Open Archive Toulouse Archive Ouverte (OATAO)

To cite this version :

Case-Based Reasoning system for mathematical modelling options and resolution methods for production scheduling problems: Case representation, acquisition and retrieval

Tibor Kocsis [a,b], Stéphane Negny [a,*], Pascal Floquet [a], Xuân Meyer [a], Endre Rév [b]

[a] *Université de Toulouse, LGC UMR 5503 – INPT ENSIACET, 4 allée Emile Monso, BP 44 362, 31030 Toulouse Cedex 4, France*
[b] *Budapest University of Technology and Economics, Department of Chemical and Environmental Process Engineering, Budafoki út 8, 1111 Budapest, Hungary*

ABSTRACT

Thanks to a wide and dynamic research community on short term production scheduling, a large number of modelling options and solving methods have been developed in the recent years both in chemical production and manufacturing domains. This trend is expected to grow in the future as the number of publications is constantly increasing because of industrial interest in the current economic context. The frame of this work is the development of a decision support system to work out an assignment strategy between scheduling problems, mathematical modelling options and appropriate solving methods. The system must answer the question about which model and which solution method should be applied to solve a new scheduling problem in the most convenient way. The decision support system is to be built on the foundations of Case Based Reasoning (CBR). CBR is based on a data base which encompasses previously successful experiences. The three major contributions of this paper are: (i) the proposition of an extended and a more exhaustive classification and notation scheme in order to obtain an efficient scheduling case representation (based on previous ones), (ii) a method for bibliographic analysis used to perform a deep study to fill the case base on the one hand, and to examine the topics the more or the less examined in the scheduling domain and their evolution over time on the other hand, and (iii) the proposition of criteria to extract relevant past experiences during the retrieval step of the CBR. The capabilities of our decision support system are illustrated through a case study with typical constraints related to process engineering production in beer industry.

1. Introduction

This research is focused on the issue of capitalizing simulation modelling knowledge to efficiently develop relevant models for short term scheduling applications. Indeed, the supply chain of a company is a complex network which involves the integration of information, transportation, procurement of raw materials, inventory, transformation into finished products, warehousing, material handling and distribution of finished products to end users. The goal is to reach a high end user satisfaction level at a low cost.

One of the main functions in the supply chain is the production which aims to use available production capacities to produce the desired products. The coordination of the production is one way to achieve high efficiency with low cost. This can be done through the production planning. But planning refers to a wide range of activities with different decision levels and different time scales. Among them, scheduling is a crucial step which is a short term planning dealing with the allocation of resources to tasks (assignments and sequencing of tasks to units) over time with one or various objectives to optimize.

However, the growing worldwide competition in the current context imposes new industrial strategies based on more and more flexible processes affording a greater reactivity to remain competitive in the global marketplace. Indeed, for the manufacture of chemicals or materials, the production process or the demand pattern is likely to change. The inherent operational flexibility of industrial plants provides the platform for great saving in good

---

production schedules as it is the core of production management. This flexibility is increased because of the demand for greater product customization and diversification. As a consequence production processes become more complex with multi products or multipurpose plants with batch mixing/splitting, multi units, multi recipes for products constructions and an increasing set of production constraints. 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). Furthermore, the application of process engineering towards new areas, such as food, biotechnological, electronic or pharmaceutical industries is generating new production and scheduling problems with additional resource constraints, batching decisions, process restrictions, handling mixing and splitting streams. To address all of these scheduling issues, the process system engineering community has developed several models, and resolution methods.

The research area on scheduling has been broadly studied by both the industry and the academia world resulting in significant advances in relevant modelling and solution techniques. Numerous research studies have been made of this area, e.g. (Blazewicz, Ecker, Pesch, Schmidt, & Weglarz, 2007; Esquirol & Lopez, 1999; Maravelias, 2012). Scheduler show inventiveness to propose new modelling options and improved mathematical methods to address to these complex highly combinatorial problems. The increasing number of research articles dedicated to short term scheduling problems bears out this trend. Accordingly, plenty of possibilities of association between modelling options, and between models and resolution methods are thus available. More and more difficult and larger problems than those studied years ago can be now solved, sometimes even to optimality in a 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. This results in different possibilities to model a scheduling problem, but also multiple mathematical formulations for the same model. The diversification of modelling options, the combination and creation of resolution methods are increasing and will going to grow in the following years in order to enlarge the classes and the size of the problems treated. For instance in Sundaramoorthy and Maravelias (2008) the number and size of batches are now included as decision variables (material based approach) in a job shop problem whereas before they were fixed (batch based approach).

To show the richness of the modelling options developed to deal with the increasing complexity of problems, we call reference here to the well known and widely used example: the chemical process problem proposed by Kondili, Pantelides, and Sargent (1993). As shown in Fig. 1, three raw materials (A, B and C) are required, three intermediates and two final products are produced through five stages: heating, reactions 1 3, and separation.

In their work Kondili et al. (1993) have solved this problem by applying a discrete time representation model with the following assumptions: process times are invariable and independent of size, products of the same task can arrive in different times, all transit/changeover times are included into process times or neglected. Other models have been successfully applied to the same problem by Maravelias and Grossmann (2003) and Ierapetritou and Floudas (1998) with the assumptions that products of the same task arrive in the same time, and process times can be size dependant. They applied a global and a unit specific time event based model respectively. Pan, Li, and Qian (2009) reused the Kondili problem and applied six different models successfully. They also tested the problem with different objectives: makespan minimization and profit maximization respectively. They performed a comparative
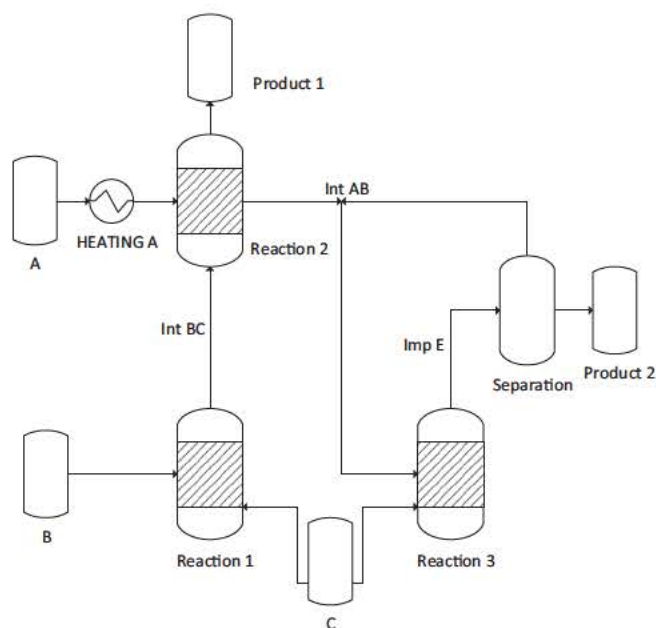

Fig. 1. Example of Kondili et al. (1993).

analysis of this example, showing that numerous models are available to the same problem, without a unique "best one".

Table 1 summarized some of the above research studies but the list is far from being exhaustive, not mentioning yet the different possible solving methods: Hegyhati and Friedler (2011) precise that most of the published approaches are based on a mixed integer programming formulation and they analyze the combinatorial nature of batch scheduling problems. Therefore, this mere example underlines that in order to choose between modelling options and solving methods strategies, we need a decision support system, especially as the process and manufacturing industries gather a wide range of applications leading to a variety of processing characteristics and constraints to take into account. Accordingly, the number of research papers has increased to develop new model options and numerical methods to account for these specific constraints, reinforcing the need for a decision support system. The goal of this decision support system is to help user in choosing the modelling options and the appropriate solving methods thanks to a detailed description of the faced scheduling problem. But in front of the difficulty to build such a system and the huge interest of the process engineering community to mathematical approaches, in the rest of the study we voluntary limit this work to a decision support dedicated to mathematical approaches.

Zhou, Son, Chen, Zhang, and MA (2007) have explained that the model development is a time consuming and knowledge intense activity that require skills from three different fields: domain expertise, modelling and simulation, and implementation. For the development of models, Meyer (2004) has formalized a process commonly used in process system engineering (PSE), Fig. 2. This process clearly demonstrates that the development of a model is an activity which relies on the skills and experiences of a working group composed of experts with diverse background and knowledge: domain of application (for instance physics, chemistry, biology), PSE, computer science. Indeed, to facilitate the resolution it is often necessary to realize a preliminary work for structuring the system of equations or to give it a specific form to easy initialization and have a stable, accurate and robust resolution. Moreover, as Zhou, Chen, He, and Chen (2010) have underlined most of simulation models developed are often customized and specific ones

Table 1
Different solution strategies to the Kondili-problem.

Paper	Assumptions	Model (type)
Kondili et al. (1993)	Process times are invariable and independent from size Products of the same task can arrive in different times No transit/changeover, but sequence/frequency dependant cleaning are considered	Discrete time representation, based on time intervals of equal lengths (MILP)
Maravelias and Grossmann (2003)	Process times depend on the quantity of material Sequence dependent changeover times Utility constraints are considered	Continuous time representation based on global time events (MILP)
Ierapetritou and Floudas (1998)	Process times depend on the quantity of material No transit/changeover, but clean-up requirements and multiple due dates are considered	Continuous time representation based on unit-specific time events (MILP)
Sundaramoorthy and Karimi (2005)	Storage and idle waiting are considered as special tasks	Continuous time representation, based on time intervals (slots) of variable lengths (MILP)
Pan et al. (2009)	Profit maximization and makespan minimization are considered	Six different continuous time representation models (MILP)

Process for Model Development

1- Identification of the principal purpose of the modeling
2- Characterisation of the system studied: kinetics, thermodynamic...
3- Identification of the principal phenomenon (limiting phenomenon...)
4- Selection of a theorical basis: model assumptions
5- Equations formulation
6- Degree of freedom analysis
7- Choice of the resolution method
8- Parameters estimation
9- Model validation with experimental data
10- Model documentation
11- Model implementation in a simulation environment
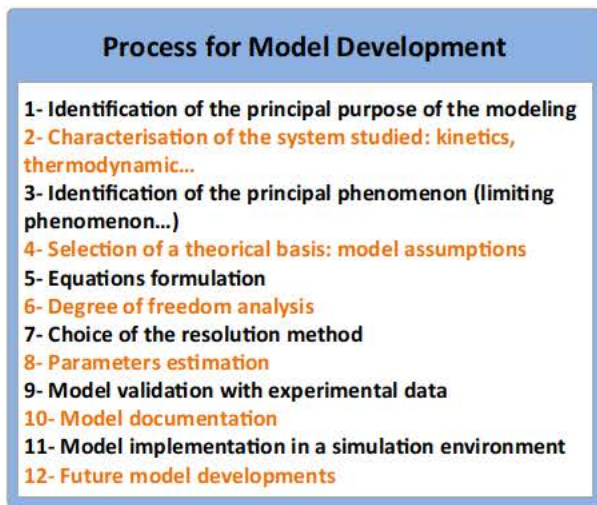12- Future model developments

Fig. 2. Model development.

but PSE experts try to develop generic models that can be easily reused and/or adapted. Consequently PSE experts continuously propose new inventive modelling options and solving methods to increase reusability and to deal with the increasing complexity of the problems treated.

A model is richer in knowledge than the one expressed through the system of equations. This knowledge is not always clearly express but is crucial to reach relevant model and solutions. The challenge is to ensure knowledge engineering in order to reuse well known and optimized past experiences to increase quality of solution and modelling decision, decrease the time of model development. Besides, in model development knowledge and skills of experts are difficult to formalize and capitalize because of their unstructured nature. However, some AI methods seem to be appropriate to construct such a decision aid system. Among AI methods for knowledge management, we decide to use Case Based Reasoning (CBR) to capitalize and reuse past experiences of PSE expert because of its ability and facility for knowledge formulation, knowledge acquisition, and knowledge maintenance.

In order to elaborate such a decision support system the existing knowledge has to be extracted, modelled, adapted, diffused, maintained and actualized. Some Artificial Intelligence (AI) methods were developed to manage knowledge used and deployed in a domain and to provide assistance to a process engineer in the development and the resolution of short term scheduling models. Accordingly, the purpose of this article is twofold. First it presents the basis of a decision support system to propose relevant and suit

able modelling options and resolution method for scheduling problem. This implies the development of a classification scheme of the modelling approaches to describe general problem. Consequently, the second purpose deals with the creation and operation of a past experiences memory to solve new problems.

The remainder of this article is structured as follows: the second part presents and discusses the existing AI approaches and especially CBR to deal with a computer aided system. In this part a classification and a notation are proposed to represent a scheduling problem and its associated solution in terms of scheduling options and solution methods. In the subsequent part, the issue of case base filling is discussed and a method for bibliographic analysis is proposed in order to extract relevant research past experiences. Part 4 deals with the problem of case retrieval and more precisely the similarity measurement and introduces the concept of adaptability. Before to draw the conclusion, in part 5, a case study related to beer production is proposed to illustrate the main steps of the approach.

2. Basic concepts of the decision-support system

2.1. Artificial Intelligence approaches in scheduling

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. The general desire of AI approaches is to make use of past experiences, and every knowledge based system tries to record and reuse an earlier episode where a problem was totally or partially solved. Most of AI approaches encapsulate knowledge gained from human experts and apply it automatically to make decisions. The process of acquiring expert knowledge and to manage it requires considerable skills to perform successfully. Among AI approaches, expert systems imitate human reasoning, considering it as being decomposable into elementary steps. An expert system is made up of a base of rules and a base of facts regrouping the properties that are "true"; condition and a consequence part (IF THEN rules). 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. Additionally, the knowledge applied to scheduling problem does not seem to really fit to a binary schema such as the "simple" production rules. Expert systems sound also inappropriate because of its difficulty to manage and maintain complex rules.

There were also attempts with neural networks. The goal is not to imitate the human reasoning but the capabilities to learn of the human brain. The "knowledge" is stored in the connection weights. The complexity of the decision process makes it difficult to build a sufficiently complex neural network to model the resolution strategy too.

CBR is an alternative to rule based systems. CBR tries to find a solution to a given problem with the help of the solution of a similar problem, solved in the past. In this approach the central element is a case, which represents a contextual experience composed of the problem description, the solution description and the environment of a problem. Numerous cases are stored in a case memory, i.e. case base. Then, when a new problem is met, the solution of a retrieved case is adapted in order 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 to solve the faced problem.
- To be able to adapt the retrieved solution.
- To store enough cases in the memory to have the maximal coverage of the problems and solutions spaces to ensure CBR efficiency.

These AI approaches are more appropriated to model local knowledge in order to imitate the human behaviour to make choice on more efficient methods, e.g. constraints programming. Accordingly, these methods are mainly used to justify priority choices in a specific context or to set some general variables of the scheduling problem. But they are also used to create a complete solution to a scheduling problem.

As explained before, due to current mathematical, numerical and computer evolutions there is a global trend to develop and solve Mixed Integer Linear Programming (MILP) and Mixed Integer Non Linear Programming (MINLP) models for scheduling problems both in manufacturing and in chemical processes communities. In this context, AI approaches can also be used as method to create a decision aided system dedicated to the first steps of the model elaboration: assumptions, time representation, objective function, numerical methods. To create such a decision aided system, CBR is relevant because it deals with a symbolic representation while neural networks used numerical one. In CBR systems this task requires significantly less knowledge acquisition effort since it searches to collect a set of past experiences without trying to formulate a domain model from these ones. CBR is also suitable because numerous modelling options become recurrent and past experiences can easily be reused to reduce significantly the model elaboration. Moreover, CBR has the advantage to make knowledge easily accessible, understandable and reusable.

In the literature the application of CBR in scheduling starts with the works of Miyashita (1995) and Schmidt (1996), Schmidt (1998) who tried to find a solution for a scheduling problem assigning jobs over time to machines and possibly additional resources. Despite these works gave the first building blocks for a CBR system, they were limited in terms of practical application. With the same aim to elaborate a complete schedule but for project, Dzeng and Lee (2004) proposed a generalized framework to represent schedule knowledge to analyze project scheduling and to give corrective advise on a potential errors. Dzeng and Tommelein (2004) developed a tool to help project scheduler to retrieve and reuse parts of existing schedule to generate new one. More recently Mikulakova, König, Tauscher, and Beucke (2010) developed a knowledge system based on CBR for project schedule generation but they go deeper by including an evaluation module to help the choice among alternative.

Another way to use CBR in production scheduling is to find the promising sequence of jobs processing as in the work of Dong and Kitaoka (1994). Priore, de la Fuente, Puente, and Parreño (2006) compared CBR and inductive learning and back propagation neural networks to extract the "best" dispatching rules to dynamically schedule jobs in flexible manufacturing systems. Chang, Hsieh, and Liu (2006) have presented a genetic algorithm and CBR hybridization for a single machine with release time to minimize the total weighted completion time. When a new problem emerges, the CBR retrieved cases that are used to be part of the initial population and injected during generations to the pool of chromosomes in the genetic algorithm.

CBR is also used to the parameterization of metaheuristics for the resolution of dynamic scheduling problems because parameter tuning is not obvious. As the value for metaheuristic parameters depend mainly on the problem, the search time to solve it, the required quality of the solution, Pereira and Madureira (2013) have established a learning module, based on CBR, for an autonomous parameterization.

2.2. Case Based Reasoning

Different models were proposed to represent the various sequential steps (knowledge representation, knowledge reasoning, knowledge interpretation and reusing) of the CBR process: (Allen, 1994; Hunt, 1995; Leake, 1996). Currently the R^5 cycle proposed by Finnie and Sun (2003) is commonly accepted, Fig. 3. This cycle is an extension of the R^4 model introduced by Aamodt and Plaza (1994).

In the CBR cycle, once the new problem described in the Representation step, the most similar cases to the new problem specifications are retrieved from the case base with the help of a similarity function. The Reuse step is the copy or the modification of the solution of the retrieved cases with the aim to solve the initial problem. The Revision step is the adaptation of the reused case to withdraw the remaining discrepancies. The Retain step is the incorporation of the new case into the existing case base once it has been confirmed or validated (Pal & Shiu, 2004). Each of these steps involves a number of more specific and complex sub processes, for instance retain implies: integrate (return problem, update general knowledge, and adjust indexes), index (generalize
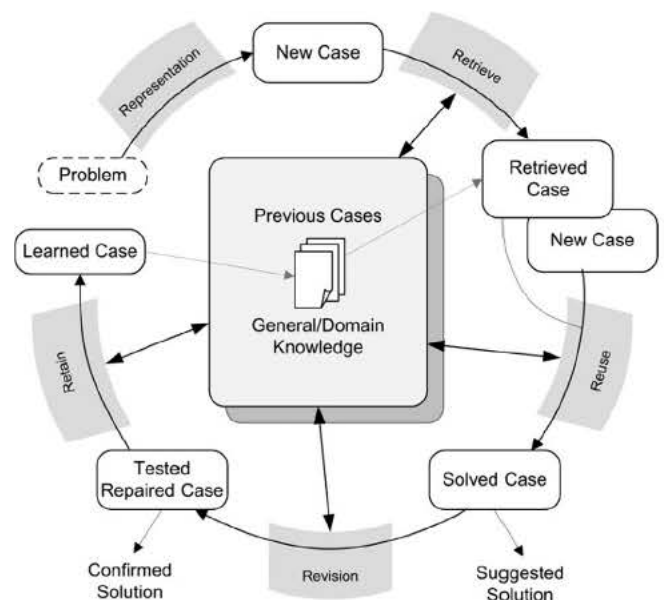

Fig. 3. Case-based reasoning cycle.

indexes, determine indexes), extract (extract solution method, extract justification, extract relevant descriptor, and extract solution). To develop a CBR system it is necessary to find an efficient solution to the following issues: representation of a case, retrieval of the most similar case, reusing and revision of an existing solution. Adaptation and storage are two other questions coming from the three above.

In the literature review no paper aims to use of CBR for scheduling model development, consequently we have to define the CBR architecture and all the different steps of the CBR cycle. The rest of this part is mainly focused on the key milestone of case representation for scheduling problems as the remainder of the CBR cycle is strongly dependent on this step. Case representation is the task of enabling the system to recognize, store, and process past contextualized experiences. Traditional case representation methods can be categorized into three groups (Pal & Shiu, 2004): the relational, the object oriented and the predicate based approaches. In our CBR system the cases are structured with attributes values pairs by using the object oriented representation because of its compact case representation ability, and the associated software reusability. Once the case representation method selected, the next step is to extract the relevant attributes to fill in order to describe a case.

2.3. Toward a general classification scheme

In the introduction, we underline that there is a dynamic research activity on short term scheduling problems both in the process system engineering and manufacturing production communities. Of course, there are obvious similarities between problems belonging to both communities. In discrete chemical production, when a batch maintains its identity through the whole production process, and it is transformed in a sequential way, then the problem can be addressed, modelled and solved with approaches developed in the manufacturing community (and inversely). However each community has its own specificities. The major difference is that in chemical engineering batch mixing and splitting are often encountered, this implies that the batches do not preserve their identity and network processing arises (Fig. 1 for example). Although, the PSE community has developed two evolutionary representations of network processes in order to account for material balances, i.e. the State Task Network (STN) and the Resource Task Network (RTN).

Besides, even if manufacturing and chemical processes share numerous objectives functions and constraints, like changeovers or adjustment of devices, some of them are more specific or predominant in one community rather than in the other. In manufacturing some researches deal with overlaps, availability or maintenance constraints that are less addressed in the PSE literature. In PSE, the production plant often needs utilities or specific auxiliary units (storage for example), connectivity restrictions and cleaning constraints. Moreover as recently noted in the analysis of Maravelias (2012), in process production scheduling it remains some ambiguities in the terms and concepts used. He gives two revealing examples:

Multipurpose has been used to describe workshops where batch mixing and splitting are allowed but also for product specific sequence of operation and with mixing and splitting.
Order and batch are used interchangeably while they do not refer to the same thing.

Consequently, the first step of the approach was to clarify the terms and concepts, for this we relied on the work of Maravelias (2012). Despite significant differences, we have to propose a unified classification which can be applied to problems arising in both communities but flexible enough to account for specificities for

each one, as they regularly proposes breakthroughs on modelling options and resolution methods. To address this problem of case representation, we must propose a standardized classification scheme meeting the following requirements; to be compatible with the existing one, to be used in our CBR system, to be available for both process and manufacturing scheduling, and easily understandable by users.

Graham, Lawler, Lenstra, and Rinnooy Kan (1979) had proposed such a notation system composed of three fields $\alpha|\beta|\gamma$ which respectively denote the machines environment, resource and task characteristics, and objective function. Blazewicz, Lenstra, and Rinnooy Kan (1983), Blazewicz et al. (2007), have expended this classification and proposed a systematic notation to give a basis for a classification scheme. The general notation system of Graham and Blazewicz had greatly facilitated the description and discussion of scheduling problems (Brucker, Drexl, Möhring, Neumann, & Pesch, 1999; Kutil, Sucha, Capek, & Hanzalek, 2010). In process system engineering, the definition of the constitutive elements of the traditional triplet was enlarged by Maravelias, 2012 to define class of problem encountered in this domain, Fig. 4. In his work, he changes the scope of α which denotes production environment rather than machines. As explain before β encompasses some specifics characteristics but also multiple entries. Concerning γ, it is only enlarged to include new criteria.

Based on these previous researches, we propose to rehash and expand this classification. In the proposed decision aided system, the existing triplet $\alpha|\beta|\gamma$ allows to list the relevant features to fill to describe in details the problem part of a case. It borrows and improves the lists proposed by Blazewicz, Lenstra, and Rinnooy Kan (1983), Blazewicz et al. (2007), Pinedo (2008), Mendez, Cerdá, Grossmann, Harjunkoski, and Fahl (2006) and Maravelias (2012). Indeed the previous studies have identified some processing constraints and characteristics but most are specific to chemical process production. Furthermore even inside each specific community, the description of the traditional fields α and β are not exhaustive.

2.4. Classification and notation system: application to case representation

The goal of this part is two folds. First, it presents a general and unified classification available for scheduling problem and modelling options both in the manufacturing and chemical productions domains. Second these classification and notation will be used as a framework for case representation in our CBR system.

By adding the solution part to our case description, the aforementioned triplet is enlarged by completing the three initial fields with the fields $\delta|\varepsilon$ which respectively detail the features to describe the modelling options for a scheduling problem and the solving method. As a result, the proposed classification scheme is
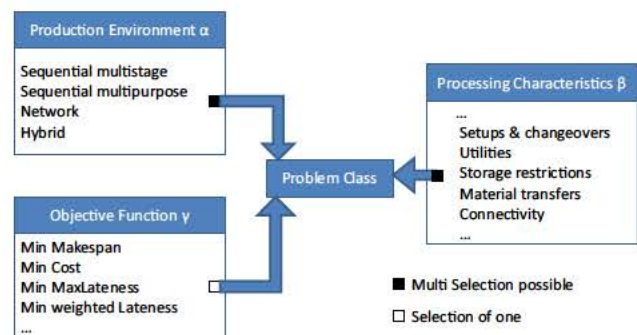

Fig. 4. Problem classes in chemical production scheduling Maravelias (2012).

composed of five fields: $\alpha|\beta|\gamma|\delta|\varepsilon$, where the first three allows to describe the problem part and the last two the solution part of a case.

2.4.1. Problem description: $\alpha|\beta|\gamma$

The field α is decomposed into two sub fields; α_1 for machines environment and α_2 for the number of workstations. In the sim plest case, each workstation gathers only a single machine but in flexible workshop, workstations can encompass several parallel machines. In accordance with the previous notation, the machines environment can be described with the following elements:

Single machine: all the jobs have to be realized in one operation on the same machine or unit. This simple case is a priori, far from industrial concerns but it has its practical importance when all the problems are concentrated on the same bottleneck machine.

Parallel machines: in a workstation several machines are avail able to perform operations simultaneously. We can distinguish three classes of problems depending on the processing time on the machines. If the processing time of an operation is the same for all the machines in the workstation, the machines are called identical (*noted P*). If the processing time can be related to the machine performance with an efficiency factor, we speak about uniform machines (*noted Q*). If the processing time of an opera tion varies independently from machine to machine we speak about independent machine (*noted R*). It is to notice that com plexity of problems goes increasing from P to R.

The next category of problems includes workshops with m dif ferent workstations. The production is divided into elementary operations each one being executed on a single machine belonging to one workstation. According to the recipe of the production, three classes of workshop can be distinguished:

Flow Shop (noted F): jobs visit the same set of workstations besides the sequence is the same for all of them (unidirectional flow). Flow Shops are also called multi product batch plant.
Job Shop (noted J): jobs go 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 the scheduling literature this type is referred as multipurpose batch plant.
Open Shop (noted O): for Flow Shop and Job Shop problems the sequences are known and immutable. But when the operations of a job can be done in any order, we speak about walk free, or open shop problem. The predetermined routing and the recipe are relaxed.

When several machines are available in workstations we talk about *hybrid functionality*, or flexible plant. It is the combination of a m different workstations workshop (F, J or O) with the three classes of parallel machines environment (P, Q and R).

The new notation scheme has to account for all these different flow possibilities, illustrated in Fig. 5. Accordingly, 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: F, J, O or single) from field α_{1b} (corresponding to the machines in a workstation). The reason of this discrimination is to be able to extend the definitions of flow shop, job shop and open shop to workstations, rather than to machines.

The second aspect of the classification deals with resource and processing constraints, field β. The notation system of Blazewicz et al. (2007), is modified and completed to separate and to detail the resources and processing constraints. Consequently, this field

is enlarged and the main types of functionality constraints and the corresponding notation are shown in Fig. 6.

These main classes are decomposed into sub classes in order to precisely describe the faced problem. For instance, β_8 details inven tory considerations for intermediate or final products, Fig. 7. In the Graham Blazewicz notation β_8 only describes a no wait property which can be Ø 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 con sumption times of raw materials, intermediates or products usu ally induce important limitations on storage that have to be included in the new notation. There are four types of inventory considerations: *capacity* (β_{8a}), *time* (β_{8b}), *no idle constraints* (β_{8c}) and *storage properties* (β_{8d}). In some workshops, the storage capac ity is not a constraint because the number and the size of the stor age tanks are greater than the need ($\beta_{8a} = $ Ø). The inventory capacity may also be limited ($\beta_{8a} = $ LC) by the number and the size of the storage tanks, or worse there is no storage capacity ($\beta_{8a} = $ ZC). In this condition, the product (or intermediate) is stored on the processing machine. The storage capacity considerations are also linked with transit constraints. Indeed, the transfer of a prod uct from its storage device to its next operation can occur only if the level inside the storage device is superior to a fixed threshold ($\beta_{8a} = $ MC). Interesting examples can be found for process schedul ing with finite intermediate storage in Ku and Karimi (1988), and with no intermediate storage in Suhami and Mah (1981).

For *time* consideration on inventory, we recall that the waiting time described above refers to the time between consecutive oper ations of a job. In the industry this waiting time is often restricted, for example in food industry in the case of a sensible product or when warranty periods are specified, but also in metallurgy in case of liquid metal for energy consumption reasons. Here we can draw a parallel with the inventory capacity features because the waiting time can be without restriction ($\beta_{8b} = $ Ø), not exceed an upper limit for time storage ($\beta_{8b} = $ LWT), or the operation should flow without waiting time (zero wait, $\beta_{8b} = $ ZWT). An inferior limit (minimal waiting time) can be also possible due to operating conditions rea sons ($\beta_{8b} = $ MWT).

No idle constraints (β_{8c}) 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.

The last storage restriction relates to the ability to share or not the storage (β_{8d}). At the upper level, the storage can be used by all the materials. But storage can also be specific to a product, raw material or intermediate for example due to security reasons or for quality or traceability constraints that prohibits to store different batches of a same product in the same vessel, we must have a specific storage tank for each batch. As a conclusion the storage policy is the combination of these four previous elements.

The detail description of all the β sub fields is not discussed in the core of this article but it is important to highlight the field β_{13}. Indeed, the decisions on the number and size of batches are often considered as a planning decision in manufacturing problems, and thus they are integrated as given parameter in the model. But these decisions can also be done at the scheduling level which appears frequently in process engineering. Therefore, they are introduced as decision variables in the model, with their associated con straints. More generally, this comment on decisions that can be taken at a higher strategic level and thus imposed at the scheduling one is also available for constraints. As a consequence, they must be included in the proposed notation because they do not appear in the previous one.
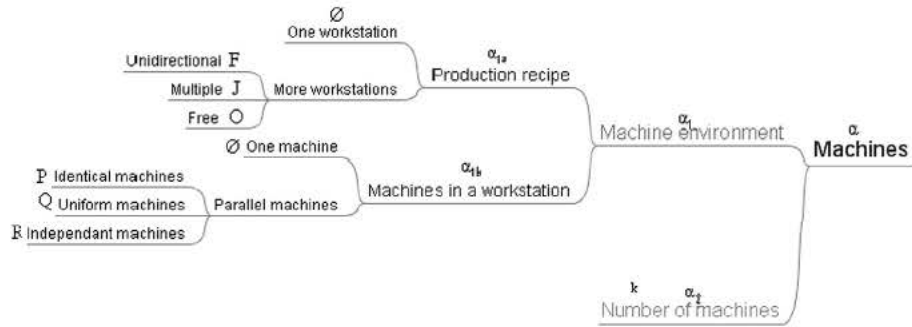
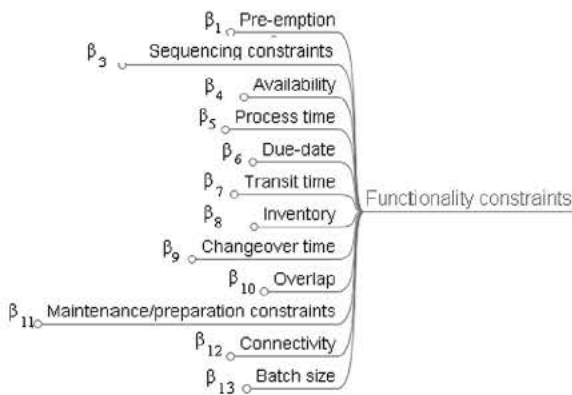**Fig. 5.** Production recipe and workstation characterization.
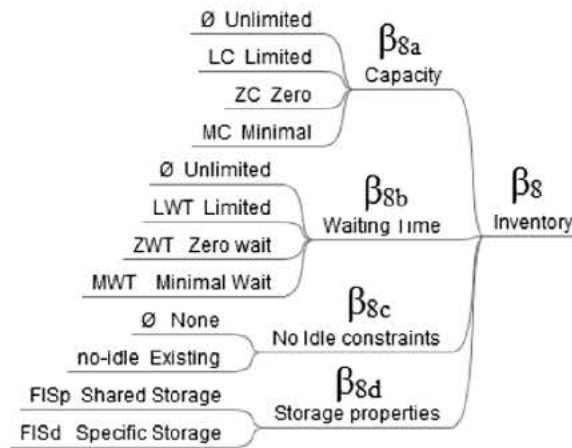

Fig. 6. Functionality constraints.

Fig. 8. Objective functions.

Fig. 7. Inventory considerations.
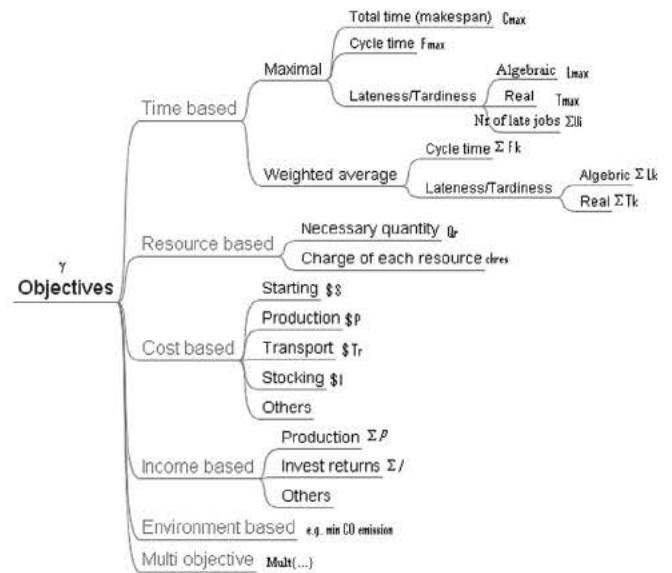
represent a market price of the product. In the literature, less frequent objectives are environmental (Dessouky, Rahimi, & Weidner, 2003) or security based ones, usually they are described (explicitly or implicitly) in the constraints. In the current context with sustainable development concerns, the environmental impact will become an important criterion to minimize. But this should not come at the expense of economic or deadlines objectives, leading to multiobjectives optimization.

2.4.2. Solution part $\delta|\varepsilon$

Currently, no notation system has been found for mathematical models. To complete the aimed decision support system it seems necessary to propose a notation scheme for both mathematical models and solving methods to capitalize the knowledge on the compatibility of problem types, model types and solving methods.

Depending on the modelling options, several effective mathematical formulations were developed to model process scheduling problems. These model characteristics influence directly the computational performances, the capabilities, the strengths and weaknesses of an optimization model. Properties of mathematical model are described by the field δ, and detailed in Fig. 9.

Mendez et al. (2006) proposed a first classification of the mathematical models, frequently used to represent scheduling problems. Among the model features, time representation is one of the most important decisions, on particular on the choice between discrete and continuous time representation (δ_1). As a consequence it has been studied and discussed widely in the process systems

Scheduling of a production plan is realized according to one or more objectives as in the work of Pan et al. (2009) for a bi criteria flow shop. There are several types of objective functions, such as time based, resource based, cost based, income based, environmental ones and multi objectives, shown in Fig. 8. Wide spread time based objective functions are: makespan minimization ($\gamma = C_{max}$), average or maximum lateness, tardiness, but special cases can be found, minimizing deviation from a common due date for example (Gowrishankar, Rajendran, & Srinivasan, 2001). The cost , resource or income based approaches are typical in the area of process scheduling problems. For cost considerations, usually the most frequent objective is the profit maximization. The reason is that in multipurpose and multiproduct batch plants, and in process scheduling in general, the quantity of materials can easily
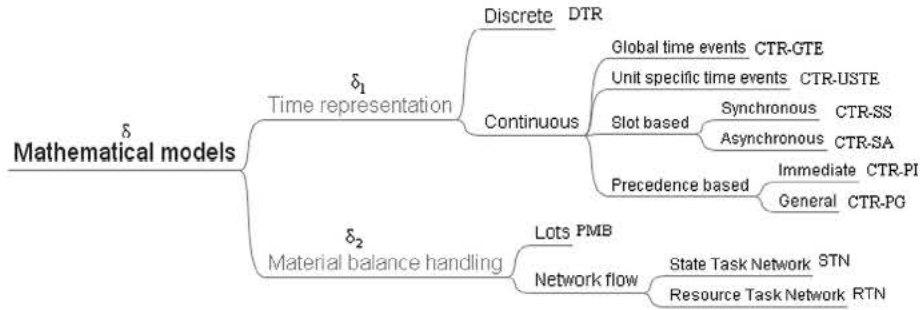
Fig. 9. Mathematical model options.

engineering literature. Detailed analysis of the different time representation options were performed by Mendez et al. (2006) and Pan et al. (2009).

Concerning material balance handling (δ_2), for sequential processes the models assume that the batching decision 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 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: State Task Network (δ_2 = STN) and Research Task Network (δ_2 = RTN). Both monolithic approaches were created in order to deal with complex product recipes. Schilling and Pantelides (1996) have extended the STN to the RTN framework where processing equipment, storage, material transfer and utilities are described as resources in a unified way.

It is important to notice that the choice of some modelling options restricts the possibility or sometimes imposes choices on other ones. For instance the RTN model was always used with time grid representations and not with sequence ones. But new studies are upsetting habits in order to offer new opportunities for modelling, e.g. Castro, Grossmann, and Novais (2006) used RTN model for sequential processes, or Shaik and Floudas (2008) coupled RTN model with Unit Specific Time representation. These two previous examples put in highlight the interest for an evolutionary decision support system.

A scheduling problem generally leads to combinatorial optimization problem. Despite several methods have been developed during the last decades, it is still difficult to deal with large scale ones. Most of the mathematical approaches are based on operational research techniques. They can be classified into three categories: mathematical programming, heuristics and metaheuristics. Scheduling problems generally lead to Mixed Integer Linear Programming problems. For more complex problems there are two possible ways to follow: simplifier assumptions are made in order to reduce the problem complexity, or/and empirical rules and strategies (heuristics) are applied in order to find a solution. Properties of solving methods are denoted with a field ε (Fig. 10).

For the mathematical approaches, the solving methods can be categorized into two groups: exact (if the optimality is mathematically guaranteed) and approximate methods (if a feasible solution is to be found in reasonable time). A common disadvantage of exact methods is that large scale problems cannot be treated efficiently without making simplifier assumptions often based on the strategies of decomposition. A problem can be decomposed with respect to three attributes: time, machines and constraints. For example, the so called Rolling Horizon technique is a decomposition by time: the main large scale problem is decomposed to a set of consecutive small scale problems, and successively solved in a part of the time horizon.

To deal with this combinational complexity various heuristic methods have been proposed. For instance, in the framework of permutation flow shops, Ruiz and Maroto (2005) have compared 18 heuristic methods divided into two groups: constructive heuristics (ε = HC) or ameliorative ones (ε = HA), for instance. With the former, users try to find a feasible solution from scratch, the solution is built step by step with respect to rules (like in Gupta, Krüger, Lauff, Werner, and Sotskov (2002) or Rad, Ruiz, and Boroojerdian (2009)). The latter start from an already existing schedule that it is used as a skeleton, and then they try to ameliorate it (Efstathiou, 1996).

Metaheuristics are general heuristic methods but they conducted a more exhaustive exploration of the solution space, to ensure that the solution is not a local minimum. Many of them implement some form of stochastic optimization, without guaranteeing an optimal solution. In scheduling, metaheuristics are used for combinational optimization to reach an "optimal solution" over a discrete search space. Metaheuristics based on decomposition techniques are also useful to address to large size and high combinational problems. Among them, the most popular are Simulated Annealing (ε = SA), Taboo Search (ε = TS), Genetic Algorithm (ε = GA), Ant Colony Optimization (ε = ACO). But there are also other possible metaheuristics and hybrid methods as well (combination, ε = Comb). For instance, Widmer, Hertz, and Costa (2001) have given an example of combined methods between metaheuristics and domain local knowledge, modelled with Artificial Intelligence methods. Besides, the performances of metaheuristics are strongly dependent on many factors such as the characteristics of different problems, and how the tuning parameters of the methods are set; significant effect on the quality of solution, and on the computational time. This aspect is not included in our decision support system because it is too dependent of the case studied and this knowledge is difficult to formalize and to combine with the purpose of our CBR. As explained before, a very recent paper addresses to this issue by developing a CBR system specifically dedicated to an autonomous parameterization of metaheuristics (Pereira & Madureira, 2013).

3. Case base filling

3.1. CBR data base

Efficiency of CBR systems is strongly dependent on the ability of the case memory to cover the problem and solution spaces. In the CBR community there is a debate with the opposition of two distinct views on the number of cases to be gathered in the base and the way to manage them. Some researchers point to an abundance of cases to fully cover the two previous spaces, designing CBR systems without adaptation phase (or with a low effort of adaptation). To keep certain efficiency, the discrimination of the most relevant case requires a very precise similarity measure, a
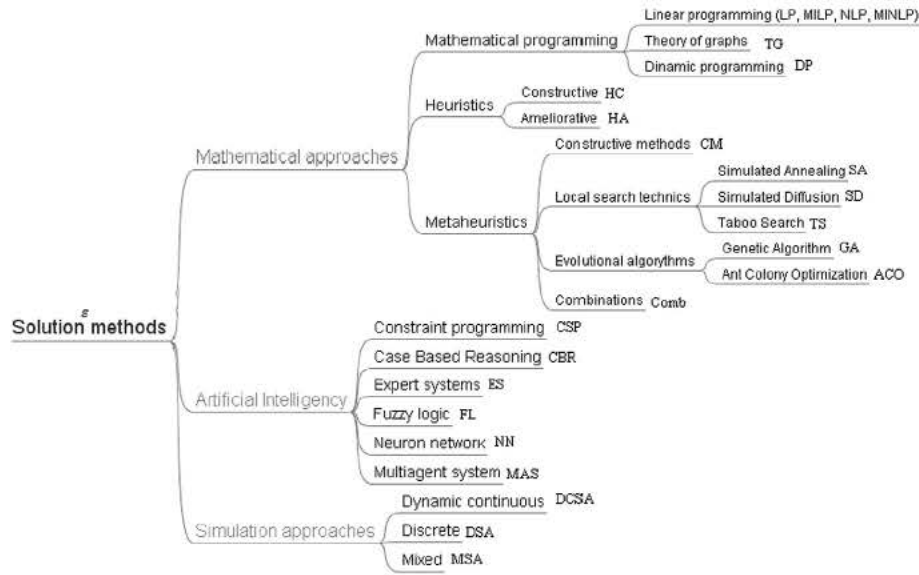
Fig. 10. Classification scheme of solving methods to scheduling problems.

structuration of the case base very specific to the application addressed and an efficient search algorithm. Furthermore such systems need a tremendous effort for knowledge and case base maintenance. The opposite point of view assumes that the case base encompass a limited number of cases but sufficiently representative. With this view, the system's effectiveness to solve problem relies on the adaptation step. For generality reason and the use of the created tool to other application, the second approach was chosen. Accordingly, to construct such a decision support system, it is important to collect information and obtain a landscape on the situation of research in this area of scheduling problems, in order to extract very relevant cases from the literature to feed the case base.

3.2. Literature analysis methods

Obviously, the expert past experiences are mandatory to fill the case base but the research literature is another important source of past experiences the most relevant and representative can be extracted. Scientific methods of bibliographic analysis are useful to perform a deep study on a specific domain, especially if a rich database of papers is available as in short term scheduling problems. Methods of bibliographic analysis can be classified into two groups (Osareh, 1996): methods of bibliometrics and citation based approaches. The former are usually applied to evaluate the scientific work or to characterize research intensity of an author, an institute, etc. The number of publications is frequently used in descriptive statistics, however in order to perform sophisticated studies it becomes insufficient. Bibliometrics methods can also be implemented to analyze research tendency, e.g. Sitarz, Kraslawski, and Jezowski (2010) used a method based on words co occurrence in article abstracts to identify thematic clusters in the distillation research area by applying financial analysing techniques. For the latter, the central assumption is that if a paper cites another one thus it indicates a relationship between them. Developed on this assumption, citation based approaches attempt to group the papers with respect to their mutual relation. For instance, Greene, Freyne, Smyth, and Cunningham (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 corresponding to each CBR steps.

In order to identify relevant papers in a precise research domain, citation based approaches are the most appropriate. Two categories of 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 cited papers. Two documents are co cited when they are jointly cited in one or more published papers. To illustrate the concept of the two methods a set of five fictive papers has been created, illustrated in Fig. 11, where P_i denotes paper i, and arrows denote citations (i.e. P_1 cites P_3 is denoted as $P_1 \rightarrow P_3$).

The papers P_1 and P_2 are bibliographically coupled because they both cite articles P_3 and P_4. In this example, bibliographic coupling determines a relationship between P_1 and P_2, while co citation analysis suggests a relationship between P_3 and P_4 (jointly cited by P_1 and P_2) and also a weaker relationship between P_4 and P_5 (jointly cited only by P_2). 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 relationship is dynamic as the strength of the connection is maintained or increased only as long as they continue to be co cited. Because of the dynamic evolution of the scheduling literature the co citation analysis based approach is the more appropriate choice to study the bibliography for our application.
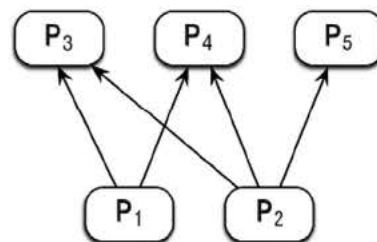

Fig. 11. Citation links between papers from Greene et al. (2008).

3.3. Clustering techniques

Once the database of papers is established and the previous citation network created, the next step is the regrouping of the papers dealing with similar research subject. A group of papers with multiple connections to each other is called a cluster and the regrouping process is called clustering. This step of our literature analysis method is borrowed from Greene et al. (2008). A cluster is determined by the connection rules of the network. In order to perform the clustering, 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 (Eq. (1)) is a particularly effective choice for clustering co citation data. Compared to Co citation Score, the other approaches have several drawbacks: 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 with example of Fig. 11. Suppose that the initial data base of papers (seemed papers) is the set {P3; P4; P5} and we attempt to determine the Co citation Score values based on the citing papers {P1; P2}. Let denote the co citation count of two papers i and j with C_{ij}, and let 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 the example are shown in Table 2.

The Co citation Score (S_{ij}) 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 between the respective citation counts C_{ii} and C_{jj}.

$$S_{ij} = \frac{C_{ij}^2}{\min(C_{ii}, C_{jj}) \cdot \text{mean}(C_{ii}, C_{jj})} \quad \forall i \neq j \tag{1}$$

To explain the normalization by $min(C_{ii}, C_{jj}) * mean(C_{ii}, C_{jj})$, let us suppose that paper i is cited much more than paper j (asymmetrical pairing). Then the normalization based on the mean citation counts between C_{ii} and C_{jj} underestimates the importance of each co citation for paper i. Conversely, if 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 Eq. (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 score is now in the range [0, 1], where a larger value indicates a stronger association between papers.

Once the measure of significance is chosen, 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 and combined techniques. Traditional methods use hierarchical algorithms to identify 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 is a tree of clusters. Agglomerative clustering is based on the following principle: find the two clusters with

the smallest similarity value, and 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). Divisive algorithms begin with an initial cluster containing the whole set of papers and proceed to divide this cluster successively into smaller clusters.

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 for 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 X is factorized into two matrices:

$$NMF(X) \rightarrow W, H$$

where the original matrix X is the product between matrices W and H. Different matrix factorization methods have been developed, e. g. principal component analysis and singular value decomposition (Lee & 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.

The major 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. Concerning the matrix factorization, the drawbacks are notably its sensitivity to the choice of an algorithm parameter, and the difficulty in interpreting the clusters generated. In order to eliminate the drawbacks of both approaches important attempts have been made to work out combined strategies, e.g. the Ensemble NMF algorithm used in our method. The Ensemble NMF clustering algorithm Greene et al. (2008) 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 is useful when the research area under examination is complex and papers can naturally relate to more than one cluster, like in scheduling domain. Firstly an 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.

3.4. Relevant cases identification

In Fig. 12, the next step of the workflow of the literature analysis is focused on the clusters analysis. As the number of cases stored in the base must be limited to ensure CBR effectiveness, we need to extract relevant papers in each cluster. The cornerstone of this method is to define what a relevant paper is. In a first approach a statistical analysis of papers gathered in a cluster can

Table 2
Co-citation counts for the example.

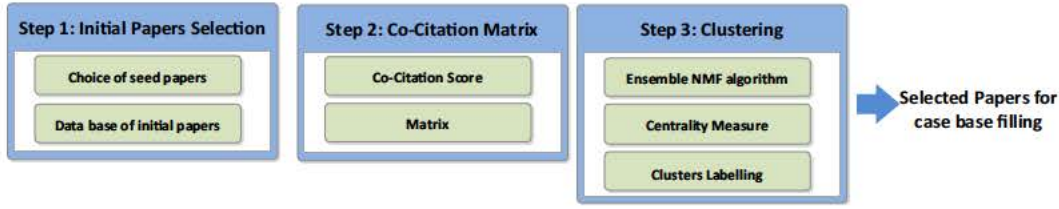|	P3	P4	P5
P3	2	2	1
P4	2	2	1
P5	1	1	1

Fig. 12. Workflow of the literature analysis method.

be helpful in widening our understanding of the cluster. For example, the most cited papers can be considered as important papers to analyze but they are not necessary the most relevant ones. Compared to Greene et al. (2008) approach, our method goes deeper in the analysis by making more progress in using social network analysis. Indeed, papers form a citation network where papers are regarded as vertices, while links traduced citing and the cited relationships between papers. Social network analysis is a mathematical method based on graph theory, which is used to understand the interactions among vertices in a network, i.e. papers in our case study. To examine the importance and value of each paper in a network, different measures exist for ranking papers, most of them are based on graph metrics. In our method, a centrality analysis is performed because it is related to each vertex but it is calculated in considering the whole network (Choe, Lee, Seo, & Kim, 2013). Several types of centrality can be defined to analyse the influence of a member (represented by a vertex) on a network (represented as a graph with relationships between vertex) for instance: betweenness centrality that indicates if the vertex is in many shortest paths between other vertices, closeness centrality that indicates the mean distance of the vertex from the rest of the network, Eigenvector centrality that indicates the influence of a node in a network. To qualify a paper i, we use the degree centrality value ($Cd(i)$) calculated with the following formula:

$$Cd(i) \quad \sum_{j\ 1}^{n} a(V_i, V_j)/(n \quad 1) \tag{2}$$

where the binary $a(V_i, V_j)$ is 1 if V_i and V_j are connected, otherwise is 0. n is the total number of vertices in the network, thus $(n \quad 1)$ is the maximum degree of a vertex. The degree centrality of a vertex in a network 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. As clusters encompass different number of papers, in order to compare them, the degree centrality values have been normalized with respect to the total number of co citation pairs. 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.

Till now, we have aggregated papers according to their research topic, and then we have identified the most relevant one in these clusters. The next step is dedicated to the labelling of clusters, i.e. finding the research topic(s) targeted by the papers inside the cluster. Another major contribution is the improvement of the method of Greene et al. (2008) by performing a word frequency analysis for all the papers belonging to a cluster. This method is more accurate as each paper is considered and analyzed. Words, occurring more often than others in a cluster, receive a higher score and supposed to be more indicative to the cluster. In order to label the meta clusters the RapidMiner software is used to perform word frequency analysis. Two different manners to proceed with the word frequency analysis are possible. The simplest is to label the cluster with the word that occurs most frequently. Another

option is to first establish and impose a predefined list of words to find the nest candidate among the clusters which corresponds to one word of the list. The latter option was used and the list of predefined words is composed of the possible values for each field α, β, γ, δ and ε. This list corresponds also to the possible values for case features. For example, if we would like to identify the cluster that most closely matches the scheduling problems dealing with deadlines, then the most probable candidate is the one whose list contains at the top the word "deadline" or "due date". In this cluster, the most relevant papers are included in the case base (obviously after expert validation) in order to widen the cover of the problems dealing with this scheduling constraints. The different steps of the method are sum up on Fig. 12.

3.5. Application to scheduling literature

The above described method has been applied to scheduling research area in order to draw up a landscape on the research situation of this domain and to extract relevant cases. To apply our method 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 have been used as the initial database. The sort based on the citation count was made in order to obtain a sufficiently high number of citation links for the analysis. Next, the papers citing the 100 seed papers have been collected, 4839 results have been recorded and the number of co citations has been determined. 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, 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 a splitting factor criterion thus they 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 restrictive threshold, 40 meta clusters remain. Among them, 18 meta clusters contained few papers (less than 10), worse the papers inside these clusters have a very low centrality (near zero). The poor population of these meta clusters indicates that the number of papers is not high enough to successfully draw conclusions from applying the centrality theory. Therefore these clusters were also eliminated. For the reason of consistency the original numbering of clusters has been maintained through all the process.

In the remainder of this section, the results are illustrated with bubble diagrams like in Fig. 13. In these diagrams, each bubble is related to one paper. Besides, each diagram gives three kinds of information to interpret:
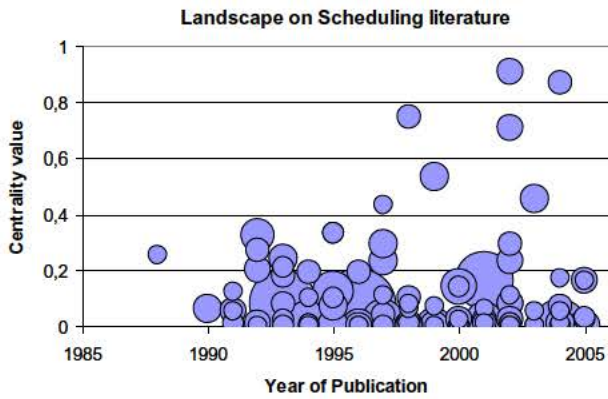
Landscape on Scheduling literature

Fig. 13. Evolution of scheduling literature during the last decades.

- The evolution along time. By this evolution, we can identify if there is a dynamic activity on the research topic illustrated, when the scheduling features was first treated in the models.
- The size of the bubble. It represents the total number of citations of the associated paper (information coming from the statistical analysis).
- The position of the bubble along the centrality axis. It traduces the relevance of a paper inside the cluster (information from the social network analysis).

These diagrams help to the recognition of new cases to introduce in the case base in order to extend the cover of the space of problems or solutions. Indeed, papers with largest area or with highest centrality are papers to examine preferentially and to potentially include in the case base. Of course bubbles with both large area and high centrality value correspond to the most relevant papers. As clusters are correlated to case features, themselves directly related to the classification items, the figures permit to easily identify the best modelling and resolution options to build a model that aims to include the scheduling constraint targeted by the cluster. Firstly, some general remarks will be made on the research activity in scheduling literature by representing on Fig. 13 the bubble diagram of the 100 seemed papers. Fig. 13 illustrates the wideness and variety of the scheduling domain.

In Fig. 13 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. Moreover, scheduling research is in permanent evolution and remains a challenging domain. The identification process is restricted to identify some specific clusters corresponding to well defined areas of process scheduling. To illustrate the results of the clustering algorithm based on the above detailed method, the terms "resources", and "due dates" have been voluntarily chosen because they are representative of the type of figures encountered in the literature analysis.
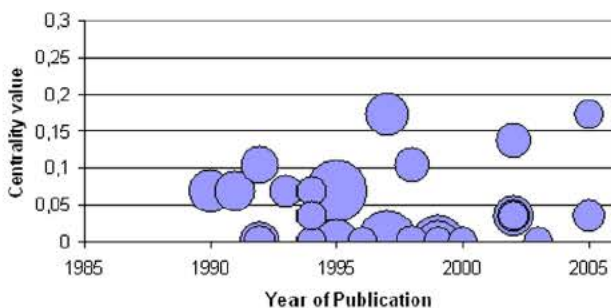
In Fig. 14 continuous, stable research effort can be seen, with several significant papers. Resource handling requires constructing complex mathematical models. Since lot sizing is frequently considered and mixing/splitting operations are allowed, these types of problems are very difficult to solve. 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. Numerous modelling options can be used to represent resources considerations. Consequently for this specific cluster, papers with centrality greater zero and those with a large number of citations are submitted to expert analysis in order to be introduced or not in the case base.

Fig. 15 is a representative illustration of the majority of clusters with some central papers and other ones frequently cited. In the particular case of paper dealing with "deadlines" there is a continuously challenging with significant papers from the nineties. It can be thus deducted 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. Indeed, these constraints were the first considered in models because they were and remain the principal constraints to respect in industry. In other words, general considerations of this aspect have been deeply analyzed, but no improvement or specific application approaches were developed since these studies. Further researches have tended to focus on the introduction of new constraints coupled with deadlines considerations rather than improving the way to model them.

The performance of the full text analysis is studied, in a meta cluster example. The application of word frequency analysis on the full text of the available papers was based on the cluster's key words: queue, protocol, grid, workload, balancing, network. The cluster's thematic has been identified as "network protocols, net working workload balancing". It is shown in Fig. 16 that the interest towards the thematics started in the early nineties, probably because of the spread of personal computers. From 1994 to 1996 significant research breakthroughs have been performed, founding further research in this area. These central papers deal with work load 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.

3.6. Discussion

As mentioned before co citation base analysis is dynamic thus the strength of connection changes with time. As a consequence we have an "image" at the studied time that can evolve. However, the kinetic of the change is quite slow so the study does not need to be repeated too frequently.

As it can be noticed in Figs. 13 16, the articles considered in the method are all published before 2006, although more recent article were included in the study (from 2012). Indeed, as for a paper its
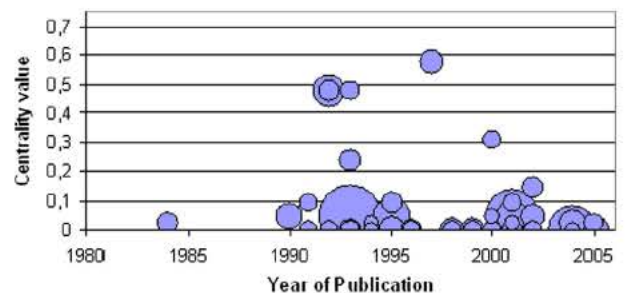

Fig. 14. Cluster of resource considerations.
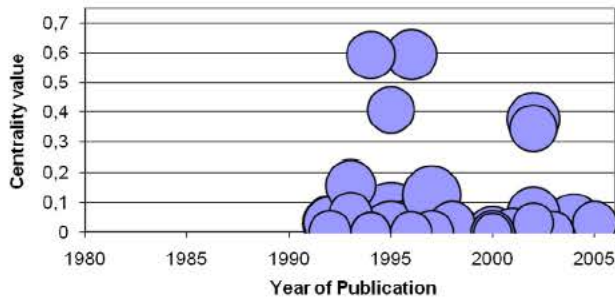

Fig. 15. Cluster of deadlines.

Fig. 16. Cluster of network protocols and networking workload balancing.

bubble size is based on the citation count and its degree centrality on the number of vertex connected to it (influence of the paper in the cluster), it must be cited or co cited several times to appear and be important in the cluster according to these two criteria. Conse quently very recent articles cannot stand out as major article in the field and underrepresented. But from the tendency of earlier pub lications the trend of research can be followed, and the activity in the area can be mapped, which satisfies the original intention.

More annoying, recent studies offering new opportunities for modelling are not considered while they open new promising ways. For example the works of Castro et al. (2006) who used the RTN model for sequential processes, or Shaik and Floudas (2008) who coupled RTN model with unit specific time representa tion. These breakthroughs obviously useful in a knowledge base for building a model would appear later in the literature analysis. Nev ertheless, this drawback can also be seen as a filter because even if a paper proposes a new evolution, there is no evidence that it will become a major breakthrough in the field. Indeed, it offers model ling options that can be innovative but limited to the particular case treated and none transferable to other problems. As a conse quence, a recent article bringing new knowledge about the field of study, cannot be directly included in the case base. Its introduc tion is carried out only if the expert considers that it increases the coverage of the problems space or the solutions one but also on condition that this new case is relevant for future uses. But the expert cannot examine all the papers from the last years (2010 to 2013 in our case) to validate if the model can be included or not in the base. To help him to identify the interesting papers in the last period we use the word frequency analysis performed in the clusters labelling step. Indeed, papers dealing with a new com bination of modelling options, a new combination of a modelling option and a solving method, a new or an improved solving meth ods are extracted to be reviewed. This process cannot miss critical cases while avoiding tedious analysis of all articles published in the last period.

Finally, Fig. 17 illustrates all the possible sources for case base filling: expert own experience, literature analysis, CBR system with the retain step and new article with promising new approaches. Regardless the knowledge source, before to be included in the case base, a new case must be validated by the expert. Indeed it is incor porated, only if it has a real added value for the CBR system. For example, after a CBR resolution the new case is not automatically retained because it can be too close to an existing one or its solu tion must be easily derived from a solution gathered in the case base. Expert knowledge goes further than case validation or case creation, he also brings its knowledge for the other CBR steps, e.g. in case retrieval with similarity measure.

4. Case retrieval

The purpose of case retrieval is to identify and to select a set of relevant cases that can be reused to find a solution to the face problem. Case retrieval needs a selection criterion (or criteria)
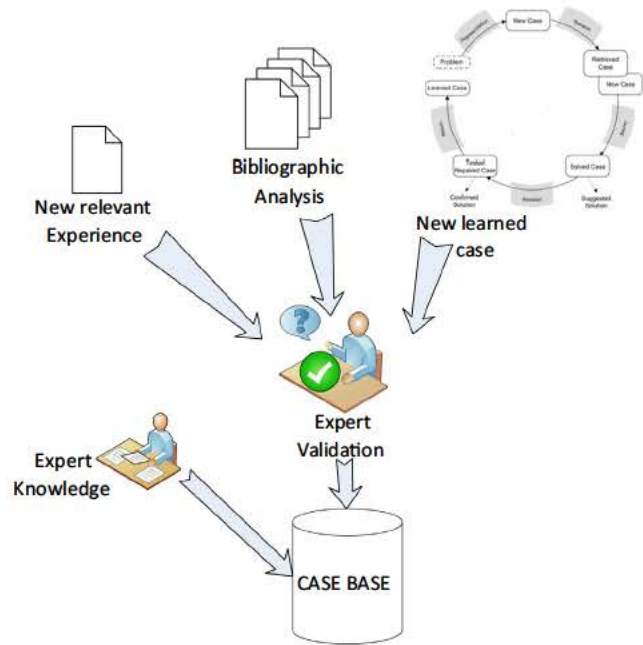

Fig. 17. Sources of knowledge for case base filling.

determining how a case is judged to be appropriate for retrieval. Consequently, the quality of the results mainly depends on the similarity measures used to extract cases. However, apart from the similarity measure, there are other factors to consider as well, when choosing a method of retrieval, e.g. the number of cases to be searched, the amount of domain knowledge available, and if one wants to look further: the time and resources required for adapta tion. The retrieval process is illustrated in Fig. 18. This step is detailed and discussed by Behbahani, Saghaee, and Noorossana (2012).

For the remainder of this section, it is assumed that the case structure does not evolve, i.e. the features of the descriptions of the problem and solution parts are the same for old and new cases. As a result, the CBR system loses in flexibility in order to ensure system performance in responses to changes in problem and solu tion environments. This assumption is widely stated in CBR sys tems because changes can have numerous impacts on the different tasks of a CBR system such as: similarity measures, cases indexation, adaptation mechanisms. Nevertheless, for the specific application aimed, this assumption is not too restrictive as the
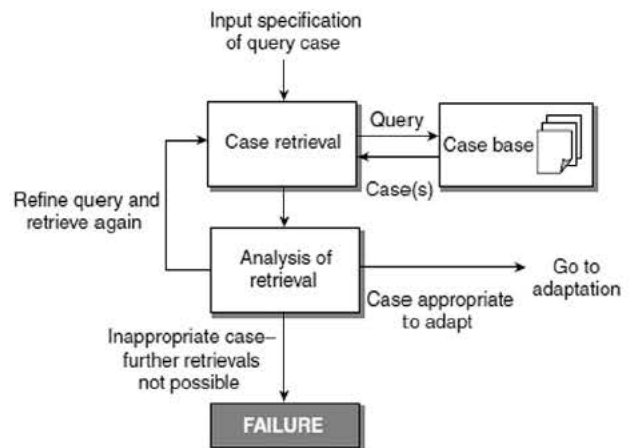

Fig. 18. Retrieval process from Pal and Shiu (2004).

problem and the solution description features does not evolve strongly. Indeed, the numerous research studies in this research domain have allowed to have an exhaustive list of the principal characteristics needed to describe the workshop specificities.

4.1. Similarity measure

During retrieval, the new problem is matched against the stored problems through the comparison of two cases and the determination of their degree of similarity. It is a numerical value expressing how two cases are similar. The effectiveness of a similarity measurement is determined by the usefulness of a retrieved case in solving a new problem. This task includes computing the degree of similarity between corresponding features, describing the new problem and the ones in the case base, and assigning importance weights (w_i) to each feature to customize the similarity measure in order it corresponds to the search purpose. The global similarity (SIM) between two cases X and Y is calculated through the weighted average (formula (3)) of local similarity (sim). These ones are calculated for each problem feature i by comparison of the feature value of the faced problem (x_i) with its corresponding one for the problems stored in the base (y_i). np refers to the total number of features.

$$SIM(X, Y) = \frac{\sum_i^{np} w_i sim(x_i, y_i)}{\sum_i^{np} w_i} \qquad (3)$$

However, as problem features are described with different types of values, the local similarity calculation must take into account these specificities. In our CBR application, the majority of problem features are filled with nominal values. Consequently a specific local similarity measure must be established in order to be sufficiently accurate to discriminate the potential candidate cases. Indeed the conventional local similarity measure is binary:

$$sim(x_i, y_i) = \begin{cases} 1 & \text{if } x_i = y_i \\ 0 & \text{if } x_i \neq y_i \end{cases} \qquad (4)$$

In our CBR system, some features of the problem description can be considered as set of elements (e.g. inventory) rather than individual nominal features. Indeed, with this way to calculate the distance, we could substitute the binary matching by a more precise measure with the Levenshtien formula (Cunningham, 2009). The distance between two sets is determined by the number of none common elements in the sets.

$$sim(x_i, y_i) = 1 - \frac{|x_i \cap y_i|}{\max(|x_i|, |y_i|)} \quad \text{where } x_i \text{ and } y_i \text{ represents sets} \qquad (5)$$

This way to estimate local similarity is still too general, thus we need to go deeper and to add domain specific knowledge to improve this measure. As each problem feature is decomposed into sub features, e.g. Fig. 7 for the inventory option, a hierarchical structure tree can be built to describe the relation between sub features. Fig. 19 sketches this hierarchical structure for inventory considerations. The first level node in the tree corresponds to the basic feature (inventory option present or not), the daughter nodes correspond to sub features such as: capacity, waiting time, storage properties. A numerical value is assigned to each tree nodes, then for two problems matching the local similarity value for the feature depends on the first common node in the tree. The deeper the common node is, the higher the local similarity. For example, for the same modelling option the local similarity is 1, and is 0.7 for two different options belonging to the same sub feature.

The weights assigned to each feature are dependent of the search goal. The user can assign himself these weights according
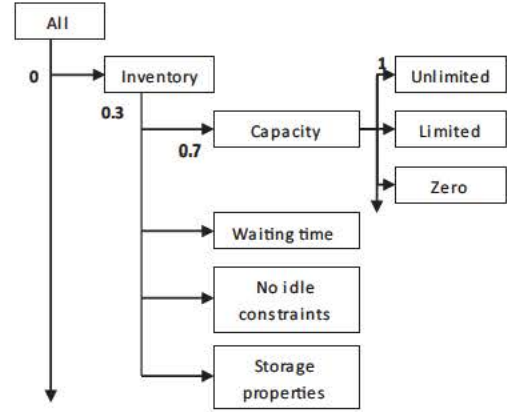

Fig. 19. Local similarity measure for nominal feature.

to its purpose, or he can be helped by asking him to select features by their order of importance. Features with rank 1 are the most important, and two features can have the same rank. Then, each weight is calculated with the following formula:

$$w_i = 1 - \frac{rank_i - 1}{Max(rank_i)} \qquad (6)$$

4.2. Adaptability

The retrieval step is crucial reach an efficient CBR system because it strongly influences the remainder of the solving process. But the most similar is not necessary the most appropriate for solving propose and more particularly for the adaptation step. Indeed the similarity is estimated on problem features but not on the ability of a solution to solve another problem, i.e. a retrieved problem can be the most similar but its solution can be difficult, worse impossible, to be reused. To guarantee an efficient retrieval, additional knowledge must be acquired and formalized. Smyth and Kean (1998) have proposed to improve similarity by introducing an adaptability criterion to link retrieval and adaptation requirements. The key issue is to estimate the adaptation potential of a case during the retrieval step. The adaptability tries to evaluate if a case would be easily modified and adapted or not. Based on this idea, we proposed an adaptability measure estimated on the solution feature. When a new case is stored in the base, the expert examined its solution part and for each solution feature, he indicates the other possibilities that can be covered by this solution, e.g. for the same problem different time representations can be used. Consequently, the expert does not characterize a solution feature with one and only one value but with a distribution of possibility modelled with fuzzy set theory. This distribution of possibility represents the additional expert knowledge needed for an effective retrieval. The intersection of this distribution of possibility with the modeller search purposes gives an adaptation space. This space represents the possible values that a solution feature can take while remaining compatible with the modeller goals. The shape of this adaptation space traduces the easiness of adaptation for the selected case: the more the shape is wide (fuzzy or imprecise), the more it contains values, the highest the adaptability of this feature. The specificity of a fuzzy set allows to evaluate the degree to which a fuzzy set contains one and only one element:

$$S_p(F) = \int_0^1 \frac{1}{\sup F_\alpha - \inf F_\alpha} d\alpha \qquad (7)$$

$Sup\, F_\alpha$ (resp. $Inf\, F_\alpha$) represents the upper (resp. lower) bound of an α cut on the domain. The specificity value is in the range [0; 1], with

$Sp = 1$ for a set with one and only one possible value. Therefore the local adaptability can be calculated with the following formula:

$$ad_i \quad 1 \quad S_p(F) \tag{8}$$

The global adaptability of a case is calculated with weighted average of the local ones:

$$ad_s \quad \sum_{i=1}^{ns} w'_i ad_i / \sum_{i}^{ns} w'_i \tag{9}$$

The details on adaptability calculation, on case base structuration and on a new search algorithm to extract relevant cases from the base are presented in Negny, Riesco, and Le Lann (2010).

Despite the adaptability measure and an important number of cases in the memory, it can remain some situations where no sufficiently adequate case fitted to a new problem can be extracted. To overcome this drawback, the CBR system must contain an efficient adaptation module which needs to add new knowledge. For knowledge extraction, two types of source are identified: external or internal to the CBR system. Internal adaptation knowledge can be extracted from the differences between the cases stored in the case memory. Among the external sources the most obvious is the expert, who can be solicited to formulate the new necessary knowledge. Currently in the tool we introduce a method base on internal source. This adaptation method is based on the main idea that the relative distances between the target problem and some source problems in the problem space are transferred in the solution space. This leads to a minimization problem as detailed in Negny and Le Lann (2008). It is important to underline that the proposed solution is just a first estimation, and then it must be modified after some additional validation tests.

5. Case study

5.1. Developed tool

We develop a software tool, which has two principal human machine interfaces. One dedicated to CBR expert, for case base administration where he can manage:

The case base indexation: a query sphere algorithm was developed to index case in the base and to search relevant case thanks to an algorithm based on the discretisation, detailed in Negny et al. (2010).
Similarity measures: in this paper we present some similarities measures but new ones can be added.

The retain step or case validation: it accepts or not if a new case should be added to the memory base.
A new bibliographic analysis: the method detailed in part 3 is included in the CBR and an expert can run a new analysis because of the dynamic evolution of the scheduling literature.

The second interface is for the user, where he can: describe its problem, choose the similarity measure, specify weights, see retrieval results (similarity and adaptability criteria), and select cases.

5.2. Problem description

The case study, first studied by Czuczai, Farkas, Rev, and Lelkes (2009), concerns an industrial process for beer production that is composed of 4 workstations: a draft beer tank, a filter, a bright beer tank, and a workstation with three package lines, Fig. 20. 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 RBT. 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. During filtration, the beer flows continuously through the filter, and as a consequence the RBT and the BBT must be available through the whole filtration task. The beer can be accumulated in the BBT from several filtration operations during a time period for which an upper bound is 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. 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. For evaluating and ranking the candidate solutions, the objective function to be minimized is the time at which all the product orders are satisfied.
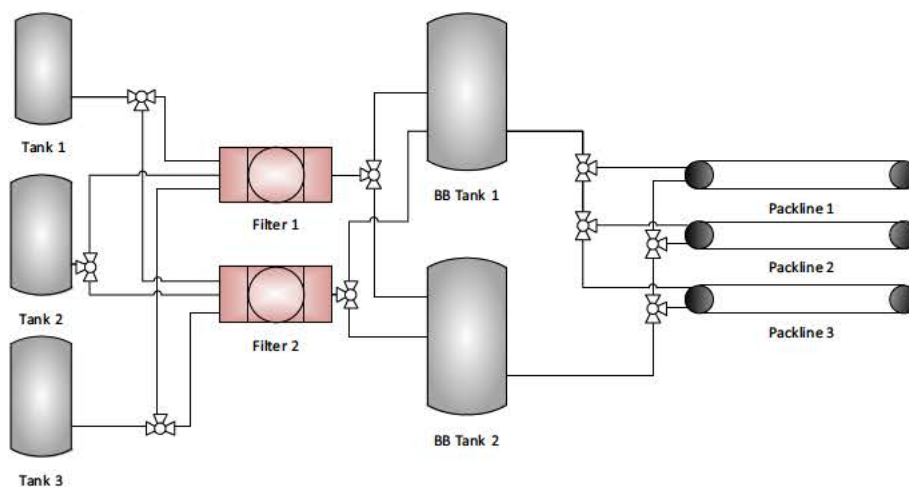

Fig. 20. Process flowsheet of the case study.

This case study has been chosen because it deals with some specific constraints not fully treated in literature like storage, connectivity restrictions and continuous task. This assess the ability of our system to be able to describe very specific problems, but also its ability to propose modelling options suited to these kinds of problems.

As illustrated in Fig. 20, the production workshop is composed of ten units distributed in four workstations: three RBT (tank 1 3) in workstation 1, two filters in workstation 2, two BBT in workstation 3 and three pack lines in workstation 4. The characteristics are summarized in Table 3.

5.3. Problem notation

For the process studied, the characterization of the machine environment is not as easy as it seems. Obviously we have a hybrid flow shop with identical machines in each workstation. Traditionally storage issues are not handled with the creation of a specific workstation but are included as constraints in the model (except when the unit is also the storage tank). But here we have storage time constraints: materials can be released/received during storage. Some contributions consider that the material can be received only at the start of the storage and released only at its end. In the example, the material can be received or released at any time of the storage process. Thus, in order to handle these storage time constraints, the storage must be considered as a specific task and consequently as a workstation in the problem description. In practice, the duration of this storage operation is not fixed but a maximum process time is imposed. Furthermore, the storage time

constraints imply an implicit zero wait constraint to be applied for operations preceding the storage operation. This constraint must be added to the problem notation although it does not clearly appear in the description.

The other constraints are clearly explained in the problem description, thus the representation of the case study with our notation scheme is summarized in Table 4.

Although the notation is sufficient to describe a problem, it is not detailed enough for the problem formulation in the decision support system. Indeed, to efficiently develop valid models for scheduling applications, the size and the combinatorial aspect of the problem must also be taken into account in the problem description because they significantly affect the model implementation and the simulation results. In addition to the preceding features, some other ones are added in the CBR system like: the number of machines in each workstation, the number of products and the number of orders.

5.4. Case retrieval

For demonstrating the capability of our new approach, and to check the performance of the different modules of the retrieval step, different queries were run. Each one simulates the potential use of the tool depending on the level of expertise of the user. In the first search, the behaviour of a novice user, i.e. user with no skill in modelling of short term scheduling problem, is simulated. Expert who is aware of the problem characteristics and constraints that require special attention in the modelling is simulated in search 2. For searches 1 and 2, the extraction of a case from the base is only based on the similarity criteria. In search 3, to support the expert decision for selecting a relevant case, we add the adaptability criteria. To present the results, we use part of the notation related to the description of the solution part of a case, i.e. the fields δ and ε. Table 5 depicts the results for the three searches. The last column of Table 5, i.e. the work of Czuczai et al. (2009) gives the description of the modelling option used to solve the case study.

Table 3
Problem general characteristics.

Process topology	Sequential
Production purpose	Multiproduct
Production mode	Semi-continuous

Table 4
Notation for the case study.

Machines	Machine environment	Production recipe	Flow shop	α_{1a} = F
		Machine in workstation	Identical parallel machines	α_{1b} = P
	Number		4 Workstations	α_2 = 4
Tasks and resources	Preemption		Not allowed	β_1 = Ø
Characteristics	Secondary resources	Resource Type	Consumable resources with variable initial amount	β_{2a} = cons$_V$
		Resource availability	Continuous	β_{2b} = CRA
	Sequencing Constraints	Precedence synchronization constraints	Chain	β_{3a} = chain
			Start-end and End-Start	β_{3b} = SE/ES
		Closed circle	No	β_{3c} = Ø
	Availability	Jobs	At t = 0	β_{4a} = Ø
		Machines	Continuously available	β_{4b} = Ø
	Process time		Quantity dependent	β_5 = P$_S$
	Due dates		Deadlines impose	β_6 = d$_k$
	Transit time		Not Considered	β_7 = Ø
	Inventory	Capacity	Limited	β_{8a} = LC
		Waiting time	Limited and zero wait	β_{8b} = LWT and ZWT
		No idle constraints	None	β_{8c} = Ø
		Storage properties	Shared	β_{8d} = FIS$_P$
	Changeover time		None	β_9 = Ø
	Overlap		Possible	β_{10} = overlap
	Maintenance/ preparation		None	β_{11} = Ø
	Connectivity		Restrictions	β_{12} = con
	Batch size		Fixed	β_{13} = fix
Objective function	Time based	Maximal	Total time	γ = C$_{max}$

Table 5
Comparison between the 3 searches results and the model used.

Fields	Search 1: same weights and similarity	Search 2: different weights and similarity	Search 3: different weights, similarity and adaptability	Model used by Czuczai et al. (2009)
δ_1	DTR	CTR-USTE	CTR-SA	CTR-SS
δ_2	RTN	RTN	RTN	RTN
ε	GA	MILP	MILP	MILP

5.4.1. General comment

For all the solutions δ_2 = RTN, which is not surprising at first glance. Indeed the RTN model can easily account for two major characteristics of the initial problem: storage and material transfer of resourced in a unified way. But to go deeper in the analysis, some modelling options are not compatible with each other. Particularly, we can focus on the options materialized by δ_1 and δ_2, and it is not obvious to associate RTN framework with continuous time representation (CTR). By construction the RTN framework was limited to be always used with grid time representation (as in search 1) and not with sequential one (as in search 2 and 3). But studies, for instance the pioneer work of Shaik and Floudas (2008), proposed new opportunities to use the RTN model with continuous time representation, and more precisely the Unit Specific Time representation for the previous authors (result of search 2). While this approach allows enlarging the possibilities and the compatibilities between modelling options, it does not emerge from the bibliographic analysis. This highlights the interest to not only based the case base filling only on a single source of knowledge such as literature but with the ability to diversify sources in order to include innovative approaches.

5.4.2. Search 1: novice user

As the user has no specific skills on scheduling problem, all the problem features have the same weights and after retrieval we select the most similar case.

In search 1, the main reasons why the modelling options ε and δ_2, are very different from the results of the other searches and from the options retained, is the iso weighting. Indeed the retrieved case has an important number of workstations, and consequently an important number of constraints. But as discussed previously, this feature strongly influences the size of the model to solve and consequently the modelling option and the resolution method. 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 a predefined grid which reduces model complexity, obviously depending on the number of time intervals. But the number of time interval also affects computational efficiency and solution accuracy. This point 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. But in the case study we need accuracy (industrial example). Furthermore, with this option it is difficult to handle storage constraint and continuous tasks, and the number of binary variables required increases as the problem size increases. In conclusion, this option is not suitable for the modelling of our problem. However, it is important to notice that the solution method ε = GA (Genetic Algorithm) is consistent with the modelling options proposed. As a consequence, the use of such a CBR system requires a minimum of knowledge about the application domain to make relevant queries. The first way to include this knowledge is to customize the similarity measure.

5.4.3. Search 2: expert

In this search, expert assigns higher weights to major problem features and selects a case based on the similarity measure. Concerning the problem features, we decide to allocate the highest weights to the unusual constraints or the constraints of prime importance according to our purpose. Ranked in order of importance the first three are: connectivity restrictions, synchronization constraints and the number of workstations (due to its strong influence, as we can see in the previous part).

Concerning connection issues, the literature models conventionally apply a material balance (amounts of a material produced or consumed are collected). Unfortunately, such models are inappropriate when units must be connected to only one other unit at the same time. Moreover, the process studied gathers continuous operations, thus the unit in which the operation is processed must be connected during the whole processing time to the other unit. The model must directly handle the connectivity restrictions. An analysis of the bubble diagram of the cluster "Connectivity Restriction" shows a poor consideration of these constraints in the literature. Indeed, the graph contains only eleven papers among which the work of Prasad, Maravelias, and Kelly (2006) has the greater centrality. In the present work, to consider this above mentioned issues we need a general integrated treatment to model the existence of this kind of connections. Both low consideration and connection specificities force us to put the higher weight to this feature.

Like in most of the short term scheduling model there are numerous precedence constraints. In the case study, it is not the number of this kind of constraints that it is specific but the variety of different precedence links. For instance there are precedencies between: two batch operations, two continuous operations, storage and batch operations, batch and storage, storage and loading operation... This diversity on the precedence relations obliges us to consider it as an important feature for the model building. As we have seen, the choice of the major features needs some experiences on the application domain.

As we have an initial problem with a small number of workstations (α_2 = 4), and as we need accuracy, all the cases at the top list of the retrieved case are solved by a MILP method. Besides, all the ten most similar cases retrieved use a continuous time representation which reduces the size of the model, and thus facilitates the use of a MILP method.

Despite the progress on the time representation option (from discontinuous to continuous), the representation δ_1 = CTR USTE is different with the option use in the real case (last column of Table 5). Worse δ_1 = CTR USTE is not recommended for the initial problem because of the number of event point needed for modelling release and due times. Furthermore, as Prasad et al. (2006) have explained, the requirement of connectivity restrictions between devices impose additional binary variables, resulting in a large number of this type of variables. Although it is possible to modify the solution proposed by the retrieved case to handle the initial problem, the preliminary analysis showed that it could be hardly effective for the reasons explained before. Moreover, the new model created must be flexible and general to study different process options: increasing number of devices, new connectivity, additional constraints (for instance batch size not fixed, Maintenance/preparation tasks). With δ_1 = CTR USTE representation, it will become difficult to manage these evolutions, because new problems with the same number of operation as in the case study have larger number of integer and binary variables, leading to much longer solution times.

5.4.4. Search 3: expert and adaptability

Here again the expert assign the higher weights on the same major features as before but he selects cases both on similarity and adaptability measures. To support his choice, a Pareto front with these two criteria is drawn. Of course the graphic does not include all the cases in the case base but only on a reduced list whose selection is based on the similarity measure (the number of cases to put in the list is fixed by the user).

The solution features of the retrieved case for this search are described in fourth column of Table 5. As we have discussed in the previous paragraphs, the most similar cases extracted after searches 1 and 2 have very low adaptability values. Here we put in highlight the contribution of the adaptability criteria because it allows to extract a relevant case with a solution very similar to that used. Both solutions used a slot based representation, the difference is that the retrieved case in search 3 proposes an asynchronous representation $(\delta_1 = \text{CTR SA})$ because the corresponding problem has specific constraints that can be only modelled with different time slots. Despite the fact that the asynchronous modelling provides more flexibility, in $\delta_1 = \text{CTR SS}$ the time slots are identical for all the machines and it allows to easier handle shared resources as the storage tank in the initial problem of the case study. Concerning the model evolution (as stated in search 2), here again the consideration of additional constraints leads to a higher resolution time but in a less steeply manner with the $\delta_1 = \text{CTR SS}$ than with $\delta_1 = \text{CTR USTE}$.

6. Conclusion

During last years, numerous publications were addressed to short term scheduling problems both in the process system engineering and manufacturing production research literature, (Kocsis, Negny, Floquet, Meyer, & Rev, 2010). We identify that there is a global trend to improve existing modelling options and solution methods in this domain. This trend is going to grow in the future especially because of industrial interests in the current economic context. Accordingly, in this paper we first presented the foundation of a decision support system to assist expert and non expert in developing a relevant model that takes into account most of the production constraints of their workshop, but also the appropriate solution method.

The proposed decision support system is based on Case Based Reasoning approach to model, represent and reuse the knowledge developed in this research area. One of the crucial steps of such a system relies on the knowledge representation and more particularly on problem and solution description. The first contribution of this paper is the proposition of an open classification and notation scheme for scheduling problems. The proposed classification system relies on the works of Blazewicz et al. (2007), Mendez et al. (2006) and Maravelias (2012) which describe classes of problems in terms of machine environment, resource and constraint characteristics, and objective function respectively represented with the triplet $\alpha|\beta|\gamma$. The field α has been reformulated from a workstation based point of view. New aspects of classification have been added to the field β, for instances: overlap, maintenance and preparation constraints, connectivity restrictions, batch size considerations. An updated classification of objective functions has been introduced in the field γ. The proposed classification and notation scheme offer new insights: mathematical models of scheduling problems (new field δ), and a new field ε related to the appropriate solving methods. These two additional fields describe the solution part of our decision support system. The main advantage of this classification is that it can be addressed to both manufacturing and chemical production problems, thereby we can take advantage and transfer the advances and breakthroughs from one domain to the other especially in discrete scheduling. As a perspective, we can improve the representation by including the work of Hai, Theiayen, and Marquardt (2011) on an ontology based approach. Their ontology is only focused on process system engineering and it must be enlarged to deal with both the process and manufacturing modelling aspects. In the same manner, we can also imagine to create a specific ontology dedicated to the formulation of mathematical model for scheduling problems whatever the application domain.

The second contribution deals with a bibliographic analysis method to extract relevant past experiences in the domain of application in order to fill the case base. Indeed, among the knowledge sources, the research literature is an important one but only if past experiences can be identified and the most relevant and representative extracted. The proposed scientific method of bibliographic analysis is used to perform a deep study to fill the case base on the one hand, and to examine the topics the more or the less examined in the scheduling domain and their evolution over time on the other hand.

This paper examines also another issue of the CBR cycle, namely the case matching. We discuss the criteria to extract a relevant case from the base. First, we studied the similarity measure and proposed several functions to evaluate this similarity based on the type of features. Then the limits of this criterion are discussed and we have improved similarity measure by introducing an additional criterion to link retrieval and adaptation requirements, namely adaptability. To guarantee an efficient retrieval, the adaptability criterion indicates the potential of a solution to respond to a wide range of problems, i.e. ease of reuse of the solution.

The capabilities of our CBR system are illustrated through a case study with typical constraints related to the building of a model for a batch process dedicated to beer production. The case study was chosen because it includes specific production constraints such as connectivity restrictions, continuous task, and storage.

There are four main perspectives for future research investigations from the point of view of CBR. First, the case base can be expanded in order to improve the efficiency of the CBR cycle. Actually the analysis is restricted to flow shop problems. The extension of study to other types of workshop is therefore an important step forwards. The second important perspective is to elaborate an adaptation module in order to modify the selected case to propose a solution that matches the problem requirements. To be satisfactory and effective, the adaptation phase needs some additional adaptation knowledge that must be extracted, stored in the base and then coded in the CBR system. We implement an introspective approach where cases provide a source from which representative adaptation knowledge can be extracted. This approach is easy to operate and implement, but it does not allow inferring explicable knowledge and it remains the question of confidence we can have to the knowledge extracted. We can improve the confidence in the adaptation knowledge acquired thanks to the elicitation of expert knowledge. Currently, a research is driven in order to formulate this expert knowledge as a constraint satisfaction problem.

As introduced before, the third perspective deals with case base maintenance to ensure CBR system sustainability. There are two kinds of maintenance: qualitative and quantitative. The former focusses on the relevance, completeness and correctness of cases. The latter concerns problem solving efficiency; the number of cases stored, reorganization of the case base (new indexation), the changes of case representation structure, retrieval algorithm. For the last perspective, we can imagine to go deeper in the CBR system by proposing model formulations. But this kind of decision support system necessitate further knowledge because various formulations can be possible for the same model, e.g. various ways to express constraints.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundation issues, methodological variations and system approaches. *Artificial Intelligence Communications, 7*, 39–59.

Allen, B. P. (1994). Case based reasoning: Business applications. *Communications of the ACM, 37*, 40–45.

Behbahani, M., Saghaee, A., & Noorossana, R. (2012). A case based reasoning system development for statistical process control: Case representation and retrieval. *Computers & Industrial Engineering, 63*, 1107–1117.

Blazewicz, J., Ecker, H., Pesch, E., Schmidt, G., & Weglarz, J. (2007). *Handbook on scheduling from theory to applications*. Springer-Verlag.

Blazewicz, J., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics, 5*, 11–24.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource constrained project scheduling: Notation, classification, models and methods. *European Journal of Operational Research, 112*, 3–41.

Castro, P. M., Grossmann, I. E., & Novais, A. Q. (2006). Two new continuous time models for scheduling of multistage batch plants with sequence dependent changeovers. *Industrial & Engineering Chemistry Research, 45*, 6210–6226.

Chang, P. C., Hsieh, J. C., & Liu, C. H. (2006). A case-injected genetic algorithm for single machine scheduling problems with release time. *International Journal of Production Economics, 103*(2), 551–564.

Choe, H., Lee, D. H., Seo, I. W., & Kim, H. D. (2013). Patent citation network analysis for domain of organic photovoltaic cells: Country, institution and technology field. *Renewable & Sustainable Energy Reviews, 26*, 492–505.

Cunningham, P. (2009). Taxonomy of similarity mechanisms for case based reasoning. *IEEE Transactions on Knowledge and Data Engineering, 21*(11), 1532–1543.

Czuczai, B., Farkas, T., Rev, E., & Lelkes, Z. (2009). New mixed integer linear programming model for solving scheduling problems with special characteristics. *Industrial & Engineering Chemistry Research, 48*, 5321–5335.

Dessouky, M., Rahimi, M., & Weidner, M. (2003). Jointly optimizing cost, service and environmental performance in demand-responsive transit scheduling. *Transportation Research Part D: Transport and Environment, 8*, 433–465.

Dong, Y., & Kitaoka, M. (1994). Implementation of a case-based production scheduling system in C language. *Computers & Industrial Engineering, 27*, 265–268.

Dzeng, R. J., & Lee, H. L. (2004). Critiquing contractors' scheduling by integrating rule-based and case-based reasoning. *Automation in Construction, 13*(5), 665–678.

Dzeng, R. J., & Tommelein, I. D. (2004). Product modeling to support case-based construction planning and scheduling. *Automation in Construction, 13*(3), 341–360.

Efstathiou, J. (1996). Any time heuristic schedule repair in manufacturing industry. *IEEE Proceedings Control Theory Application, 143*(2), 114–120.

Esquirol, P., & Lopez, P. (1999). *L'ordonnancement*. Paris: ECONOMICA.

Finnie, G., & Sun, Z. (2003). R^5 model for case-based reasoning. *Knowledge-Based Systems, 16*, 59–65.

Gowrishankar, K., Rajendran, Z., & Srinivasan, G. (2001). 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, 132*, 643–665.

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

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

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics, 5*, 287–296.

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

Gupta, J. N. D., Krüger, K., Lauff, V., Werner, F., & Sotskov, Y. N. (2002). Heuristics for hybrid flow-shops with controllable processing times and assignable due dates. *Computers & Operations Research, 29*, 1417–1439.

Hai, R., Theiayen, M., & Marquardt, W. (2011). An ontology based approach for operational process modeling. *Advanced Engineering Informatics, 25*, 748–759.

Hegyhati, M., & Friedler, F. (2011). Combinatorial algorithms of the S-graph framework for batch scheduling. *Industrial & Engineering Chemistry Research, 50*, 5169–5174.

Hunt, J. (1995). Evolutionary case based design. In Watson (Ed.), *Progress in case based reasoning, LNAI 1020* (pp. 17–31). Berlin: Springer.

Ierapetritou, M. G., & Floudas, C. A. (1998). Effective continuous-time formulation for short-term scheduling 1. Multipurpose batch process. *Industrial & Engineering Chemistry Research, 37*, 4341–4359.

Kocsis, T., Negny, S., Floquet, P., Meyer, X., & Rev, E. (2010). An application of a cocitation-analysis method to find further research possibilities on the area of scheduling problems. *CACE, 28*, 1165–1170.

Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1993). A general algorithm for short-term scheduling of batch operations. Part I: MILP formulation. *Computers & Chemical Engineering, 17*, 211–227.

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

Kutil, M., Sucha, P., Capek, R., & Hanzalek, Z. (2010). Optimization and scheduling toolbox. In *Matlab – Modelling, Programming and Simulations*. Sciyo.

Leake, D. (1996). *Case-based reasoning: Experiences, lessons, and future directions*. Menlo Park AAAI Press/MIT Press.

Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference* (pp. 556–562). MIT Press.

Maravelias, C. T. (2012). General framework and modeling approach classification for chemical production scheduling. *AIChE Journal, 58*(6), 1812–1828.

Maravelias, C. T., & Grossmann, I. E. (2003). New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Industrial & Engineering Chemistry Research, 42*, 3056–3074.

Mendez, C. A., Cerdá, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering, 30*, 913–946.

Meyer, M. (2004). *Habilitation à Diriger des Recherches*. Institut National Polytechnique de Toulouse.

Miyashita, K. (1995). Case-based knowledge acquisition for schedule optimization. *Artificial Intelligence in Engineering, 9*, 277–287.

Mikulakova, E., König, M., Tauscher, E., & Beucke, K. (2010). Knowledge-based schedule generation and evaluation. *Advanced Engineering Informatics, 24*(4), 389–403.

Negny, S., Riesco, E., & Le Lann, J. M. (2010). Effective retrieval and new indexing method for case based reasoning: Application in chemical process design. *Engineering Applications of Artificial Intelligence, 23*(6), 880–894.

Negny, S., & Le Lann, J. M. (2008). Case based reasoning for chemical engineering design. *Chemical Engineering Research and Design, 86*, 648–658.

Osareh, F. (1996). Bibliometrics, citation analysis and co-citation analysis. A review of literature. *Libri, 46*, 149–158.

Pal, S. K., & Shiu, S. C. K. (2004). *Foundations of soft case-based reasoning*. Wiley.

Pan, M., Li, X., & Qian, Y. (2009). Continuous-time approaches for short-term scheduling of network batch processes: Small-scale and medium-scale problems. *Chemical Engineering Research and Design, 87*, 1037–1058.

Pereira, I., & Madureira, A. (2013). Self-optimization module for scheduling using case-based reasoning. *Applied Soft Computing, 13*(3), 1419–1432.

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

Prasad, P., Maravelias, C. T., & Kelly, J. (2006). Optimization of aluminium smelter casthouse operations. *Industrial & Engineering Chemistry Research, 45*, 7603–7617.

Priore, P., de la Fuente, D., Puente, J., & Parreño, J. (2006). A comparison of machine-learning algorithms for dynamic scheduling of flexible manufacturing systems. *Engineering Applications of Artificial Intelligence, 19*(3), 247–255.

Rad, S. F., Ruiz, R., & Boroojerdian, N. (2009). New high performing heuristics for minimizing makespan in permutation flow-shops. *Omega, 37*, 331–345.

Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research, 165*, 479–494.

Schilling, G., & Pantelides, C. C. (1996). A simple continuous-time process scheduling formulation and a novel solution algorithm. *Computers & Chemical Engineering, 20*, S1221–S1226.

Schmidt, G. (1996). Modeling production scheduling systems. *International Journal of Production Economics, 46–47*, 109–118.

Schmidt, G. (1998). Case-based reasoning for production scheduling. *International Journal of Production Economics, 56–57*, 537–546.

Shaik, M. A., & Floudas, C. A. (2008). Unit specific event based continuous time approach for short term scheduling of batch plants using RTN framework. *Computers & Chemical Engineering, 32*, 260–274.

Sitarz, R., Kraslawski, A., & Jezowski, J. (2010). Dynamics of knowledge flow in research on distillation. *Computer Aided Chemical Engineering, 28*, 583–588.

Smith, K (1981). Cocitation analysis. In W. G. Potter, *Bibliometrics* (pp. 83–106). Illinois.

Smyth, B., & Kean, M. T. (1998). Adaptation-guided retrieval: Questioning the similarity assumption in reasoning. *Artificial Intelligence, 102*(2), 249–293.

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

Sundaramoorthy, A., & Maravelias, C. T. (2008). Modeling of storage in batching and scheduling of multistage processes. *Industrial & Engineering Chemistry Research, 47*, 6648–6660.

Suhami, I., & Mah, R. S. H. (1981). An implicit enumeration scheme for the flow-shop problem with no intermediate storage. *Computers & Chemical Engineering, 5*, 83–91.

Widmer, M., Hertz, A., & Costa, D. (2001). Les métaheuristiques. *Hermes Edition*, 55–93.

Zhou, M., Chen, Z., He, W., & Chen, X. (2010). Representing and matching simulation cases: A case based reasoning approach. *Computers & Industrial Engineering, 59*, 115–125.

Zhou, M., Son, Y., Chen, Z., Zhang, Q., & MA, J. (2007). Conceptual simulation modeling: Patterns and knowledge representations. *International Journal of Industrial Engineering, 14*(1), 73–83.