks of the matrix factorization are notably its sensitivity to the choice of parameter, and the difficulty in interpreting the clusters produced by the decomposition procedure. In order to eliminate the drawbacks of the two approaches important attempts have been made to work out combined strategies, as e.g. the Ensemble NMF algorithm detailed in Section 2.2. These techniques construct the clusters by applying their own method which is different from technique to technique.

2.1.3. Ranking papers

After building the clusters, the importance of a paper belonging to a given cluster has to be determined. Different measures exist for ranking the importance of a paper; the most of them are based on graph metrics, like the theory of centrality. The degree centrality of a vertex is the number of edges that are connected to the vertex. In social network analysis the degree centrality is commonly used as a means of assessing importance to a vertex, as the greater the degree of a vertex is, the higher influence it will potentially have in the network. Other types of centrality can also be defined which are dedicated to be used in special cases, e.g. betweenness centrality (indicates if the vertex is in many shortest paths between other vertices), closeness centrality (indicates the mean distance of the vertex from the rest of the network, i.e. the mean of shortest paths from the vertex to all other vertices reachable from it), etc... In the following, we use the degree centrality value.

Centrality based approaches require to construct a graph of connected papers. As the centrality value measures the potential influence of a vertex in the graph it indicates the importance value of a paper belonging to a given cluster.

In order to compare different clusters it is reasonable to normalize centrality values. The normalization can be performed with respect to the total number of pairs of papers in the given cluster, or with respect to the total number of citations occurring in the corresponding cluster.

2.1.4. Labelling clusters

As usually a bibliographic analysis creates a relevant number of clusters, it would be time and effort consuming to label the different clusters manually. The most frequent techniques to perform cluster labelling are based on word frequency analysis of the papers belonging to the cluster. Words, occurring more often than others in a cluster, receive a higher score and supposed to be more indicative to the cluster.

Unfortunately, the whole text is rarely available for all paper. Therefore, either the analysis has to be restricted to abstract, title and keywords, or the identification of a cluster is based only on a subset of the papers belonging to it.

2.1.5. Conclusion

In order to reach a landscape on research situation of the domain of scheduling problems a bibliographic analysis method has been applied. Bibliographic methods have been shortly introduced. The methods of bibliographic analysis are regrouped into two main categories: bibliometrics and citation-based approaches. To establish and map a connection network among papers published in the scheduling thematic, the citation based approaches seem to be the most appropriate. These methods can be categorized in two groups: bibliographic

coupling and co-citation based approaches. Because of their dynamic attribute, co-citation based approaches are preferred in order to study a domain under dynamic evolution, like scheduling problems.

Next, the clustering techniques have been classified and briefly discussed. These techniques attempt to regroup the papers into sets (clusters) consisting of similar papers, based on a similarity measure and a cluster constructing method. Based on the literature (Gmür, 2003) the Co-citation Score values have been selected as similarity measure. Cluster constructing methods are regrouped in three categories: traditional methods, matrix-factorization techniques and combined approaches. In this work the Ensemble NMF algorithm (Greene et al., 2008) has been chosen, because it combines the ability of matrix-factorization techniques to accurately identify overlapping structures with the interpretability and visualization benefits of hierarchical techniques.

Finally the ranking of paper importance and cluster labelling techniques have been discussed. The ranking of paper importance is usually based on the centrality value (e. g. degree centrality), coming from graph theory. The centrality value indicates the importance of a vertex in a graph representing a network of connected papers. In this work the degree centrality is applied. Cluster labelling techniques are most frequently based on word frequency analysis, performed either on the full text of the papers belonging to a given cluster, or on abstract title and keywords. In this work the application of the first approach is tested on an example cluster, and the application of the second approach is examined.

2.2. The Ensemble NMF algorithm

The Ensemble NMF clustering algorithm (Greene et al., 2008) is belonging to the group of combined methods. The algorithm is based on co-citation of papers, and uses Co-citation Score values as a basis for measuring the similarity between papers. Based on the decomposition of the matrix of Co-citation Score values the algorithm provides a “soft” hierarchical clustering, where papers can belong to more than one cluster. This attribute is useful when the examined problematic is complex and papers can naturally relate to more than one research area, like in scheduling domain.

Firstly the initial database of papers is constructed. The elements of this database are the seed papers, collected e. g. from queries sent to scientific search engines. Then the papers citing the seed papers and the citation links are determined. Next, based on the co-citation counts the matrix of Co-citation Score values is calculated.

The rest of the algorithm can be separated into two phases: a **generation** phase and an **integration** phase. In the generation phase, a matrix decomposition technique is applied iteratively to the Co-citation Score matrix, while the integration phase is the construction of meta-clusters. A **membership vector** is associated to each meta-cluster, indicating the weight of membership of papers to a given meta-cluster.

The final step is the association of papers to the obtained meta-clusters. A paper j is associated to a meta-cluster M_a if the element j of the associated membership vector v_a is higher or equal to a threshold value. This threshold value is to be determined according to the given structure.

2.2.1. Application of the method to scheduling literature

The above described method has been applied to scheduling research area in order to gain a landscape on the research situation of this domain.

2.2.1.1. Construction of the database of papers

To apply the Ensemble NMF method to scheduling problems, queries have been sent to ISI Web of Knowledge in order to obtain papers and citation data. A research with the keywords “process scheduling” provided 8158 results. The results have been refined to the subjects “Computer Science Theory & Methods” and “Industrial Engineering” obtaining thus 2406 results. Finally these papers have been sorted with respect to their citation counts and the first 100 articles (from 1984 to 2005) have been used as the initial database (presented in the appendix of this chapter). The sort based on the citation count was made in order to obtain a sufficiently high number of citation links to the analysis. Next, the papers citing the 100 seed papers have been collected (4839 results have been recorded) and the number of co-citations have been determined.

2.2.1.2. Clustering

After the building of the matrix of co-citations and then the matrix of Co-Citation Score values, the Ensemble NMF algorithm has been performed. The value of the parameter k has been varied increasingly from 3 to 10, leading to 52 basis vectors and initial meta-clusters. Applying the agglomerative clustering to the gained meta-clusters, 103 nodes have been obtained. 12 of these nodes proved to be instable based on the splitting factor criterion performed with a threshold value 0.9 (the value proposed by Greene et al., 2008). The instable nodes have been eliminated.

With the remainder 91 meta-clusters, the papers have been associated with respect to the adequate membership vector values based on a threshold value of 0.45. This value is higher than in the case of Greene et al. (2008), because our intention was more to identify papers corresponding to different clusters rather than the interdisciplinary ones.

Based on this strong threshold, 51 clusters proved to be “empty” (no paper associated), resulting to 40 meta-clusters. Finally, 18 meta-clusters contained few papers (less than 5), most of them with zero centrality. The poor population of these meta-clusters indicates that the number of papers belonging to that cluster is not high enough to successfully draw conclusions from applying the centrality theory. Therefore these clusters were also eliminated, and 22 clusters were stored.

For the reason of consistency the original numbering of clusters has been maintained through all the process. In Figure 2-4 the hierarchical tree gained from the algorithm is presented.

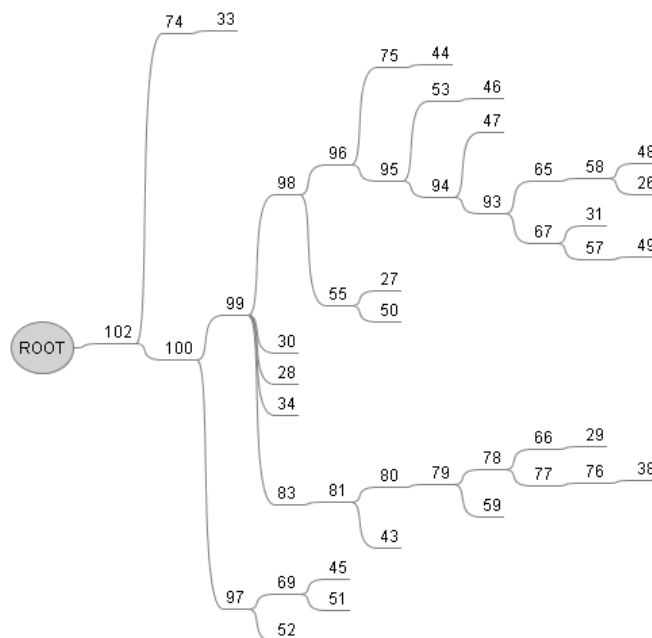


Figure 2-3 – NMF tree structure of the identified clusters

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

Figure 2-4 illustrates the wideness and variety of the scheduling domain. Due to the faced problems in the labelling of clusters (detailed in section 2.2.2.3) the detailed interpretation of this tree structure requires further research work.

2.2.1.3. Analysis and labelling of clusters

The importance of a paper in a meta-cluster has been defined as the degree centrality of paper in the corresponding meta-cluster. In order to associate degree centrality values to each paper – meta-cluster pair, the network of interconnected member papers has been created for each meta-cluster. The degree centrality of a paper in a meta-cluster is then defined as the degree centrality of the vertex associated to the paper in the network associated to the meta-cluster. The degree centrality values have been normalized with respect to the total number of co-citation pairs.

In order to label the meta-clusters the RapidMiner software is used, proposed by Sitarz et al. (2010) and available from <http://rapid-i.com> to download. This software among other functions allows to perform word frequency analysis.

The lists of the most frequent words in title, abstract and keywords have been created for each meta-cluster. However, a serious problem has been faced: the obtained terms proved to be too general (e. g. “new, well performing method”, “paper”, “scheduling” ...), and most of the words and expressions have no specificity to the area represented by the meta-cluster, which makes it impossible to draw sophisticated conclusions.

Therefore, an other possible point of views had to be examined. Firstly, if some previous knowledge is already available on the concerned area it is possible to find the best candidate among the clusters to describe a given thematic. For example: if one would like to identify the cluster which corresponds the most to scheduling problems dealing with deadlines then the most probable candidate is the cluster whose list contains in high rank the words “deadline”, or “due-date”. This approach is applied in Section 2.3.2.

On the other hand, as applying a word frequency analysis based on title, abstract and keywords the identification of the different meta-clusters did not prove to be successful, the application possibilities of full text analysis have been examined in Section 2.3.3.

2.3. Discussion of results

Firstly, some general remarks will be made on the research activity in scheduling literature. Then applying the above described two labelling approach will be discussed on some illustrative examples.

2.3.1. General remarks

Scheduling research is a dynamic, challenging area, being in permanent evolution. In Figure 2-5 the centrality values of the articles of the initial database are shown in the time. Each bubble corresponds to one of the 100 seemed papers. The size of each bubble corresponds to the total number of citation of the associated paper.

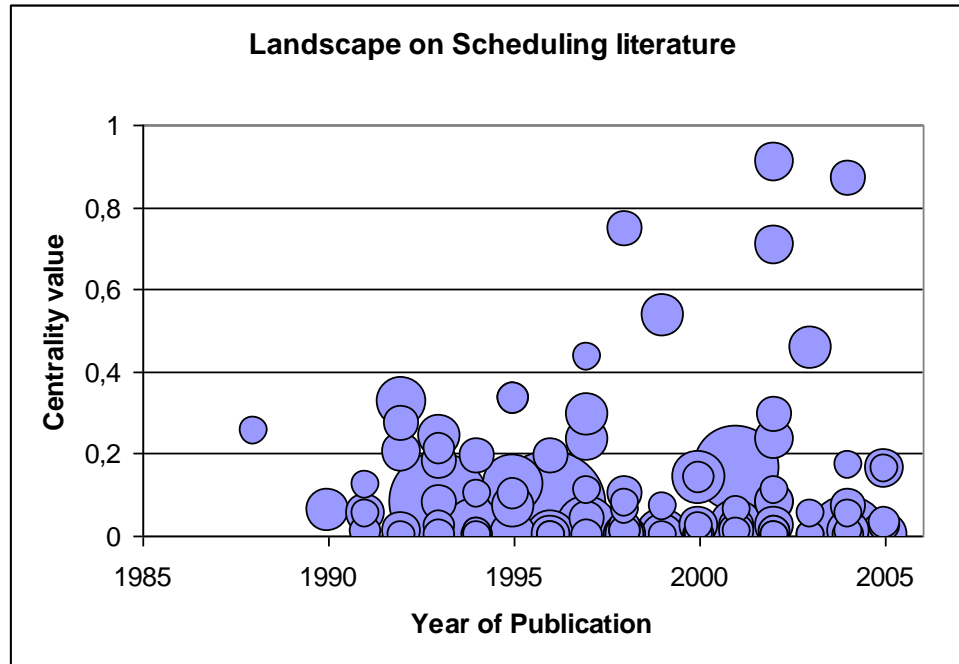


Figure 2-4 – Evolution of scheduling literature during the last decades

In Figure 2-5 it can be seen that there is a significant, continuous research activity which indicates that the trend of the dynamic evolution of the area continues. However, there is no information about the last few years. The explanation of this inconvenience is that (in order to obtain a database which is rich in citation links and thus potentially in co-citation information as well) during the creation of the initial database the papers had been sorted with respect to their citation counts. This sorting led to the drawback, that recent, and thus yet not frequently cited papers are underrepresented. On the other hand, from the tendency of earlier publications the trend of research can be followed, and the activity in the area can be mapped, which satisfies the original intention.

2.3.2. Finding the most appropriate meta-cluster to a keyword

In this section the identification process is restricted to identify some specific clusters corresponding to well-defined areas of process scheduling. Voluntarily, the terms “resources”, and “due-dates” have been chosen. The cluster whose list contains in high rank the words “resource” has been identified as the cluster dealing with

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

resource considerations, while the cluster where the occurrences of “due-date” (or its synonym “deadline”) are high, has been assigned to the papers dealing with deadline.

To illustrate the results of the clustering algorithm based on the above detailed method, two example clusters are presented in Figure 2-6 and Figure 2-7. The first cluster is the one which is the best candidate to assign to resource considerations and the second cluster is the one who corresponds mostly to the due-date attribute.

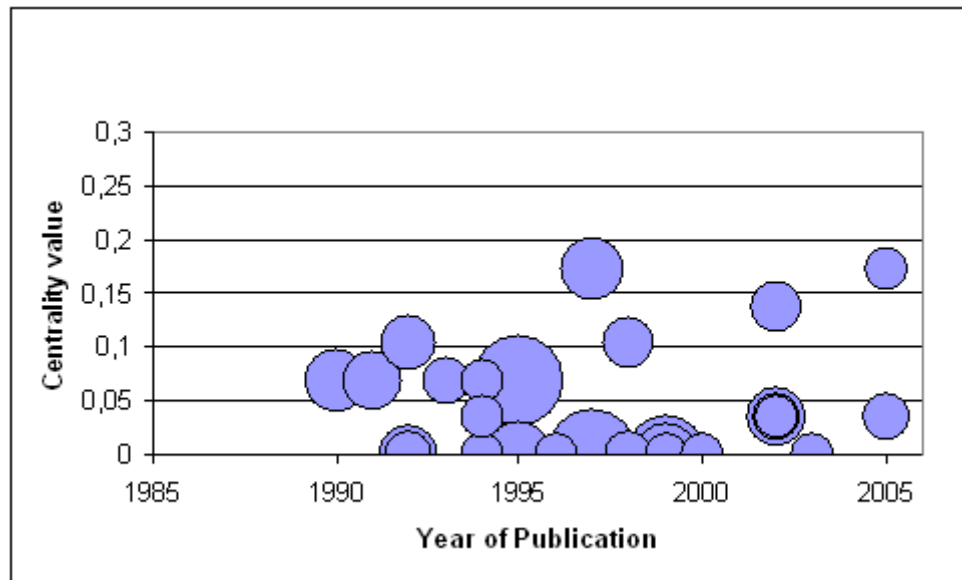


Figure 2-5 – Cluster of Resource Considerations

In Figure 2-6 continuous, stable research effort can be seen, with several significant papers. Resource handling requires constructing complex mathematical models. Since lot-sizing is frequently considered and mixing/splitting operations are allowed, these types of problems are very difficult to solve. No “central” article can be determined (all centrality values are below 0.25), which means that there is no general best solution that is able to deal with any type of this problem.

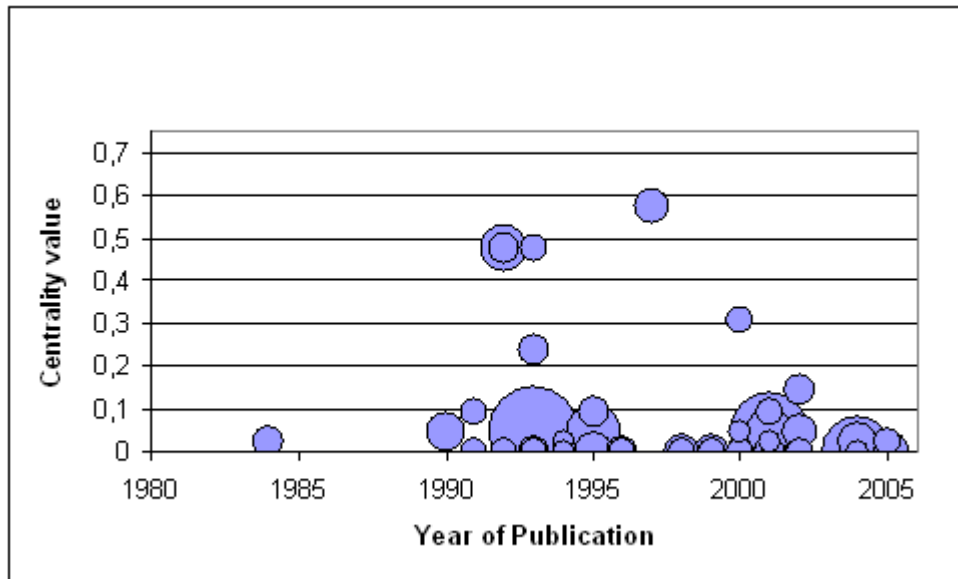


Figure 2-6 – Cluster of Deadlines

In Figure 2-7 dynamic, continuously challenging research can be seen, with some central, significant papers from the nineties. It can be thus deduced that the main foundation of the area has been established during the last decade of the twentieth century, and these results are continuously used and regularly referred by new research workers. With other words, general considerations of this aspect have been deeply analyzed, but improvement, or specific application approaches require further work.

2.3.3. Cluster analysis based on the full text of papers

In this section the performance of the full-text analysis is studied, in an example meta-cluster (Cluster n°26). 34 papers belong to this meta-cluster, from which 22 are available. The application of word frequency analysis on the full text of these 22 available papers was successful, and based on the cluster's keywords (queue, protocol, grid, workload, balancing, network...) the cluster's thematic has been identified as "network protocols, networking workload balancing".

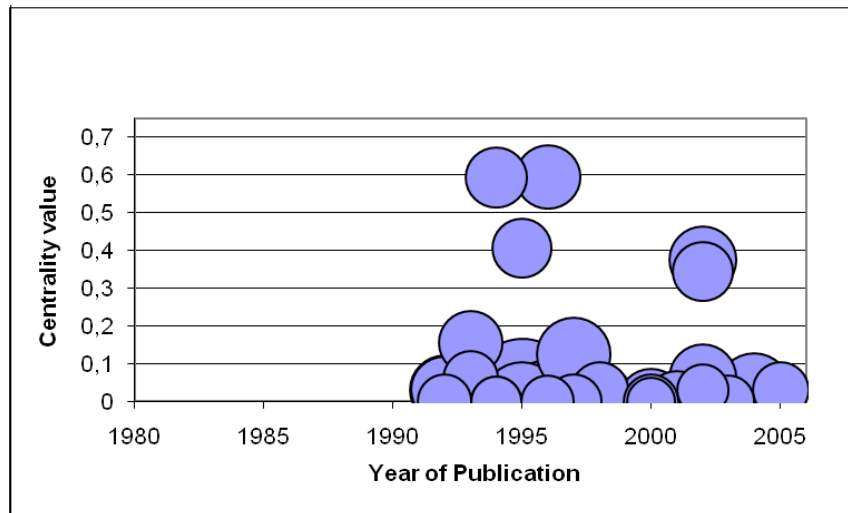


Figure 2-7 – Cluster of network protocols and networking workload balancing

It is shown in Figure 2-7 that the interest towards the scheduling of network protocols and networking workload balancing started in the early nineties, probably because of the spread of personal computers. From 1994-1996 significant research breakthroughs have been performed, founding further research in this area. These central papers deal with workload control concepts, just in time production, and integrated process planning. Another intensive phase can be noticed in 2002, the corresponding central papers deal with integrated process planning and scheduling.

2.4. Conclusion

In order to obtain a landscape on the situation of research concerning scheduling problems, a bibliographic study has been made on this area. To perform the analysis a scientific bibliographical approach has been chosen, the Ensemble NMF algorithm of Greene et al. (2008). This algorithm belongs to the group of combined methods based on co-citation analysis.

Generally a dynamic and strong research tendency has been found which justifies further research work in scheduling problems area. Two approaches have been introduced in order to identify clusters: the word frequency analysis based on title, keywords and abstracts, and the analysis based on full text. The former one proved to be not effective enough to identify and label the clusters, however, it was able to associate clusters to some predetermined thematic keywords. The inconvenience of the latter is the limitation to a subset of papers, due to the constrained availability to full texts of papers. Nevertheless, the latter approach was still able to identify the thematic of the cluster presented as illustrative examples.

To illustrate the first approach two thematic have been chosen: resource considerations, and deadline attributes. The research activity in the assigned clusters has been presented in centrality – time diagrams. To the former a continuous strong research activity has been found, with no evident breakthrough, but stable activity. To the latter a continuous, but dynamically challenging research activity has been identified, indicating that the

main foundation of the area has been established, and thus current research is focusing to amelioration and specific application approaches.

To illustrate the second approach, an example cluster has been chosen. The analysis has been performed on the full text of the 22 available papers from the 34 belonging to the cluster. The identification was successful, and the cluster has been associated to network protocols, network workload balancing thematic. Intensive research has been found from the early nineties in this area, which seems to be continued.

To go further towards the goal of constructing a decision support system the next step is to identify the associations between problems, models and methods. For this reason, and also to better understand the results of the bibliographic analysis a classification of scheduling problems, mathematical models, and adapted solving methods is presented in the next chapter. Once extended for all clusters, the results of the bibliographic analysis performed in this chapter will facilitate the construction of an effective similarity function for the CBR, which is a perspective of this work.

2.5. References of Chapter Two

Osareh, F. Bibliometrics, Citation Analysis and Co-citation Analysis. A review of literature. *Libri* **1996**, 46, 149-158.

Smith, K. Cocitation analysis. In: *W.G. Potter: Bibliometrics* Illinois **1981**, 83-106.

Sitarz, R., Kraslawski, A., Jezowski, J. Dynamics of Knowledge Flow in Research on Distillation. *Computer Aided Chemical Engineering* **2010**, 28, 583-588.

Gmür, M. Cocitation analysis and the search for invisible colleges A methodological evaluation. *Scientometrics* **2003**, 57, 27-57.

Greene, D., Freyne, J., Smyth, B., & Cunningham, P. An analysis of research themes in the CBR conference literature. In: *K-D Althoff et al: ECCBR LNAI 5239 Berlin, Springer* **2008**, 18-43.

Lee, D. D.; Seung, H. S. Algorithms for Non-negative Matrix Factorization. In: *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference. MIT Press*, **2001**, 556–562.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

2.6. Appendix

Appendix Table – References of the seed papers

Paper number	Title	Authors	Source Title	Publication Year	Volume
1	Efficient fair queuing using deficit round-robin	Shreedhar, M; Varghese, G	IEEE-ACM TRANSACTIONS ON NETWORKING	1996	4
2	HIGH-SPEED SWITCH SCHEDULING FOR LOCAL-AREA NETWORKS	ANDERSON, TE; OWICKI, SS; SAXE, JB; et al.	ACM TRANSACTIONS ON COMPUTER SYSTEMS	1993	11
3	Applying the design structure matrix to system decomposition and integration problems: A review and new directions	Browning, TR	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	2001	48
4	Solving unsymmetric sparse systems of linear equations with PARDISO	Schenk, O; Gartner, K	FUTURE GENERATION COMPUTER SYSTEMS	2004	20
5	AN EXTENDIBLE APPROACH FOR ANALYZING FIXED PRIORITY HARD REAL-TIME TASKS	TINDELL, KW; BURNS, A; WELLINGS, AJ	REAL-TIME SYSTEMS	1994	6
6	Eliminating receive livelock in an interrupt-driven kernel	Mogul, JC; Ramakrishnan, KK	ACM TRANSACTIONS ON COMPUTER SYSTEMS	1997	15
7	A SURVEY ON THE RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM	OZDAMAR, L; ULUSOY, G	IIE TRANSACTIONS	1995	27
8	Recent developments in evolutionary computation for manufacturing optimization: Problems, solutions, and comparisons	Dimopoulos, C; Zalzal, AMS	IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION	2000	4
9	Adversarial queuing theory	Borodin, A; Kleinberg, J; Raghavan, P; et al.	JOURNAL OF THE ACM	2001	48
10	The timed asynchronous distributed system model	Cristian, F; Fetzer, C	IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS	1999	10
11	DYNAMIC DISPATCHING ALGORITHM FOR SCHEDULING MACHINES AND AUTOMATED GUIDED VEHICLES IN A FLEXIBLE MANUFACTURING SYSTEM	SABUNCUOGLU, I; HOMMERTZHEIM, DL	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	1992	30
12	Task clustering and scheduling for distributed memory parallel architectures	Palis, MA; Liou, JC; Wei, DSL	IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS	1996	7
13	Stable scheduling policies for fading wireless channels	Eryilmaz, A; Srikant, R; Perkins, JR	IEEE-ACM TRANSACTIONS ON NETWORKING	2005	13
14	P(3)L - A STRUCTURED HIGH-LEVEL PARALLEL LANGUAGE, AND ITS STRUCTURED SUPPORT	BACCI, B; DANELUTTO, M; ORLANDO, S; et al.	CONCURRENCY-PRACTICE AND EXPERIENCE	1995	7
15	Minimizing makespan on a single batch processing machine with dynamic job arrivals	Lee, CY; Uzsoy, R	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	1999	37

16	A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling	Kim, YK; Park, K; Ko, J	COMPUTERS & OPERATIONS RESEARCH	2003	30
17	A ROBUST ADAPTIVE SCHEDULER FOR AN INTELLIGENT WORKSTATION CONTROLLER	CHO, H; WYSK, RA	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	1993	31
18	Minimizing total completion time in a two-machine flowshop with a learning effect	Lee, WC; Wu, CC	INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS	2004	88
19	Bidding-based process planning and scheduling in a multi-agent system	Gu, P; Balasubramanian, S; Norrie, DH	COMPUTERS & INDUSTRIAL ENGINEERING	1997	32
20	SCHEDULING IN A SEQUENCE-DEPENDENT SETUP ENVIRONMENT WITH GENETIC SEARCH	RUBIN, PA; RAGATZ, GL	COMPUTERS & OPERATIONS RESEARCH	1995	22
21	GENETIC ALGORITHMS AND JOB SHOP SCHEDULING	BIEGEL, JE; DAVERN, JJ	COMPUTERS & INDUSTRIAL ENGINEERING	1990	19
22	Efficient schemes for nearest neighbor load balancing	Diekmann, R; Frommer, A; Monien, B	PARALLEL COMPUTING	1999	25
23	Efficient dispatching rules for scheduling in a job shop	Holthaus, O; Rajendran, C	INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS	1997	48
24	Economical evaluation of disassembly operations for recycling, remanufacturing and reuse	Johnson, MR; Wang, MH	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	1998	36
25	SIMULTANEITY IN DISCRETE-TIME SINGLE-SERVER QUEUES WITH BERNOULLI INPUTS	GRAVEY, A; HEBUTERNE, G	PERFORMANCE EVALUATION	1992	14
26	Minimizing the makespan on a batch machine with non-identical job sizes: an exact procedure	Dupont, L; Dhaenens-Flipo, C	COMPUTERS & OPERATIONS RESEARCH	2002	29
27	A genetic algorithm to minimize maximum lateness on a batch processing machine	Wang, CS; Uzsoy, R	COMPUTERS & OPERATIONS RESEARCH	2002	29
28	Steel-making process scheduling using Lagrangian relaxation	Tang, LX; Luh, PB; Liu, JY; et al.	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	2002	40
29	AUTOMATED 2-MACHINE FLOWSHOP SCHEDULING - A SOLVABLE CASE	KISE, H; SHIOYAMA, T; IBARAKI, T	IIE TRANSACTIONS	1991	23
30	Advanced planning and scheduling with outsourcing in manufacturing supply chain	Lee, YH; Jeong, CS; Moon, C	COMPUTERS & INDUSTRIAL ENGINEERING	2002	43
31	Determining optimum Genetic Algorithm parameters for scheduling the manufacturing and assembly of complex products	Pongcharoen, P; Hicks, C; Braiden, PM; et al.	INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS	2002	78
32	Pull-driven scheduling for pipe-spool installation: Simulation of lean construction technique	Tommelein, ID	JOURNAL OF CONSTRUCTION ENGINEERING AND MANAGEMENT-ASCE	1998	124
33	Implementation and performance of integrated	Cao, P; Felten, EW; Karlin, AR; et al.	ACM TRANSACTIONS ON COMPUTER	1996	14

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

	application-controlled file caching, prefetching, and disk scheduling		SYSTEMS		
34	INTELLIGENT SCHEDULING WITH MACHINE LEARNING CAPABILITIES - THE INDUCTION OF SCHEDULING KNOWLEDGE	SHAW, MJ; PARK, S; RAMAN, N	IIE TRANSACTIONS	1992	24
35	A constraint-based method for project scheduling with time windows	Cesta, A; Oddi, A; Smith, SF	JOURNAL OF HEURISTICS	2002	8
36	Generation of efficient nested loops from polyhedra	Quillere, F; Rajopadhye, S; Wilde, D	INTERNATIONAL JOURNAL OF PARALLEL PROGRAMMING	2000	28
37	A simulation-based process model for managing complex design projects	Cho, SH; Eppinger, SD	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	2005	52
38	Workload control concepts in job shops - A critical assessment	Land, M; Gaalman, G	INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS	1996	46
39	DYNAMIC SCHEDULING - A SURVEY OF RESEARCH	SURESH, V; CHAUDHURI, D	INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS	1993	32
40	SCHEDULING IN A 2-STAGE MANUFACTURING PROCESS	NARASIMHAN, SL; PANWALKAR, SS	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	1984	22
41	Robust and reactive project scheduling: a review and classification of procedures	Herroelen, W; Leus, R	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	2004	42
42	EXPERIMENTAL INVESTIGATION OF FMS MACHINE AND AGV SCHEDULING RULES AGAINST THE MEAN FLOW-TIME CRITERION	SABUNCUOGLU, I; HOMMERTZHEIM, DL	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	1992	30
43	Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing	Melouk, S; Damodaran, P; Chang, PY	INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS	2004	87
44	Optimizing preventive maintenance for mechanical components using genetic algorithms	Tsai, YT; Wang, KS; Teng, HY	RELIABILITY ENGINEERING & SYSTEM SAFETY	2001	74
45	Approximation techniques for average completion time scheduling	Chekuri, C; Motwani, R; Natarajan, B; et al.	SIAM JOURNAL ON COMPUTING	2001	31
46	A simulation study on lot release control, mask scheduling, and batch scheduling in semiconductor wafer fabrication facilities	Kim, YD; Lee, DH; Kim, JU; et al.	JOURNAL OF MANUFACTURING SYSTEMS	1998	17
47	Flowshop scheduling/sequencing research: A statistical review of the literature, 1952-1994	Reisman, A; Kumar, A; Motwani, J	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	1997	44
48	Minimizing mean flow times criteria on a single batch processing machine with non-identical jobs sizes	Ghazvini, FJ; Dupont, L	INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS	1998	55
49	LOAD-ORIENTED MANUFACTURING	BECHTE, W	PRODUCTION PLANNING &	1994	5

	CONTROL JUST-IN-TIME PRODUCTION FOR JOB SHOPS		CONTROL		
50	INTERACTIVE DYNAMIC LAYOUT PLANNING	TOMMELEIN, ID; ZOUEN, PP	JOURNAL OF CONSTRUCTION ENGINEERING AND MANAGEMENT-ASCE	1993	119
51	Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain	Moon, C; Kim, J; Hur, S	COMPUTERS & INDUSTRIAL ENGINEERING	2002	43
52	Shift scheduling for steppers in the semiconductor wafer fabrication process	Kim, S; Yea, SH; Kim, B	IIE TRANSACTIONS	2002	34
53	A HIERARCHICAL BICRITERION APPROACH TO INTEGRATED PROCESS PLAN SELECTION AND JOB-SHOP SCHEDULING	BRANDIMARTE, P; CALDERINI, M	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	1995	33
54	A PROGRESSIVE APPROACH FOR THE INTEGRATION OF PROCESS PLANNING AND SCHEDULING	HUANG, SH; ZHANG, HC; SMITH, ML	IIE TRANSACTIONS	1995	27
55	Discrete-time multiserver queues with priorities	Laevens, K; Bruneel, H	PERFORMANCE EVALUATION	1998	33
56	Automatic storage management for parallel program	Lefebvre, V; Feautrier, P	PARALLEL COMPUTING	1998	24
57	SCHEDULING OF REPETITIVE PROJECTS WITH COST OPTIMIZATION	MOSELHI, O; ELRAYES, K	JOURNAL OF CONSTRUCTION ENGINEERING AND MANAGEMENT-ASCE	1993	119
58	PREDICTION-BASED DYNAMIC LOAD-SHARING HEURISTICS	GOSWAMI, KK; DEVARAKONDA, M; IYER, RK	IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS	1993	4
59	New grid scheduling and rescheduling methods in the GrADS Project	Berman, F; Casanova, H; Chien, A; et al.	INTERNATIONAL JOURNAL OF PARALLEL PROGRAMMING	2005	33
60	Analysis and design of an adaptive minimum reasonable inventory control system	Towill, DR; Evans, GN; Cheema, P	PRODUCTION PLANNING & CONTROL	1997	8
61	INTELLIGENT WORKSTATION CONTROLLER FOR COMPUTER-INTEGRATED MANUFACTURING - PROBLEMS AND MODELS	CHO, HB; WYSK, RA	JOURNAL OF MANUFACTURING SYSTEMS	1995	14
62	AN EFFICIENT 4-PHASE HEURISTIC FOR THE GENERALIZED ORIENTERING PROBLEM	RAMESH, R; BROWN, KM	COMPUTERS & OPERATIONS RESEARCH	1991	18
63	A dynamic and reliability-driven scheduling algorithm for parallel real-time jobs executing on heterogeneous clusters	Qin, X; Jiang, H	JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING	2005	65
64	Managing a portfolio of interdependent new product candidates in the pharmaceutical industry	Blau, GE; Pekny, JF; Varma, VA; et al.	JOURNAL OF PRODUCT INNOVATION MANAGEMENT	2004	21
65	An available-to-promise system for TFT LCD manufacturing in supply chain	Jeong, B; Sim, SB; Jeong, HS; et al.	COMPUTERS & INDUSTRIAL ENGINEERING	2002	43

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

66	Wiener model based nonlinear predictive control	Gerksic, S; Juricic, D; Strmcnik, S; et al.	INTERNATIONAL JOURNAL OF SYSTEMS SCIENCE	2000	31
67	New dispatching rules for shop scheduling: a step forward	Jayamohan, MS; Rajendran, C	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	2000	38
68	Heuristics for minimizing tool switches when scheduling part types on a flexible machine	Hertz, A; Laporte, G; Mittaz, M; et al.	IIE TRANSACTIONS	1998	30
69	TEMPORAL PROOF METHODOLOGIES FOR TIMED TRANSITION-SYSTEMS	HENZINGER, TA; MANNA, Z; PNUELI, A	INFORMATION AND COMPUTATION	1994	112
70	AN EFFICIENT DYNAMIC DISPATCHING RULE FOR SCHEDULING IN A JOB-SHOP	RAGHU, TS; RAJENDRAN, C	INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS	1993	32
71	Planning and control of rework in the process industries: a review	Flapper, SDP; Fransoo, JC; Broekmeulen, RACM; et al.	PRODUCTION PLANNING & CONTROL	2002	13
72	A tabu search approach for the resource constrained project scheduling problem	Thomas, PR; Salhi, S	JOURNAL OF HEURISTICS	1998	4
73	CHARACTERIZING THE MANUFACTURING SCHEDULING PROBLEM	PARUNAK, HV	JOURNAL OF MANUFACTURING SYSTEMS	1991	10
74	Temporal representation and reasoning for workflow in engineering design change review	Chinn, SJ; Madey, GR	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	2000	47
75	Lot streaming and scheduling multiple products in two-machine no-wait flowshops	Sriskandarajah, C; Wagneur, E	IIE TRANSACTIONS	1999	31
76	Modeling reentrant manufacturing systems with inspection stations	Narahari, Y; Khan, LM	JOURNAL OF MANUFACTURING SYSTEMS	1996	15
77	A comprehensive study of the complexity of multiparty interaction	Joung, YJ; Smolka, SA	JOURNAL OF THE ACM	1996	43
78	A HEURISTIC OF SCHEDULING PARALLEL TASKS AND ITS ANALYSIS	WANG, QZ; CHENG, KH	SIAM JOURNAL ON COMPUTING	1992	21
79	Parallel machine scheduling under a grade of service provision	Hwang, HC; Chang, SY; Lee, K	COMPUTERS & OPERATIONS RESEARCH	2004	31
80	The effect of various operators on the genetic search for large scheduling problems	Nearchou, AC	INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS	2004	88
81	A decoupled scheduling approach for Grid application development environments	Dail, H; Berman, F; Casanova, H	JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING	2003	63
82	A new adaptive neural network and heuristics hybrid approach for job-shop scheduling	Yang, SX; Wang, DW	COMPUTERS & OPERATIONS RESEARCH	2001	28
83	A genetic algorithm based procedure for more realistic job shop scheduling problems	Candido, MAB; Khator, SK; Barcia, RM	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	1998	36
84	HEURISTIC SOLUTION TO THE SCHEDULING PROBLEMS IN FLEXIBLE MANUFACTURING SYSTEM	MUKHOPADHYAY, SK; MAITI, B; GARG, S	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	1991	29

85	PROCESS PLANNING AND SCHEDULING - A METHOD OF INTEGRATION FOR PRODUCTIVITY IMPROVEMENT	SUNDARAM, RM; FU, SS	COMPUTERS & INDUSTRIAL ENGINEERING	1988	15
86	A type system for lock-free processes	Kobayashi, N	INFORMATION AND COMPUTATION	2002	177
87	Stack and queue layouts of directed acyclic graphs: Part I	Heath, LS; Pemmaraju, SV; Trenk, AN	SIAM JOURNAL ON COMPUTING	1999	28
88	An experimental evaluation of heuristics for scheduling in a real-life flowshop with sequence-dependent setup times of jobs	Parthasarathy, S; Rajendran, C	INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS	1997	49
89	MODELING WORK SPACE TO SCHEDULE REPETITIVE FLOORS IN MULTISTORY BUILDINGS	THABET, WY; BELIVEAU, YJ	JOURNAL OF CONSTRUCTION ENGINEERING AND MANAGEMENT-ASCE	1994	120
90	SCHEDULING MANUFACTURING SYSTEMS WITH WORK-IN-PROCESS INVENTORY CONTROL - MULTIPLE-PART-TYPE SYSTEMS	BAI, SX; GERSHWIN, SB	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	1994	32
91	Model driven service composition	Orriens, B; Yang, J; Papazoglou, MP	SERVICE-ORIENTED COMPUTING - ICSOC 2003. LECTURE NOTES IN COMPUTER SCIENCE	2003	2910
92	Concurrent process planning and scheduling in distributed virtual manufacturing	Wu, SH; Fuh, JYH; Nee, AYC	IIE TRANSACTIONS	2002	34
93	A branch and bound algorithm for minimizing mean flow time on a single batch processing machine	DuPont, L; Ghazvini, FJ	INTERNATIONAL JOURNAL OF INDUSTRIAL ENGINEERING-APPLICATIONS AND PRACTICE	1997	4
94	TOWARDS A BETTER UNDERSTANDING ORDER ACCEPTANCE	TENKATE, HA	INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS	1994	37
95	PROCESSING REAL-TIME TRANSACTIONS IN A REPLICATED DATABASE SYSTEM	ULUSOY, O	DISTRIBUTED AND PARALLEL DATABASES	1994	2
96	Evolutionary algorithm for advanced process planning and scheduling in a multi-plant	Moon, C; Seo, Y	COMPUTERS & INDUSTRIAL ENGINEERING	2005	48
97	An integrated agent-based approach for responsive control of manufacturing resources	Lim, MK; Zhang, DZ	COMPUTERS & INDUSTRIAL ENGINEERING	2004	46
98	A flexible decision support system for steel hot rolling mill scheduling	Cowling, P	COMPUTERS & INDUSTRIAL ENGINEERING	2003	45
99	Implicit coscheduling: Coordinated scheduling with implicit information in distributed systems	Arpaci-Dusseau, AC	ACM TRANSACTIONS ON COMPUTER SYSTEMS	2001	19
100	Coordination approaches and systems - Part II: An operational perspective	Coates, G; Whitfield, RI; Duffy, AHB; et al.	RESEARCH IN ENGINEERING DESIGN-THEORY APPLICATIONS AND CONCURRENT ENGINEERING	2000	12

CHAPTER THREE

Classification of scheduling problems, mathematical models and solving methods

3.1. Introduction

Scheduling is an important issue in process operations to improve production performance. In the last twenty years there has been significant research effort regarding this area and several excellent reviews have been published recently (Mendez et al., 2006; Pinedo, 2008; Blazewicz et al., 2007).

This thesis is focusing on how to work out a resolution strategy to scheduling problems. The area is rich and wide; the concerning research is under dynamic evolution. In order to elaborate a resolution strategy it is necessary to develop a way to characterise each step of the resolution, i.e. to characterise scheduling problems, the corresponding mathematical models, and the appropriate solving methods.

Examples of classification and notation attempts to scheduling problems can be found in the literature. The need of a common notation system of scheduling problems appeared already in the seventies: Graham et al. (1979) proposed a notation system composed from three fields (machine environment, resource and task characteristics, and objective function) which has been expanded by Blazewicz et al. (1983) and later by Blazewicz et al. (2007), who took into account the resource characteristics and attributes and proposed a systematic notation to give a basis for a classification scheme. He underlines that such a notation of problem types would greatly facilitate the presentation and discussion of scheduling problems. The general notation system of Graham and Blazewicz became widely used in the scheduling community, facilitating greatly the presentation and discussion of scheduling problems. Brucker et al. (1999) dealt with resource constrained project scheduling, and based on this notation system provided a unified classification and notation scheme for project and production scheduling. The Graham – Blazewicz notation has been applied for example by Kutil et al. (2010) who developed an Optimization and Scheduling Toolbox for MATLAB. This toolbox is developed to solve different scheduling problems in the MATLAB environment, and based on the notation system it is decided whether a solution algorithm is applicable or not. The domain of activity characteristics can be widely enlarged towards process and production scheduling, especially batch scheduling problems which are frequently met in chemical engineering.

In this work we go further and propose a general and detailed classification which fits to production scheduling problems including batch production scheduling problems as well. The existing notation scheme is completed to be more precise in the problem formulation by including new kind of constraints but also extended to mathematical models and solving methods.

Firstly, in this chapter, an open classification scheme of scheduling problems and their characteristics is proposed on the basis of works of Blazewicz et al. (2007) and Mendez et al. (2006). The aspects of classification of a scheduling problem are machine environment, secondary resources, constraints (mostly functionality constraints, but also technical and environmental ones), and the objective function.

In order to develop the characterization and description of a case, a notation scheme associated to the classification of scheduling problems has been proposed. Next, this classification and notation scheme is extended to mathematical models and appropriate solving methods.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

3.2. Aspects of classification of a scheduling problem

Scheduling consists of foreseeing the sequence of every elementary operation necessary to realize fabrication orders on production units, taking into account the secondary resources (such as operators), and some external or internal constraints with respect to an objective function to optimize. Thus the main aspects of all existing classification are machines (workshop configuration), secondary resources and constraints, and objective function as presented in Figure 3-1.



Figure 3-1 – General classification scheme for scheduling problems

The proposed classification scheme is composed of the scheduling problem (machine environment, constraints, and objective function), its mathematical model and associated solution methods which are characterized through five fields: $\alpha|\beta|\gamma|\delta|\epsilon$.

The initial Graham – Blazewicz notation system is only composed of the three fields $\alpha|\beta|\gamma$ qualifying the scheduling problems. This notation system is summarized in Table 3-1. To maintain compatibility, this notation is used as a core to build up the new one but the proposed classification and notation scheme go further by completing the three initial fields with adding the fields $\delta|\epsilon$ describing the mathematical model of the scheduling problem and the solving method respectively.

Table 3-1 – Notation scheme of Blazewicz et al. (2007)

Field	Possible values	Meanings
$\alpha = \alpha_1\alpha_2$ machine environment		
α_1 type of machine	\emptyset or 1	single machine
	P	identical machines
	Q	uniform machines
	R	unrelated (independent) machines
	F	flow-shop
	J	job-shop
	O	open-shop
α_2 number of machines	\emptyset	the number of machines is assumed to be variable
	k	number of machines

$\beta = \beta_1\beta_2\beta_3\beta_4\beta_5\beta_6\beta_7\beta_8$		
task and resource characteristics		
β_1 preemption	\emptyset	no preemption
	<i>pmtn</i>	preemption is allowed
β_2 secondary resources	\emptyset	no secondary resource
	<i>res</i>	resource constraints
β_3 precedence constraints	\emptyset	independent tasks
	<i>prec</i>	general precedence constraints
	<i>tree</i>	precedence constraints forming a tree
	<i>chains</i>	precedence constraints forming a chain
	<i>network</i>	precedence constraints illustrated by a network structure
β_4 ready times	\emptyset	all ready times are zero
	r_j	ready times differ from job to job
β_5 processing times	\emptyset	arbitrary process times
	$p_j = p$	equal processing times for all tasks
	$p^* < p_j < p^*$	interval of processing times meaning that no p_j is out of the interval
β_6 deadlines	\emptyset	no deadlines are assumed
	d_k	deadlines are imposed
β_7 maximal number of tasks constituting a job (in case of job-shop system)	n	maximal number of tasks constituting a job (in case of job-shop system)
β_8 no-wait property	\emptyset	no zero-wait property
	<i>no-wait</i>	after finishing a task of a job, the next task has to be started immediately
γ objective function		
γ	C_{max}, L_{max}, \dots	the optimality criterion

In the followings the examples introduced in chapter one will be used to illustrate the different aspects of the classification.

3.2.1. Machines

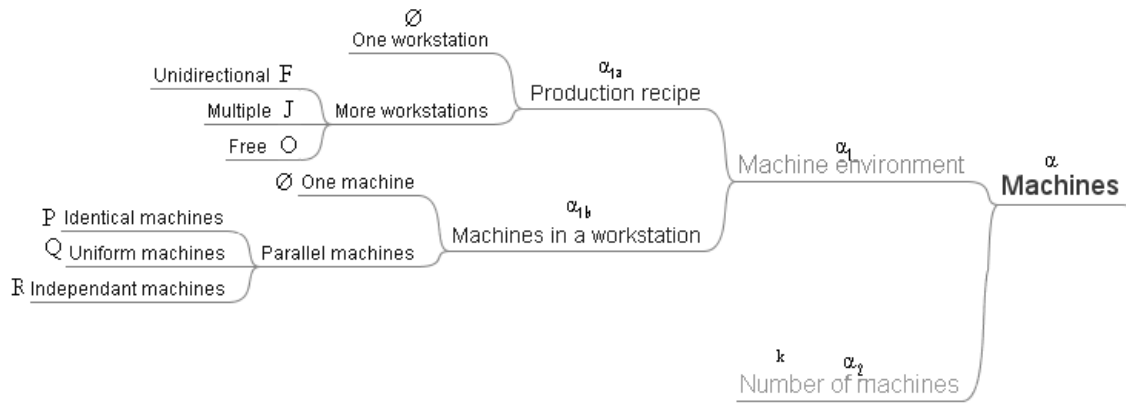


Figure 3-2 – Production recipe and workstation characterization

The first aspect of the classification deals with production recipe and workstation characterization. The main problem types are shown in Figure 3-2, denoted by the field α in the notation scheme. The possible values of α are given in Table 3-2 and in the appendix of this chapter.

Table 3-2 – Notation and possible values for α

Field	Possible values	Meanings
$\alpha_1 = \alpha_{1a}\alpha_{1b}$		
α_{1a} Production recipe	\emptyset	single workstation
	F	Unidirectional (flow-shop)
	J	Multiple direction (job-shop)
	O	Free (open-shop)
α_{1b} Machines in a workstation	\emptyset or I	single machine
	P	identical machines
	Q	uniform machines
	R	independent machines
α_2 number of machines	\emptyset	the number of machines is assumed to be variable
	k	the number of machines is equal to k (k is a positive integer)

On Table 3-2 it is shown that the field α_1 of the Graham – Blazewicz notation system has been reorganized to clearly distinguish the fields α_{1a} (corresponding to the production recipe) from field α_{1b} (corresponding to the

machines in a workstation). The reason of making this regrouping is to be able to extend the definitions of flow-shop, job-shop and open-shop to the workstations, rather than to the machines.

Scheduling problems can be classified into two main groups regarding to the number of necessary workstations to realize each product (α_{1a}). The problems where every product needs only one workstation are in the first group. The second one concerns the problems where products pass through different successive workstations. A workstation can contain one machine, or more parallel machines (α_{1b}).

The study of a **single machine** environment ($\alpha_1 = \emptyset$, $\alpha_2 = 1$) can appear as a case study, but in some cases it has practical importance: when the problems are all concentrated on the same machine (so-called bottleneck machine). Even if the factory includes more machines, this feature gives an opportunity to simplify the scheduling problem by considering it as a single machine problem, as it has been studied by Valente, 2007 who described this type of problem in detail. Note that single machine problems can be complicated if changeover costs/times and resource constraints (described later) are introduced.

In the case of **parallel machines** in one single workstation ($\alpha_{1a} = \emptyset$, $\alpha_{1b} \neq 1$), several machines are available to perform the operations simultaneously. The durations of operations can be variable on the parallel machines. If the duration time of an operation is the same for all machines designed to that operation, then the machines are called identical ($\alpha_{1b} = P$). If the duration time of an operation varies uniformly (i.e. an efficiency factor can be defined for each machine which is to be multiplied with a predefined duration time of the corresponding operation to get the real duration time of that operation on the used machine), we speak about uniform machines ($\alpha_{1b} = Q$). If the duration time of an operation varies independently from machine to machine, we speak about independent machines ($\alpha_{1b} = R$).

The next great category of problems includes the workshops with **m different workstations** ($\alpha_{1a} \neq \emptyset$). The production is divided into elementary operations each one being executed on a machine belonging to one workstation. Once the number and type of machines are described, we need to characterize the sequences. According to the recipe of the production three main classes can be distinguished: flow-shop, job-shop and open-shop (see Taillard, 1993).

In the case of **flow-shop** ($\alpha_{1a} = F$) the jobs visit the same set of workstations besides the sequence is the same for all jobs (unidirectional flow). This is the case in Example 1-1: each product is produced on the same way: heating – mixing – packaging. Flow-shops are also called multi-product batch plant in batch scheduling literature (see e.g. Mendez et al., 2006). These types of workshops are generally very productive, but poorly flexible. The production can be continuous like in refinery or discrete like in pharmaceutical processes.

In a **job-shop** ($\alpha_{1a} = J$), job goes through the workshop with respect to a predefined recipe, but, unlike the flow-shop, the sequence of operations can be different for each job (multi-directional flow). Sometimes in batch scheduling literature this type is referred as multi-purpose batch plant. This is the case in Example 1-2, where P_1 and P_2 have two different sequences. Usually these types of workshops are more flexible but less productive than flow-shops. The transports between the machines are hardly possible to automatize and the workstations may need a reconfiguration between two operations (cleaning, change of utilities...).

For flow-shop and job-shop problems the sequences are known in advance and are immutable. However, this modeling framework may be too restrictive in some contexts. When the operations of a job are usually ordered, but some sub-assemblies (or even all) can be done in any order, we speak about walk free, or **open-shop** problems ($\alpha_{1a} = O$). The recipe is not fixed a priori. Open-shop problems rarely occur in process engineering, and

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

almost never in chemical engineering, because the transformation of a material usually has to be performed in a well predefined order.

When several machines are available in workstations we talk about hybrid functionality, or **hybrid path**. In Example 1-2, the workstation of reactors is composed of two identical parallel reactors, thus it can be denoted with $\alpha_1 = JP$ (where $\alpha_{1a} = J$, $\alpha_{1b} = P$) and as there are four machines in the workshop (the heater, the two reactors and the distillation column) $\alpha_2 = 4$. Thus $\alpha = \alpha_1\alpha_2 = JP4$.

The use of the term “workstation” is important if the presence of replacement machines or secondary units is frequent (for security and productivity reasons). If the definition of a flow-shop system is applied directly to the machines and not to the workstations, then if for even one machine type a secondary machine becomes available, the system can no longer called a flow-shop (due to the existence of a machine which does not necessarily participate in the production chain). However, the important attributes of workshops (flexibility, productivity) do not change significantly due to the appearance of this secondary machine. Therefore the workstation based interpretation fits better to process industrial productions.

3.2.2. Resources and constraints



Figure 3-3 – Resources and constraints

The second aspect of the classification deals with resource and constraint characteristics. Main classes of the classification scheme can be seen in Figure 3-3.

In the notation system of Blazewicz et al. (1983), the resource and constraint considerations are not separated from each other: the field β_2 denotes the secondary resource characteristics. Consequently, instead of introducing three fields for the three branches – as it has been done in the case of the field α – the corresponding fields of the original notation have been used.

3.2.2.1. Functionality constraints

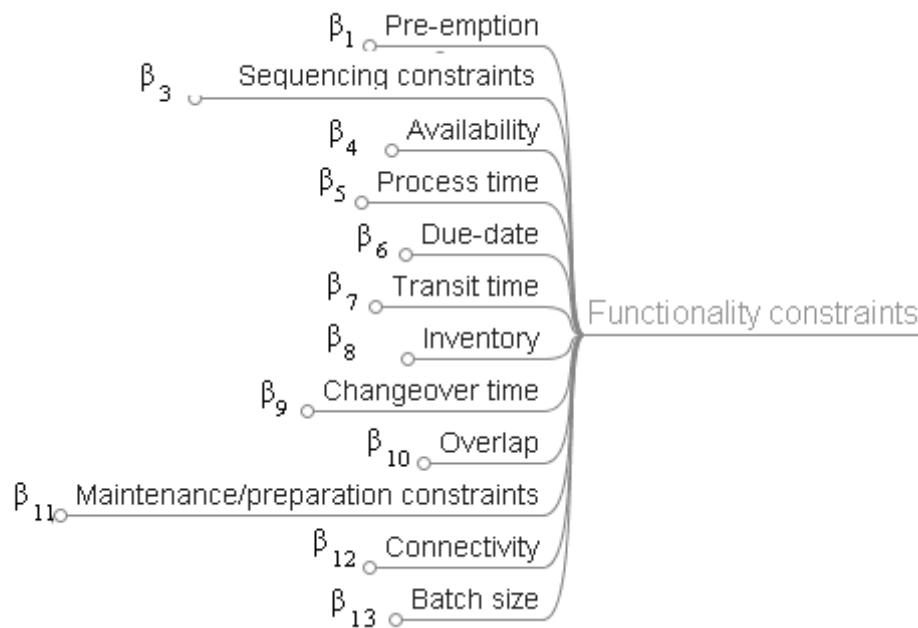


Figure 3-4 – Functionality constraints

The main types of functionality constraints and the corresponding notation can be seen on Figure 3-4.



Figure 3-5 – Preemption (β_1)

Preemption (β_1) means that a task can be stopped and resumed before end (Figure 3-5). Preemption of tasks is usually possible in manufacturing processes, and impracticable in process engineering due to flux, sub-products or cleaning problem, for example. The notation field is the same than the Blazewicz's notation.

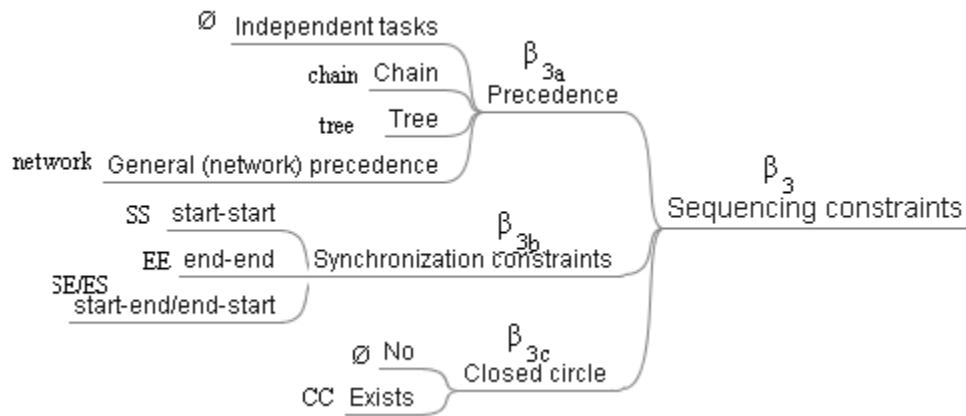


Figure 3-6 – Sequencing constraints (β_3)

On the occasion of production recipe, some restrictions about operation relations have been already mentioned. In a flow-shop or a job-shop system the sequence of operations belonging to the same job are fixed. **Sequencing constraints** can be extended with precedence constraints, synchronization constraints, and closed cycle property (Figure 3-6).

Operation i precedes operation j (or operation j succeeds operation i) if and only if operation A is started and completely performed before operation B (see Figure 3-7).

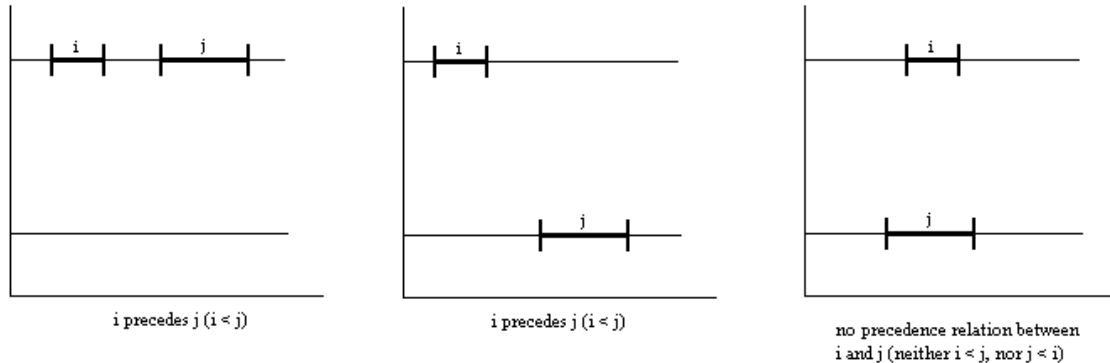


Figure 3-7 – Precedence relations (β_{3a})

The main types of precedence relations are: chain precedence ($i < j < k < \dots$), tree precedence ($i < k, j < k, \dots$), network precedence (or general precedence).

Two operations without a precedence constraint are not necessarily independent. Starting or ending times can be related as it is shown in Figure 3-8.

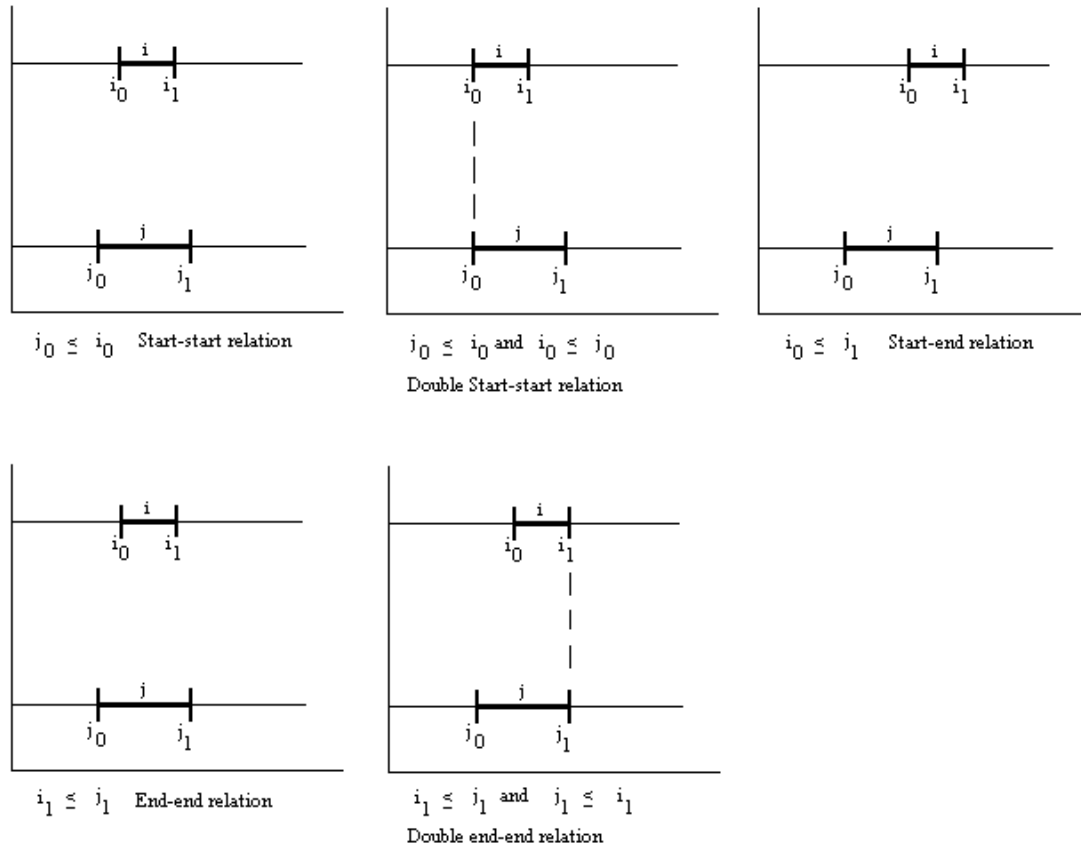


Figure 3-8 – Starting and ending time relations (β_{3b})

As it is illustrated in Figure 3-8, the equalities are expressed by two inequalities. A double end-end relation can be noticed in Example 1-1: heating A and heating B operations have to be finished in the same time to avoid maintaining liquid in hot form.

Closed circle (β_{3c}) means that a product can visit the same machine several times. In the metallurgical industry for example, a piece may locally follow a pass through an oven to soften the metal (annealing), be formed by a press, and once again through the oven to harden the metal (quenching). In Example 1-2 the intermediate IntAB from operation Separation has to be recycled.

In Blazewicz's notation β_3 refers to precedence constraints but does not contain synchronization constraints and closed circle (β_{3b} and β_{3c}) added in the present work. This information can have important effect during the choice of mathematical model and solution method and may need specific treatment as it has been illustrated by Czuczai et al. (2009).

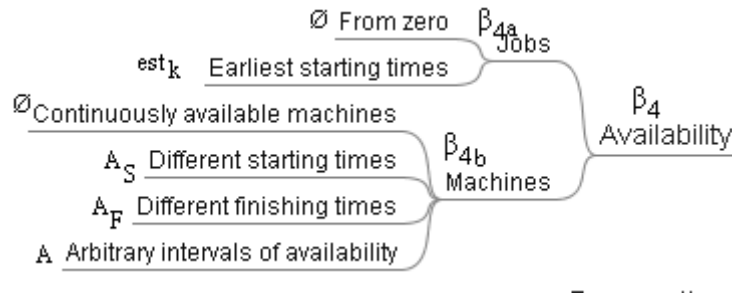


Figure 3-9 – Availability constraints

Availability constraints (β_4) can be interpreted both for jobs and machines as shown on Figure 3-9. An availability constraint with respect to a job (β_{4a}) or a machine (β_{4b}) means that the correspondent job (or machine) is only available in a part of the entire time horizon. The earliest possible starting time of a job may differ from zero for several reasons, such as waiting for a raw material, or intermediate coming from another factory. This is the case for the raw material B_1 and B_2 in Example 1-1.

A machine can also be unavailable during a certain time, e.g. it needs an installation time (starting time differs from zero), or a shutdown time (finishing time differs from the length of time horizon), or both (intervals of availability).

The field β_4 of the Graham – Blazewicz notation corresponds to the field β_{4a} of the present work. However, as it has been shown by Saïdy and Taghavi-Fard, 2008 the availability intervals of machines are frequently limited which justifies the introduction of the field β_{4b} .

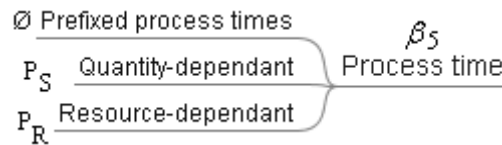


Figure 3-10 – Process times

Process times (β_5) (also called duration of an operation) can be prefixed (or invariable), quantity-dependant or can depend on a secondary resource used (Figure 3-10).

The process time of a heating operation for example depends on the quantity of the material to heat ($\beta_5 = P_S$) and also on the quantity of hot vapor used ($\beta_5 = P_R$).

In the case of more parallel machines in a workstation, the process times can also be unit-dependent ($\alpha_{1b} = Q$ and R).

The Graham – Blazewicz notation does not take into account quantity- or resource-dependence. However, this aspect becomes important during the choice of a mathematical model to the scheduling problem. Some models are hard to be adapted to quantity or resource dependence (see chapter four). As the present work makes efforts to extend the notation system to the mathematical models and the solving methods too, this aspect has to be considered.

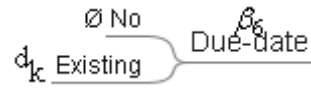


Figure 3-11 – Due dates

β_6 refers to deadlines (due dates). The notation field (Figure 3-11) is the same as in the Graham – Blazewicz system.



Figure 3-12 – Transit time

The duration of transport of material between two machines is called **transit time** or **transportation time** (β_7). We distinguish not considered, invariable, or way-dependent transit time (Figure 3-12). Transit time is called invariable if it is constant from any machine to any other. Transit time is called way-dependent if it depends on the corresponding machines.

The Graham – Blazewicz notation does not deal with transit times, as in most cases the transit time can be ignored. However it is an important aspect with respect to the modeling (Behnamian et al., 2011) therefore transit times have been included to the notation system.

The field β_7 of the notation Blazewicz et al. (1983) describes the maximal number of tasks constituting a job in case of job-shop systems. This aspect is specific to job-shop problems; however, in this work a general notation scheme is proposed, therefore this aspect is decided to be neglected.

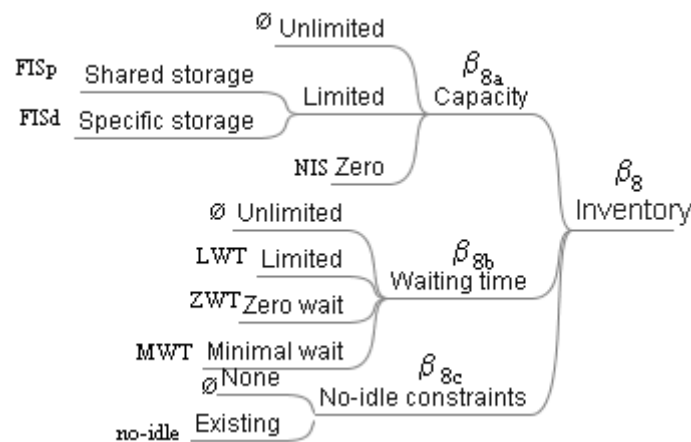


Figure 3-13 – Inventory considerations

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

Due to the presence of intermediates and products, **inventory** considerations (β_8) have to be taken into account in process engineering (Figure 3-13). There are three types of inventory considerations, but two of them are the most significant: *capacity* (β_{8a}) and *time* (β_{8b}): the capacity of intermediate storage can be limited ($\beta_{8a} \neq \emptyset$) without restrictions on time ($\beta_{8b} = \emptyset$) and waiting time can be limited ($\beta_{8b} \neq \emptyset$) without capacity constraints ($\beta_{8a} = \emptyset$). As the inventory can be forbidden or permitted on the machine itself, two cases are to be distinguished: the storage in the processing machine and the storage by releasing the machine. In the first case the machine is occupied by the product material, thus the machine is immobilized. In the second case the machine is released, and the material is stored in storage tanks. On capacity restrictions we mean only the second case, as storage in the machine is treated with waiting time constraints. The inventory *capacity* may be unlimited, limited or zero. The intermediate storage can be shared or specific. Shared storage means that the same storage tank is used by all materials. This is the case of a warehouse for example. In the case of specific storage there are specific storing units for each material, with their own capacities. This is the case when different materials need to be stored separately. In Example 1-2 an unlimited storage is applied for raw materials, but finite storage for intermediates. Other interesting examples can be found for process scheduling with finite intermediate storage (Ku and Karimi, 1988), and with no intermediate storage (Suhami and Mah, 1981).

The second category of constraints deals with the *time* on the inventory. In the industry the waiting time between the consecutive operations of a material is often restricted, for example in food industry in the case of a sensible product, or in metallurgy in case of liquid metal for energetic reasons. The superior limit of the waiting time can be infinite (unlimited waiting time), a positive number (limited waiting time), or zero (zero wait problems). The inferior limit can be zero, or a positive number (minimal waiting time) Minimal waiting time exists in Example 1-1 between mixing and packaging tasks. We recall the attention that the waiting time above refers to the time between consecutive operations of a job.

Finally, *no-idle constraints* means that a machine which has started to work cannot be interrupted until it finishes all its operations (Goncharov and Sevastyanov, 2009) due to the high operational costs of a machine. Consider, for example, a unit which needs a preparation procedure requiring a lot of energy like a reactor that has to be heated to a high working temperature.

In the Graham – Blazewicz notation the field β_9 describes a no-wait property which can be \emptyset or *no-wait* with respect to the presence of at least one zero-wait restriction in the problem. However, especially in pharmaceutical processes or food industrial problems, the consumption times of raw materials, intermediates or products usually induce important limits on storage. As this work is focusing mainly on chemical engineering problems, this important aspect has to be represented in the applied notation.

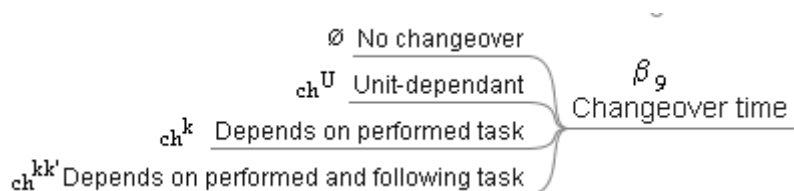


Figure 3-14 – Changeover time

A waiting time between consecutive operations on the same shared machine is called **changeover time** (β_9). The changeover time (Figure 3-14) can depend on the unit, on the performed task, or both on the performed and on the following task (sequence dependency). For example the unit has to be cooled before refill due to security reasons. In Example 1-1 the restriction 4 (after being used, each reactor has to be cleaned before its next operation and the cleaning time depends both on the former and the current operation) is a changeover constraint of type ch^{kk} .

The field does not exist in the general notation of Graham and Blazewicz, but it becomes important in the case of chemical engineering problems. In chemical industry units are frequently used in a multipurpose way, and not taking into account the necessary time for changeover could lead to undesired interactions between the different materials. That aspect is very important for example in delicate processes, like pharmaceutical industry. Let us consider for example a fermentor where an antibiotic A was produced earlier by bacteria X. If now we desire to produce antibiotic B by bacteria Y, the fermentor has to be sterilized before the new production process; otherwise the residual quantity of antibiotic A may fertilize bacteria Y who is resistant only for antibiotic B.

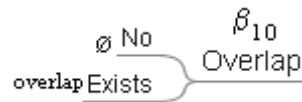


Figure 3-15 – Overlap

Sometimes certain subsets of operations are in connection with one another, and must be conducted according to a special plan (e.g. operation B must start when operation A is performed in 90%). This relation type is similar to the synchronization constraints (β_{3b}) but the restriction is interpreted to an intermediate time of the operation rather to the starting or ending time. This phenomenon is called **overlap** (β_{10}). Overlap is illustrated in Figure 3-16, deducted with a little modification from Figure 3-8:

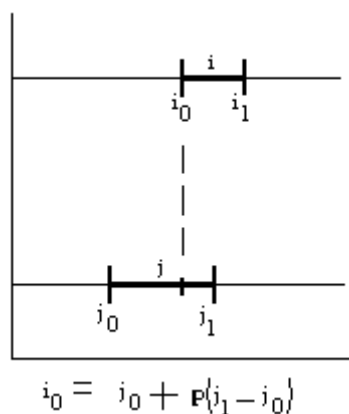


Figure 3-16 – Overlapping operations

A special case of overlap is when the two overlapping operations belong to the same job sequence – which means that the following operation of the job has to be started before (and not after) the end of its previous operation.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

The field β_{10} does not exist in the Graham – Blazewicz notation. However, in chemical engineering it is an important aspect: let us consider for example a distillation process (denoted with operation j) and a cooling process (denoted with operation i). On Figure 3-16 it is shown that operation i has to be started exactly at an intermediate time point of operation j . The reason for this is that operation j produces an intermediate B which has to be cooled immediately. Suppose that the intermediate B appears half hour after the starting time of the operation j . Now, if the duration time of the operation j is two hours, then the cooling operation has to be started exactly at the 25% of the realization of operation j , thus $i_0 = j_0 + 0.25(j_1 - j_0)$.



Figure 3-17 – Maintenance and preparation constraints

The next field refers to **maintenance/preparation constraints** (β_{11}). Maintenance and preparation constraints are related to the machine. We distinguish process, unit and time dependant maintenance.

Both the process dependant and the unit dependant maintenance/preparation mean that the machine has to be prepared before use and reuse. But, in the process dependant case the necessary time of maintenance depends on the type of operation to be performed, while in the unit dependant case the necessary time of maintenance depends on the machine. In the case of parallel machines in a workstation ($\alpha_{ib} \neq \emptyset$), the necessary maintenance time can be different to each unit. Typical examples from chemical engineering: preheating a reactor before adding the reactant, preheating a gas-chromatograph before injecting the sample. Notice the difference between changeover time (β_9) and maintenance/preparation time (β_{11}): in the former case the waiting is caused by the previous operation, while in the latter case the reason is either the next operation or the machine itself. Time dependant maintenance means that the machine has to be stopped and maintained regularly, independently of the performed operations.

Notice, that the different maintenance/preparation cases can occur simultaneously, thus the field β_{11} may take more values, for example if both time and process dependant maintenance are required then (concatenating M_P and M_T) $\beta_{11} = M_{PT}$.

The field β_{11} does not exist in the Graham – Blazewicz general notation. However, this work is focusing mainly on process and chemical engineering problems and the attribute encoded with the field β_{11} is important in process engineering domain, especially in chemical industry.

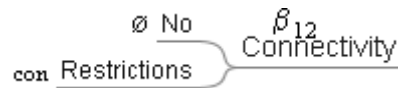


Figure 3-18 – Connectivity restrictions

Forbidden connections between machines are called “machine precedence” or “**connectivity restrictions**” (β_{12} on Figure 3-18). In example 1-1, several connectivity restrictions are represented by the dashed lines. For example, Reactor 1 cannot be connected to Packager 2 due to incompatibility.

The field β_{12} does not exist in the general notation of Graham and Blazewicz. However, as it is shown by Czuczai et al. (2009), it is frequent in several process engineering problems. In order to treat this attribute in modeling phase, special constraints have to be introduced which complicate the solution process.

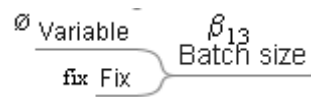


Figure 3-19 – Batch size restrictions

Finally, regarding to **batch size** (β_{13}), the production can be performed in lots, or the batch size can be variable. A product which has to be packed and the size of package is fixed is an example to the former one (see Example 1-3).

The field β_{13} is specific to batch scheduling problems (appearing frequently in process and chemical engineering) therefore it does not exist in the general notation of Graham and Blazewicz.

The whole corresponding notation is presented in the appendix of this chapter.

3.2.2.2. Secondary resources (β_2)

In order to remain coherent with the Graham – Blazewicz notation the classification aspect of secondary resources is represented by the field β_2 (Figure 3-20 and Table 3-3). However, based on the classification of this work, the set of functionality constraints are discussed before the secondary resources. That is why β_2 appears between β_{13} and β_{14} .

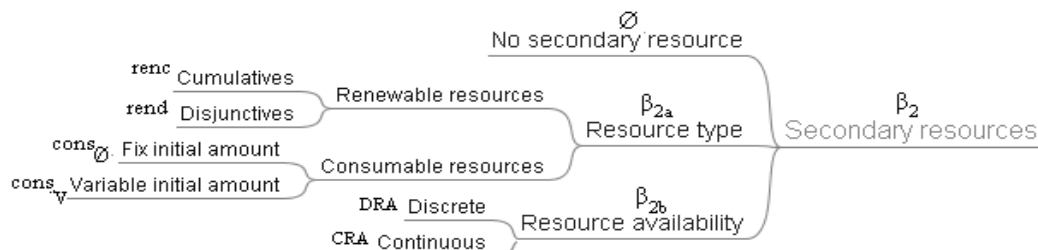


Figure 3-20 – Secondary resources

Table 3-3 – Secondary resources notation

Field	Possible values	Meanings
β_2 secondary resources	\emptyset	no secondary resource
	$\beta_{2a}\beta_{2b}$	there are specified resource constraints
β_{2a} resource type	<i>renc</i>	renewable, cumulative resource
	<i>rend</i>	renewable, disjunctive resource
	<i>cons₀</i>	consumable resource with fix initial amount
	<i>cons_v</i>	consumable resource with variable initial amount
β_{2b} resource availability	<i>DRA</i>	resource available in discrete time points (delivery)
	<i>CRA</i>	resource (with its limits) is available continuously

Secondary resources – if considered – can be **renewable**, or **consumable**.

A resource is **renewable** if, after having been used by one or more operations, it is available again in the same quantity (men, machines, space, equipments...) limited by its capacity. Renewable resources can be cumulative or disjunctive. **Cumulative resources** can be used by several tasks simultaneously (e.g. team workers, several machines in a workstation, etc.); nevertheless the total quantity of this resource is limited. **Disjunctive resources** can perform just one task at once (e.g. machine tool, robot manipulator).

A typical example where renewable secondary resources are met is the employee timetabling. In their work Artigues et al. (2009) treated the case when the problem of machine scheduling and employees timetabling are to be solved simultaneously. They further and called the attention to the fact that this case, although it appears often in the practice, needs further research investigation.

A resource is **consumable** if being used once the resource cannot be used again. The global consumption of a consumable resource is limited over the time (e.g. raw materials, energy, or budget). A consumable resource is thus doubly constrained: its actual availability and its global consumption are both limited.

A resource can be continuously available (such as heating or electricity), or available in discrete time points (delivery).

Both Example 1-1 and Example 1-2 need raw materials to perform the operations, thus consumable resources are met.

In the Graham – Blazewicz notation parameter β_2 refers to the resource constraints that have been introduced by Blazewicz et al. (1983). The presence of resources is indicated by a value \emptyset or $res\lambda\sigma\rho$ where λ stands to the number of resources, σ denotes the capacity of resources (all resource capacities are considered to be constant and equal to σ), and ρ denotes the requirement of resources (all resource requirements have a constant upper bound equal to ρ). They precise that many types of resource constraints are not represented by this classification, however, varying the values λ , σ and ρ , this notation still generates most of the relevant and previously studied

types of problems. On the other hand, if it is needed for a specific domain there is no limit to enlarge and enrich the proposed notation scheme – so did the present work.

3.2.2.3. Other constraints (β_{14})

The technical constraints are very miscellaneous and include for example: quality, standard or other technical problems.

Security and environmental (so-called societal) constraints can be observed (e.g. policy restriction on CO₂ emission, legal working hours etc.). Societal considerations are often implicitly taken in account. However, it is possible that in special situations they have to be treated explicitly.

Sometimes, when the objective function does not consider economical requirements, the economical aspects are taken into account by constraints. This is the case for example with the objectives like “energy minimization”. (Of course the energy needed is minimal if there is no production at all. However, this case would not respect economical considerations.) Restriction on the costs of used resources or other economic considerations are also possible.

The field β_{14} does not exist in the Graham – Blazewicz notation system. This domain is complex and not very well studied yet. In future research, this field could be examined more deeply and in a more sophisticated way. In this work, only the existence is noted in order to be able to store this property for the case-based reasoning system, i.e. the possible values of β_{14} are 1 or \emptyset whether it is considered or not, respectively.

3.2.3. Objective function

Scheduling of a production plan is realized according to one or more objectives. So, it is possible to classify scheduling problems with respect to the objective function. The notation scheme related to the objective functions to be optimized is detailed in this part, denoted by the field γ .

There are several types of objective functions, such as time-based, resource-based, cost-based, income-based, environmental ones and multi-objectives.

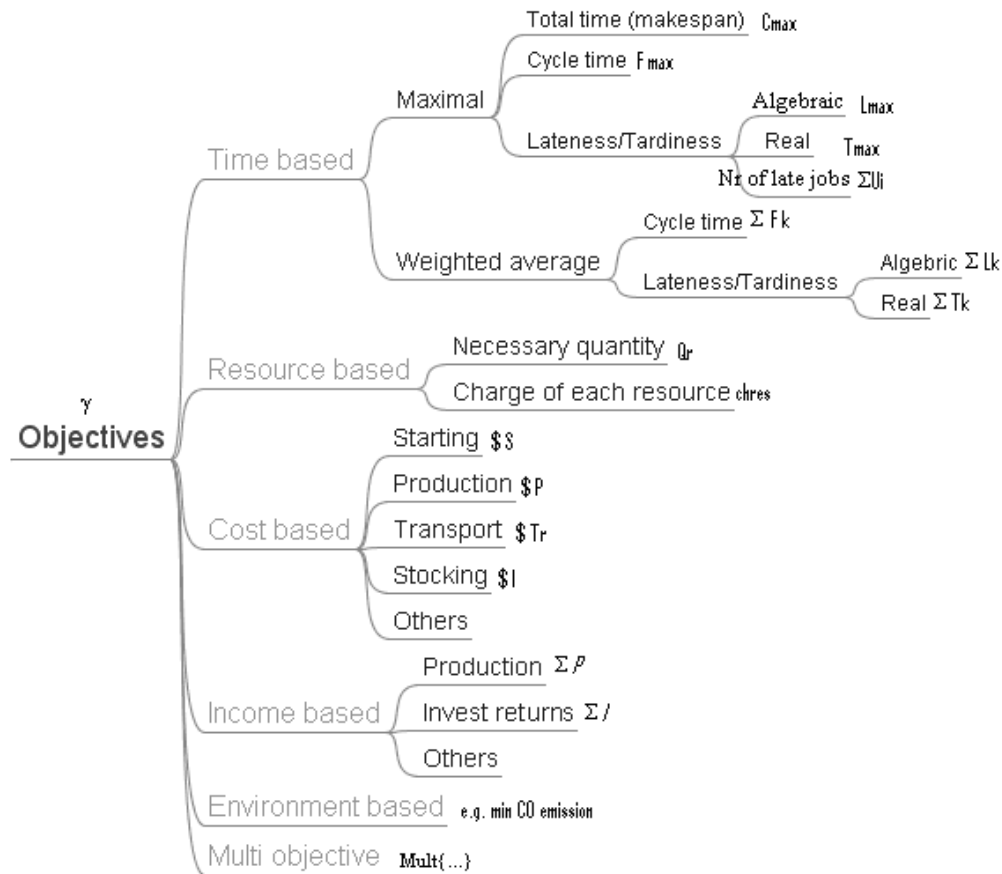


Figure 3-21 – Objective functions (γ)

Figure 3-21 and the notation table in Appendix show the different objective function classes. The corresponding classification field has been denoted with γ as in the Graham – Blazewicz notation. The time-based approach is typical for production scheduling problems. The most frequent one is the makespan minimization ($\gamma = C_{\max}$). The time-based objective functions based on lateness and tardiness are also widespread. Lateness of a job is calculated as the difference between the finishing time and the deadline of that job. Thus lateness can have a negative value, provided that a job is realized before its deadline. Tardiness (also called real lateness) on the other hand can only take positive value or zero, and defined by the formula: $\max\{\text{lateness}, 0\}$. The objective functions based on lateness/tardiness are either the maximal lateness/tardiness or a weighted average of lateness and tardiness values. Special objective function types can be found too, e.g. for minimizing deviation from a common due-date (Gowrishankar et al., 2001).

The cost-, resource- or income-based approaches are typical in the area of process scheduling, and batch scheduling problems, and usually the most frequent one is the profit maximization. The reason of this phenomenon is that in multipurpose and multiproduct batch plants, and in process scheduling in general, there are material balances, prices of materials. Consequently the quantity can easily represent a market price of the product. Mendez et al. (2006) classify these objective functions in their review.

In the literature, less frequent objectives are the environmental-based ones (Dessouky et al., 2003). Security considerations rarely appear in the objective function; usually they are described (explicitly or implicitly) in the

constraints. Finally, we have to mention here the multi-objective approach, e.g. bi-criteria flow-shop (Pan et al., 2009a).

This field already exists in Blazewicz's notation; however, the different objective functions are not classified.

3.2.4. Application to illustrative examples

In the previous paragraphs the fields $\alpha|\beta|\gamma$ of the notation system have been detailed which correspond to the scheduling problem. Before to go further with the description of the fields of mathematical model (δ) and solving method (ϵ) the application of the classification and notation system to scheduling problems is illustrated.

The classification scheme of Mendez et al., 2006 is applicable to the example problems Example 1-1, Example 1-2 and Example 1-3. By the new classification system however, we get some supplementary information: in the case of Example 1-1 there are end – end relations, earliest starting times, minimal waiting times and the resources are consumable. In the case of Example 1-2 the resources are consumable and there is a closed cycle in the system. The special characteristics of Example 1-3 can better be illustrated by the new classification system. The new point of view of workstations, as a unit regarding to the process topology and thus unifying flow-shop/job-shop/open-shop property with parallel machines property is missing; however, it is an important aspect to consider in modelling.

Using the Graham – Blazewicz notation system, Example 1-1 would be noted as a $\{F10|res, chain, r_k, p_j, d_k, no-wait|C_{max}\}$ problem. By the new system it is noted as a $\{FR10|cons, CRA, chain, EE, est_k, P_s, d_k, mwt, zw, ch^{kk'}, con, fix|C_{max}\}$ problem. The supplementary information encoded in the new notation is: the presence of independent machines in a workstation, the type of the secondary resource, the presence of end – end relations, minimal waiting time, connectivity restrictions, the size-dependency of process times, the presence of changeover time and fix batch sizes. These properties are important in order to choose an adequate mathematical model to the problem. E.g.: the property $\beta_5 = P_s$ makes the problem difficult to be treated with discrete time representation ($\delta_1 = DTR$, see in next paragraph) models. The properties $\beta_9 = ch^{kk'}$ and $\beta_{13} = fix$ need special considerations.

Example 1-2 would be noted as $\{J4|res, network, p_j, 4, no-wait|\Sigma P\}$ with the Graham – Blazewicz notation, and as $\{JP4|cons, CRA, network, CC, P_s, FIS_d, ZW|\Sigma P\}$ by the new notation system. The supplementary information encoded in the new notation is: the machines in a workstation are identical, there is closed circle in the system, the process times are size dependent, and a finite, specific intermediate storage policy is applied.

Example 1-3 would be noted as $\{F6|res, chain, d_k |C_{max}\}$ by using the notation system of Graham and Blazewicz. The supplementary information encoded in the new $\{F6|cons, CRA, chain, P_s, d_k, FIS_d, LW, overlap, con, fix|C_{max}\}$ notation is: the limited storage and waiting time properties, the overlapping attribute, the connectivity restrictions and the batch size constraints.

3.3. Mathematical models

Mendez et al., 2006 classified the mathematical models, frequently used to represent scheduling problems. However, no notation system has been found to mathematical models. To construct a decision-support system to find the adequate model and solving method to a scheduling problem it seems to be necessary to propose a notation scheme for mathematical models and solving methods as well. The application of the notation system facilitates significantly the association of models to problems, and to gather and store important information about the compatibility of problem types and model types. Thus, the notation scheme has been extended to mathematical models and solving methods.

Several effective mathematical formulations were developed to model process scheduling problems. Depending on the modeling options, there are various ways for formulating the same problem. These model characteristics influence directly the computational performances, the capabilities, the strengths and weaknesses of an optimization model.

Properties of mathematical model will be described by a field δ , and can be seen in Figure 3-22 and in the notation table of appendix.

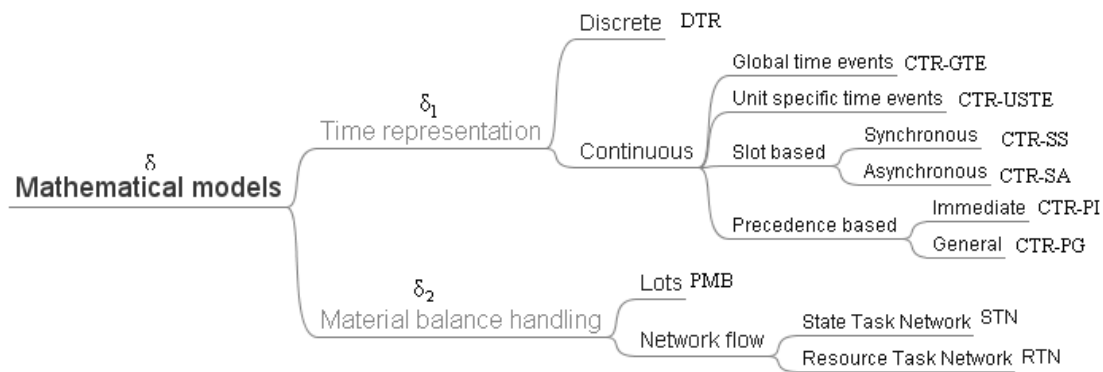


Figure 3-22 – Mathematical models

3.3.1. Time representation

Based on time representation, all existing formulations can be classified into two main categories: discrete time representation and continuous time representation. A detailed analysis of the different time representation options was performed by Mendez et al. (2006) and Pan et al. (2009b).

In discrete time representation (δ_1 =DTR), the time horizon is divided into a finite number of time intervals with predefined and equal duration. With this approach, the operations are enforced to begin and to finish exactly at intervals boundaries. Its main advantage is that the constraints are formulated at predefined grid points which reduces model complexity and simplifies the model resolution. But the number of time intervals strongly influences the size of the mathematical models, computational efficiency and the accuracy or quality of solution. Due to the time horizon division, the scheduling problem is described by the model only approximately. This results in a tradeoff between accuracy, requiring a small time interval to achieve suitable approximation, and computational effort, requiring a reasonable time interval to reduce the size of the combinatorial problem.

Despite of these limitations especially for complex industrial application (difficulty to handle variable processing times or some constraints like changeovers, inventory), this time representation can be convenient in some applications.

In order to overcome the aforementioned inherent limitations of the discrete time representation, there has been significant research on continuous time representation (δ_1 =CTR). In these approaches, variables are required to define precisely the occurrence time of events (potentially at any point in the time horizon). Consequently, the number of variables is reduced, resulting in smaller-sized models and greater flexibility. On the other hand, the possibility to have events that can take place at any time, leads to more complex structure of mathematical models (more complicated constraints definition). Based on events representations, these approaches are classified into four categories: slot based (Sundaramoorthy and Karimi, 2005), precedence based (Pan et al., 2008), global event based (Castro et al., 2001, Maravelias and Grossmann, 2003) and unit specific event based models (Ierapetritou and Floudas, 1998, Giannelos and Georgiadis, 2002). The first two are more appropriated for sequential processes, the last two for general network processes.

The concept of time slots stands for a set of predefined time intervals with unknown duration. The main idea is to assign enough time slots for each machine in order to allocate them to operations to be performed. The number of time slots is a crucial parameter for computational performances and optimality. When time slots are identical for all the machines, shared resources are easier to handle, and we speak about synchronous representation (δ_1 =CTR-SS). On the other hand, in asynchronous representation (δ_1 =CTR-AS) time slots are different from one machine to another providing more flexibility.

The representation of batch precedence enforces the sequential use of shared resources explicitly thanks to model variables and constraints. On the same machine, the immediate predecessor of an operation (δ_1 =CTR-PI) can be considered or the whole set of operations processed before for the case of general precedence (δ_1 =CTR-PG). Recent mentionable efforts have been made in the area of precedence-based models by Qian et al. (2009) who attempted to eliminate an important disadvantage of these models: the difficulties with inventory and resource limitations

Like in the slot based representation, the global event based (δ_1 =CTR-GTE) one uses shared time slots by all operations and machines. Continuous variables are introduced to determine the timing of time slots and binary variables are specified to assign beginning and ending time of operations to the time slots. The last approach, i.e. unit specific event based models (δ_1 =CTR-USTE), assigns specific event points for each machine. The event locations are different for one machine to another. Consequently different operations can start at different time machine.

3.3.2. Material balance handling

For sequential processes, the models assume that the size of each batch is known before scheduling. Therefore it is not necessary to consider mass balances explicitly. The first step consists in decomposing the quantity of the product required into individual batches. Once the batches created, the scheduling problem (δ_2 =PMB) is solved by allocating the resources to batches over the time horizon.

In general network processes the material balances are required to be established explicitly. To represent a problem's structure two different approaches exist in the literature (Schilling and Pantelides, 1996): State-Task

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

Network (δ_2 =STN) and Research-Task Network (δ_2 =RTN). Both approaches were created in order to deal with complex product recipes.

For the former one Kondili et al. (1993) proposed a general framework composed on two types of nodes: task nodes denoted by rectangles and representing process operations, state nodes denoted by circles representing raw materials, intermediate products and final products. These nodes are linked by arcs that indicate the flow of materials and the task precedence. The number beside each arc gives the fraction of a batch transferred to the next operation.

Schilling and Pantelides (1996) had extended the STN to the RTN framework where processing equipments, storage, material transfer and utilities are described as resources in a unified way. Indeed, in addition to states, circles represent also other resources and ellipses the equipment. The two approaches are illustrated in Figure 3-23 (represented here in order to facilitate the reading) which is an STN representation of example 1-2, and in Figure 3-24 which is an RTN representation of example 1-2.

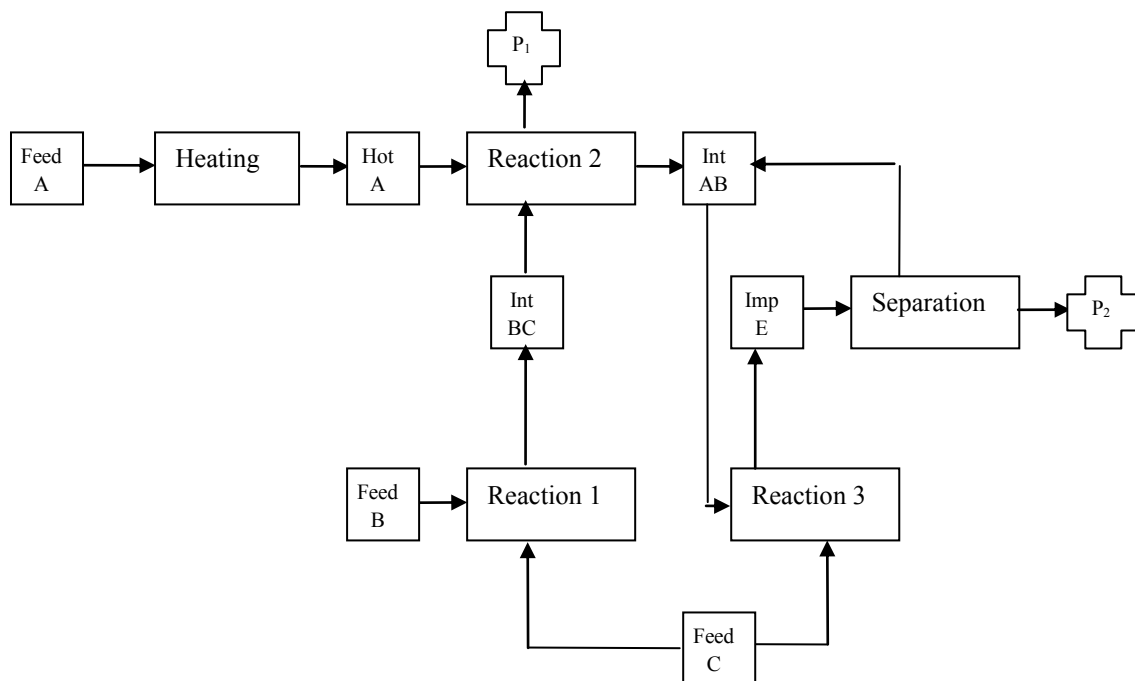


Figure 3-23 – State Task Network of example 1-2 (from Kondili et al., 1993)

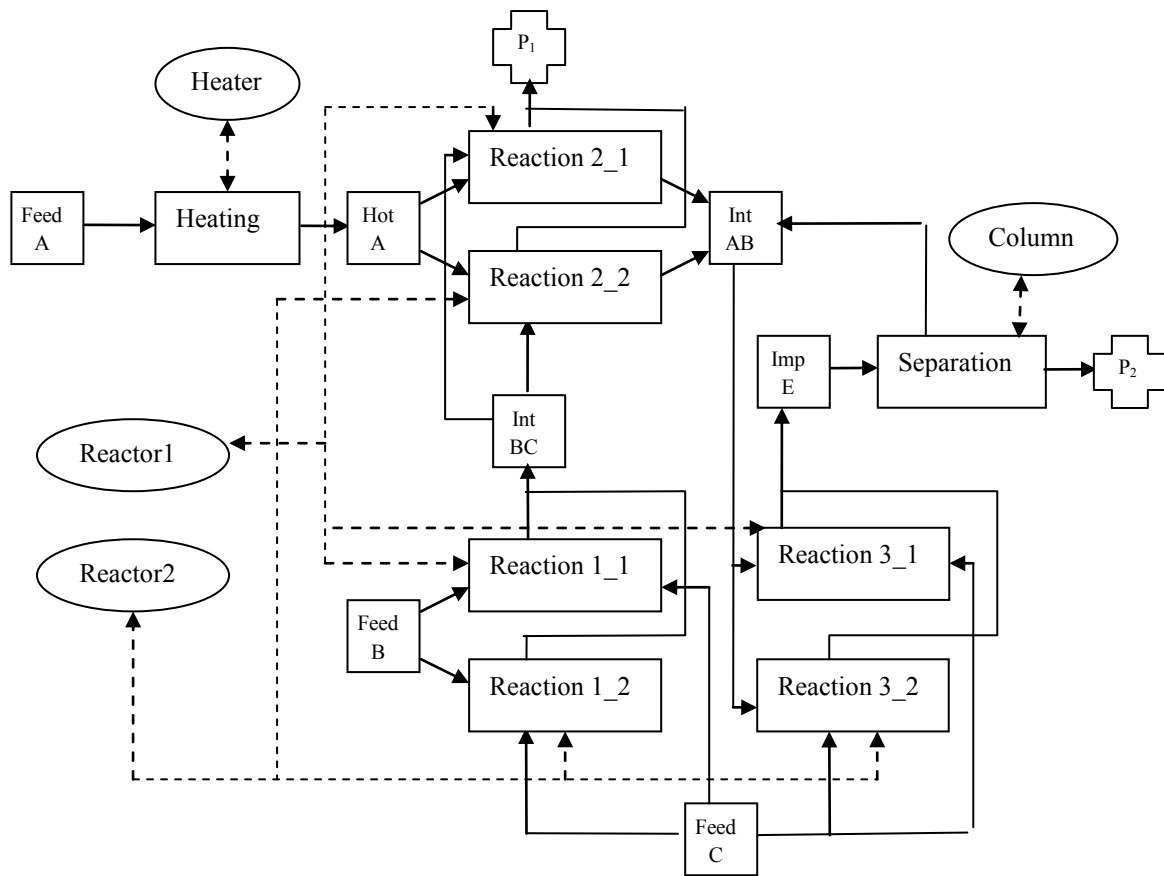


Figure 3-24 – Resource Task Network of example 1-2

3.4. Solving methods

The resolution methods of scheduling problems have been discussed in section 1.4. Among the three branches of solving methods the research is decided to be oriented to mathematical approaches as they seemed to be the most promising ones to cover the widest possible domain of scheduling problems. Therefore in this paragraph the mathematical approaches will be detailed.

Properties of solution methods are denoted with a field ε , shown in Figure 3-24 and in the notation table of appendix.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

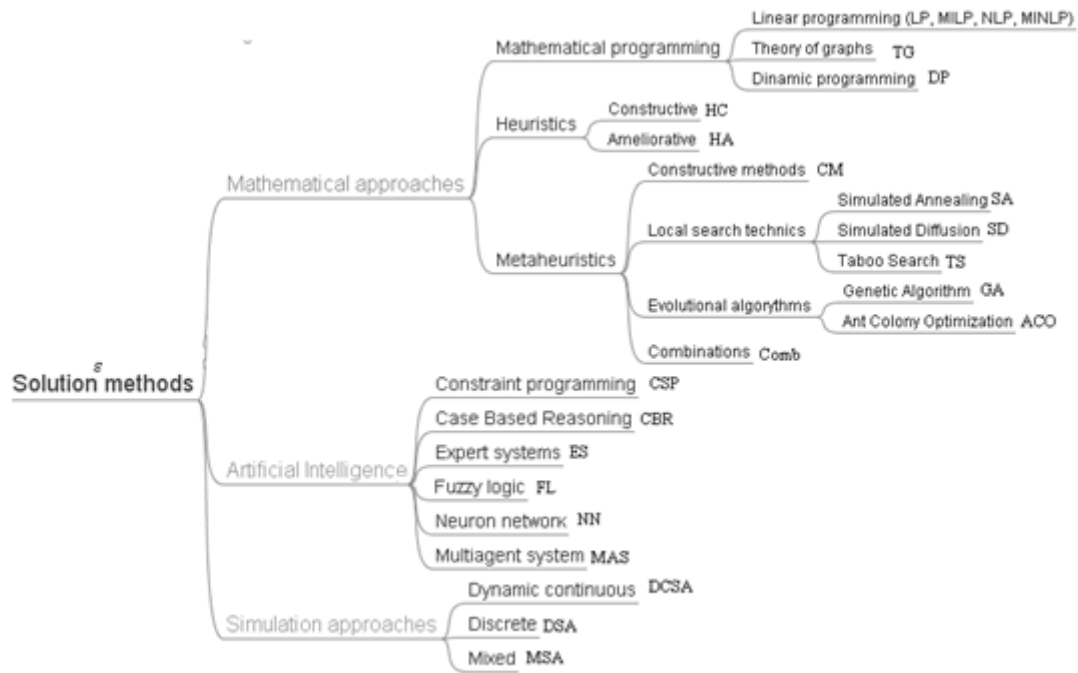


Figure 3-25 – Classification scheme of solution methods to scheduling problems

Most of the mathematical approaches are based on operational research techniques. They can be classified into three categories: mathematical programming, heuristics and meta-heuristics. Scheduling problems generally lead to mixed integer linear programming problems.

As all solving methods mathematical approaches can be categorized into two groups: exact (if the optimality of the found solution is mathematically guaranteed) and approximate methods (if a feasible solution is to be found in reasonable time). Excellent books and reviews deal with the area of exact methods, e.g. Biegler et al. (1997). A common disadvantage of exact methods is that large-scale problems cannot be treated efficiently without making simplifier assumptions.

The majority of simplifier assumptions are based on the strategies of decomposition. A problem can be decomposed with respect to three attributes: time, machines and constraints (corresponding to operations or resources). For example, the so-called Rolling Horizon technique (Bassett et al., 1996) is decomposition by time: the main large-scale problem is decomposed to a set of consecutive small scale problems, and these problems are to be solved after each other in a part of the time horizon.

Despite of the continuing evolution of computational resources, it exists always a critical limit (depending on the computational environment, the applied mathematical model, and other parameters) beyond which it becomes prohibitive to use exact methods. Facing these difficulties, some specialists have oriented their research towards **Heuristics Methods**. They use technical experiences to propose rapidly feasible solutions. Heuristic methods can be constructive ($\epsilon = HC$) or ameliorative ($\epsilon = HA$). In the case of constructive methods, the solution is constructed step by step with respect to heuristic rules. On the other hand, with applying ameliorative methods, the first step is to create an initial solution as a skeleton in order to ameliorate it step by step. Due to their large number, it would be impossible to give an exhaustive list of all the heuristic methods. A famous example, in production scheduling area (permutation flow-shop), is the so-called NEH heuristic (Kalczynski and

Kamburowski, 2007), but there are several other effective constructive (Gupta et al., 2002, Rad et al., 2009), and ameliorative (Efsthathiou, 1996) heuristic methods as well. Heuristic methods can also be classified with respect to their main rules whether they are based on due-date or process-time property. For example Earliest Due Date (EDD) is a due date based heuristic, while Shortest Process Time first (SPT) is a process time based one. The main disadvantage of these methods is that they are often specific to a given problem (this point explains their large number).

Compared to heuristic methods, **Meta-Heuristic** ones conducted a more exhaustive exploration of the solution space, to ensure that the solution is not a local minimum. **Meta-Heuristic Methods** (Xhafa and Abraham, 2008) are usually inspired by an analogy of the nature. The analogy can be a physical one (ϵ = SA, simulated annealing, ϵ = SD, simulated diffusion), a biological one (ϵ = GA, genetic algorithms, ϵ = TS, taboo search), or from the area of ethology (ϵ = ACO, ant colony optimization). Constructive methods, local search techniques (ϵ = SA, ϵ = TS), evolutionary algorithms (ϵ = GA, ϵ = ACO) and combined methods (ϵ = Comb) can be distinguished (Widmer et al., 2001). A common disadvantage of these methods is the presence of tuning parameters. These parameters have a significant effect to the quality of solution, and to the computational time, and thus they need to be well chosen, according to the actual problem in order to work efficiently. In other words, meta-heuristic methods need to be adapted to the concrete problem type. Meta-heuristic methods usually need one or more solution in order to ameliorate them. Consequently, meta-heuristics can be used to ameliorate the solutions obtained from heuristic methods, for example.

3.5. Conclusion

In this chapter an open classification and notation scheme of scheduling problems, mathematical models and appropriate solving methods has been proposed. This classification and notation scheme has been developed in the frame of constructing a decision-aid system on the foundations of Case-Based Reasoning approach in order to find the most appropriate resolution strategy to different scheduling problems.

The proposed classification system is based on the works of Blazewicz et al. (2007) and Mendez et al. (2006). The notation scheme of Graham et al. (1979) and Blazewicz et al. (1983) is composed of three fields $\alpha|\beta|\gamma$ which correspond to machine environment, resource and constraint characteristics and objective function respectively. The field α has been reformulated from a workstation-based point of view. New aspects of classification have been added to the field β : transit time, changeover time, overlap, maintenance and preparation constraints, connectivity restrictions, batch size considerations, resource types and availability, synchronization constraints, presence of closed cycle, machine availability, quantity and resource dependency of process times, capacity constraints, limited and minimal waiting time, no-idle constraints. A classification of objective functions has been introduced in the field γ .

The notation system has also been extended to mathematical models of scheduling problems by a field δ , which follows the classification scheme of Mendez.

A new field ϵ has been added corresponding to the appropriate solving methods.

The application of the new classification and notation system has been shown on three illustrative examples. The integrated classification and notation system facilitates the study of the different resolution strategies of scheduling problems.

3.6. References of Chapter Three

Artigues, C.; Gendreau, M.; Rousseau, L-M.; Vergnaud A. Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch and bound. *Computers & Operations Research* **2009**, *36*, 2330.

Bassett, M. H.; Pekny, J. F.; Reklaitis, G. V. Decomposition techniques for the solution of large-scale scheduling problems. *AIChE Journal* **1996**, *42*, 3373.

Behnamian, J.; Fatemi Ghomi, S. M. T.; Jolai, F.; Amistaheri, O. Realistic two-stage flow-shop batch scheduling problems with transportation capacity and times. *Applied Mathematical Modelling* **2012**, *36*(2) 723. (available online 22 July 2011)

Biegler, L. T.; Grossmann, I. E.; Westerberg, A. W. *Systematic methods of chemical process design*; Prentice Hall, International Series in the Physical and Chemical Engineering Science, 1997.

Blazewicz, J.; Lenstra J. K.; Rinnooy Kan, A. H. G. Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics* **1983**, *5*, 11.

Blazewicz, J.; Ecker, H.; Pesch, E.; Schmidt, G.; Weglarz, J. *Handbook on Scheduling From Theory to Applications*; Springer-Verlag, 2007.

Brucker, P.; Drexel, A.; Möhring, R.; Neumann, K.; Pesch, E. Resource constrained project scheduling: Notation, classification, models and methods. *European Journal of Operational Research* **1999**, *112*, 3.

Castro, P.; Barbosa-Povoa, A. P. F. D.; Matos, H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. *Industrial and Engineering Chemistry Research* **2001**, *40*, 2059.

Czuczai, B.; Farkas, T.; Rev, E.; Lelkes, Z. New Mixed Integer Linear Programming Model for Solving Scheduling Problems with Special Characteristics. *Industrial Engineering and Chemistry Research* **2009**, *48*, 5321.

Dessouky, M.; Rahimi, M.; Weidner, M. Jointly optimizing cost, service and environmental performance in demand-responsive transit scheduling. *Transportation Research Part D: Transport and Environment* **2003**, *8*, 433.

Efstathiou, J. Any time heuristic schedule repair in manufacturing industry *IEEE Proceedings control theory application* **1996**, *143* (2), 114.

Esquirol, P.; Lopez, P. *L'ordonnancement*; Economica, 1999.

Giannelos, N. F.; Georgiadis, M. C. A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes. *Industrial and Engineering Chemistry Research* **2002**, *41*, 2178.

Goncharov, Y.; Sevastyanov, S. The flow-shop problem with no-idle constraints: A review and approximation. *European Journal of Operational Research* **2009**, *196*, 450.

Gowrishankar, K.; Rajendran, Z.; Srinivasan, G. Flow-shop scheduling algorithms for minimizing the completion time variance and the sum of squares of completion time deviations from a common due date. *European Journal of Operational Research* **2001**, *132*, 643.

Grabot, B. Ordonnancement d'ateliers manufacturiers. *Techniques de l'ingénieur, AG3015* **2006**.

Graham, R. L.; Lawler, E. L.; Lenstra, J. K.; Rinnooy Kan, A. H. G. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics* **1979**, *5*, 287.

Gupta, J. N. D.; Krüger, K.; Lauff, V.; Werner, F.; Sotskov, Y. N. Heuristics for hybrid flow-shops with controllable processing times and assignable due dates. *Computers & Operations Research* **2002**, *29*, 1417.

Ierapetritou, M. G.; Floudas, C.A. Effective continuous-time formulation for short-term scheduling 1. Multipurpose batch process. *Industrial and Engineering Chemistry Research* **1998**, *37*(11), 4341.

Kalczynski, P. J.; Kamburowski, J. On the NEH heuristic for minimizing the makespan in permutation flow-shops. *Omega* **2007**, *35*, 53.

Kondili, E.; Pantelides; C. C.; Sargent, R. W. H. A general algorithm for short-term scheduling of batch operations. Part I: MILP formulation. *Computers & Chemical Engineering* **1993**, *17*, 2211.

Ku, H-M.; Karimi, I. Scheduling in serial multiproduct batch processes with finite intermediate storage: a mixed integer nonlinear programming formulation. *Industrial Engineering and Chemistry Research* **1988**, *27*, 1840.

Kutil, M.; Sucha, P.; Capek, R.; Hanzalek, Z. Optimization and Scheduling Toolbox. In: *Matlab – Modelling, Programming and Simulations*; Sciyo, 2010.

Maravelias, C. T.; Grossmann, I. E. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Industrial and Engineering Chemistry Research* **2003**, *42*, 3056.

Mendez, C. A.; Cerdá, J.; Grossmann, I. E.; Harjunkoski, I.; Fahl, M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering* **2006**, *30*, 913.

Pan, M.; Qian, Y.; Li, X. A novel precedence-based and heuristic approach for short-term scheduling of multipurpose batch plants. *Chemical Engineering Science* **2008**, *63*, 4313.

Pan, Q.K.; Wang, L.; Qian, B. A novel differential evolutionary algorithm for bi-criteria no-wait flow-shop. *Computers & Operations Research* **2009**, *36*, 2498.

Pan, M.; Li, X.; Qian, Y. Continuous-time approaches for short-term scheduling of network batch processes: Small-scale and medium-scale problems. *Chemical Engineering Research and Design* **2009**, *87*, 1037.

Pinedo, M. Scheduling: Theory, algorithms and systems; Prentice Hall, 2008.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

Qian, Y.; Pan, M.; Huang, Y. Modeling and optimization for scheduling of chemical batch processes. *Chinese Journal of Chemical Engineering* **2009**, *17*(1), 1.

Rad, S. F.; Ruiz, R.; Boroojerdian, N. New high performing heuristics for minimizing makespan in permutation flow-shops. *Omega* **2009**, *37*, 331.

Saidy, H. R. D.; Taghavi-Fard, M. T. Study of scheduling problems with machine availability constraints. *Journal of Industrial and Systems Engineering* **2008**, *1*(4), 360.

Schilling, G.; Pantelides, C. C. A simple continuous-time process scheduling formulation and a novel solution algorithm. *Computers and Chemical Engineering* **1996**, *20*, S1221.

Suhami, I.; Mah, R. S. H. An implicit enumeration scheme for the flow-shop problem with no intermediate storage. *Computers & Chemical Engineering* **1981**, *5*, 83.

Sundaramoorthy, A.; Karimi, I. A. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chemical Engineering Science* **2005**, *60*, 2679.

Taillard, E. Benchmarks for basic scheduling problems. *European Journal of Operational Research* **1993**, *64*, 278.

Valente, J. M. S. Dispatching heuristics for the single machine early/tardy scheduling problem with job independent penalties. *Computers & Industrial Engineering* **2007**, *52*, 434.

Widmer M.; Hertz A.; Costa D.; Les métaheuristiques, chapter 3, pp 55-93, Hermes Edition, 2001.

Xhafa, F.; Abraham, A. Metaheuristics for scheduling in industrial and manufacturing applications; Springer 2008.

3.7. Appendix

Appendix Table 1 – Notation and possible values for α

Field	Possible values	Meanings
$\alpha_1 = \alpha_{1a}\alpha_{1b}$		
α_{1a} Production recipe	\emptyset	single workstation
	F	flow-shop system
	J	job-shop system
	O	open-shop system
α_{1b} Machines in a workstation	\emptyset or I	single machine
	P	identical machines
	Q	uniform machines

	R	independent machines
α_2 number of machines	\emptyset	the number of machines is assumed to be variable
	k	the number of machines is equal to k (k is a positive integer)

Appendix Table 2 – Notation and possible values for β

Field	Possible values	Meanings
β_1 possibility of preemption	\emptyset	no preemption allowed
	$pmtn$	preemption is allowed
β_2 additional resources	\emptyset	no additional resource
	$\beta_2 = \beta_{2a}\beta_{2b}$	there are specified resource constraints
β_{2a} resource type	$renc$	renewable, cumulative resource
	$rend$	renewable, disjunctive resource
	$cons_{\emptyset}$	consumable resource with fix initial amount
	$cons_V$	consumable resource with variable initial amount
β_{2b} resource availability	DRA	resource available in discrete time points (delivery)
	CRA	resource (with its limits) is available continuously
β_3 sequencing constraints	$\beta_3 = \beta_{3a}\beta_{3b}\beta_{3c}$	
β_{3a} precedence	\emptyset	independent tasks
	$chain$	precedence constraints forming a chain
	$tree$	precedence constraints forming a tree
	$network$	precedence constraints illustrated by a network structure
	$prec$	general precedence constraints
β_{3b} synchronization constraints	\emptyset	no connection
	SS	start-start relations
	EE	end-end relations

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

	SE/ES	start-end or end-start relations
β_{3c} closed circle	\emptyset	no closed circle appears
	CC	a unit has to be visited more than one times
β_4 availability constraints	$\beta_4 = \beta_{4a}\beta_{4b}$	
β_{4a} earliest starting times of jobs	\emptyset	all earliest starting times are zero
	est_k	earliest starting times differ from job to job
β_{4b} machine availability	\emptyset	all machines are continuously available
	A_S	machines have different start times of availability
	A_F	machines have different finish times of availability
	A	machines have arbitrary intervals of availability
β_5 process times	\emptyset	tasks have arbitrary, invariable processing times
	P_S	processing time depends on size (quantity in unit)
	P_R	task processing time depends on other resource (utility) used by the unit
β_6 due dates	\emptyset	no deadlines are assumed
	d_k	deadlines are imposed on the performance of a job set
β_7 transit time	\emptyset	no transit time considered
	T	transit time is invariable
	$T_{it'}$	transit time is way-dependent (unit to unit)
β_8 inventory considerations	$\beta_8 = \beta_{8a}\beta_{8b}\beta_{8c}$	
β_{8a} capacity	\emptyset	buffers of unlimited capacity are assumed
	FIS_d	finite intermediate storage is assumed with specific storage
	FIS_p	finite intermediate storage is assumed with shared storage

	<i>NIS</i>	there is no intermediate storage
β_{8b} waiting time	\emptyset	unlimited waiting time is permitted
	<i>LWT</i>	there is a limited waiting time
	<i>ZWT</i>	no waiting time, zero-wait property
	<i>MWT</i>	there is a minimal waiting time, which must be fulfilled before the next task of the job (but the unit is released)
β_{8c} no-idle constraints	\emptyset	no such constraint
	<i>no-idle</i>	there are no-idle constraints
β_9 changeover time	\emptyset	no need to wait
	ch^U	changeover time depends on unit
	ch_k	changeover time depends on performed task
	$ch_{kk'}$	changeover time depends on performed and following task (we have a changeover time matrix)
β_{10} possibility of overlap	\emptyset	no overlap
	<i>overlap</i>	overlap can occur
β_{11} maintenance and preparation/cleaning constraints	\emptyset	no such constraint is taken into account
	M_P	process dependent
	M_U	unit dependent
	M_T	time dependent
β_{12} connection between machines	\emptyset	free connectivity
	<i>con</i>	machines have connection restrictions represented by a connection graph
β_{13} batch size restrictions	\emptyset	variable batch size, there is no restriction
	<i>fix</i>	fix batch size, production is realized in lots
β_{14} non-functionality constraints	\emptyset	no other than functionality constraint
	<i>technics</i>	technical constraints
	<i>societals</i>	societal constraints

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

	<i>economics</i>	economic considerations
	<i>life-cycle</i>	life-cycle constraints

Appendix Table 3 – Notation and possible values for γ

Field	Possible values	Direction	Meanings
γ objective function	C_{max}	min	makespan (maximal completing time, schedule length)
	F_{max}	min	cycle time
	L_{max}	min	maximal lateness
	T_{max}	min	maximal tardiness
	$\sum U_i$	min	number of late jobs
	$\sum F_k$	min	average cycle time
	$\sum L_k$	min	average lateness
	$\sum T_k$	min	average tardiness
	Q_r	min	necessary quantity of resource
	Ch_r	min	charge of resource
	$\$S$	min	starting costs
	$\$P$	min	Production costs
	$\$T$	min	transport costs
	$\$I$	min	inventory costs
	$\sum P$	max	total profit
	$\sum I$	max	investment retour
	$minCO$	min	CO emission

Appendix Table 4 – Notation and possible values for δ

Field	Possible values	Meanings
δ_1 time representation	<i>DTR</i>	discrete time representation
	<i>CTR-GTE</i>	time events based formulation with global time events
	<i>CTR-USTE</i>	time events based formulation with unit-specific time events
	<i>CTR-SS</i>	slot based formulation with synchronous time slots
	<i>CTR-SA</i>	slot based formulation with asynchronous time slots
	<i>CTR-PI</i>	precedence based formulation with immediate precedence
	<i>CTR-PG</i>	precedence based formulation with general precedence
δ_2 material balance handling	<i>PMB</i>	material balances are not considered explicitly
	<i>STN</i>	State Task Network
	<i>RTN</i>	Resource Task Network

Appendix Table 5 – Notation and possible values for ε

Field	Possible values	Meanings
ε solution methods	<i>LP, NLP, MILP, MINLP</i>	methods of mathematical programming
	<i>TG</i>	theory of graphs
	<i>DP</i>	dynamic programming
	<i>HC</i>	constructive heuristic
	<i>HA</i>	ameliorative heuristic
	<i>CM</i>	constructive meta-heuristic
	<i>SA</i>	simulated annealing
	<i>SD</i>	simulated diffusion
	<i>TS</i>	taboo search
	<i>GA</i>	genetic algorithm
	<i>ACO</i>	ant colony optimization
	<i>Comb</i>	combined meta-heuristics
	<i>CSP</i>	constraint programming
	<i>CBR</i>	case based reasoning
	<i>ES</i>	expert system
	<i>FL</i>	fuzzy logic
	<i>NN</i>	neuron network
	<i>MAS</i>	multi-agent system
	<i>DCSA</i>	dynamic continuous simulation approach
	<i>DSA</i>	discrete simulation approach
	<i>MSA</i>	mixed simulation approach

CHAPTER FOUR

Association rules between flow-shop scheduling problems, corresponding mathematical models and appropriate solving methods

4.1. Introduction

To solve a scheduling problem denoted by $\alpha|\beta|\gamma$, a mathematical model (δ) and a solution method (ε) have to be proposed. Thus, in the following an $\alpha|\beta|\gamma|\delta|\varepsilon$ group is referred as a **solution strategy**. From the detailed description of these notation fields in chapter three it follows that the number of the possible problem-model-method triplet combinations is very high. Therefore the analysis in this chapter is limited to flow-shop problems ($\alpha = \text{Fn}$).

Referring to Figure 1-7, it can be noted that the modeling phase, the choice of the numeric method and the resolution cannot be separated from each other completely. Each model has its limits and cannot fit every type of problem. For example a $\beta_2 \neq \emptyset$ problem needs to be treated with taking into account the resource constraints, therefore in the most of the cases a $\delta_2 = \text{PMB}$ model does not suit.

Firstly in this chapter the domain of association between scheduling problem, mathematical model and solution method will be discussed. Then, some association rules will be proposed, based on computational experiments. Finally, a preliminary decision support system will be presented inspired by a Case-Based Reasoning approach.

4.2. Adaptation of mathematical models

First, some definitions will be presented in order to facilitate the reading and understanding.

The **adaptation** of a mathematical model to a scheduling problem is the modification of a general mathematical model – by changing the objective function, modifying or introducing variables and constraints – in order to take into account the special characteristics of the problem to be modelled. Adaptation techniques of mathematical models can be divided into the following categories:

- changing the objective function,
- introducing new continuous variables,
- modifying or introducing new constraints on continuous variables,
- introducing new binary variables,
- modifying or introducing new constraints on binary variables.

An adaptation of the model which does not concern its objective function or results in the introduction of new binary variables or binary constraints is called **minor modification**. Otherwise we speak about major modification.

The illustration of resolution strategies is presented in two parts. First, the adaptation of the same mathematical model with minor modifications will be presented on the disjunctive graph model ($\delta = \text{CTR-PG}$, PMB, illustrated in Figure 4-1 from Esquirol and Lopez, 1999), and the effects of these minor modifications will be shown. In the second part, some literature mathematical model will be applied to three example problems.

4.2.1. Adaptation of a mathematical model by minor modifications

In order to illustrate the adaptation by minor modifications, the following properties have been introduced to the flow-shop example $\text{F10}|\text{chain}|C_{\max}$ of the chapter one (Example 1-4):

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

1. earliest starting times ($\beta_4 = \text{est}_k$),
2. minimal waiting times ($\beta_8 = \text{MWT}$),
3. due-dates ($\beta_6 = d_k$),
4. job-dependent changeovers ($\beta_9 = \text{ch}_k$),
5. job-to-job-dependent changeovers ($\beta_9 = \text{ch}_{kk'}$),
6. and unit-dependent changeovers ($\beta_9 = \text{ch}^U$) independently.

The meanings of these terms were explained in chapter three.

In each case ten flow-shop problems of 10 jobs and 10 machines have been generated using the algorithm proposed by Taillard (1993), i.e. the durations are random numbers with uniform distribution between 1 and 100. Each job consists of 10 operations, which have to be performed in a predefined order, each one on its corresponding machine. Thus each job has to follow the same route from the first machine to the tenth respectively as it was illustrated in Figure 1-10.

4.2.1.1. The mathematical model

The applied mathematical model (disjunctive graph model) is based on the representation of the scheduling problem with a graph, where the vertices correspond to the operations, and the edges correspond to the priorities between these operations. These priorities can be predefined (conjunctive edges, denoted with unidirectional arrows) or not (disjunctive edges, denoted with bidirectional arrows). Usually the subset of conjunctive edges (U) represents the precedence constraints for consecutive operations of a job, and the subset of disjunctive edges (D) is associated to the conflicts of using a non-shareable resource (i.e. a machine). Figure 4-1 illustrates the scheme of the disjunctive graph model on a problem which contains three jobs (J1, J2, and J3) and two machines (M1 and M2).

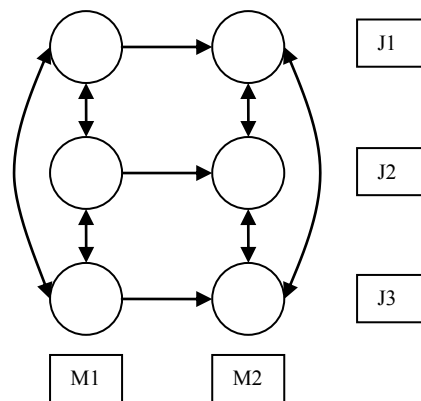


Figure 4-1 – Illustration scheme of the disjunctive graph model

The circles on Figure 4-1 denote the unit operations. The lines J1, J2 and J3 correspond to the jobs and the columns M1 and M2 correspond to the operations of a job and – in the same time – to the adequate machines. J1, J2 and J3 have to be performed firstly by M1, then by M2. Thus an implicit assumption has just been made (which is also an important restriction of the applicability of the disjunctive graph model): each operation has its one machine and only one on which it can be performed. Let us denote with J the number of jobs, and A the

number of machines. The graph is thus composed of $J \cdot A$ vertices, $J \cdot (A-1)$ conjunctive edges and $\frac{1}{2} \cdot A \cdot J \cdot (J-1)$ disjunctive edges.

A mathematical representation of the disjunctive graph model with makespan objective ($\gamma = C_{\max}$) has been proposed by Esquirol and Lopez (1999). This mathematical model is a MILP problem, due to the presence of binary variables $x_{kp,k'}$ (Equation 4-1):

$$\begin{aligned}
 & \min C_{\max} \\
 (1) \quad & C_{\max} - Ts_{k,p} \geq T_{k,p} \quad \forall kp \in X \\
 (2) \quad & Ts_{k,p'} - Ts_{k,p} \geq T_{k,p} \quad \forall \langle p, k'p' \rangle \in U \\
 (3) \quad & \left. \begin{aligned} Ts_{k',p} - Ts_{k,p} &\geq T_{k,p} + M(x_{kp,k'} - 1) \\ Ts_{k,p} - Ts_{k',p} &\geq T_{k',p} - Mx_{kp,k'} \end{aligned} \right\} \quad \forall \langle p, k'p' \rangle \in D \\
 (4) \quad & x_{kp,k'} \in \{0,1\} \quad \forall \langle p, k'p' \rangle \in D \\
 (5) \quad & Ts_{k,p} \geq 0 \quad \forall kp \in X
 \end{aligned}$$

Equation 4-1 – Mathematical representation of disjunctive graph model with makespan objective

A nomenclature of this mathematical representation is presented in Table 4-1.

Table 4-1 – Nomenclature of the disjunctive graph model

A	Number of machines
C_{\max}	Total makespan (finishing time of the last operation)
D	Set of disjunctive edges
$G(X, U, D)$	Graph of vertices X , conjunctive edges U and disjunctive edges D
J	Number of jobs
k, k'	Index of jobs
M	Parameter “big-M”, a sufficiently large positive number
p	Index of operations
$Ts_{k,p}$	Starting time of operation p of job k
$T_{k,p}$	Duration time of operation p of job k
U	Set of conjunctive edges
$x_{k,p,k'}$	Binary variable determining if job k precedes job k' on the machine performing operation p
X	Set of vertices

The constraint (1) defines that the makespan value cannot be inferior to the finishing time of any operation. The finishing times are not represented explicitly in the model, but they are expressed as the sum of starting times ($Ts_{k,p}$) and duration times ($T_{k,p}$). Because of the minimization the objective function value reach its possible minimum: the finishing time of the last operation.

The constraint (2) expresses the precedence between two consecutive operations of the same job represented by a conjunctive edge from kp to $k'p'$.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

The pair of constraints (3) interpreted to disjunctive edges from k_p to k'_p corresponds to priority of operations on the unit performing the operation p . Only one constraint is active from the pair (3) at a time. If k precedes k' then the first constraint is active, if k' precedes k then the second one. The role of the parameter M , which is a sufficiently large number (also known as big- M in the practice of optimization) is to neutralize the inactive constraint of the pair (3).

The decision variable x_{kp} is forced to be binary by constraint (4).

Finally, constraint (5) forces the continuous variable Ts_{kp} to be nonnegative.

The number of variables and the number of constraints are presented in Table 4-2:

Table 4-2 – Number of variables and constraints for Equation 4-1

Number of variables	$1+J \cdot A+J \cdot (J-1) \cdot A = J^2 A+1$
Number of constraints	$J \cdot A+J \cdot (A-1)+A \cdot J \cdot (J-1)+J \cdot (J-1) \cdot A+J \cdot A = J \cdot (2A \cdot J+A-1)$

4.2.1.2. The applied solving methods

In order to solve the problems heuristic and meta-heuristic methods have been applied. Heuristic methods are working with operation order rules therefore they do not require a mathematical model. However, in order to evaluate the solution corresponding to the proposed operation orders (i.e. to gain the makespan value and the starting and finishing time of operations) it is a convenient way to solve a mathematical model with prefixed decision variables. That is, a LP problem is solved, which does not need important computational time.

The problems have been solved with $\varepsilon = \text{HC-SPT}$ and $\varepsilon = \text{HC-LPT}$ heuristics. These heuristics are based on the operation duration times. In the case of $\varepsilon = \text{HC-SPT}$ (shortest process time first) the priority order of jobs on each machine prefers the operations with shorter durations. E.g. if the operation 3 of job 7 takes less time to perform than operation 3 of job 5, then job 7 precedes job 5 on the machine 3. In the case of $\varepsilon = \text{HC-LPT}$ it is the contrary, the operations with longer durations are preferred, thus the order will be the opposite.

Then the problems have been solved with $\varepsilon = \text{ACO}$, $\varepsilon = \text{TS}$, $\varepsilon = \text{SA}$ and $\varepsilon = \text{GA}$ meta-heuristics.

Ant-colony optimization ($\varepsilon = \text{ACO}$) is inspired by an ethological analogy. The principal idea of this method is based on the behavior of ants, helping each other to find food by emitting an essence (pheromone). As the path to the food is followed by more and more ants, the pheromone track becomes more and more intensive. To mimic this behavior, the application of the $\varepsilon = \text{ACO}$ method requires to define a pheromone model describing the different paths leading to the solution (Xhafa and Abraham, 2008). Several successful applications to scheduling problems have been published, e.g. Rajendran and Ziegler (2004) and Shyu et al. (2004). In this work the pheromone values are associated to the preference of each job on a machine, thus a path followed by the ants corresponds to the choices of the disjunctive arcs (Figure 4-2).

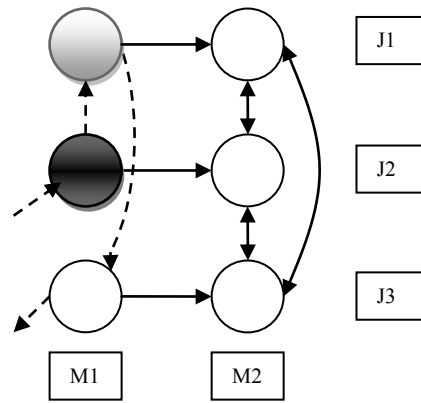


Figure 4-2 – Illustration scheme of the disjunctive graph model

In Figure 4-2 the process is illustrated on machine 1. The intensity of the pheromone track is illustrated with the intensity of black colour in the vertices and the path followed by the ants is shown with the dashed arrows ($J2 \rightarrow J1 \rightarrow J3$). Pheromone values are updated after each new solution. Two update rules are applied: the evaporation of the pheromone and the increase or decrease of the pheromone value depending on the gained improvement.

Taboo Search ($\epsilon = TS$) is inspired by mechanism of human memory enabling to avoid previously made mistakes (Xhafa and Abraham, 2008). Firstly, mutated solutions are generated from an initial feasible solution. Then taboo search choose the best one (with respect to the gained objective function value). To prevent cycling and encourage greater movement through the solution space, a taboo list is maintained of solutions already faced during the last iterations. It is forbidden to choose a solution presented on the taboo list. This list is the “short-term memory” of the algorithm. A “long-term memory” is also constructed, by generating more than one initial solution to begin with. If after several iterations the solution is not improving, then a new initial solution is chosen to restart with. To scheduling problems taboo search is most frequently applied as local search procedure within more effective methods, e.g. Huang and Liao (2008). In this work the mutated solutions are generated with applying exchanges in the priority order of jobs on a machine.

Simulated Annealing ($\epsilon = SA$) is inspired by the freezing of a thermodynamic system (Xhafa and Abraham, 2008). In analogy with the physical process the objective function to minimize is referred as the energy of the system (E). The temperature (T) of the system is also introduced as a fictive parameter. Starting with an initial solution and an initial temperature the solution is mutated. If the new solution provides a lower objective function value than the previous one, then the new solution is accepted, else it is accepted with a probability proportional to $\exp(-\Delta E/T)$. In scheduling domain there are examples both for the application of simulated annealing alone (Satake et al., 1999) both for the application combining with other methods (Nearchou, 2004). In this work the mutation of a solution is generated with applying exchanges in the priority order of jobs on a machine.

Genetic Algorithm ($\epsilon = GA$) is inspired by the biological evolution (Xhafa and Abraham, 2004). In analogy with the biological process, the method starts with an initial population. Each individual of the population has a

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

fitness value which corresponds to its quality (e.g. the objective function in case of maximization or its reciprocal). The algorithm constructs a new generation of the population maintaining the number of individuals. The method attempts to increase the fitness value of the individuals applying two procedures: selection and reproduction. The selection ensures that the individuals with higher fitness value are preferred to reproduce, and the reproduction creates new entities by recombination. In scheduling area, similarly to Taboo Search, Genetic Algorithm is usually combined with other methods (Sadeegheigh et al., 2006). In this work the priority order of jobs on a machine of the “child” is calculated based on the priority order of jobs on a machine of the “parents”. That is, if job k precedes job k' on machine m in both parent schedule, then this precedence is maintained in the child schedule too. Suppose for example that the order of jobs on machine m is [1 5 2 3 4] for parent 1 and [5 4 1 3 2] for parent 2. Then, the preference value of job k is defined as the sum of the orders of job k on machine m in the parent schedules. Thus, for job 1 the preference value is $1+3 = 4$, as Job 1 figures at the first place in parent 1 and at the third place in parent 2 (Table 4-3). The child is created with establishing an order according to increasing preference values.

Table 4-3 – Preference values

Job	Preference value
1	$1+3 = 4$
2	$3+5 = 8$
3	$4+4 = 8$
4	$5+2 = 7$
5	$2+1 = 3$

Based on the preference values in the child the order of jobs is [5 1 4 2 3] or [5 1 4 3 2]. In this work in the case of equal precedence values a lexicographic order has been applied, thus the order of jobs is [5 1 4 2 3].

4.2.1.3. Calculation results

The following equations and diagrams show the adapted forms of the disjunctive graph model to each problem, and the Box-Whisker plot of obtained makespan values by different solving methods.

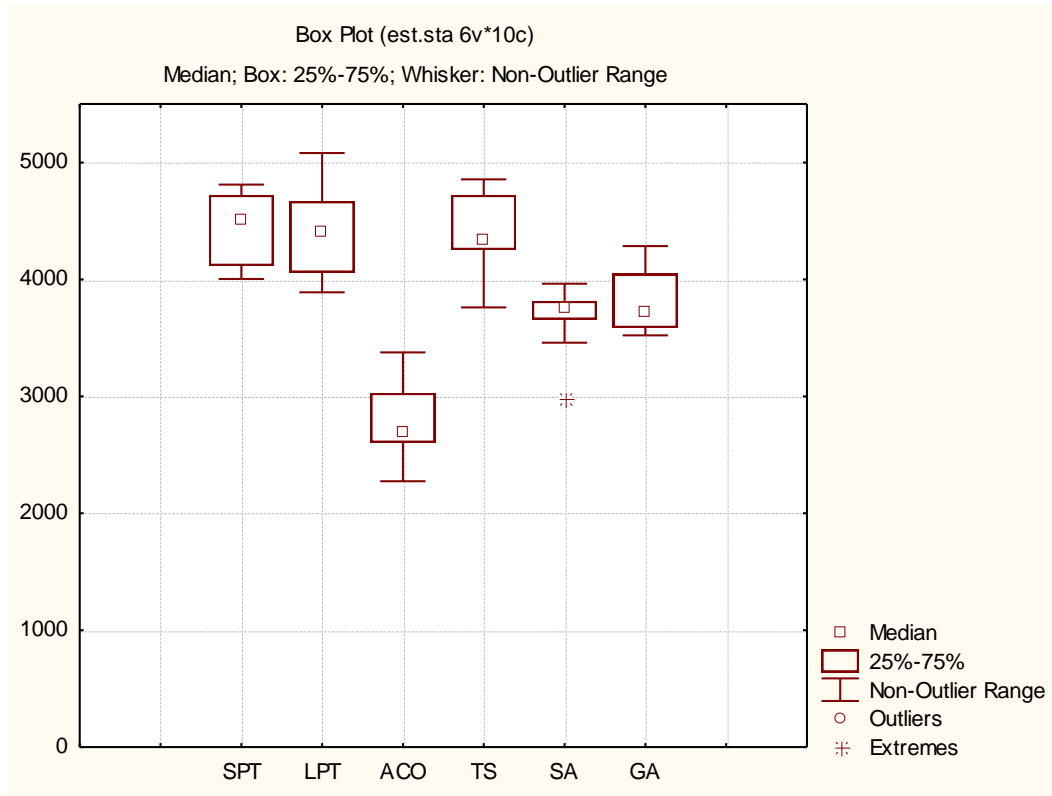


Figure 4-3 – Box-Whisker plot of makespan values obtained to the F10|chain, $est_k|C_{max}$ problem

On Figure 4-3 it is shown that every solving method was able to find a solution. It can also be deduced that meta-heuristics were able to gain makespan values less than the ones given by the heuristics.

Although the average makespan value obtained by $\varepsilon = \text{ACO}$ method were better than the ones provided by $\varepsilon = \text{TS}$, $\varepsilon = \text{GA}$ or $\varepsilon = \text{SA}$, it does not necessarily mean that for F10|chain, $est_k|C_{max}|CTR\text{-}PG$, PMB problem-model association the former solving method are better than the latter ones. Meta-heuristics propose a guideline which has to be always adapted to the concerned problem, and therefore it is very difficult to compare the different meta-heuristics. With other words: the more we work on a meta-heuristic method (adjusting the parameters, changing the way of interpretation) the better results can be obtained.

Nevertheless, in order to have a comparison which is at least reasonable, the number of evaluated schedules has been fixed to the same number (50) for each method. It is still not a valid comparison of meta-heuristic methods, but just of the adaptations of these methods performed by this work, in order to illustrate their applicability and capability to obtain better results than heuristics.

A nomenclature used in the adapted models is summarized in Table 4-4:

Table 4-4 – Nomenclature of the adapted models

EST_k	Earliest starting time of job k
$MWT_{k,p}$	Minimal waiting time after operation p of job k
DD_k	Due date (deadline) of job k
L_k	Lateness of job k
ch_k	Changeover time from job k to any other job
$ch_{k',k}$	Changeover time from job k to job k'
ch_p	Changeover time from operation p of any job

The model has been adapted by introducing a continuous constraint (Equation 4-2):

$$\begin{aligned}
 & \min C_{\max} \\
 (1) \quad & C_{\max} - Ts_{k,p} \geq T_{k,p} \quad \forall kp \in X \\
 (2) \quad & Ts_{k,p'} - Ts_{k,p} \geq T_{k,p} \quad \forall \langle p, k' \rangle \in U \\
 (3) \quad & \left. \begin{aligned} Ts_{k',p} - Ts_{k,p} &\geq T_{k,p} + M(x_{kp'k'} - 1) \\ Ts_{k,p} - Ts_{k',p} &\geq T_{k',p} - Mx_{kp'k'} \end{aligned} \right\} \quad \forall \langle p, k' \rangle \in D \\
 (4) \quad & Ts_{k,p} \geq EST_k \quad \forall kp \in X \\
 & x_{kp'k'} \in \{0, 1\} \quad \forall \langle p, k' \rangle \in D \\
 & Ts_{k,p} \geq 0 \quad \forall kp \in X
 \end{aligned}$$

Equation 4-2 – Disjunctive graph model adapted to F10|chain, est_k|C_{max}

Constraint (4) fixes that every starting time (for all operation) has to be superior or equal to the earliest possible starting time belonging to job k.

The number of variables and the number of constraints are shown in Table 4-5:

Table 4-5 – Number of variables and constraints for Equation 4-2

Number of variables	J^2A+1
Number of constraints	$J \cdot (2A \cdot J + A - 1) + J \cdot A$

Notice that if EST_k values are compulsory to be defined (i.e. $EST_k = 0$ for immediately available jobs) then the $Ts_{k,p} \geq 0$ constraint becomes useless and can be removed from the model.

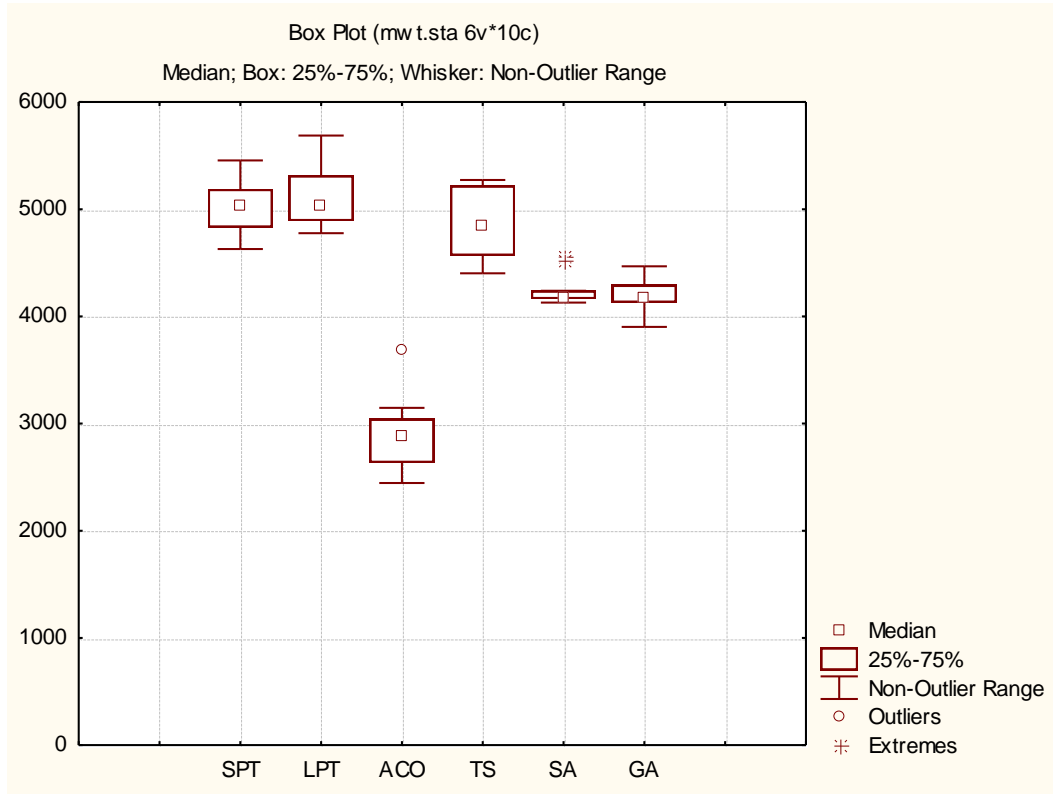


Figure 4-4 – Box-Whisker plot of makespan values obtained to the F10|chain, MWT|C_{max} problem

As it is shown in Figure 4-4 the behavior of the solving methods did not change with respect to the previous case by the introducing of minimal waiting times.

The model has been adapted by modifying a continuous constraint (Equation 4-3):

$$\begin{aligned}
 & \min C_{\max} \\
 (1) \quad & C_{\max} - Ts_{k,p} \geq T_{k,p} \quad \forall kp \in X \\
 (2) \quad & Ts_{k,p'} - Ts_{k,p} \geq T_{k,p} + MWT_{k,p} \quad \forall \langle p, k'p' \rangle \in U \\
 (3) \quad & \left. \begin{aligned} Ts_{k',p} - Ts_{k,p} &\geq T_{k,p} + M \langle x_{kp'k'} - 1 \rangle \\ Ts_{k,p} - Ts_{k',p} &\geq T_{k',p} - Mx_{kp'k'} \end{aligned} \right\} \quad \forall \langle p, k'p' \rangle \in D \\
 & x_{kp'k'} \in \{0, 1\} \quad \forall \langle p, k'p' \rangle \in D \\
 & Ts_{k,p} \geq 0 \quad \forall kp \in X
 \end{aligned}$$

Equation 4-3 – Disjunctive graph model adapted to F10|chain, MWT|C_{max}

The minimal waiting times are taken into account in the constraint (2). By adding the value $MWT_{k,p}$ it is ensured that no following operation p' of job k can start before this necessary minimal waiting time.

The number of variables and the number of constraints are shown in Table 4-6:

Table 4-6 – Number of variables and constraints for Equation 4-3

Number of variables	J^2A+1
Number of constraints	$J \cdot (2A \cdot J + A - 1)$

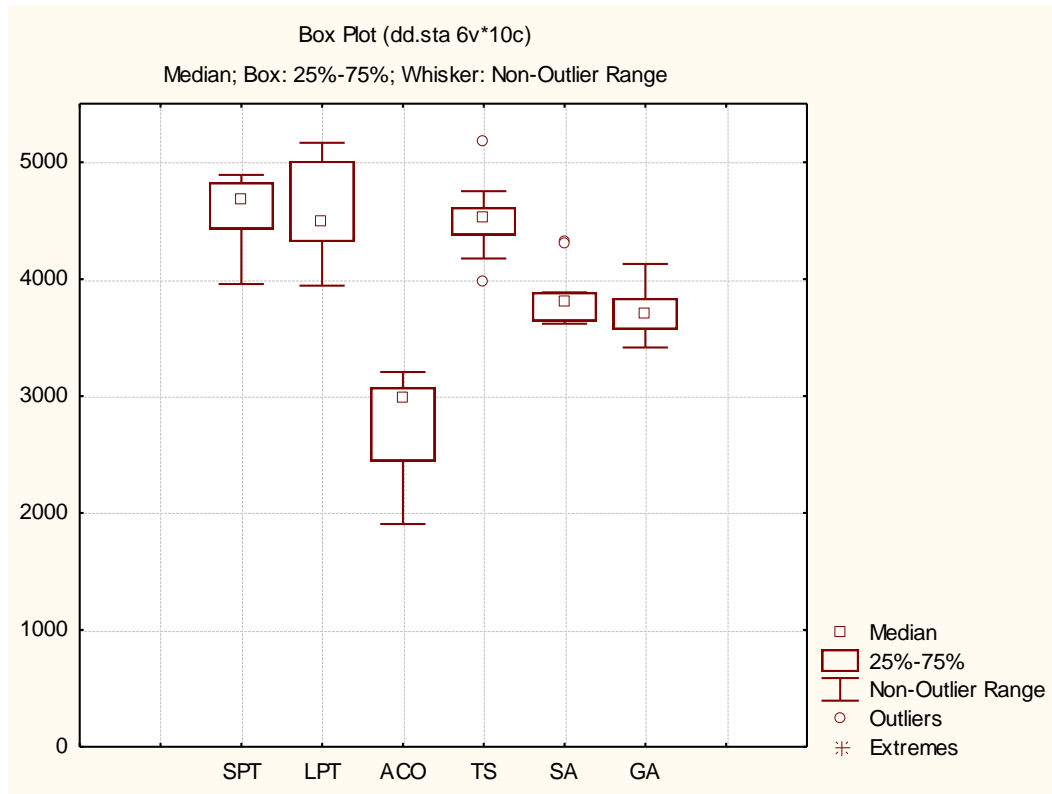


Figure 4-5 – Box-Whisker plot of makespan values obtained to the F10|chain, d_k | C_{\max} problem

Figure 4-5 shows the results obtained in the case of the F10|chain, d_k | C_{\max} problem. The model is adapted by introducing a new continuous variable L_k and a continuous constraint (4). The adapted model is given by Equation 4-4. The introduction of L_k (the lateness) is advised in order to avoid infeasibility in the case of those problems where it is impossible to respect all of the deadlines.

$$\begin{aligned}
& \min C_{\max} \\
(1) \quad & C_{\max} - Ts_{k,p} \geq T_{k,p} \quad \forall kp \in X \\
(2) \quad & Ts_{k,p'} - Ts_{k,p} \geq T_{k,p} \quad \forall (p, k'p') \in U \\
(3) \quad & \left. \begin{aligned} Ts_{k',p} - Ts_{k,p} &\geq T_{k,p} + M(x_{kp'k'} - 1) \\ Ts_{k,p} - Ts_{k',p} &\geq T_{k',p} - Mx_{kp'k'} \end{aligned} \right\} \quad \forall (p, k'p') \in D \\
(4) \quad & Ts_{k,p} + T_{k,p} \leq DD_k + L_k \quad \forall kp \in X \\
& x_{kp'k'} \in \{0, 1\} \quad \forall (p, k'p') \in D \\
& Ts_{k,p} \geq 0 \quad \forall kp \in X
\end{aligned}$$

Equation 4-4 – Disjunctive graph model adapted to F10|chain, d_k | C_{\max}

The constraint (4) is introduced in order to ensure that the finishing time of operations be inferior to the corresponding deadline with a tolerance L_k .

The number of variables and the number of constraints are shown in Table 4-6:

Table 4-7 – Number of variables and constraints for Equation 4-4

Number of variables	J^2A+1+J
Number of constraints	$J \cdot (2A \cdot J + A - 1) + J \cdot A$

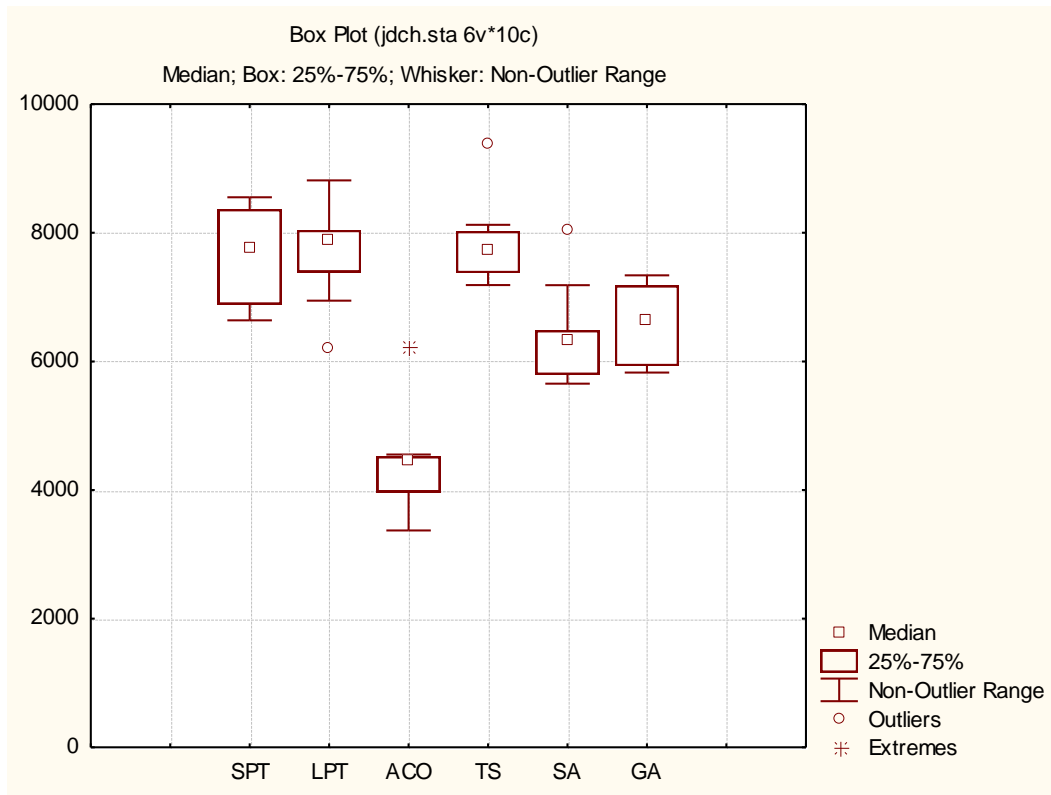


Figure 4-6 – Box-Whisker plot of makespan values obtained to the F10|chain, ch_k | C_{\max} problem

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

In Figure 4-6 the obtained results can be seen in the case of job dependant changeovers. The model is adapted by modifying binary constraints (Equation 4-5):

$$\begin{aligned}
 & \min C_{\max} \\
 (1) \quad & C_{\max} - Ts_{k,p} \geq T_{k,p} \quad \forall kp \in X \\
 (2) \quad & Ts_{k,p'} - Ts_{k,p} \geq T_{k,p} \quad \forall \langle p, k'p' \rangle \in U \\
 (3) \quad & \left. \begin{aligned} Ts_{k',p} - Ts_{k,p} &\geq T_{k,p} + ch_k + M(x_{kp'k'} - 1) \\ Ts_{k,p} - Ts_{k',p} &\geq T_{k',p} + ch_{k'} - Mx_{kp'k'} \end{aligned} \right\} \quad \forall \langle p, k'p' \rangle \in D \\
 & x_{kp'k'} \in \{0,1\} \quad \forall \langle p, k'p' \rangle \in D \\
 & Ts_{k,p} \geq 0 \quad \forall kp \in X
 \end{aligned}$$

Equation 4-5 – Disjunctive graph model adapted to F10|chain, ch_k|C_{max}

In order to take into account the changeover time, the pair of constraints (3) has to be modified. Thus, if the constraint is active, i.e. a machine performs job k' after job k then the corresponding changeover time is added to the finishing time of the operation.

The number of variables and the number of constraints are shown in Table 4-8:

Table 4-8 – Number of variables and constraints for Equation 4-5

Number of variables	J^2A+1
Number of constraints	$J \cdot (2A \cdot J + A - 1)$

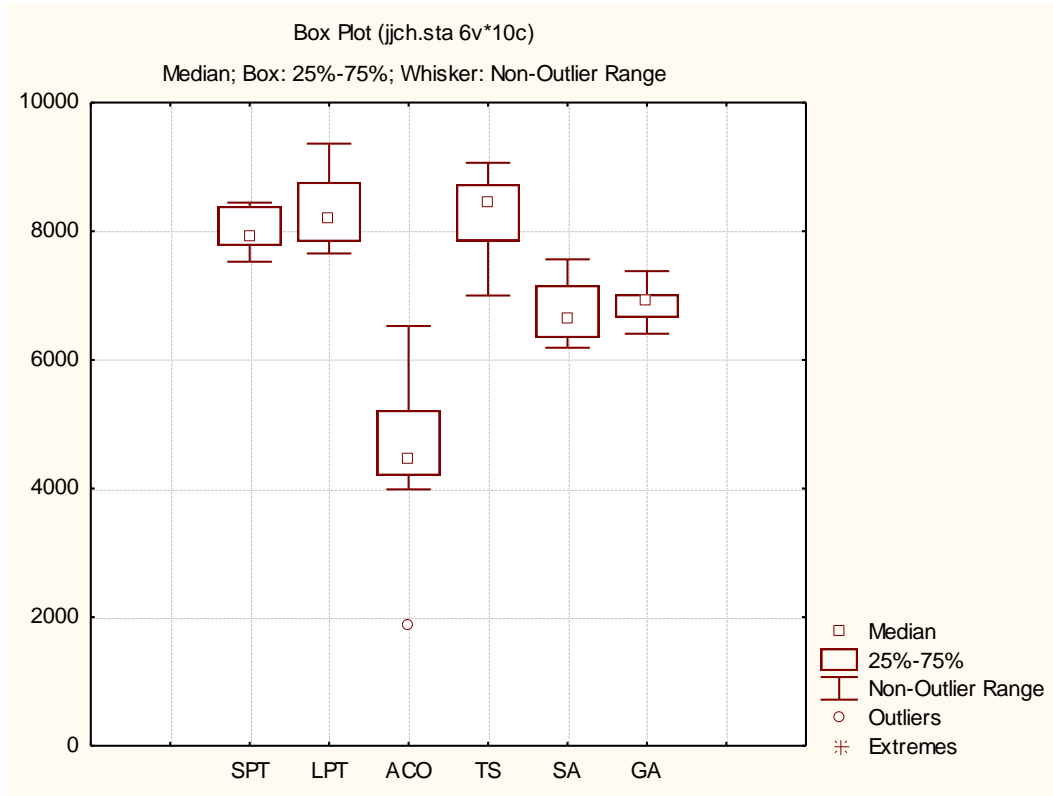


Figure 4-7 – Box-Whisker plot of makespan values obtained to the F10|chain, $ch_{kk'}$ | C_{max} problem

Job-to-job dependant changeovers lead to similar results to the job-dependant changeover ones (Figure 4-7). The model is adapted in the same way, by modifying binary constraints (Equation 4-6):

$$\begin{aligned}
 & \min C_{\max} \\
 (1) \quad & C_{\max} - Ts_{k,p} \geq T_{k,p} \quad \forall kp \in X \\
 (2) \quad & Ts_{k,p'} - Ts_{k,p} \geq T_{k,p} \quad \forall \langle p, k' \rangle \in U \\
 (3) \quad & \left. \begin{aligned} Ts_{k',p} - Ts_{k,p} &\geq T_{k,p} + ch_{kk'} + M(\langle k, k' \rangle - 1) \\ Ts_{k,p} - Ts_{k',p} &\geq T_{k',p} + ch_{k'k} - Mx_{kp'k'} \end{aligned} \right\} \quad \forall \langle p, k' \rangle \in D \\
 & x_{kp'k'} \in \{0, 1\} \quad \forall \langle p, k' \rangle \in D \\
 & Ts_{k,p} \geq 0 \quad \forall kp \in X
 \end{aligned}$$

Equation 4-6 – Disjunctive graph model adapted to F10|chain, $ch_{kk'}$ | C_{max}

The new term $ch_{kk'}$ in constraints (3) plays the same role as ch_k in Equation 4-5. The only difference lies in the fact that in Equation 4-6 the changeover time depends on both jobs k and k' , and not only on job k . But, this does not mean any problem to the disjunctive graph model, as the pairs of constraints (3) already exist for every disjunctive edge.

The number of variables and the number of constraints are shown in Table 4-9:

Table 4-9 – Number of variables and constraints for Equation 4-6

Number of variables	J^2A+1
Number of constraints	$J \cdot (2A \cdot J + A - 1)$

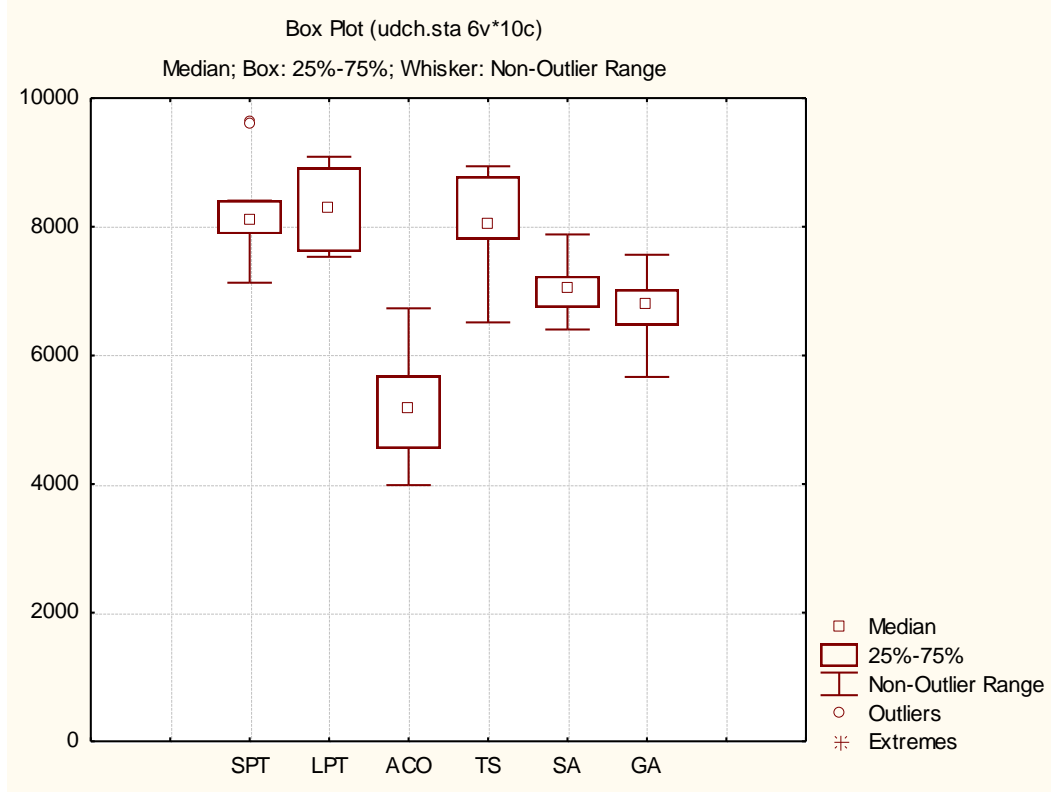


Figure 4-8 – Box-Whisker plot of makespan values obtained to the F10|chain, ch^U | C_{\max} problem

From the implicit assumption of disjunctive graph model which says that every operation p of any job k has its own only one machine to be performed on, it follows that the unit-dependant changeover time (Figure 4-8) can be described by a term ch_p which depends on the order number of the operation (regardless to its job). The model is thus adapted by modifying binary constraints (3) as it is shown in Equation 4-7:

$$\begin{aligned}
 & \min C_{\max} \\
 (1) \quad & C_{\max} - Ts_{k,p} \geq T_{k,p} \quad \forall kp \in X \\
 (2) \quad & Ts_{k,p'} - Ts_{k,p} \geq T_{k,p} \quad \forall \langle p, k' \rangle \in U \\
 (3) \quad & \left. \begin{aligned} Ts_{k',p} - Ts_{k,p} &\geq T_{k,p} + ch_p + M(x_{kp'k'} - 1) \\ Ts_{k,p} - Ts_{k',p} &\geq T_{k',p} + ch_p - Mx_{kp'k'} \end{aligned} \right\} \quad \forall \langle p, k' \rangle \in D \\
 & x_{kp'k'} \in \{0, 1\} \quad \forall \langle p, k' \rangle \in D \\
 & Ts_{k,p} \geq 0 \quad \forall kp \in X
 \end{aligned}$$

Equation 4-7 – Disjunctive graph model adapted to F10|chain, ch^U | C_{\max}

The adaptation way is the same as in the case of $\beta_9 = ch_k$ or $\beta_9 = ch_{kk}$ – the corresponding changeover time appears in the pair of constraints (3).

The number of variables and the number of constraints are shown in Table 4-10:

Table 4-10 – Number of variables and constraints for Equation 4-7

Number of variables	J^2A+1
Number of constraints	$J \cdot (2A \cdot J + A - 1)$

As it is illustrated with Equations 4-1 – 4-7, none of the above tested properties leads to the introduction of new binary variables or constraints on binary variables. Figures 4-3 – 4-8 show that computational behavior did not changed due to these minor modifications.

What are then the characteristics having important effects to the behavior and complexity of a mathematical model? Pan et al., 2009 showed that the objective function is a very important property in the case of network batch processes. Also, the introduction of new binary variables and binary constraints increase the problem's combinatorial complexity and therefore has an important effect to the solvability and behavior of a mathematical model. Assume e.g. that instead of each machine in $F10|chain|C_{max}$ we dispose a workstation of two machines which are identical to each other. Let us now try to apply the disjunctive graph model to this $FP20|chain|C_{max}$ problem. Due to the implicit assumption of this model that every operation has its corresponding machine, the presence of additional machines to realize the same tasks makes it very hard to adapt the model.

4.2.2. Application of different mathematical models

In Section 4.2.1 the same model has been adapted to different problems. The second question is: how different models can be applied to the same problem?

The more complex a scheduling problem is (i.e. material balances need to be treated, resources have to be taken into account, splitting/merging is authorized, etc.) usually the more complex the representing mathematical model will be.

To study the applicability of different mathematical models to the same problem, three sequential, multiproduct batch plant examples have been examined.

4.2.2.1. The scheduling problems

The first and the second examples have been proposed by Epperly et al. (1997). The third example has been created by this work, enlarging these two problems.

Example 4-1 is a 3-stage 2-product batch plant (Figure 4-9), whose characteristics are summarized in Table 4-11.

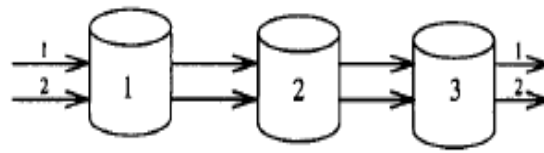


Figure 4-9 (from Epperly et al., 1997) – Example 4-1

Table 4-11 – Problem characteristics of Example 4-1

Process topology:	Sequential
Production purpose:	Multiproduct
Production mode:	Batch
Time constraints:	-
Resource constraints:	Final intermediate storage

The objective function to maximize is the total profit. Thus, Example 4-1 is an F3|cons, CRA, network, FIS_d|ΣP problem according to the notation scheme presented in chapter three.

The data of Example 4-1 is shown in Table 4-12.

Table 4-12 – Data of Example 4-1

PROBLEM DATA			Working quantity	
<i>Job</i>	<i>Operation</i>	<i>Process time</i>	Min	Max
1	1	8	500	4500
1	2	20	500	4500
1	3	8	500	4500
2	1	16	500	4500
2	2	4	500	4500
2	3	4	500	4500
Products		Require	Price	
1		200	10	
2		100	10	

In Table 4-12 it is shown that the problem is composed of 2 jobs, each of them 3 operations, with different, invariable process times, which are known in advance. Working quantity is limited by both inferior and superior limits. The prices of the two products are the same, but the required quantities are different.

Example 4-2 is a 6-stage 4-product batch plant, illustrated in Figure 4-10. The data of Example 4-2 can be found in the Appendix of this chapter.

In Example 4-2 there are four jobs to be performed, each of them has 6 operations. The process times are fix and known in advance. There are requirement constraints for each product (i.e. a minimal quantity to produce) and the capacity of the stages enforces the working quantity to be between inferior and superior limits. The prices of products are different.

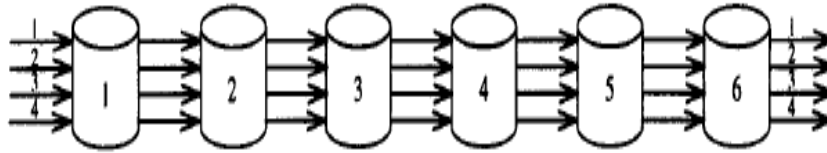


Figure 4-10 (from Epperly et al. 1997) – Example 4-2

The characteristics of Example 4-2 are summarized in Table 4-13. The objective function to maximize is the total profit. Thus, Example 4-2 is an F6|cons, CRA, network, FIS_d|ΣP problem according to the notation scheme presented in chapter three.

Table 4-13 – Problem characteristics of Example 4-2

Process topology:	Sequential
Production purpose:	Multiproduct
Production mode:	Batch
Time constraints:	-
Resource constraints:	Final intermediate storage

The recipe of Example 4-3 is illustrated on Figure 4-11. The data of this example can be found in Appendix 1 of this chapter.

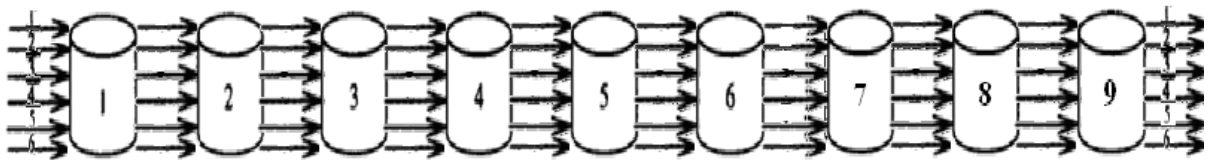


Figure 4-11 – Example 4-3

The characteristics of Example 4-3 are summarized in Table 4-14. The objective function to maximize is the total profit. Thus, Example 4-1 is an F9|cons, CRA, network, FIS_d|ΣP problem according to the notation scheme presented in Chapter Three.

Table 4-14 – Problem characteristics of Example 4-3

Process topology:	Sequential
Production purpose:	Multiproduct
Production mode:	Batch
Time constraints:	-
Resource constraints:	Final intermediate storage

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

Example 4-3 is a 9-stage-6-product batch plant. There are six jobs to be performed and each of them has 9 operations. The process times are fix and known in advance. The process delivers 6 products and the production is to be realized in a batch plant with 9 stages. There is a minimal required quantity for each product and the capacity of the stages enforces the working quantity to be between inferior and superior limits. The prices of products are the same. The data of Example 4-3 is presented in Appendix 2 of this chapter.

4.2.2.2. The mathematical models and the solving method

The example problems have been modeled by 4 models: the discrete time representation model of Kondili et al. (1993) ($\delta = \text{DTR, STN}$), a global time event based model of Maravelias and Grossmann (2003) ($\delta = \text{CTR-GTE, STN}$), a unit-specific time event based model of Ierapetritou and Floudas (1998) ($\delta = \text{CTR-USTE, STN}$) and a slot-based model of Sundaramoorthy and Karimi (2005) ($\delta = \text{CTR-SS, STN}$). Exact solution method was applied for each case ($\varepsilon = \text{MIP}$). The time horizon is 72h.

The solutions for each model have been obtained by applying the default MIP solver of the software AIMMS[®] (Advanced Interactive Multidimensional Modeling System). The AIMMS software system is designed for modeling and solving large-scale optimization and scheduling-type problems. The computations have been performed on a Dell[®] Latitude D510 Intel Pentium 1.73GHz personal computer.

4.2.2.3. Calculation results

The optimal schedule for Example 4-1 is shown in Figure 4-12:

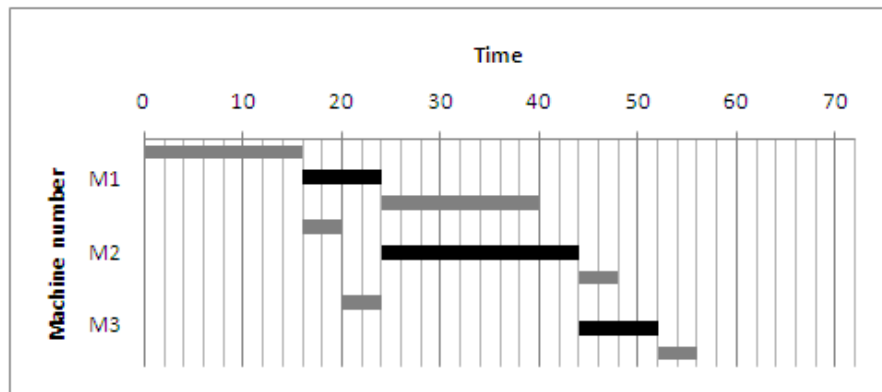


Figure 4-12 – Optimal schedule for Example 4-1

On Figure 4-11 the operations of Job 1 are colored with black, and the operations of Job 2 are colored with gray. It can be seen that after having finished Job 1 and Job 2 there is enough time to run again one of the jobs (in this case Job 2) and sell thus two times more from the obtained product.

Table 4-15 – Solutions obtained for Example 4-1

Model	Objective function (\$)	Computational time (s)	Number of variables	Number of constraints
$\delta_1 = \text{CTR-SS}$	90000	0.80	1248	2630
$\delta_1 = \text{DTR}$	135000	0.73	944	1007
$\delta_1 = \text{CTR-GTE}$	135000	0.27	648	1089
$\delta_1 = \text{CTR-USTE}$	135000	0.37	798	1553

Table 4-15 presents the objective function values obtained by solving the different models, the computational times, the number of variables and the number of constraints.

The CTR-SS model – unless the other ones – did not find the optimal solution of 135000. The reason of this fact is that the model does not allow selling products before the end of the time horizon. Due to this inconvenience the solution proposed by this model does not run Job 2 again, and thus the objective function value does not reach the optima of 135000 but only 90000 (as the maximal working quantities are 4500 for both product and the prices are 10, the total realized sold will be $2 \cdot 4500 \cdot 10 = 90000$).

In order to represent the time horizon of 72h the discrete time representation based model of Kondili et al. (1993) needs an important number of time intervals. In this case, as all duration times can be divided by 4, the problem can be modeled in a horizon divided into 18 time intervals. On the other hand, as the number of operations is not great, the continuous time representation based formulations can model the problem with less time event point or slot (in this case concretely 10 event points were declared to be available for the model and it proved to be enough). Thus, the event-point based formulations (CTR-GTE and CTR-USTE) were the most adequate to this example.

Table 4-16 – Solutions obtained for Example 4-2

Model	Objective function (\$)	Computational time (s)	Number of variables	Number of constraints
$\delta_1 = \text{CTR-SS}$	166500	1.49	9088	19550
$\delta_1 = \text{DTR}$	211500	2.78	6220	6357
$\delta_1 = \text{CTR-GTE}$	211500	1.13	4988	7949
$\delta_1 = \text{CTR-USTE}$	211500	44.14	10388	22041

In the case of Example 4-2 the time slot based model met the same problem as for Example 4-1 due to the same reason (Table 4-16). With the increasing number of machines, the unit-specific time event based model needs significantly more computational time than the global time event based model, which remains to be the most effective. The number of time intervals is 18, the number of event points is 20 and the number of time slots is 20.

Table 4-17 – Solutions obtained for Example 4-3

Model	Objective function (\$)	Computational time (s)	Number of variables	Number of constraints
$\delta_1 = \text{CTR-SS}$	270000	5.30	59980	129542
$\delta_1 = \text{DTR}$	315000	11.2	19716	19939
$\delta_1 = \text{CTR-GTE}$	315000	5.89	16620	25981
$\delta_1 = \text{CTR-USTE}$	-----	>360	49290	107905

Table 4-17 illustrates that in the case of Example 4-3 with the time-slot based, discrete time representation based, and the global time event based models found feasible solutions in less than a minute. As by applying the unit-specific time event based model no solution has been obtained in ten minutes, we decided to interrupt the solution process. In Table 4-17 it is shown that the CTR-USTE model has more variables and more constraints than the DTR and the CTR-GTE model, which explains the worse computational behavior. On the other hand, the CTR-SS model is still applicable despite of the huge number of constraints and variables. This contradiction can be explained by the relaxation properties of this model, i.e. the CTR-SS model of Sundaramoorthy and Karimi (2005) is formulated without big-M constraints.

The reason of the lower objective function obtained by the time slot based formulation has already been discussed above. Between the two models achieving the optima the continuous time formulation needed less computational time.

Resolving Example 4-1, there was no significant difference in computational times. However, to solve the larger problems Example 4-2 and 4-3 the unit-specific event-based continuous formulation based model was significantly slower than the other models. Based on the computational results the global time event-based formulation is advised to apply.

It has been shown that $F10|chain|C_{\max}$ problems can be associated with the disjunctive graph ($\delta = \text{CTR-PG}$, PMB) model and solved by numerous different solving methods. This property holds even with several modifications in the field β apart from $\beta_2 \neq \emptyset$ (presence of secondary resources) and some other special cases.

It can be seen too, that to the complicated $\beta_2 \neq \emptyset$ problems $\delta_2 = \text{STN}$ models had to be applied. On the other hand, the typical objective function in these cases was $\gamma = \sum P$ which is better to deal with than the $\gamma = C_{\max}$, and therefore we can find successful examples to the application of $\varepsilon = \text{MILP}$ exact methods.

Based on the above results and literature experiences the following scheme can be proposed to construct a solution for a scheduling problem (Figure 4-13).

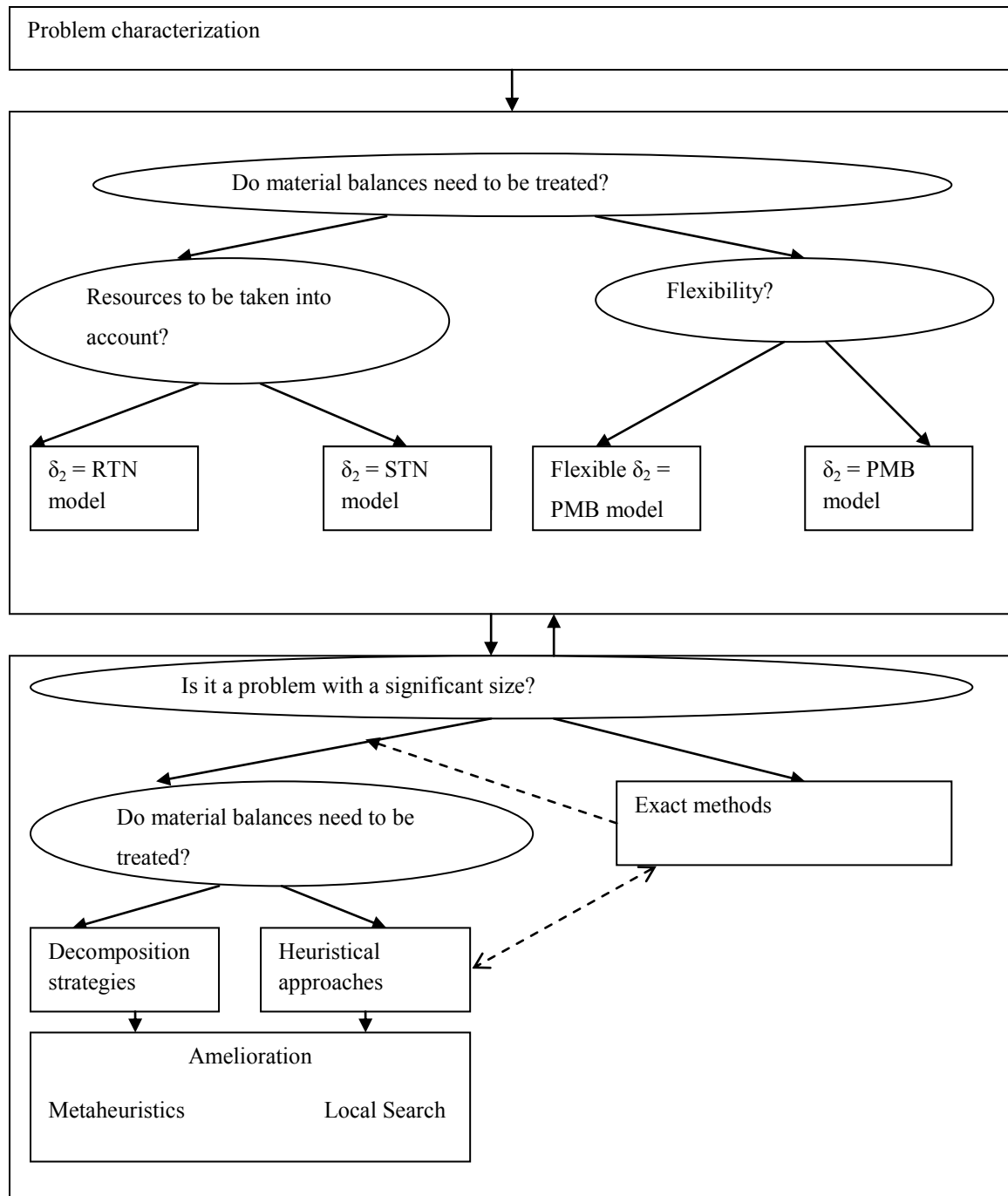


Figure 4-13 – Methodology scheme of proposal of solution strategy to scheduling problems

Obviously this methodology scheme could be go further and be more detailed, e.g. with respect to the objective function type the behaviour of the different mathematical models and numerical methods will not be the same. However, as this research is focusing on the application of CBR, here just some basic general rules are summarized.

To identify a possible resolution strategy, the first step to do is to characterize the problem. The aspects of characterization have been detailed in chapter three.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

In the modelling phase the first aspect to verify is the presence of material resources (if the answer is positive ($\beta_2 \neq \emptyset$) then we need a model which is able to deal with material balances). If there is no material balance handling, the preference is to apply the simplest possible model, e.g. the disjunctive graph model.

If there are resources to deal with, a State Task Network ($\delta_2 = \text{STN}$) model is recommended. If not only materials but even secondary resources have to be taken into account ($\beta_{2a} \neq \text{cons}, \emptyset$) then a Resource Task Network ($\delta_2 = \text{RTN}$) based model will be proposed.

In the resolution phase the first question is whether the application of exact methods ($\varepsilon = \text{MIP}$) is possible. The term “reasonable time” always has to be defined by the user. Several aspects influence whether the problem is hopeful or not to be solved by exact methods in reasonable time. Main attributes to take into account: number of binary variables, number and formulation of constraints, type of the objective function...

If exact methods do not lead to success, then in the cases without material balance handling usually heuristics and meta-heuristics are applied. On the other hand, in the case of problems with material balance handling decomposition strategies are more frequently used.

4.3. Rule based strategy to associate model/method to a scheduling problem

In Figure 4-13 a methodology scheme has been proposed to associate mathematical model and solution method to a scheduling problem. The summary of these rules is presented here.

4.3.1. Summary of the rules of association

The following eight rules summarize the above discussed association strategy.

1. Identify the problem applying the notation scheme of chapter three.
2. If $\beta_2 \neq \emptyset$ then go to step 4.
3. If $\alpha_{1b} \neq \emptyset$ then apply a flexible $\delta = \text{CTR-PG}$, PMB model else apply the disjunctive graph model.
4. If $\beta_{2a} \neq \text{cons}, \emptyset$ (i.e. apart from the materials there are other resources as well) then apply a $\delta_2 = \text{RTN}$ model, else a $\delta_2 = \text{STN}$ model.
5. If $\beta_5 \neq \emptyset$ then most $\delta_1 = \text{DTR}$ models are not applicable. Apply a $\delta_1 = \text{CTR}$ model.
6. If the problem size allows try $\varepsilon = \text{MIP}$. If there is success, stop.
7. If $\beta_2 \neq \emptyset$ try $\varepsilon = \text{HC}$ with $\delta = \emptyset$. Else continue $\varepsilon = \text{MIP}$ with decomposition approaches (e.g. Rolling Horizon technique).
8. Try to ameliorate the solution by $\varepsilon = \text{Comb}$ meta-heuristics.

4.3.2. Example cases

The strategy described above will be applied to the three examples of chapter one. Applying the notation system to Example 1-1, the notation obtained is: $\{\text{FR10|cons, CRA, chain, EE, est}_k, P_s, d_k, \text{mwt, zw, ch}^{kk}, \text{con, fix|C}_{\max}\}$.

According to $\beta_2 \neq \emptyset$, the process is continued with step 4. As $\beta_{2a} = \text{cons}$, \emptyset a $\delta_2 = \text{RTN}$ model is to be applied. Due to $\beta_5 \neq \emptyset$, a $\delta_1 = \text{CTR}$ model has to be chosen, either an event-based, either a slot-based one. If it is permitted to realize the selling before the end of the time horizon, then the event time point based formulation is advised.

In the case of Example 1-2 the notation is: $\{\text{JP4}|\text{cons}, \text{CRA}, \text{network}, \text{CC}, \text{P}_s, \text{FIS}_d, \text{ZW}|\Sigma\text{P}\}$. As $\alpha_{1a} \neq \text{F}$, the concerned problem is not a flow-shop, therefore the algorithm will not be applied.

Finally, to Example 1-3 the notation is: $\{\text{F6}|\text{cons}, \text{CRA}, \text{chain}, \text{P}_s, d_k, \text{FIS}_d, \text{LW}, \text{overlap}, \text{con}, \text{fix}|\text{C}_{\max}\}$. Similarly to Example 1-1 a $\delta = \text{CTR}$, STN model is proposed.

4.4. Case based reasoning based strategy to associate model/method to a scheduling problem

A rule based association strategy has been proposed in the previous chapter. However, it has been shown that a simple rule-based strategy is not the best solution. The most important disadvantages of such a strategy are that it could not be rich enough to describe all special types of problems, and could not be enough flexible to follow the dynamic evolution of the scheduling area. Therefore, in order to go further and construct a better, more detailed, more effective strategy, another artificial intelligence approach is chosen, namely the case based reasoning. The reason of this choice has been explained in Chapter One.

4.4.1. Introduction

Case-Based Reasoning approach possesses a learning capability. The continuous dynamic challenge of the scheduling research domain makes this approach an ideal choice in our case.

To elaborate a case-based reasoning system the following important problems are to be solved: representation of a case, find the most similar case, and adaptation of an existing solution. Revision and storage are two other questions coming from the three above.

In this paragraph the elaboration of the foundations of a CBR strategy applied to the above problematic will be discussed.

4.4.2. Main steps of case-based reasoning and their realization

In this paragraph the main aspects of a case-based reasoning will be discussed, namely: the case base, the description of a case, the retrieval of a case (based on a similarity function), and finally the adaptation of a case.

In order to describe a case, it is characterized using the classification scheme proposed in Chapter Three, and represented by the proposed notation scheme. In the Appendix 3 of this chapter a database regrouping 73 cases is detailed. The corresponding notation values to the case base are also presented in Appendix 4 of this chapter. The term “case” thus refers to the $\alpha|\beta|\gamma|\delta|\epsilon$ resolution strategy of the $\alpha|\beta|\gamma$ scheduling problem, the applied (if any) δ model and the used ϵ solving method.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

For example: Example 4-1 is modeled by the global event based model of Maravelias and Grossmann (2003) and solved by using a MIP solver will be denoted as an $\alpha|\beta|\gamma|\delta|\varepsilon = F3|cons, CRA, network, FIS_d|\Sigma P|CTR-GTE, STN|MIP$ case, where $\alpha|\beta|\gamma = F3|cons, CRA, network, FIS_d|\Sigma P$ denotes the problem, $\delta = CTR-GTE$, STN denotes the applied mathematical model type, and $\varepsilon = MIP$ denotes the solving method used. If a case has already been solved, then information on the quality of the gained solution can also be recorded with the case.

4.4.3. Case retrieval – Similarity function

In order to maintain the case base and to perform the retrieval of similar cases, an object-oriented framework encoded in Java for building Case-Based Reasoning (CBR) systems has been applied: the Java COLIBRI (<http://gaia.fdi.ucm.es/index.html>). The abbreviation COLIBRI stands for Cases and Ontology Libraries Integration for Building Reasoning Infrastructures. It includes a complete Graphical User Interface that guides the user in the design of a CBR system and fulfills the similarity analysis as well. The program developed by the Group of Artificial Intelligence Applications (GAIA) is available for download from the given site.

COLIBRI tools allow building a CBR system without writing a single line of code and support many features like graphical interfaces, description logics and ontologies, textual CBR, evaluation, etc.

The similarity measure (see Section 1.5) used to retrieve the most similar cases is configured by the user. COLIBRI implements also several similarity functions that can be used depending on the type of the attribute (integers, strings, etc.), own similarity measures can also be defined.

4.4.4. Adaptation of a case

Once the most similar case is found, its solution must be adapted to the new problem before being reused. It has been shown that a minor modification on the model does not concern significantly the behaviour of the solution process. Consequently, if the problem of the new case can be obtained from the one of the most similar case through a minor modification then the advised adaptation is simple: introduce the necessary new variables and or constraints to the model, and apply the solving method used in the solved case.

For such cases that cannot be solved via a minor modification, an intervention will be necessary. Then these cases (resolved with the help of the user) will be learnt by the system and thus enrich the case memory.

4.4.5. Examples of application

An initial database of 73 cases (Appendix 4) has been created. The case base contains the problems generated by the method of Taillard (1993) presented and solved in Sections 1.4 and 4.2, problems Example 4-1, 4-2 and 4-3 with the mathematical models and solving method applied, and Examples 1-1, 1-2 and 1-3 with associated mathematical model and solving methods. The database has been saved in the COLIBRI software.

Applying the default Euclidean distance based similarity function in the retrieval process (with no weighting parameters), the program can find the most similar cases to a new problem helping the user to associate a

mathematical model and a solution method to it. Thus, the first, preliminary version of the decision-aid system – even if it is not very strong yet – is born.

Let us for example apply the program in order to find a resolution strategy to an hypothetical $F5|d_k|C_{\max}$ problem. With the help of the graphical interface it is easy to enter the problem representation (the query is shown in Figure 4-14):

Parameter	Value	Weighting
a1	F	1.0
a2	5	1.0
b1		1.0
b2		1.0
b3		1.0
b4		1.0
b5		1.0
b6	dk	1.0
b7		1.0
b8		1.0
b9		1.0
b10		1.0
b11		1.0
b12		1.0
b13		1.0
b14		1.0
g	Cmax	1.0

Figure 4-14 – Query of new problem

In Figure 4-14 the fields a_i , b_i and g correspond to the fields α_i , β_i and γ of the notation system presented in Chapter Three, respectively. The parameters next to these fields are the weighting parameters applied in the similarity function.

The program engine now finds the most similar case to the problem. The result is given by an output screen, shown in Figure 4-15:

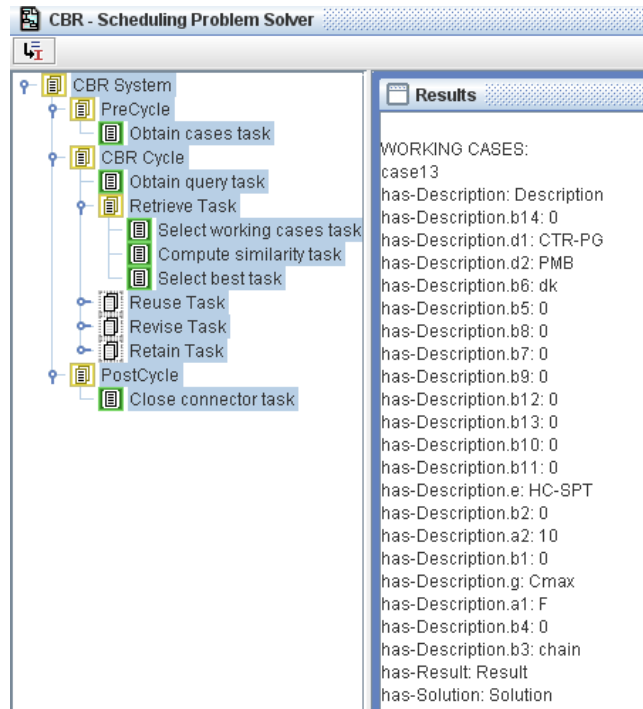


Figure 4-15 – Simulation result to example problem $F5|d_k|C_{max}$

It can be seen that case 13 is found to be the most similar to this problem. The representation of case 13 is: $F10|chain, d_k|C_{max}|CTR-PG, PMB|HC-SPT$. The found case is also a flow-shop, but with 10 machines instead of 5. This case contains also the deadline property, but also a chain precedence restriction. The objective function is the total makespan, as in the new case as well. The model is a $\delta = CTR-PG, PMB$ one, and the applied solving method is the shortest process time first heuristic ($\epsilon = HC-SPT$).

Reusing the case 13, the model and method proposed to solve $F5|d_k|C_{max}$ will be $CTR-PG, PMB|HC-SPT$. This is a possible and thus an acceptable solution to this new case.

Finally the solved new case can be stored in the case base as $F5|d_k|C_{max}|CTR-PG, PMB|HC-SPT$ for future reuse.

4.5. Conclusion

In this chapter the association between scheduling problems, mathematical models and corresponding solution methods has been studied.

The question of adaptation of a mathematical model is examined, and illustrated through a simple example. “Minor modification” of a mathematical model has been defined, i.e. a modification of a mathematical model which does not lead to the introduction of new binary variables or binary constraints. To illustrate the adaptation process by minor modifications the same model has been adapted to similar problems, and has been successfully solved with different solving methods. It has been concluded that a minor modification of a mathematical model does not concern significantly the difficulty level of solving.

In order to perform a comparative study on applicability, different mathematical models have been applied to the same problems, and the computational results have been analyzed. In the case of little problems no significant difference has been recorded in computational times. However, in the case of larger problems the

global time event based continuous formulation proved to be the most effective one to $F_n|cons$ CRA, network, $FIS_d|\Sigma P$ problems.

According to the time representation (δ_1) – despite there is no trivial “best” model to choose – the tendency seems to be advancing from discrete to continuous time based representations. The advantage of the latter ones is the more precise description of duration times and the better applicability in the case of variable (e.g. quantity dependant) duration times.

A methodology scheme has been established to obtain a resolution strategy to scheduling problems and an association strategy has been proposed which is summarized in eight rules. The application of this rule based strategy has been illustrated on three examples.

Despite some rules are possible to deduct, it became clear that there is no self-understanding resolution strategy to follow. Scheduling research is in dynamic evolution and scheduling problems are very rich in characteristics, which makes it very difficult to make general conclusions.

However, it can be stated that a simple rule-based association does not fit to this rich and large domain. The problem needs to be treated on a higher level, i.e. a dynamic decision-aid system has to be constructed which has the ability of learning. An Artificial Intelligence inspired approach has been constructed based on case-based reasoning (CBR).

The case-based reasoning system has been created using the COLIBRI software which is a tool developed to the purpose of easily creating CBR systems.

The initial efforts of applying a CBR based approach in order to find the adequate resolution strategy to scheduling problems show that the method can be well applied to the problematic. Even with a relatively small database the method seems to be able to identify the most similar problems and propose mathematical models and adequate solution methods. The retrieval process of this preliminary decision-aid software has been tested on a little $F5|d_k|C_{max}$ example problem with success: a possible and thus applicable resolution strategy has been proposed by the software.

4.6. References of Chapter Four

- Esquirol, P.; Lopez, P. *L'ordonnancement*; ECONOMICA Paris, 1999.
- Epperly, T. G. W.; Ierapetritou, M.G.; Pistikopoulos, E. N. On the global and efficient solution of stochastic batch plant design problems. *Computers and Chemical Engineering* **1997**, *21* (12), 1411.
- Huang, K. L.; Liao, C. J. Ant colony optimization combined with taboo search for the jobshop scheduling problem. *Computers & Operations Research* **2008**, *35*, 1030.
- Ierapetritou, M. G.; Floudas, C.A. Effective continuous-time formulation for short-term scheduling 1. Multipurpose batch process. *Industrial and Engineering Chemistry Research* **1998**, *37*(11), 4341.
- Kondili, E.; Pantelides; C. C.; Sargent, R. W. H. A general algorithm for short-term scheduling of batch operations. Part I: MILP formulation. *Computers & Chemical Engineering* **1993**, *17*, 2211.

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

Maravelias, C. T.; Grossmann, I. E. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Industrial and Engineering Chemistry Research* **2003**, *42*, 3056.

Nearchou, A. C. A novel metaheuristic approach for the flow shop scheduling problem. *Engineering Applications of Artificial Intelligence* **2004**, *17*, 289.

Pan, M.; Li, X.; Qian, Y. Continuous-time approaches for short-term scheduling of network batch processes: Small-scale and medium-scale problems. *Chemical Engineering Research and Design* **2009**, *87*, 1037.

Rajendran, C.; Ziegler, H. Ant-colony algorithms for permutation flow-shop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research* **2004**, *155*, 426.

Satake, T.; Morikawa, K.; Takahashi, K.; Nakamura, N. Simulated annealing approach for minimizing the makespan of the general job-shop. *International Journal of Production Economics* **1999**, *60-61*, 515.

Shyu, S. J.; Lin, B. M. T.; Yin, P. Y. Application of ant colony optimization for no-wait flowshop scheduling problem to minimize the total completion time. *Computers & Industrial Engineering* **2004**, *47*, 181.

Sundaramoorthy, A.; Karimi, I. A. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chemical Engineering Science* **2005**, *60*, 2679.

Xhafa, F.; Abraham, A. *Metaheuristics for Scheduling in Industrial and Manufacturing Applications*. Springer Verlag, 2008.

4.7. Appendix

Appendix Table 1 – Data of Example 4-2

PROBLEM DATA			Working quantity	
<i>Job</i>	<i>Operation</i>	<i>Process time</i>	Min	Max
1	1	7	500	4500
1	2	8,3	500	4500
1	3	6	500	4500
1	4	7	500	4500
1	5	6,5	500	4500
1	6	8	500	4500
2	1	6,8	500	4500
2	2	5	500	4500
2	3	6	500	4500
2	4	4,8	500	4500
2	5	5,5	500	4500
2	6	5,8	500	4500
3	1	4	500	4500
3	2	5,9	500	4500
3	3	5	500	4500
3	4	6	500	4500
3	5	5,5	500	4500
3	6	4,5	500	4500
4	1	2,4	500	4500
4	2	3	500	4500
4	3	3,5	500	4500
4	4	2,5	500	4500
4	5	3	500	4500
4	6	2,8	500	4500
Products		Require	Price	
1		150	10	
2		150	8	
3		180	9	
4		160	10	

Appendix Table 2 – Data of Example 4-3

PROBLEM DATA			Working quantity	
<i>Job</i>	<i>Operation</i>	<i>Process time</i>	Min	Max
1	1	7	500	4500
1	2	8,3	500	4500
1	3	6	500	4500
1	4	7	500	4500
1	5	6,5	500	4500
1	6	8	500	4500
1	7	7,5	500	4500
1	8	6	500	4500
1	9	6,3	500	4500
2	1	6,8	500	4500
2	2	5	500	4500
2	3	6	500	4500
2	4	4,8	500	4500
2	5	5,5	500	4500
2	6	5,8	500	4500
2	7	6,2	500	4500
2	8	5,4	500	4500
2	9	6	500	4500
2	1	4	500	4500
3	2	5,9	500	4500
3	3	5	500	4500
3	4	6	500	4500
3	5	5,5	500	4500
3	6	4,5	500	4500
3	7	6	500	4500
3	8	5,5	500	4500
3	9	4,5	500	4500
4	1	2,4	500	4500
4	2	3	500	4500
4	3	3,5	500	4500
4	4	2,5	500	4500
4	5	3	500	4500

4	6	2,8	500	4500
4	7	3	500	4500
4	8	3,5	500	4500
4	9	2,5	500	4500
5	1	6,8	500	4500
5	2	5	500	4500
5	3	6	500	4500
5	4	4,8	500	4500
5	5	5,5	500	4500
5	6	5,8	500	4500
5	7	6,2	500	4500
5	8	5,4	500	4500
5	9	6	500	4500
6	1	7	500	4500
6	2	8,3	500	4500
6	3	6	500	4500
6	4	7	500	4500
6	5	6,5	500	4500
6	6	8	500	4500
6	7	7,5	500	4500
6	8	6	500	4500
6	9	6,3	500	4500
Products		Require	Price	
1		150	10	
2		150	10	
3		150	10	
4		150	10	
5		150	10	
6		150	10	

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

Appendix Table 3 – Cases of the case base

<i>Case n°</i>	<i>Scheduling problem</i>	<i>Mathematical model</i>	<i>Solution method</i>
1	Example 1-4	Disjunctive graph model	SPT heuristic
2	Example 1-4	Disjunctive graph model	LPT heuristic
3	Example 1-4	Disjunctive graph model	Ant Colony Optimization
4	Example 1-4	Disjunctive graph model	Taboo Search
5	Example 1-4	Disjunctive graph model	Simulated Annealing
6	Example 1-4	Disjunctive graph model	Genetic Algorithm
7	Example 1-4a	Disjunctive graph model	SPT heuristic
8	Example 1-4a	Disjunctive graph model	LPT heuristic
9	Example 1-4a	Disjunctive graph model	Ant Colony Optimization
10	Example 1-4a	Disjunctive graph model	Taboo Search
11	Example 1-4a	Disjunctive graph model	Simulated Annealing
12	Example 1-4a	Disjunctive graph model	Genetic Algorithm
13	Example 1-4b	Disjunctive graph model	SPT heuristic
14	Example 1-4b	Disjunctive graph model	LPT heuristic
15	Example 1-4b	Disjunctive graph model	Ant Colony Optimization
16	Example 1-4b	Disjunctive graph model	Taboo Search
17	Example 1-4b	Disjunctive graph model	Simulated Annealing
18	Example 1-4b	Disjunctive graph model	Genetic Algorithm
19	Example 1-4b	Disjunctive graph model	EDD heuristic
20	Example 1-4c	Disjunctive graph model	SPT heuristic
21	Example 1-4c	Disjunctive graph model	LPT heuristic
22	Example 1-4c	Disjunctive graph model	Ant Colony Optimization
23	Example 1-4c	Disjunctive graph model	Taboo Search
24	Example 1-4c	Disjunctive graph model	Simulated Annealing
25	Example 1-4c	Disjunctive graph model	Genetic Algorithm
26	Example 1-4d	Disjunctive graph model	SPT heuristic
27	Example 1-4d	Disjunctive graph model	LPT heuristic
28	Example 1-4d	Disjunctive graph model	Ant Colony Optimization
29	Example 1-4d	Disjunctive graph model	Taboo Search
30	Example 1-4d	Disjunctive graph model	Simulated Annealing
31	Example 1-4d	Disjunctive graph model	Genetic Algorithm
32	Example 1-4e	Disjunctive graph model	SPT heuristic
33	Example 1-4e	Disjunctive graph model	LPT heuristic
34	Example 1-4e	Disjunctive graph model	Ant Colony Optimization
35	Example 1-4e	Disjunctive graph model	Taboo Search
36	Example 1-4e	Disjunctive graph model	Simulated Annealing
37	Example 1-4e	Disjunctive graph model	Genetic Algorithm
38	Example 1-4f	Disjunctive graph model	SPT heuristic
39	Example 1-4f	Disjunctive graph model	LPT heuristic
40	Example 1-4f	Disjunctive graph model	Ant Colony Optimization
41	Example 1-4f	Disjunctive graph model	Taboo Search
42	Example 1-4f	Disjunctive graph model	Simulated Annealing
43	Example 1-4f	Disjunctive graph model	Genetic Algorithm
44	Example 1-4g	Disjunctive graph model	SPT heuristic
45	Example 1-4g	Disjunctive graph model	LPT heuristic
46	Example 1-4g	Disjunctive graph model	Ant Colony Optimization
47	Example 1-4g	Disjunctive graph model	Taboo Search
48	Example 1-4g	Disjunctive graph model	Simulated Annealing
49	Example 1-4g	Disjunctive graph model	Genetic Algorithm
50	Example 4-1	Kondili model	MILP solver
51	Example 4-1	Maravelias model	MILP solver
52	Example 4-1	Ierapetritou model	MILP solver
53	Example 4-1	Sundaramoorthy model	MILP solver

54	Example 4-2	Kondili model	MILP solver
55	Example 4-2	Maravelias model	MILP solver
56	Example 4-2	Ierapetritou model	MILP solver
57	Example 4-2	Sundaramoorthy model	MILP solver
58	Example 4-3	Kondili model	MILP solver
59	Example 4-3	Maravelias model	MILP solver
60	Example 4-3	Ierapetritou model	MILP solver
61	Example 4-3	Sundaramoorthy model	MILP solver
62	Example 1-1	Kondili model	MILP solver
63	Example 1-1	Maravelias model	MILP solver
64	Example 1-1	Ierapetritou model	MILP solver
65	Example 1-1	Sundaramoorthy model	MILP solver
66	Example 1-2	Kondili model	MILP solver
67	Example 1-2	Maravelias model	MILP solver
68	Example 1-2	Ierapetritou model	MILP solver
69	Example 1-2	Sundaramoorthy model	MILP solver
70	Example 1-3	Kondili model	MILP solver
71	Example 1-3	Maravelias model	MILP solver
72	Example 1-3	Ierapetritou model	MILP solver
73	Example 1-3	Sundaramoorthy model	MILP solver

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

Appendix Table 4 – Representation of cases

<i>Case n°</i>	<i>Representation using the notation system</i>
1	F10 chain C _{max} CTR-PG, PMB HC-SPT
2	F10 chain C _{max} CTR-PG, PMB HC-LPT
3	F10 chain C _{max} CTR-PG, PMB ACO
4	F10 chain C _{max} CTR-PG, PMB TS
5	F10 chain C _{max} CTR-PG, PMB SA
6	F10 chain C _{max} CTR-PG, PMB GA
7	F10 chain, est _k C _{max} CTR-PG, PMB HC-SPT
8	F10 chain, est _k C _{max} CTR-PG, PMB HC-LPT
9	F10 chain, est _k C _{max} CTR-PG, PMB ACO
10	F10 chain, est _k C _{max} CTR-PG, PMB TS
11	F10 chain, est _k C _{max} CTR-PG, PMB SA
12	F10 chain, est _k C _{max} CTR-PG, PMB GA
13	F10 chain, d _k C _{max} CTR-PG, PMB HC-SPT
14	F10 chain, d _k C _{max} CTR-PG, PMB HC-LPT
15	F10 chain, d _k C _{max} CTR-PG, PMB ACO
16	F10 chain, d _k C _{max} CTR-PG, PMB TS
17	F10 chain, d _k C _{max} CTR-PG, PMB SA
18	F10 chain, d _k C _{max} CTR-PG, PMB GA
19	F10 chain, d _k C _{max} CTR-PG, PMB HC-EDD
20	F10 chain, MWT C _{max} CTR-PG, PMB HC-SPT
21	F10 chain, MWT C _{max} CTR-PG, PMB HC-LPT
22	F10 chain, MWT C _{max} CTR-PG, PMB ACO
23	F10 chain, MWT C _{max} CTR-PG, PMB TS
24	F10 chain, MWT C _{max} CTR-PG, PMB SA
25	F10 chain, MWT C _{max} CTR-PG, PMB GA
26	F10 chain, ch _k C _{max} CTR-PG, PMB HC-SPT
27	F10 chain, ch _k C _{max} CTR-PG, PMB HC-LPT
28	F10 chain, ch _k C _{max} CTR-PG, PMB ACO
29	F10 chain, ch _k C _{max} CTR-PG, PMB TS
30	F10 chain, ch _k C _{max} CTR-PG, PMB SA
31	F10 chain, ch _k C _{max} CTR-PG, PMB GA
32	F10 chain, ch _{kk'} C _{max} CTR-PG, PMB HC-SPT
33	F10 chain, ch _{kk'} C _{max} CTR-PG, PMB HC-LPT
34	F10 chain, ch _{kk'} C _{max} CTR-PG, PMB ACO
35	F10 chain, ch _{kk'} C _{max} CTR-PG, PMB TS
36	F10 chain, ch _{kk'} C _{max} CTR-PG, PMB SA
37	F10 chain, ch _{kk'} C _{max} CTR-PG, PMB GA
38	FP20 chain C _{max} CTR-FLEX, PMB HC-SPT
39	FP20 chain C _{max} CTR-FLEX, PMB HC-LPT
40	FP20 chain C _{max} CTR-FLEX, PMB ACO
41	FP20 chain C _{max} CTR-FLEX, PMB TS
42	FP20 chain C _{max} CTR-FLEX, PMB SA
43	FP20 chain C _{max} CTR-FLEX, PMB GA
44	FP20 chain, ch ^U C _{max} CTR-FLEX, PMB HC-SPT
45	FP20 chain, ch ^U C _{max} CTR-FLEX, PMB HC-LPT
46	FP20 chain, ch ^U C _{max} CTR-FLEX, PMB ACO
47	FP20 chain, ch ^U C _{max} CTR-FLEX, PMB TS
48	FP20 chain, ch ^U C _{max} CTR-FLEX, PMB SA
49	FP20 chain, ch ^U C _{max} CTR-FLEX, PMB GA
50	F3 cons, CRA, network, FIS _d ΣP DTR, STN MILP
51	F3 cons, CRA, network, FIS _d ΣP CTR-GTE, STN MILP
52	F3 cons, CRA, network, FIS _d ΣP CTR-USTE, STN MILP
53	F3 cons, CRA, network, FIS _d ΣP CTR-SS, STN MILP

54	F6 cons, CRA, network, FIS _d ΣP DTR, STN MILP
55	F6 cons, CRA, network, FIS _d ΣP CTR-GTE, STN MILP
56	F6 cons, CRA, network, FIS _d ΣP CTR-USTE, STN MILP
57	F6 cons, CRA, network, FIS _d ΣP CTR-SS, STN MILP
58	F9 cons, CRA, network, FIS _d ΣP DTR, STN MILP
59	F9 cons, CRA, network, FIS _d ΣP CTR-GTE, STN MILP
60	F9 cons, CRA, network, FIS _d ΣP CTR-USTE, STN MILP
61	F9 cons, CRA, network, FIS _d ΣP CTR-SS, STN MILP
62	FR10 cons, CRA, chain, EE, est _k , P _S , d _k , MWT, ZWT, ch ^{kk} , con, fix C _{max} DTR, STN MILP
63	FR10 cons, CRA, chain, EE, est _k , P _S , d _k , MWT, ZWT, ch ^{kk} , con, fix C _{max} CTR-GTE, STN MILP
64	FR10 cons, CRA, chain, EE, est _k , P _S , d _k , MWT, ZWT, ch ^{kk} , con, fix C _{max} CTR-USTE, STN MILP
65	FR10 cons, CRA, chain, EE, est _k , P _S , d _k , MWT, ZWT, ch ^{kk} , con, fix C _{max} CTR-SS, STN MILP
66	JP4 cons, CRA, network, CC, P _S , FIS _d , ZWT ΣP DTR, STN MILP
67	JP4 cons, CRA, network, CC, P _S , FIS _d , ZWT ΣP CTR-GTE, STN MILP
68	JP4 cons, CRA, network, CC, P _S , FIS _d , ZWT ΣP CTR-USTE, STN MILP
69	JP4 cons, CRA, network, CC, P _S , FIS _d , ZWT ΣP CTR-SS, STN MILP
70	F6 cons, CRA, chain, P _S , d _k , FIS _d , LW, overlap, con, fix C _{max} DTR, STN MILP
71	F6 cons, CRA, chain, P _S , d _k , FIS _d , LW, overlap, con, fix C _{max} CTR-GTE, STN MILP
72	F6 cons, CRA, chain, P _S , d _k , FIS _d , LW, overlap, con, fix C _{max} CTR-USTE, STN MILP
73	F6 cons, CRA, chain, P _S , d _k , FIS _d , LW, overlap, con, fix C _{max} CTR-SS, STN MILP

CONCLUSIONS AND PERSPECTIVES

CONCLUSIONS

This work is focused on the application of Case Based Reasoning (CBR) in order to propose resolution strategies for solving scheduling problems. Firstly, the importance and the role of scheduling problems in industry have been presented. The basic definitions and terminology of this area have been illustrated by three illustrative examples. Through these examples, the complexity of the solution process has been shown as well. The resolution methods of scheduling problems have been briefly discussed. Because of the high interest shown by scheduling community to mathematical solving approaches, the research has been restricted to this area. However, due to the wide complexity of the resolution process, it is reasonable and necessary to apply a decision support system in order to facilitate the choices made by the expert during the resolution strategy. This decision support system is based on methods of Artificial Intelligence. Among the Artificial Intelligence methods the Case-Based Reasoning has been chosen as the most appropriate candidate and detailed. Previously solved cases (i.e. problem-model-method triplets stored in a case memory) will be used to propose solutions (i.e. model and method) to a faced scheduling problem. To establish such a case memory it is necessary to extract the existing knowledge on the domain. In order to collect and extract the existing expert knowledge and study the evolution of scheduling research we have decided to perform a scientific bibliographical analysis. The methods of bibliographical analysis have been introduced, and a co-citation based method has been chosen, namely the Ensemble NMF algorithm of Greene (2008).

Usually, a dynamic and strong research tendency has been found which indicates further research work in scheduling problems area. 22 clusters have been obtained by the clustering algorithm. Two approaches have been introduced in order to identify clusters: the word frequency analysis based on title, abstract and keywords and the word frequency analysis based on the full text of the available papers. The application of both approaches has been illustrated on some illustrative examples: the attributes “resource consideration” and “deadline” have been associated to the most appropriate clusters with applying the first approach, and an example cluster has been successfully labeled based on the second approach.

In order to work out an efficient case representation system in the CBR a classification of scheduling problems, mathematical models, and adapted solution methods has been proposed. This classification scheme is based on the works of Blazewicz et al. (2007) and Mendez et al. (2006). A notation scheme has been proposed to the classification as well. The core of this notation scheme is the notation scheme of Graham et al. (1979) and Blazewicz et al. (1983), which has been completed with the following aspects and properties: resource type and availability, synchronization constraints, presence of closed cycle, machine availability, quantity and resource dependency of process times, capacity constraints, limited and minimal waiting time, no-idle constraints. The field describing machine environment (α) has been reformulated from a workstation-based point of view. New aspects of classification have been added to the field describing resources and constraints (β): transit time,

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

changeover time, overlap, maintenance and preparation constraints, connectivity restrictions, batch size considerations. A detailed classification of objective functions has been introduced in the field of objective functions (γ). The notation system has also been extended to mathematical models of scheduling problems by a field δ , and to the solving methods by a field ϵ .

The application of these new classification and notation scheme has been shown on illustrative examples.

Due to the hugeness of the research domain the research on elaborating an assignment strategy has been restricted to flow-shop problems.

Finally, the resolution strategy of flow-shop scheduling problems by means of mathematical approaches has been studied. As a first step, a simple rule-based association has been proposed. However, scheduling research is in dynamic evolution and scheduling problems are very rich in characteristics, and therefore a simple rule-based association does not fit the expectations. The problem needs to be treated in a higher level, i.e. a dynamic system is to be constructed which has a learning ability. An extension of these rules has been proposed, creating the foundations of a CBR system.

The initial efforts of applying a CBR based approach to scheduling problems show that the method can be successfully applied. Even with a relatively small database the method was able to find similar cases to the problem and efficiently solve it. Mathematical models and adequate solution methods could also be proposed by a simple reuse strategy without special adaptation.

PERSPECTIVES

This study on the applicability of CBR-based approaches on scheduling problems gives the opportunity to investigate further research efforts in this area. There are several perspectives of this research work.

Firstly, concerning the bibliographical analysis in Chapter Two, a comparative study on the available bibliographical methods, evaluating the differences between the results would be an interesting perspective. It is also desirable that the analysis be completely automatized.

Next, from the point of view of CBR there are three main perspectives for future research investigation:

1. Extension of the case base

The case base can be expanded in order to improve the effectiveness of the CBR cycle. The analysis of Chapter Four has been restricted to flow-shop problems. The application of the performed analysis to all the resource and constraint characteristics of flow-shops and the

extension of the study to other types of workshops (especially to job-shops, as they are frequently faced in chemical industry) is therefore an important step forwards.

Supplementary information about the cases can be stored too in order to classify the obtained solutions with respect to their quality (e.g. user provided value on a satisfaction scale 0-10, etc...).

2. Elaboration of an adaptability measure

Most frequently in case retrieval processes the previously solved case is retrieved based on a similarity measure. However, the most similar case is not necessarily the most adaptable. For example: Suppose that the CBR system uses Euclidean distance based similarity function without weighting (see Chapter One). Suppose that our case memory contains a resolution strategy $RS1 = F5|cons, CRA, chain|C_{max}|CTR-SS, STN|MILP$ and a resolution strategy $RS2 = F8|chain, est_k, mwt_k, ch_k|C_{max}|CTR-PG, PMB|ACO$. To a new problem $\alpha|\beta|\gamma = F10|cons, CRA, chain, est_k, mwt_k, ch_k|C_{max}$ the similarity function provides $RS2$, and therefore the CBR proposes a $\delta = CTR-PG, PMB$ model which is not adaptable due to the presence of resources. On the other hand, a well developed adaptability measure takes into account that $\beta_2 \neq \emptyset$ and consequently excludes the cases where $\delta_2 = PMB$.

The importance of the field β_2 may be obvious for an expert, and can be taken into account by adequate weighting of the similarity function as well, but for a non-expert an adaptability measure permits to avoid a potential source of difficulties.

3. Development of an adaptation method

This work focused on the retrieval part of the CBR cycle. There are, however, interesting further possibilities in the other parts, e.g. the case adaptation. The adaptation is a crucial step of CBR and is difficult to elaborate, because it requires an important, domain-specific expert knowledge. Important research efforts have been made in order to reduce the complexity of adaptation process, e.g. introduction of the above discussed adaptability, or significant efforts on similarity function. Nevertheless, the adaptation remains a critical part with respect to the effectiveness of the CBR cycle. Adaptation methods can be regrouped into three categories: unifying attempts, knowledge acquisition and combinations of the former two (also referred as knowledge catalogs). The former one attempts to propose general adaptation models (principles, algorithms) and it is difficult to apply for a domain flexible and rich in characteristics like scheduling problems. However, knowledge acquisition is an interesting adaptation strategy, and may be suitable for our domain if it is completed with an interaction cycle with the expert. If this interaction cycle is activated, then the solution has not been accepted by the user. That is, firstly the reasons of the failure have to be identified. The solution proposed by the CBR is thus sent to an expert of the

Study on Application Possibilities of Case-Based Reasoning on the Domain of Scheduling Problems

domain for performing the necessary corrections. Then, the solution found and commented by the expert is proposed to the user. Accepted by the user, the solution is stored as adapted solution in the case memory, otherwise the interaction cycle is reactivated.

The meta-heuristics have been applied, in this study, without parameter adjustment strategy. Therefore a study on the determination of the optimal parameters of these methods is an important perspective. In addition, the optimal parametric values could be encoded to the CBR information as well, in order to propose adequate parameter values if a meta-heuristic solving method is proposed. In order to collect the required information, a sensitivity analysis on the parameters of meta-heuristic methods is to be performed.