Contribution to the Optimization and Flexible Management of Chemical Processes

Sergio Ferrer Nadal

Chemical Engineer

Barcelona, June 2008

Departament d'Enginyeria Química

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

Universitat Politècnica de Catalunya

Contribution to the Optimization and Flexible Management of Chemical Processes

Thesis

In partial fulfillment of the requirements for the degree of Doctor of Philosophy at the Universitat Politècnica de Catalunya, directed by Prof. Lluis Puigjaner Corbella and Dr. Moisès Graells Sobrè, in Barcelona, June 2008.

Sergio Ferrer Nadal

Chemical Engineer

To my family and all those who have stood by my side.

Research consists of seeing what everybody has seen, and thinking what nobody has thought.

Albert Szent-Györgi(1893-1986)

Summary

The chemical industry has become increasingly competitive over the past decades. Companies are required to adapt to changing market conditions and meet stricter product specifications. While globalization has opened new markets for the chemical industry, it has also increased the competitor pool, giving an advantage to companies with more efficient and highly integrated plants.

In this context, the main aim of this thesis is to demonstrate new concepts and computational methods that exploit process flexibility to enhance plant profitability under transient operating conditions. These methods ensure that safety and product quality requirements are consistently met. This thesis makes contributions to the optimization and management of production in plants ranging from small batch plants to large capacity continuous processes.

First, this thesis addresses the management of continuous processes, in which similar products are mass produced. Continuous processes can achieve the highest consistency and product quality by taking advantage of economies of scale and reduced manufacturing costs and waste. However, in order to remain competitive in the market, plants are required to dynamically adapt their processes to fit the continuously changing market and operating conditions. The supervisory control system presented in this part of the thesis decreases the system reaction time to incidences and re-optimizes the production in real time if the opportunity for improved performance exists.

Next, this thesis addresses the management of semicontinuous processes, which allow more customized and flexible operation. Semicontinuous processes run with periodic start-ups and shutdowns to accommodate frequent product transitions. This thesis proposes an optimization model that creates improved production schedules by introducing a new concept of flexible manufacturing

that allows production rate profiles to be programmed within each operation campaign.

The major part of the research work of this thesis deals with the operational management of batch processes, which are mainly used for the production of high value-added chemicals. Batch processing offers the advantage of increased flexibility in product variety, production volume, and the assortment of operations that can be processed by a particular piece of equipment. However, the trade-off is that production scheduling is significantly complicated by the large number of batches involved with different production paths. In order to avoid the complexity of managing transfer operations, the assumption of negligible transfer times is generally accepted in batch scheduling. Conversely, this thesis highlights the critical role that transfer operations play in the synchronization of tasks and in determining the feasibility of production schedules.

Continuing to focus on batch plant operation, this thesis demonstrates that the use of the concept of flexible recipes enhances the operation batch plants within an uncertain environment. Recipe flexibility is considered as an additional opportunity for reactive scheduling as well as a proactive way to reduce the risk of meeting unfavorable scenarios.

Finally, this thesis examines pipeless plants as an alternative to batch plants. In the search for more competitive and effective ways of production, flexibility of batch plants for producing a large number of products is limited due to the need for equipment, piping and frequent cleaning tasks. Pipeless plants have enhanced flexibility over batch plants, because the material is moved along its production path through moveable vessels. This part of the thesis contributes to the optimization of the management of pipeless plants by proposing an alternative formulation for solving short-term scheduling problems.

In summary, this thesis provides novel modeling approaches and solution methods aimed at supporting the decision-making process in plant production scheduling which exploit the existing flexibility in chemical processes. The main advantages of each contribution are highlighted through case studies.

Resumen

La industria química ha experimentado en las últimas décadas un aumento en la competencia por la cual las empresas se ven obligadas a adaptarse a un mercado cambiante y cada vez más exigente. Aunque la globalización ha abierto nuevos mercados, ha incrementado también el número de competidores, de tal manera que sólo las empresas que usen las plantas más integradas y eficientes podrán mantenerse en el negocio.

En este contexto global, el principal propósito de esta tesis es desarrollar métodos que exploten la flexibilidad de los procesos, con el objetivo de aumentar la eficiencia de las plantas y asegurar los requerimientos de seguridad y calidad de los productos. Esta tesis contribuye a la optimización y a la gestión de la producción desde pequeñas plantas que usen procesos discontinuos hasta grandes plantas de procesado continuo.

En primer lugar, esta tesis trata la gestión de los procesos continuos en los que suelen fabricar productos muy similares a gran escala. La gran ventaja de los procesos continuos es que pueden conseguir mayor consistencia en la calidad de los productos y que pueden aprovechar las economías de escala que reducen los costes y residuos. Sin embargo, la industria química para mantenerse competitiva necesita adaptar continuamente sus procesos a las condiciones del mercado y de operación. El sistema de control supervisor presentado en esta parte de la tesis disminuye el tiempo de reacción frente a incidentes en los procesos continuos y re-optimiza la producción en tiempo real, si existe posibilidad de mejora.

A continuación, esta tesis trata la gestión de los procesos semicontinuos que permiten una operación más flexible y personalizada. Los procesos semicontinuos operan con puestas en marcha y paradas periódicas para acomodar las frecuentes transiciones entre diferentes productos. Esta tesis presenta un

nuevo concepto de fabricación flexible que permite programar perfiles variables de velocidad de producción dentro de cada campaña de producción.

La mayor parte del trabajo de investigación de esta tesis se dedica a la planificación de la producción en los procesos discontinuos por lotes, utilizados principalmente en la producción de productos químicos con alto valor añadido. Estos procesos ofrecen varias ventajas respecto a los procesos continuos y semi-continuos debido a la mayor flexibilidad para acomodar diversos productos, diferentes capacidades de producción, y la posibilidad de realizar operaciones completamente diferentes en los mismos equipos. Sin embargo, la obtención del plan de producción óptimo se complica al aumentar la complejidad de la planta y/o el número de lotes a planificar. La simplificación de considerar tiempos de transferencia despreciables es generalmente aceptada en la literatura para evitar la complejidad del manejo de las operaciones de transferencia. En cambio, esta tesis pretende resaltar el papel crítico que juegan las operaciones de transferencia en la sincronización de tareas, y en la consiguiente determinación de planes de producción factibles.

Siguiendo con los procesos por lotes, esta tesis demuestra que el uso del concepto de recetas flexibles mejora la operación de los procesos en ambientes de producción con mucha incertidumbre. La flexibilidad de las receta se considera como una oportunidad adicional, tanto para la planificación de la producción reactiva como proactiva, reduciendo el riesgo de llegar a resultados económicamente desfavorables.

Finalmente, esta tesis presenta las plantas discontinuas sin tuberías como una alternativas a las plantas por lotes clásicas. En la búsqueda de formas más competitivas y efectivas de producción, la flexibilidad para producir un elevado número de productos en plantas por lotes es limitada debido a la necesidad de equipos fijos conectados por tuberías y frecuentes tareas de limpieza. Las plantas sin tuberías presentan una mayor flexibilidad ya que el material se transfiere entre estaciones de procesamiento usando equipos que se mueven dentro de la planta. El trabajo presentado en esta parte de la tesis contribuye a la mejora en la gestión de este tipo de plantas proponiendo una formulación alternativa a las encontradas en la literatura que resuelve el problema de la planificación de la producción.

En resumen, esta tesis desarrolla nuevas estrategias de modelado y métodos de resolución encaminados al soporte de la toma de decisiones que explotan la flexibilidad intrínseca de los procesos químicos. Las principales ventajas de cada una de las contribuciones de esta tesis se demuestran mediante su aplicación a diferentes casos de estudio.

Acknowledgments

If I have seen further it is by standing on the shoulders of giants.

Isaac Newton (1642-1727)

Working as a Ph.D. student during the last four years at the Chemical Engineering Department of Universitat Politècnica de Catalunya has been a wonderful and challenging experience for me. I am in debt to many people, without whose support this thesis would not have been completed. Here is my small tribute to all of these people.

First of all, I wish to express my gratitude to my thesis supervisors, Emeritus Prof. Luis Puigjaner and Dr. Moisés Graells, for their guidance and constant support throughout my research work. I would like to show my appreciation to all my colleges from the CEPIMA group for their friendship and collaboration. Furthermore, I would like to thank Prof. Ignacio Grossmann from Carnegie Mellon University and his group for their kind welcome that contributed to my pleasant stay in Pittsburgh. I also owe to my warmest gratitude to my family for their affection; especially my Dad, Mom, Auntie Pili and my brother Emilio, to Katie for her writing suggestions as well as to all my friends.

Finally, I sincerely acknowledge the financial support received from Departament d'Educació i Universitats de la Generalitat de Catalunya and the European Social Fund by means of an FI grant.

Agradecimientos

Si he visto más lejos es porque estoy sentado sobre los hombros de gigantes.

Isaac Newton (1642-1727)

Trabajar estos últimos cuatro años como estudiante de doctorado en el Centro de Ingeniería de Procesos y Medio Ambiente (CEPIMA) ha sido una gran experiencia para mí. Me considero en deuda con muchas personas sin las que no podría haber realizado esta tesis. Este es mi pequeño tributo para todas estas personas.

En primer lugar, me gustaría expresar mi mayor agradecimiento a mis directores de tesis, Luis Puigjaner y Moisès Graells, por sus orientaciones y constante apoyo a lo largo de mi trabajo de investigación. Me gustaría mostrar también mi aprecio a todo mis compañeros del grupo CEPIMA por su colaboración y amistad. Además, agradezco al Profesor Ignacio Grossmann y a todo su grupo por mi inolvidable estancia en Pittsburgh. Debo mi más afectiva gratitud a mi familia; especialmente a mi padre, a mi madre, a mi tía Pili y a mi hermano, a Katie por sus sugerencias en la redacción, así como a todos mis amigos.

Finalmente, agradezco sinceramente el apoyo económico recibido del Departament d'Educació i Universitats de la Generalitat de Catalunya y el Fondo Social Europeo mediante una beca FI.

Contents

Chapter 1

Introduction

In this chapter, the main concepts underlying the studies carried out in this thesis are introduced. An overview of the current challenges in the chemical industry that motivate the development of this body of work is also included.

1.1 An overview of the Chemical Industry

The chemical industry is one of the European Union's most international, competitive and fruitful industries. In 2006, worldwide chemical sales were estimated to be 1.6 billion euros, with the European Union leading exports and imports making 31% of all sales (figure 1.1). In addition, as shown in figure 1.2, five European companies are in the top ten of companies with most increased revenues in 2006.

Germany is the largest European producer, followed by France, the United Kingdom and Italy. Spain is the fifth European producer, accounting for approximately 7% of total production. The Spanish industry is made up of more than 3.600 companies, with a net revenue of 38 million euros (10% of the GDP) and more than 500.000 jobs. It is worth mentioning the important role played by Catalonia which generates almost half of the national production.

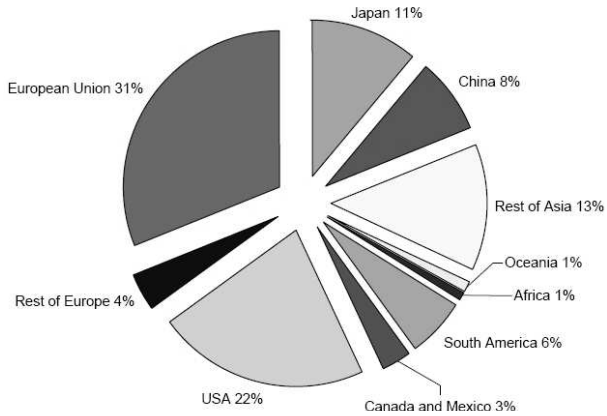

Figure 1.1: Geographical breakdown of world chemical sales.

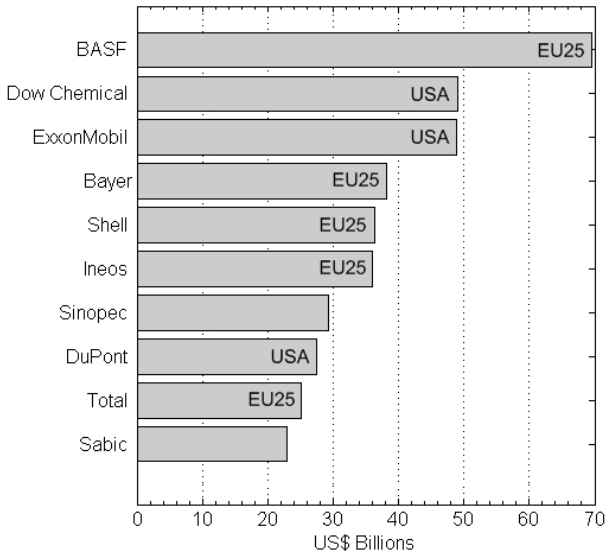

Figure 1.2: Revenues of the top ten chemical companies in 2006.

However, in spite of the continuous positive trend of European sales, global sales are growing faster. The strongest competition comes from the Asian countries, specifically Japan, China and India which have enormously increased their sales over the latest years.

The sectors that the European chemical industry most heavily relies on are basic chemicals followed by pharmaceuticals, specialty chemicals and consumer products. Basic chemicals cover petrochemicals, polymers and basic inorganics, and represents almost 43% of the sales. Pharmaceuticals accounts for 28% of the sales and specialty chemicals, which are auxiliaries for industry, dyes, pigments, crop protection, paints and inks, for 19%. The remaining 10% of production is directly sold to consumers as soaps, detergents, perfumes and cosmetics.

In view of this overall scenario, the following trends have been identified as the major issues that the current industry has to deal with.

Globalization of markets Advances in transportation and especially in communication technology have led to greater global interaction in practically every aspect of life. This globalization has opened new markets but at the same time has resulted in growing worldwide competition. This has forced companies to increase corporate internationalization and mergers to gain a competitive advantage by reducing redundancies and increasing economies of scale.

The nature of the products produced by richer countries has also changed, because simpler chemicals can be produced at a lower cost in less developed countries. For instance, the increasing competition from Eastern and Asian countries to cheaply produce basic chemicals has caused a decrease in the European production of basic chemicals from 61% in 1977 to 38% in 2007. The European industry has responded to this growing competition in the marketplace by placing a greater focus on adapting products to meet the requirements of the target market and on producing more specialty high-added value products.

Financial importance Capital markets have increased their expectations for return on investment despite the cyclical behavior inherent in most chemical industries. Increased investor's expectations are so demanding that it has become very difficult to attain the economic targets based on traditional technologies. This is another fact which is pushing the industry to work towards more efficient and leading technologies.

High customer expectations With higher education and globalization, increased consumer sophistication has become an extra motivation for the de-

velopment of new ideas, approaches and performances. Companies are also forced to establish efficient processes to achieve consistent quality in order to satisfy their customers.

Environmental concern Besides the obvious concern for the welfare of the workers, protection of the environment is putting pressure on industry to pollute less, recycle more and develop processes with close to zero waste. Reduction of CO_2 emissions ,which is the supposed main human contributor to global warming, and waste water management were two of the most important concerns at the end of the 20th century and will continue to be very important issues in the 21st century.

1.2 Information management : a new Chemical Engineering paradigm

The aforementioned issues pose many new challenges to process industries in today´s world. In order to remain competitive, chemical companies are obliged to make efforts to increase the efficiency and flexibility of their plants to adapt to the current production environment. The appearance of computers and digital systems has provided chemical engineers a powerful tool to face these challenges. Currently, digital systems provide a large a amount of information, while computers offer the possibility to rapidly analyze this information to guide the decision-making process for complex and even previously intractable problems. As a result, information management has burgeoned as an important area that opens up a set of opportunities not traditionally assigned to chemical engineers.

This technological revolution is described by Roger Sargent in a very interesting retrospective article (Sargent, 2005) in which he states that "*with the advent of the computer, the immediate task was to replace the many elegant graphical solution procedures then in general use by numerical algorithms, and engineers were faced with the need to understand available general numerical techniques or develop new ones of their own*". Sargent emphasizes that the appearance of digital computers not only allowed engineers to make calculations faster, but also was a tool for creating new approaches to problem solving. The advent of computers had greater implications than just as a fast calculator. The solutions of rudimentary solvers of the last century were no longer sufficient because of the new approaches to problem solving. No longer did

the assumptions and problem simplifications taken 50 years ago need to be made, because computers now could handle the complexity caused by the use of more constraints. This fact opened new research fields for all engineering disciplines. Again, a new paradigm had been added in the chemical engineering which allowed new problem approaches relying on an increased capacity for information management.

The challenging task of developing new approaches for transforming information into knowledge was rapidly adopted by chemical engineering. This gave birth to a new discipline called Process System Engineering (PSE) entitled to develop new methods and tools that allow industry to meet its needs (Grossmann, 2004). Pekny (2002) describes the different stages that process system engineers must accomplish in order to deal with these tasks. First, a comprehensive analysis of the problem is done in order to understand its essential features. Second, a model is formulated, which captures the essential features and provides a concise statement of the goal and system constraints. Finally, an algorithmic tool for obtaining answers from the model is developed. It is worth mentioning that computers have limited capacity, so problems decomposition and generation of algorithms and procedures to deal with these problems is necessary. Finally, these answers must be interpreted to determine the range over which they are valid and how to implement them in practice.

1.3 Process Flexibility

The current situation in industry and the newly available technologies motivate the type of problem solving methods laid out in this thesis. The goal is to improve the operational management of chemical plants by exploiting their flexibility to increase the efficiency. In this section, we review and analyze the opportunities for enhancing the flexibility of the different types for processing materials in the chemical industry.

1.3.1 Continuous processes

In the continuous processing mode, units are continuously fed and yield a constant product flow. For mass production of similar products, continuous processes can achieve higher consistent product quality, taking advantage of the economies of scale and reducing manufacturing costs and waste. Processes from the petrochemical industry are usually good examples of continuous pro-

duction. From the fifties and sixties, continuous processes were scaled up to capacities of millions of tons per year becoming a successful challenge for the chemical engineering profession. This level of production made the chemical industry one of the driving force of the worldwide economy and development.

An essential simplifying assumption in continuous processes is that the production mode is steady state due to its relatively slow dynamics. This means that at a given point in the system there is no change in process conditions over time. However, invariable conditions over time are clearly unreal due to the uncertain nature of the industrial environment which make operating conditions actually change very often. The validity of this simplification is determined depending on whether the focus is on the process design or operation. Most probably, a steady state condition is a satisfactory assumption during the synthesis and design of a process. However, in the operational management of a continuous process, the appearance of unexpected events motivates frequent transitions in the operating conditions that makes an steady state assumption no longer valid. This uncertainty in the operation of continuous processes is motivated by these two main reasons:

1. *Internal disturbances*, which are the physical drifting of the same process operating conditions, such as the fouling of a heat exchanger, deactivation of a catalyst in a reactor, temperature and flow-rate fluctuation of the feed, etc.

2. *External disturbances*, which are changes in market demands, raw materials supply and economic conditions that may result in the change of product specifications and plant schedules.

Therefore, to be competitive in the market and keep reliable profit margins, plants are required to dynamically adapt their processes to fit changing conditions. Although local controllers can guide their respective units to the given set-point, they do not take into account the effect that they have on other unit operations in the process and on the economics of the entire plant operation.

Real time optimization (RTO) has attracted much attention because of its capacity for improving the operating profit of plants beyond conventional process control. Thus, RTO becomes an essential tool to take advantage of the flexibility of continuous chemical processes when the opportunity exists for further improvement of plant operation. Marlin and Hrymak (1997) provide

an excellent survey of RTO systems and list a series of features and opportunities for a successful application which basically consists of the capacity of the process to give a positive response to the following questions:

1. *Does any degree of freedom exist for the optimization?*
   Adjustable optimization variables must exist after higher priority safety, product quality and production rate objectives have been achieved. Decision variables that are not already adequately being optimized by other applications must be available. In fact, process variables apparently available for optimization may already be optimally set by advanced process control or by the production planning.

2. *Does profit significantly change as the value of the optimization variables are changed?*
   This question implies the economic justification of a RTO system which should be consciously assessed accounting also for future scenarios.

3. *Are changes in unit operation and/or economics frequent?*
   RTO is only justified if disturbances occur frequently enough for real-time adjustments to be required. Simple operating changes are insufficient reasons to defend the implementation of a RTO system. These changes must result in frequent changes in the optimum process operation. If changes are sporadic or limited, periodic off-line open-loop optimization may be satisfactory to reset the optimum strategy by manual implementation.

4. *Is the process sufficiently complex?*
   The process must be complex enough, so that determining the proper values for the optimization variables cannot be achieved by selecting them from several operating procedures.

Under these premises, real-time optimization techniques have enjoyed significant industrial interest in past years because of their capability to boost the profitability of plants. The objective is to maintain the plant operation near an economic optimum in the face of disturbances and other external/internal changes. Selection of the correct set of operating conditions by an on-line optimization system is reported to be worth about 3-5% of the economic value of a process (Cutler and Perry, 1983). Furthermore, the cost of engineering a RTO system is more easily justified with large scale continuous plants when cash flows are available.

Although the increasing demand of higher productions make continuous processes appear to be the most efficient and productive way to produce scarce

products, in the last years this has been changing. The tendency is now moving from constructing huge continuous processes to smaller but more flexible and diversified semicontinuous and batch plants, which are next analyzed.

1.3.2 Semicontinuous processes

In contrast to the more widespread and rigid continuous production mode typical of mass production, semicontinuous processing offers a more customized operation for very dynamic and uncertain environments.

Semicontinuous operations are characterized by their processing rate, running continuously with periodic start-ups and shutdowns for frequent product transition. The processing times of semicontinuous processes are relatively long periods of time called campaigns, each dedicated to the production of a single product. Typical campaign lengths range from a few hours to several days. The products from semicontinuous processes are often used as feedstocks for downstream processes that produce more specialized final products (Papageorgiou and Pantelides, 1996). Most process plants in the chemical industry combine continuous operations and batch processes in their product processing routes thus working in semicontinuous mode.

The importance of semicontinuous and batch plants is that production is more flexible and equipment can be more efficiently utilized for the simultaneous processing of medium size amounts of several products than in a continuous facility.

1.3.3 Batch processes

In a batch process, materials are fed at the beginning and removed after the processing finishes over a finite and relatively short period of time. In a multistage plant, intermediate material undergoes a sequence of processing activities using a series of pieces of equipment until the final product is obtained. Here the production mode is considered dynamic because the composition of the species in the batch equipment changes continuously over time.

Although batch processing has been traditionally associated with specialty chemicals and products of high-added value, the demand patterns can be so unpredictable that profitability may be only achieved taking full advantage of the inherent flexibility of a batch plant. Therefore, in order to reduce the risk of new investment, batch plants are preferred as an adequate and flexible

answer to the variability in the supply of raw materials, the manufacturing of diverse products and the instability of product demands.

Activities in batch plants must be coordinated to produce a large number of products. Therefore, beside the management of the process conditions, the scheduling of these operations is another key factor for ensuring optimal management of batch processes. Tasks, resources and recipes are the components that must be considered in scheduling. A task is an operation which changes the nature of materials and/or any other resources, such as transportation, quality control, cleaning, changeovers, etc. Resources include the physical facilities used to execute specific manufacturing operations as well as any other measurable entity employed or produced in manufacturing such as materials, human labor, and utilities. A recipe specifies the sequences of tasks that must be performed for manufacturing a given product. Thus, the scheduling problem can be considered a decision making process which deals with the allocation of a set of limited resources over time to manufacture one or more products consisting of a series of tasks following a production recipe.

According to their purpose, batch processes are generally classified into multiproduct and multipurpose processes. In multiproduct plants, each product requires the same sequence of processing tasks because they all follows a similar recipe. In multipurpose plants, each product may be manufactured by a different sequence of processing tasks and there may be more than one way to produce the same product. Flow patterns are not straight lines as in the multiproduct case and some units may be used to perform non-consecutive operations for the same product (backtracking). Multipurpose plants result in more flexible operation, which can be optimized to decrease equipment idle time to more efficiently utilize critical equipment units. Due to historic association to the workshop problem in operation research, multiproduct and multipurpose plants are also called flowshop and jobshop, respectively.

Because of the staged nature of both multiproduct and multipurpose plants, a convenient intermediate storage policy turns out to be a key factor for the flexible management and debottlenecking of batch plants. Proper storage tanks allocated in strategic places of a plant can help to reduce idle times between stages by storing intermediate materials and thus increasing equipment utilization and productivity (Ku and Karimi, 1988). In this way, intermediate storage can mitigate the material flow imbalance of feedstock materials and products (Romero et al., 2004). Efficient management of storage resources must also be carried out in order to reduce the capital investment and environmental impact caused by more frequent cleaning. There exist five major categories of

managing intermediate storage:

- *Unlimited Intermediate Storage (UIS).* This case is unrestricted because the storage capacities are assumed to be unlimited. The best attainable solution with the shortest production times is obtained using this policy, so this case is considered the best case scenario upper limiting bound for all other solutions.
- *No Intermediate Storage (NIS).* In this case, there are no storage tanks available for intermediate materials, but the materials can be stored inside the processing unit, waiting to be transferred to the next unit once it has been freed.
- *Zero Wait (ZW).* This policy restricts the NIS policy by avoiding the alternative use of processing units as storage facilities for intermediate materials. This policy is usually used in cases where the materials are unstable products that must be transferred to the next processing unit immediately after completion. This is the most restrictive policy and constitutes a lower limiting bound.
- *Finite Intermediate Storage (FIS).* Here, limited storage capacity is available in terms of the number of storage units, their capacities and connections between processing units and tanks. Each storage unit can temporarily hold any product (shared), some products (dedicated) or just one product (exclusive).
- *Common Intermediate Storage (CIS).* This is a generalization of the FIS policy in which intermediate storage is commonly used throughout the whole process network.

Finally, it also very important to mention the important role of scheduling in the management of uncertainty in batch processes. The most common sources of uncertainty may be internal to the same process such as model parameters, processing times and equipment availability, or external sources such as amount of demand and/or due date and price/cost of product/raw materials. A schedule generated by a deterministic model based on nominal values of the parameters may be infeasible upon realization of the uncertain parameters. Therefore, the flexibility of batch plants is a very potent tool that can be used to deal with the uncertainty. Short-term uncertainties are usually treated through reactive scheduling (rescheduling), while longer-term uncertainties are usually addressed through stochastic optimization (proactive

scheduling). A reactive scheduling strategy may be executed on-line and aims to generate a feasible schedule as soon as possible to minimize deviations from the schedule in progress in order to reduce the disruption in the production. In contrast, the aim of a proactive scheduling strategy is to take into account the process uncertainties in order to produce robust schedules that avoid the consequences of unexpected events.

1.3.4 Pipeless plants

In looking for more competitive and effective ways of production, we may find that the flexibility of batch plants for producing a large number of products is limited due to the need for equipment, piping and frequent cleanings. Pipeless plants are the quintessence of flexibility because material is moved through moveable vessels transported by AGV (automated guided vehicles) between processing stages. Therefore, they minimize wastes by reducing piping and avoiding complex cleaning operations. Processing is usually carried out in the same vessel used to transport the materials through a number of fixed stations. This allows for multiple production tasks processed simultaneously increasing the efficiency and shareability of all the process equipment and peripheral facilities.

Pipeless plants were developed to increase the plant flexibility and scalability. Niwa (1993) draws an analogy between a pipeless batch plant and a chemical laboratory. In the laboratory, a beaker or flask is a "moveable vessel", and the laboratory's stationary equipment consists of a number of "processing station", such as weighing balances, mixers and Bunsen burners. To synthesize a product, the chemist generally uses a single flask, moving it to the appropriate processing station to carry out a specific operation.

Pipeless batch plants are already in use to manufacture products such as lubricant oils, inks and paints. Without the maze of the piping network, pipeless batch plants permit a wide range of products to be handled with frequent changeover to meet market opportunities. Furthermore, the economical benefits of pipeless plants are significant as the time in which new recipes or improved recipes are ready for commercial production is greatly reduced. Using pipeless plants allows faster market introduction of products and more flexible production based on demand.

However, the use of pipeless plants is actually restricted to small productions because of the limited capacity of the vessels which need to be transported. In fact, the higher operational cost associated to pipeless plants is only

justified by the introduction of new products in the market or to manufacture products of high added value. Furthermore, the consideration of moveable vessels and different lay-out configurations determine an additional complexity for the scheduling problem of this type of plants.

1.4 Thesis outline

The general aim of this thesis is to prove that flexible production may be exploited in plants ranging from small batch plants to large volume continuous processes. This thesis work contributes to the development of new concepts, methods and techniques to better describe and use the inherent batch process flexibility in industrial practice.

After a detailed review of the state-of-the-art of the research in chapter 2, this thesis examines different types of processes, analyzing their capacity for flexible production following the scheme shown in figure 1.3. It is demonstrated that flexibility can be exploited and monotonically increases from large-sized continuous processes, passing through semicontinuous and batch processes, until batch pipeless plants.

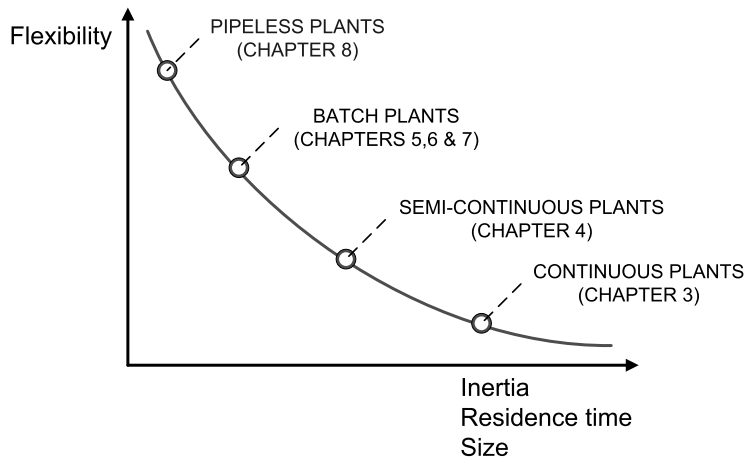

Figure 1.3: Thesis outline

Chapter 3 exploits the potential flexibility of continuous processes, presenting a supervisory control system able to give a fast and optimized response to abnormal plant events.

As a link between continuous and batch processes, Chapter 4 analyses the production management of semicontinuous processes. The work of this chapter proposes a new flexible concept of fabrication which considers production rate transitions within a single campaign.

Next, Chapter 5 addresses the challenging modeling of batch scheduling, incorporating a new approach for the synchronization of transfer activities.

Chapters 6 and 7 emphasize the role of process flexibility in the management of uncertainty within a batch plant. Flexibility is described as the ability of a system to operate under changing conditions or variations of uncertain parameters. Chapter 6 proposes a rescheduling strategy to address the uncertainty using flexible recipes with adjustable processing times. Alternatively, Chapter 7 makes use of flexible recipes to address the uncertainty of batch processes in a proactive way. This chapter also introduces the risk management associated with the decision making process under uncertainty.

Chapter 8 introduces the scheduling of pipeless plants and gives a efficient alternative for manufacturing small size amounts of chemicals of high added value.

Finally, conclusions, remarks and future work are presented in Chapter 9. Here, a summary of the results of this thesis is given, emphasizing the different level of flexibility management that have been achieved.

Chapter 2

State of the art and literature review

Flexibility can be defined as the ability of a system to keep its performance under changing conditions. The first attempt to consider the potential flexibility in a process should be made at the *design stage*. The concept of flexibility in a design has been recursively treated in the literature over the recent years. As a result, a theory was developed that allowed abstract flexibility concepts to appear during the design stage in order to be translated into mathematical terms. The aim was to obtain a process design able to maintain feasible operation and economic profitability over a set of uncertain conditions.

Following this direction, Halemane and Grossmann (1983) developed a feasibility test in order to ensure profitable operation even under the worst case scenario conditions. The varying parameters were deterministically described by an interval with lower and upper bounds. These authors suggested an enumeration approach that identified the vertices of the uncertainty region as the appropriate points to evaluate feasibility. Swaney and Grossmann (1985) proposed a branch and bound method to calculate a flexibility index, which is defined as the range in which uncertain parameters can be dealt with by a specific design or operational plan. An important idea proposed in this work was the combined consideration for the adjustment of design and control decisions in order to minimize the uncertainty impact. However, a drawback of this kind of approach is the number of optimization problems that must be

solved which increases exponentially with the number of uncertain parameters. Grossmann and Floudas (1987) reduced the calculation of the flexibility index to a MINLP problem and applied the active constraint method to evaluate the feasibility at non-vertex points. Pistikopoulos and Grossmann (1989a,b) developed a method for the retrofitting problem under uncertainty and Pistikopoulos and Ierapetritou (1995) formulated this problem as a two-stage stochastic programming optimization. Straub and Grossmann (1993) redefined the concept of flexibility as stochastic flexibility so that the uncertain parameters are described by joint probability density functions. Ostrovsky et al. (1997) developed bounding algorithms for the problem considered by Halemane and Grossmann (1983). Floudas et al. (2001) evaluated the flexibility index based on the principles of the deterministic global optimization algorithm αBB, while Bansal et al. (2002) applied piece-linear approximations. More recently, Ostrovsky and Volin (2006) tried to incorporate more operational data of the processes and Malcolm et al. (2007) took into consideration the process dynamics to carry out the flexibility analysis.

In this thesis, the focus is on a later *operational stage*, once the design of the plant is already fixed. Exploiting flexibility is a key feature for the improvement of the operational management of processes, but it can be only accomplished when the process design allows for it.

This chapter presents the state of the art and literature review of the operational management and optimization of chemical processes, which constitutes the basis for the development of this thesis.

2.1 Optimization and flexible management of continuous processes

For continuous processes, a Real Time Optimization (RTO) strategy can take advantage of process flexibility to increase plant profitability by deciding the optimum operating policy after safety and quality objectives have been satisfied.

Typically, RTO systems rely on a process model. However, when a good process model is too difficult or expensive to obtain, direct-search methods (Garcia and Morari, 1981; McFarlane and Bacon, 1989) are able to track the optimum operating trajectory estimated from the gradient of plant data. Thus, the direction that improves the plant performance is estimated and the plant is moved along that direction. One of the most applied direct search methods

is the Nelder-Mead simplex, which was proposed in 1965 and has been used extensively in many fields, especially in chemistry and chemical engineering. However, the use of direct search methods results very restricted because they are limited to very simple and small processes.

Model-based methods arise as the most powerful approach to deal with the on-line optimization of complex systems with a large number of variables and operational constraints. They make use of a mathematical model to represent the process, and the best operating conditions are determined by optimizing the plant performance subject to this model.

Conventional RTO strategy is based on steady state operation. This assumption greatly simplifies the modeling task, but raises other issues associated with model validation. A typical steady state RTO system contains the elements shown in the closed loop of figure 2.1.

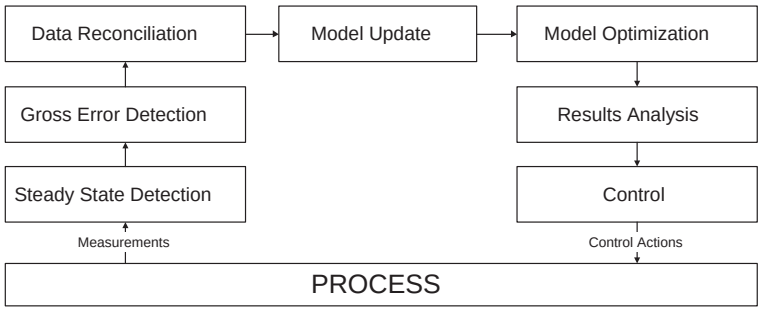

Figure 2.1: Real time optimization closed loop.

2.1.1 Steady state detection

The loop begins by ensuring that the plant is operating under steady-state conditions. Most of the available techniques for steady state detection are based on the following algorithm:

1. Selecting a data window,
2. Computing either the average, variance, or regression slope on the data window,
3. Comparing this value with that on the previous data window using an appropriate statistical test (Bethea and Rhinehart, 1991).

2.1.2 Gross error detection and data reconciliation

Once the existence of steady state is ensured, measurements are validated through Gross Error Detection (GED) and Data Reconciliation (DR). GED methods are developed based on various statistical tests to remove wrong measurements due to the malfunctioning of instruments or process errors as leaks. DR adjusts process measurements with random errors so that they satisfy the material and energy balances of the process model. Comprehensive reviews of GED and DR methods are provided by Crowe (1996) and Yang et al. (1995), respectively.

2.1.3 Model updating

Validated measurements are employed to update the model parameters by reducing the structural mismatch between the model and the plant. The key factors that enable the RTO system to closely track optimal operating conditions are the measurements used, parameters updated and the parameter estimation method (Britt and Luecke, 1973; Chen et al., 1998).

2.1.4 Optimization

The core of the RTO system is the model-based optimizer which calculates the optimal operating policy. In order to maximize accuracy, RTO usually employs fundamental process models. Along these lines, the work of Yip and Marlin (2004) analyzes the accuracy of the model to demonstrate that the plant behavior is a key factor that allows an RTO system to closely track the optimal conditions. Furthermore, due to the non-linear nature of the vast majority of chemical processes, a large scale process model will be formulated that calls for the use of adequate non-linear algorithm solvers.

In most applications, Successive Quadratic Programming (SQP) is chosen as the non-linear optimization algorithm. In fact, the use of SQP algorithms in the applied mathematics and process system engineering communities set the precedent for the development of RTO. The SQP algorithm is a generalization of Newton's method for unconstrained optimization in which a quadratic subproblem is solved at each major iteration. The first appearance of SQP can be traced to the early sixties, but numerical difficulties impeded widespread application. The application of quasi-Newton methods and analysis of exact penalty functions led to the first efficient SQP algorithms created by Han (1977). From

this starting point, the next decade saw algorithmic developments which led to advanced features. Sargent (1997) gives a detailed overview of the SQP techniques, while some of its applications are summarized in the book by Biegler et al. (1997).

2.1.5 Results analysis

Before optimal set-points are implemented in plants by means of controllers, on-line statistical analyses are required to decrease the frequency of unnecessary changes. To deal with this problem, Miletic and Marlin (1998) applied multivariable statistical hypothesis tests based on control charts in order to distinguish between high-frequency disturbances propagated through calculations and significant changes in plant optimization variables. Finally, it is important to recognize that the opportunity to accept or reject a solution may be given to the plant operator.

2.1.6 Dynamic Real Time Optimization

While the RTO based on a steady-state model is the current standard, the execution frequency of this optimization can be a severe limitation. This is specially critical in the case of plants with very long transient dynamics due to the presence of recycle loops, transportation delays, or large intermediate storage capacities. Once a change occurs, it may take a very long time for the plant to reach the new steady state. Actually, most plants are rarely in steady state conditions because of the frequency with which unexpected events take place.

To overcome the drawbacks existing in steady-state RTO, many researchers have suggested the use of dynamic RTO. This type of systems is aimed at the possibility of not only optimizing the steady state performance of continuous processes, but also the dynamic transitions to a new steady state. This transition may be either in response to new steady-state optimization results or to changes in operation requirements, such as grade changes in multiproduct plants.

However, the use of dynamic models adds significant complexity to the modeling task because steady state allows a large number of simplifications that are no longer valid to describe the dynamic transients. Thus, chemical processes must be modeled by means of a set of differential equations that describe the dynamic behavior of the system, such as mass and energy balances and al-

gebraic equations that ensure physical and thermodynamic relations (DAEs). Although significant advances have been achieved in the development of algorithms in order to solve these dynamic optimization problems (Biegler and Grossmann, 2004), the use of dynamic RTO for large-scale plant-wide problem is still very limited because the computational complexity is too great to be able to provide a fast enough response in real time.

Alternative algorithmic approaches have emerged in literature in order to perform RTO during the dynamic transient period of the plant operation in order ensure a sufficiently fast response. This is case in the Real Time Evolution (RTE) algorithm developed by Sequeira et al. (2002), which relies on the iterative solving of a steady-state model. Rather than being a formal optimization technique, RTE seeks for the improvement of the trajectory of the process during its dynamic transient.

2.2 Optimization and Flexible Management of Batch Processes

The flexibility of batch processes mainly rests on their capacity to carry out a large number of production tasks for different types of products by sharing the same equipment. The scheduling optimization of these tasks has received great attention over the last decades as a way to improve the efficiency of plants. Due to the discrete decisions involved (i.e. equipment assignment, task allocation over time), these problems are combinatorial in nature, and hence very challenging from the point of view of computational complexity (Pekny and Reklaitis, 1998). In fact, scheduling problems belong to the class of NP-hard problems because the solution time scale increases exponentially as the problem size increases. A modest growth in problem size can lead to a significant increase in the computational requirements. Thus, it is of crucial importance to develop effective tools to represent the manufacturing processes and to explore efficient solution approaches for such problems.

Alternative methodologies and problem statements with different considerations have been proposed in the literature to address the combinatorial character of these problems. Extensive reviews of scheduling can be found in Reklaitis (1992), Floudas and Lin (2004), Shah (1998) and Méndez et al. (2006). Solution methodologies reported can be mainly grouped into two broad classes: approximated and rigorous.

2.2.1 Approximated scheduling methods

Approximated methods have been historically used in industry and even now, are often the base of most of the available commercial scheduling tools. This kind of approach usually uses assumptions to make easier the problem to be solved. The main advantage is that these methods can obtain good quality solutions within reasonable computing time. However, they may not guarantee optimality or an estimation of the quality of the solution obtained.

2.2.1.1 Heuristic rules

Traditionally, the favorite method to deal with the scheduling problem has been the use of heuristics or dispatching rules. Some of these sets of rules are the shortest processing time first (SPT), the longest processing time first (LPT), the earliest due date first (EDD), the first come the first served (FCFS) and the earliest release date first (ERD). According to the scheduling rule, tasks are sequenced in decreasing priority order and then one by one assigned to processing units. These rules are best suited to processes where the production involves a prespecified sequence of tasks with fixed batch sizes. Extensive review of various heuristics (dispatching rules) can be found in Panwalkar and Iskander (1997) and Pinedo (1995).

However, the application of heuristics is not so straightforward, because there is no one single universal rule and its effectiveness strongly depends on the scheduling objective and the plant configuration. Therefore, the use of heuristics has been adapted to work in combination with mathematical programming models with the purpose of reducing the size of the models (Pinto and Grossmann, 1996; Hui and Gupta, 2001; He and Hui, 2007).

2.2.1.2 Meta-heuristics

Other types of approximated methods are the meta-heuristic or local search methods, such as simulated annealing (SA), genetic algorithms (GA), and tabu search (TA), all of which are inspired by moves arising in natural phenomena. They employ an iterative evolutionary procedure that starts with an initial schedule that is gradually improved. These techniques became very popular for solving certain types of scheduling problems because they are easy to implement and tend to perform better than conventional heuristics. Examples of application include the works of Ku and Karimi (1991) for simulated an-

nealing, Ponsich et al. (2007) for genetic algorithms and Cavin et al. (2004) for tabu search. Nevertheless, meta-heuristics may not be very suitable for certain types of large-size problems due to the complexity and constraints of some process and their incapacity to guarantee the quality of the solutions.

2.2.2 Rigorous scheduling methods

Alternatively, rigorous methods rely on an accurate representation using exact procedures to solve the scheduling problem. Furthermore, they can provide precise information about the quality of the solution. However, their main drawback is the problem size limitation.

2.2.2.1 Mathematical programming

Mathematical programming techniques arise as a powerful tool for addressing the complexity of the scheduling problem. Mathematical formulations provide a rigorous and explicit representation in an algebraic form of the main decisions that must be made in terms of continuous and discrete variables. This optimization technique maximizes or minimizes a suitable objective function subject to equality and inequality equations which represent material balances, equipment allocation, storage and resource utilization constraints.

Most scheduling applications arising in the process industry can be formulated as Mixed Integer Linear Programming (MILP) models. These MILP models can be solved to achieve global optimization through different general-purpose optimizers, commonly using linear programming based branch and bound methods (B&B). Mathematical programming decouples the modeling issue from the solution algorithm. Therefore, MILP-based methods can take advantage not only of better modeling techniques but also of significant improvements in the optimization codes.

A varying characteristic among different mathematical programming approaches is the handling of the material balances. In the literature two alternative approaches are reported for dealing with this important feature.

A first approach assumes that the number and size of batches are known in advance. This simplification permits addressing larger practical problems, especially those involving quite a large number of batch tasks related to different intermediates or final products. However, they are still restricted to processes comprised of sequential production recipes.

A second approach simultaneously addresses the optimal number of batches and size of batches (batching), the allocation and sequencing of manufacturing resources and the timing of processing tasks (batch scheduling). These methods are able to deal with arbitrary network processes involving complex recipes. Their generality usually implies large model size formulations and consequently their applications are currently restricted to processes involving a small number of processing tasks and rather narrow scheduling horizons. The state-task-network (STN) representation introduced in the work by Kondili et al. (1993a) assumes that processing tasks produce and consume states (materials). In this representation a special treatment is given to manufacturing resources in addition to equipment. Similarly, Pantelides (1994) introduced the resource-task-network representation (RTN) that employs a uniform treatment and representation framework for all available resources. The idea that processing and storage tasks consume and release resources at their beginning and ending times, respectively.

Optimization models for batch scheduling can also be classified according to their time representation; continuous or discrete. In *discrete-time* approaches, the time horizon of interest is divided into a number of time intervals of uniform duration. Then, the beginning and ending of a task are forced to take place at the boundaries of these time intervals . An early discrete model was developed by Kondili et al. (1993a), using an STN representation which was followed by other many applications (Pantelides, 1994; Zentner et al., 1994; Dedopoulos and Shah, 1995; Bassett et al., 1996). However, discrete models with uniform time intervals have a tradeoff between the quality of solution and the required computational effort for large size problems.

Alternatively, *continuous-time* formulations allow scheduling events to occur at any time point along the time horizon. These formulations reduce the number of variables, which leads to faster solutions. But at the same time, they also require more complicated constraints, which increases the model complexity.

In addition, continuous time models can be distinguished according to the way that the event points are arranged over the time horizon. The *global time point* representation corresponds to a generalization of global time intervals for discrete time models, in which the timing of the intervals is treated as new model variable (Maravelias and Grossmann, 2003). Ierapetritou and Floudas (1998a) used an *unit-specific time events* representation which only defines an event at the starting of a task. Each event point could take place at different times in each different unit. The *time slots* representation (Pinto and Gross-

mann, 1996; Lim and Karimi, 2003) considers a set of predefined time intervals with unknown durations. The model presented by Méndez and Cerdá (2003a) relies both on a continuous-time representation and the notion of *general precedence*. This model is able to achieve a great decrease of binary variables by: (a) defining a single sequencing variable for each pair of tasks, (b) using a unique binary variable to sequence both processing and storage units, and (c) predefining the sequence of multiple batches of the same product. This model exploits all these features leading to simpler models which do not compromise the global optimality of the solution. This makes this formulation very suitable for applications requiring short calculation times, i.e. in real time.

2.2.2.2 Constraint Programming

Constraint Programming (CP) is a relatively new modeling and solution paradigm, particularly effective for solving feasibility problems (Hooker, 2000; Hentenryck, 1989).

The solution of CP models is based on performing constraint propagation at each node by reducing the domains of the variables. CP basically implies the problem of selecting, from finite domains, values to be assigned to each variable, so that every element in the set of constraints is satisfied. Its solving procedure uses a tree search combined with domain reduction and constraint propagation procedures.

CP methods have proved to be quite effective in solving certain types of scheduling problems, particularly those that involve sequencing and resource constraints. However, the lack of an obvious relaxation, makes CP worse for loosely constrained problems. CP performs worse if the focus is on finding the optimal solution among many feasible ones and proving optimality, especially for solving more general optimal scheduling problems that involve assignments.

The use of CP in combination with MILP techniques (hybrid methods) is a promising solution approach because of the complementation between both methodologies. Examples of this integration are shown in the works by Jain and Grossmann (2001) for single-stage, Harjunkoski and Grossmann (2002) for multi-stage multiproduct plants, and Maravelias and Grossmann (2004) for multipurpose plants.

2.2.2.3 Graph-based methods

Another rigorous alternative for solving the scheduling problem is the S-graph framework proposed by Sanmartí et al. (2002), which relies on a graphical representation of the problem. In contrast to MILP-based methods, the S-graph embeds the modeling aspects into the solution algorithm. Recipes are represented as a recipe-graph, in which the nodes represent the production tasks and the arcs the precedence relationships among them. A schedule-graph describes a single solution of the scheduling problem. A branch-and-bound procedure may generate the optimal schedule of a scheduling problem.

An experiment carried out by Ferrer-Nadal et al. (2006) demonstrated that through the better guided search method implemented in the S-graph algorithm, very good schedules can be created with low computational effort in special cases. Nevertheless, the S-graph is still a solution technique under development, resulting in limited applications to complex plant configurations, alternative objective functions and the consideration of restricted plant resources different than the processing equipment.

2.2.3 Batch scheduling under uncertainty

Although there has been a substantial amount of work to address the problem of the design of batch plants under uncertainty, the operational issues have received relatively less attention. Next, existing approaches in the literature to deal with this problem are presented.

2.2.3.1 Reactive scheduling

The main idea of reactive scheduling is to revise the production plan whenever a variation occurs. Although rescheduling techniques have a central role in process operations, only a few studies have focused their attention on this challenging problem.

Probably one of the first approaches to this problem was taken by Cott and Macchietto (1989), who considered uncertainty in processing times within a larger computer-aided production management system. They proposed a time shifting algorithm (project operation modification algorithm, POMA) to minimize the effects of uncertainty which readjusts the task starting times depending on the maximum deviation between the expected and the actual processing times of all related processing steps. This approach did not account

for changes in plant profitability and was very limited in terms of the sets of unexpected events that it could address.

Next, Hasebe et al. (1991) proposed a reordering algorithm for the scheduling of multiproduct sequential batch plants consisting of parallel production lines with a shared unit. This algorithm allowed the insertion of a job or the exchange of two jobs, one at a time. Otherwise, the algorithm was inoperable in terms of computational time.

Kanakamedala et al. (1994) additionally considered unit unavailability in multipurpose plants. These authors proposed a heuristic algorithm based on the search tree analysis that allowed time shifting and unit replacement. This work introduced a least impact (LI) heuristics in order to emphasize the importance of maintaining the original schedule.

Later on, Huercio et al. (1995) considered task processing time variations and equipment availability. They used heuristic equipment selection rules for the modification of starting times and the re-assignment of tasks to alternative units. Sanmartí et al. (1996) continued this work accounting for unexpected equipment failures.

Rodrigues et al. (1996) developed a reactive scheduling technique for multipurpose plants based on a discrete STN representation (Kondili et al., 1993a) accounting for uncertain processing times. They introduced a rolling horizon methodology which decreased the size of the original problem through successive solutions of MILP problems of reduced dimension. Time constraints outside the rolling horizon were taken into account with a look-ahead procedure in order to avoid future possible bottlenecks or unfeasibilities.

Honkomp et al. (1999) combined a deterministic scheduler with a simulator and accounted for processing time variations and equipment breakdowns. They proposed two penalty functions in order to reduce the effect of the rescheduling actions. This was accomplished by keeping constant batch sizes and minimizing the differences between the starting times of tasks and their nominal values.

Vin and Ierapetritou (2000) presented a two-stage solution procedure based on a unit-specific time events approach (Ierapetritou and Floudas, 1998a). This procedure was aimed at reacting against machine breakdown and rush order arrival in multiproduct plants without considering any heuristic rule.

Ruiz et al. (2001) applied a supervisory rescheduling system to multipurpose plants in which a fault diagnosis system interacts with a schedule optimizer.

Roslöf et al. (2001) introduced an MILP-based heuristic algorithm to improve an existing but non-optimal schedule or update the schedule in progress in the case of changed operational parameters. Rescheduling was performed by iteratively re-allocating and re-sequencing one or two tasks at a time.

Méndez and Cerdá (2003b) developed an MILP model for reactive scheduling of single-stage batch plants. In a subsequent work, these authors extended the original approach to multi-stage multiproduct plants with limited discrete renewable resources (Méndez and Cerdá, 2004). Limited changes in batch sequencing and unit assignment were permitted in order to prevent massive rescheduling actions which could disrupt smooth plant operation. In addition, this also reduced the computational size of the problem.

2.2.3.2 Proactive scheduling

Proactive scheduling appears in the literature as an alternative approach for uncertainty management through preventive scheduling, which takes advantage of the a priori knowledge. This approach transforms the original deterministic model into a stochastic model, optimizing a performance criterion and treating the uncertainties as stochastic variables. In order to represent the uncertainty, either discrete probability distributions or the discretization of continuous probability distribution functions are used. Stochastic problems can be classified as:

- *Two-stage or multi-stage stochastic approaches*, in which first-stage decisions are taken before the actual realization of the uncertain event, and second-stage decisions (recourse decisions) are determined in response to each random outcome.
- *Chance constraints approaches*, in which the focus is more on the reliability of the system, defined as its ability to find feasibility in any uncertain scenario.

Ierapetritou and Pistikopoulos (1996) proposed a two-stage stochastic approach applicable to multiproduct continuous plants with uncertainty in product demands.

Orçun et al. (1996) proposed an MINLP formulation to account for optimal batch scheduling with uncertainty in processing times and used chance constraints to define the probability of the accomplishment of the timing constraints.

Sanmartí et al. (1997) incorporated equipment failure uncertainty into the production and maintenance problem in order to generate a more robust schedule. They defined and assigned a reliability index for each plant unit and each task, and formulated a non-convex MINLP model in order to maximize the overall scheduling reliability. Preventive maintenance tasks may be introduced to compensate for possible delays produced by the use of equipment units with low reliability indexes. The complexity of the problem was circumvented by applying a heuristic method able to provide feasible but non-optimal solutions.

Petkov and Maranas (1997) considered demand uncertainty within the multi-stage planning and scheduling problems of multiproduct plants. They reformulated the stochastic elements of the model using their equivalent deterministic forms and maximized the expected profit of the convex MINLP resultant model. Chance constraints were used to impose a limit on the probability level of demand satisfaction.

Bassett et al. (1997) applied a Monte Carlo sampling approach in order to generate random instances considering processing time, equipment reliability and/or availability, process yields, manpower and demand fluctuations. The distribution of the generated schedules for each instance were determined and statistically analyzed to define different operating policies.

Vin and Ierapetritou (2001) used a multi-stage formulation to address the scheduling problem of multiproduct and multipurpose plants under demand uncertainty. They introduced a series of metrics in order to quantify the schedule robustness.

Lee and Malone (2001) developed a hybrid Monte Carlo simulation and simulated annealing approach to deal with multiple uncertain parameters and several types of probability density functions.

Balasubramanian and Grossmann (2002) minimized the expected makespan of a flowshop plant under uncertain processing times described by discrete probability distributions. They developed a rigorous branch and bound procedure relying on the aggregated probability model used to evaluate a lower bound of the expected makespan. The same authors (Balasubramanian and Grossmann, 2003) next solved this problem using a fuzzy set theory. Later, they reformulated the problems as a multistage stochastic MILP model (Balasubramanian and Grossmann, 2004). To do this, they overcame the computational complexity of the problem by using an approximation strategy based on the solution of a series of two-stage models within a shrinking-horizon approach.

Lin et al. (2004) proposed a robust optimization method with bounded uncertainty levels of processing times, market demands and prices. They treated the uncertainty in a bounded form by considering the worst-case values of the uncertain parameters. Later, Janak et al. (2007) solved the same problem in which the uncertainty was described by known probability distribution functions.

Bonfill et al. (2005) considered uncertain idle and waiting times. They proposed a two-stage stochastic approach accounting for the minimization of a weighted combination of the expected makespan and waiting times. Furthermore, they evaluated the robustness of the schedules using three different economic risk metrics.

2.3 Optimization and Flexible Management of Semi-continuous Processes

Semicontinuous processes reside in between the continuous and batch processes. Their flexibility has been addressed in the literature by trying to efficiently manage the scheduling of the production campaigns. One of the first contributions was made by Sahinidis and Grossmann (1991), who addressed the problem of cyclic multiproduct scheduling on continuous parallel production lines without any common resource. They identified in this problem a combinatorial part related to the assignment of products to lines and their sequencing, and a continuous part accounting for the duration of production runs and frequency of production. They formulated a slot-based MINLP model which was linearized in the space of the integer variables. Inventory, production, and transition costs as well as profit contribution constituted the objective function.

Kondili et al. (1993b) dealt with the planning of a multiproduct energy-intensive continuous operation. They utilized an STN-based hybrid formulation considering discrete time periods and time slots of varying duration.

Pinto and Grossmann (1994) extended the work of Sahinidis and Grossmann (1991) and modeled the cyclic scheduling problem in multi-stage continuous processing plants. They proposed a solution method based on a Generalized Benders Decomposition (GBD) and an Outer Approximation (OA). In this solution algorithm, an MINLP subproblem was solved to determine optimal cycle times and inventory levels for a fixed sequence, while an MILP master problem determined the optimal sequence of production. Furthermore,

this model handled the inventory profiles of intermediate storage tanks using explicit inventory breakpoints.

Zhang and Sargent (1996) developed a discrete RTN-based formulation for the optimal operation of a semicontinuous production facility. Even though the resultant MINLP was linearized, this formulation still led to a very large MILP model. Schilling and Pantelides (1996) reformulated the problem as a continuous-time model. Moreover, these authors developed a branch-and-bound algorithm to overcome the large integrality gap that rendered its solution using standard algorithms. The proposed algorithm starts branching on the continuous slot lengths to get a tighter bound, and then switches to the normal branching on the binary variables.

Karimi and McDonald (1997) proposed a slot-based continuous-time MILP formulation. This formulation minimizes the inventory, transition and shortage costs under known product demands at specified due dates.

Ierapetritou and Floudas (1998b) used an STN representation to formulate the scheduling problem of continuous processes taking into account multiple intermediate due dates. This model couples task and unit events, achieving significant reductions in the number of binary variables. In this work, storage requirements were handled using an approximation of the storage task timings.

Mockus and Reklaitis (1999) proposed a global event-based MINLP, called a non-uniform discrete-time model. This model was able to handle resource constraints such as limited availability of utilities and manpower.

Giannelos and Georgiadis (2002) developed a model similar to that by Ierapetritou and Floudas (1998b), relaxing the time durations and eliminating the big-M constraints and assuming equal starting and ending times of tasks that produce or consume the same state. This simplification could lead to a suboptimal solution in the case that materials were allowed to bypass storage.

In the work by Méndez and Cerdá (2002), a continuous-time formulation based on a general precedence notion led to very compact small-size models. The formulation takes into account sequence-dependent changeover times and storage limitations. One of the major assumptions of these authors is that every intermediate or final product can be produced by a sole production campaign.

Munawar et al. (2003) used a slot-based cyclic scheduling formulation for hybrid plants consisting of serial and parallel processing and storage units. They proposed a modified definition of the time slots to account for feed losses during product transitions, and generalized the resultant MINLP model for the case of slots having zero time duration.

Castro et al. (2004) developed an MILP formulation based on the RTN representation, highlighting the benefits of using a uniform time grid continuous representation. The authors pointed out that non-uniform time grid formulations might become particularly useful in some cases, for instance, when the problem involves many changeovers because such events must be modeled implicitly. In uniform-time-grid formulations each changeover might require an additional event point to find the global optimum.

Zeballos and Henning (2006) developed a formulation based on constraint programming (CP) to address the optimal operation of multiproduct continuous processes. They considered intermediate storage tanks of limited capacity and limited manpower resources within the problem statement.

Shaik and Floudas (2007) extended the work of Ierapetritou and Floudas (1998b) to handle additional storage requirements, such as dedicated, flexible, unlimited, and non-intermediate storage policies. In this work, flexible storage refers to the general case in which finite intermediate storage is available, and for each material state several suitable storage options exist.

Sadrieh et al. (2007) addressed the scheduling problem of hybrid semicontinuous systems by integrating a pair of meta-heuristic approaches: a timed arc hybrid Petri net (TAHPN) and a genetic algorithm (GA). In this algorithm the TAHPN module is considered as a fitness function for the GA module. The GA provides the flowrates of the continuous operational units and the TAHPN module finds the optimal batch schedule with respect to the objective function.

2.4 Optimization and Flexible Management of Batch Pipeless Plants

Pipeless plants were projected to increase the flexibility of conventional batch plants by allowing materials to be transported from one processing stage to another in moveable vessels. However, this scheme poses extra difficulties in the operational management of this kind of plants that have only recently attracted the interest of the research community (Niwa, 1993).

One of the first attempts to address the scheduling problem of batch pipeless plants was carried out by Pantelides et al. (1995), who developed an MILP model relying on a uniformly discretized time horizon and an STN representation. Although this formulation was able to deal with the major characteristics of these plants, a huge number of binary variables were inevitably needed to

represent the events at the boundaries of the predefined time intervals.

Liu and Mcgreavy (1996) examined the design and operating strategies of pipeless plants, emphasizing dynamic analyses and production arrangements. They concluded that the design process is tightly coupled with the operational strategy in the case of pipeless plants.

Realff et al. (1996) used a STN-based representation with a uniformly discretized time horizon to formulate an MILP model that simultaneously considers the design, layout, and operation. However, the discrete time representation was computationally very expensive. In order to overcome this problem, these authors applied a decomposition procedure in which first, the layout structure was preselected and next, the position allocation of each station was decided.

Bok and Park (1998) modeled the scheduling problem as a matching problem and developed an alternative MILP formulation. This approach was based on a two-coordinate representation in which the time slots, symbolizing the timetable of resources, were paired with the processing stages of products. However, computational effort was highly dependent on the number of time slots, especially on those not matching any stage of a product (superfluous).

Gonzalez and Realff (1998a) developed a discrete event simulator coupled to the results of an MILP formulation. The aim was to simulate the plant operation and check its performance under varying conditions, such as vessel travel times and task durations. In their following work, Gonzalez and Realff (1998b) compared the MILP solutions with the results obtained by using local dispatching rules to manage the station operation and vessel movement. They observed that the dispatch rules generally performed quite poorly in comparison with the MILP formulation. However, these authors remarked that customized rules to particular plant layouts could achieve almost similar performance to the MILP.

Huang and Chung (2000) adressed scheduling of pipeless batch plants using constraint satisfaction techniques (CST). CST, rather than searching the entire space for a solution, exploits the constraints themselves to reduce the search space. Constraints are exploited in a constructive way such that it may lead to the deduction of other constraints and the detection of inconsistencies among possible solutions. Later on, Huang and Chung (2005) extended their work to consider the problems of both scheduling and routing.

Patsiatzis et al. (2005) considered simultaneously layout, design and production planning by manipulating the model of Realff et al. (1996). This model used the same aggregated production planning constraints, but it was based

on a continuous-time representation instead of a discrete one.

2.5 Thesis scope and objectives

In the previous sections, the current approaches for the optimization and flexible management of the chemical processes have been carefully analyzed. In general, this thesis makes a step forward from the existing approaches in order to exploit the flexible operation of chemical processes.

In particular, the objectives of this thesis can be summarized in the following points:

- Develop integrated computer aided systems for the support of the decision-making process in the operation of chemical processes, exploiting their potential flexibility.
- Propose and apply new optimization and supervisory techniques in order to achieve more robust plant operation in real time.
- Propose new solutions in order to address the drawbacks and limitations of the available scheduling formulations in flexible batch chemical plants.
- Propose alternative reactive scheduling strategies, by exploiting the concept of batch flexible recipes during the plant operation.
- Analyze the positive effect of considering flexible recipes as a manner to proactively manage the uncertainty and generate efficient and robust predictive batch schedules.
- Explore the potential use of pipeless plants as a more flexible alternative to the conventional batch plants.

On continuous processes

Chapter 3

A supervised real time optimization system

This chapter addresses flexibility in continuous processes motivated by the fact that chemical systems have to adapt to changing conditions over the time. As a result, determining the optimal operation of the plants becomes a continuous challenge.

An integrated supervisory framework is envisaged for more robust real time optimization and exception handling in continuous processes. This system combines the capabilities of Real Time Evolution (RTE) algorithm (Sequeira et al., 2002) with a Fault Diagnosis System (FDS) for managing abnormal situations. Both systems are part of a supervisory control module which is responsible for handling plant incidences and taking appropriate corrective actions. Thus, a more satisfactory on-line performance is achieved, while reaction to incidences is possible because flexibility allows the readjustment of the operation conditions in continuous processes. Finally, it should be borne in mind that the usefulness of this system is enhanced by providing combined cause-effect information to plant managers.

3.1 Introduction

Current situation of the process industry with growing competence, increasing costs and stricter environmental constraints, makes optimization a key to obtain the maximum profitability with the available resources. Furthermore, process abnormalities have significant impact, in such a way that hundreds of billion of dollars are lost by the industry due to poor abnormal situation management. However, there is a lack of systems able to systematically manage the occurrence of abnormal events and optimization in real time.

The aim of this chapter is to demonstrate the robustness of a supervisory control system that combines optimization and fault diagnosis in real time. This system provides relevant coupled and timely information for plant managers to improve the decision-making processes or generate new knowledge which could be applied to the design of new processes.

This chapter is organized as follows. The general optimization methodology applied in this chapter, Real Time Evolution, is described in section 3.2. Next, section 3.3 focuses on the advantages of developing a real time supervisory control system able to integrate on-line fault diagnosis and optimization. The proposed structure of the architecture of the system is also described in this section. A case study for its application is presented in section 3.4. Results obtained are presented and discussed in section 3.5. Finally, this chapter closes with some conclusions outlined in section 3.6.

3.2 Real Time Evolution

The application of the classic RTO systems, as described in section 2.1 of chapter 2 presents the following main limitations: a) requirement for steady state data in a common dynamic environment which may require heavy filtering and long waiting, b) detailed process models, and c) lack of a proper trajectory for the implementation of the manipulated variables to reach the new optimum.

Real Time Evolution (RTE) algorithm introduced by Sequeira et al. (2002) establishes a new approach for on-line optimization that overcomes the drawbacks reported for the classical RTO approach. This approach is an improvement rather than an optimization algorithm, which supported by a steady state model, successively identifies the direction for the improvement of an objective function or performance criterion. A very important feature of RTE is that it does not need to reach the steady state to trigger the improvement mechanism.

RTE can respond faster to disturbances by a continuous adjustment of the decision variables according to the current plant conditions. The steady-state information is used by RTE only for data reconciliation and model updating, while the core of the system is a recursive improvement, which does not require the process to be at steady state. Thus, the application of this system results in a better overall performance.

Real Time Evolution relies on four main aspects, an improvement algorithm, a steady state process model, a neighborhood for the search and a time between successive executions:

- The improvement algorithm is based on the way that better operating conditions are selected in the neighborhood of the current operating point. Despite the increasing complexity of RTO to handle non-linear constraints in a formal optimization problem, RTE can easily discard unfeasible solutions by directly excluding the unattainable paths in its search.

- The neighborhood for the search is defined as the maximum allowed changes in the decision variables to be explored around the current state. Limits for such changes are imposed by the distributed control system.

- Regarding the type of models required for on-line optimization, a greater detail in the process model improves the quality of the solution in classical RTO. However, the more complex a model is, the more computationally expensive the solution of non-linear optimization problem will be. Alternatively, RTE is much simpler because it does not need to solve any formal optimization problem. RTE just checks feasibility in the neighborhood of the search and chooses the direction for the improvement. Therefore, the RTE computational burden is significantly decreased compared to RTO for a similar model complexity. In addition, excessive detailed models are not required in RTE because the process model is just used as a tendency model.

- Finally, for a similar disturbance pattern, different RTE frequencies or intervals between consecutive executions may produce different profiles of the values of the decision variables. For that reason, as it was underlined by Sequeira et al. (2004), a successful RTE application requires an appropriate parameter tuning and selection of the RTE neighborhood and frequency.

3.3 Supervisory control system

An incorrect or null fault detection and identification of the root causes of process upsets may lead on-line optimization systems to unappropriated performance of the plant. Hence, implementing a robust and reliable fault diagnosis system (FDS) has a crucial importance to assure right responses against any upset. In the Supervised Real Time Evolution (SRTE) framework proposed in this work, RTE runs under the supervision of a FDS. The combination of both systems gives rise to a supervisory module which is the core of the on-line response against deviations from the normal behavior of the plant. This system makes the overall process more reliable and safe.

A fault diagnosis system mainly comprises two basic tasks: detection and diagnosis of the fault. In this work, incidence detection has been carried out by means of Multi-Scale Principal Component Analysis (MSPCA). MSPCA is based on Principal Component Analysis (PCA), one of the most applied multivariate statistical techniques (Kourti and MacGregor, 1995). Through the structure of MSPCA (Bakshi, 1998), the capacity of PCA to extract the relationship between variables is combined with the capacity of wavelets to separate deterministic features from stochastic processes. This FDS makes use of the monitoring statistic Squared Predicted Error (SPE), defined as the sum of the quadratic error between plant data and the model reconstructed signal. This statistic is calculated from MSPCA and continuously monitor the current state of the plant. One of the major advantages of SPE is that all the process variables and their corresponding correlations can be considered in the process monitoring with this single statistic (Kourti and MacGregor, 1995).

Once an incidence is detected, next the root causes must be found out. Incidence symptoms coming from the detection module are classified by a feed forward neural network by the proposed FDS. Variables contributions to SPE and latent scores from the MSPCA are used to classify data into different sets of performance. Data sets corresponding to the most probable incidences occurring in the plant are tested off-line or obtained from historical data to extract the enough information for training the neural network. As recent studies demonstrate (Zhao et al., 1998), radial basis function networks (RBFN) are preferred to back-propagation networks (BPN) because they provide more reliable generalization and fewer extrapolation errors. Due to this reason, a probabilistic neural network based on radial basis functions has been chosen to perform the fault isolation.

Then, the information provided by the monitoring and fault diagnosis sys-

tems is used by the supervisory control system SRTE to manage the plant. This system regularly classifies data received from plant between normal and abnormal behavior. In case of a abnormal behavior, the supervisory system differentiates the incidence into two main groups:

Disturbances , or incidences affecting the profitability of the plant but susceptible to operate under on-line optimization. A disturbance, as it is considered in this work, alters the normal performance of a process without affecting its safe operation.

Faults , or critical incidences that alter the safety and good performance of the plant. These faults must be established in a previous analysis of the possible incidences. Faults may generate different responses from the supervisory module. Typical responses comprise, for example, activating a set of corrective actions to prevent more critical situations, or monitoring different alarms to advise operators in the way they should act against this incidence.

Flexibility is regarded as the ability of the system to react against these disturbances and/or faults by adapting the plant to the new operating conditions. Accordingly, this supervisory system acts as a support decision tool that advises the operator to take the most convenient actions at every time. If deviations from the normal operation are not classified as critical faults, the system SRTE advises the operator to trigger the optimization procedure. Otherwise, knowledge acquired from a previous process hazard analysis would be used to suggest the most proper corrective actions to be adopted in case of critical faults. This knowledge must be extracted from previous off-line analysis which might be carried out by the empirical insight of the plant gained during its daily running.

3.3.1 Architecture of the system

A schematic representation of the architecture of the Supervised Real Time Evolution (SRTE) system is depicted in figure 3.1.

The SRTE system takes data continuously from the plant sensors and uses a monitoring and FDS systems to analyze the data. In case a disturbance is diagnosed, SRTE automatically activates the optimization algorithm RTE. The elapsed time between the fault is diagnosed and RTE is activated involves a trade-off between the benefits gained by an early reaction of SRTE and the

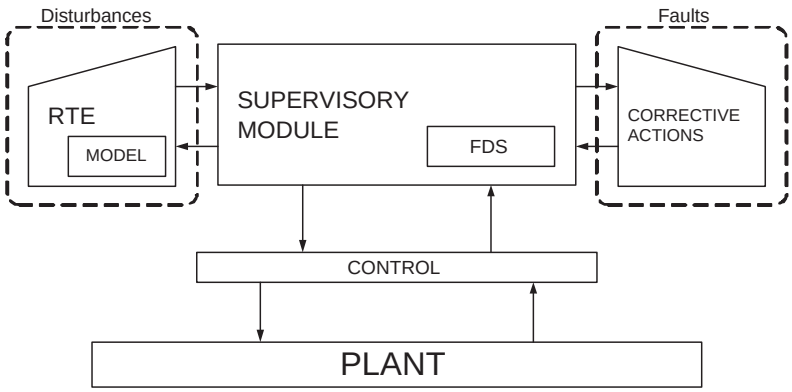

Figure 3.1: Supervised RTE Integrated Framework (SRTE).

confidence on the detection. Then, RTE works by collecting measurements of the current conditions of the plant and using a process model to explore the future performance of the process in steady state evaluated at the surroundings of the current values of the decision variables. The best objective function value is chosen and the corresponding values of the decision variables (or set-points for the controllers) are immediately applied to the plant. This mechanism is successively repeated at every sample time. When the plant reaches a new steady state, SRTE updates the steady-state process model and the fault diagnosis system in order to adapt the system to the new state of the plant.

In case a fault is diagnosed, the supervisory control may warn the operator or suggest proper course of actions. Depending on the nature of the abnormal situation, it can also execute a previously designed protocol consisting of a series of corrective actions to be executed in the process until the normal conditions are restored.

This SRTE system has been implemented using Matlab and Aspen Hysys. The supervisory module (Matlab) acts also as a data manager establishing the communication between the plant (Aspen Hysys in dynamic mode), the model (Aspen Hysys in steady state mode), the optimization algorithm RTE (Matlab) and the monitoring and FDS (Matlab). Communication between Matlab and Aspen Hysys has been achieved by mean of COM technology. Matlab creates a COM automation server which interacts with the objects exposed by Aspen Hysys.

3.4 Case Study: a debutanizer column

The debutanizer column provided by Aspentech at its website documentation (http://support.aspentech.com) has been chosen in this thesis in order to make the results easily reproducible. This multicomponent distillation column has fifteen stages and is fed by a pair of streams which consist of a mixture of light hydrocarbons. The flowsheet of this case study is depicted in figure 3.2. The operational objective of this unit is the recovery of butane and lighter hydrocarbons from the feed streams.

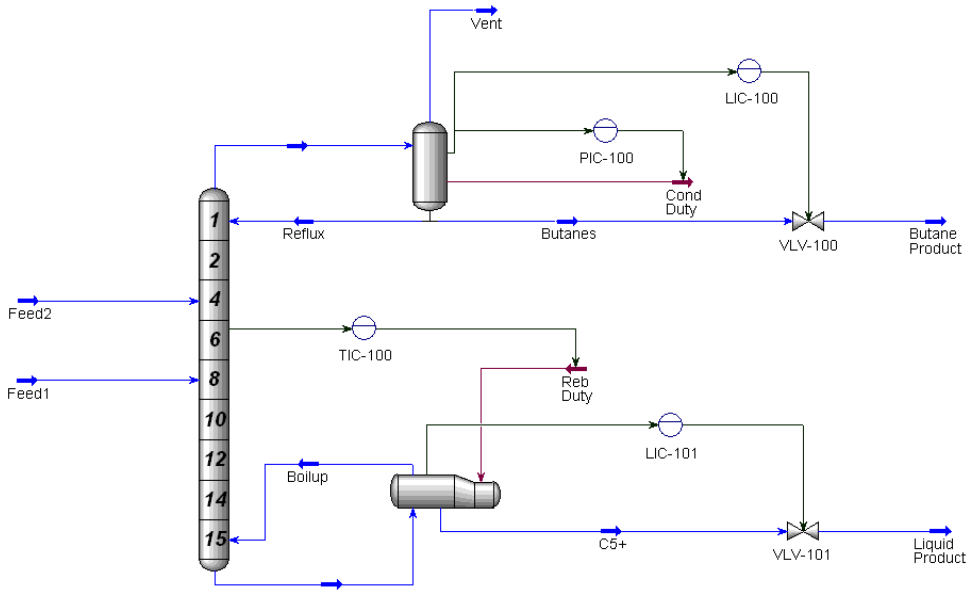

Figure 3.2: Debutanizer flowsheet.

A previous off-line analysis of the distillation column allowed to determine the most probable incidences during the normal running of this plant. Some of these possible abnormal situations are listed in table 3.1, along with a brief description of each incidence.

Firstly, the column has been simulated in Aspen Hysys in steady-state. This case serves as the steady-state model for the RTE algorithm. Next, the necessary modifications have been made in order to build the dynamic simulation. The control system consists of two level controllers for condenser and reboiler and a temperature controller taken from the sixth plate of the column. Tuning of these controllers has been made using the PID auto-tuning function

Table 3.1: Off-line analysis of the incidences.

Incidence	Description	Type
I1	Feed1 Flowrate Step -20% Fall	Disturbance
I2	Feed1 Flowrate Step +20% Rise	Disturbance
I3	Feed1 Temperature +5% Step	Disturbance
I4	Feed1 Flowrate Ramp -0.4 %/min Fall	Disturbance
I5	Feed2 Flowrate Step +20% Rise	Fault

of Aspen Hysys. These tuning parameters are shown in table 3.2. In order to carry out the experiments and evaluate the performance of SRTE, the dynamic simulation was used as the real plant.

Table 3.2: Parameters for the PI controller.

Controller	LIC-100	LIC-101	PIC-100	PIC-101
K_c	1.80	2.00	2.00	5.00
T_i, min	10.00	10.00	2.00	20.00

The instant objective function (IOF) in equation 3.1 has been proposed in order to have a measure of the economic performance of the plant at every moment. This objective function is based on the difference between the sales of the desired products and the cost of the raw materials and energy.

$$\begin{aligned} &Instant\ Objective\ Function(t) = \\ &\quad + (Flowrate\ of\ C4\ \&\ lighter_{ButaneProduct}) \cdot (Price_{ButaneProduct}) \\ &\quad + (Flowrate\ of\ C5+_{LiquidProduct}) \cdot (Price_{LiquidProduct}) \\ &\quad - (Flowrate_{Feed1}) \cdot (Price_{Feed1}) \\ &\quad - (Flowrate_{Feed2}) \cdot (Price_{Feed2}) \\ &\quad - (Q_{Condenser}) \cdot (Price_{CondenserHeat}) \\ &\quad - (Q_{Reboiler}) \cdot (Price_{ReboilerHeat}), \quad m.u./time \end{aligned} \tag{3.1}$$

The values of raw material costs and product selling prices have been taken just to formulate a representative objective function for demonstrating this methodology. These prices and costs are stated in table 3.4.

Table 3.3: Raw material, products and energy prices.

Feed1, m.u./kg	Feed2, m.u./kg	Butane P., m.u./kg	Liquid P., m.u./kg	$Q_{Condenser}$, m.u./kJ	$Q_{Reboiler}$, m.u./kJ
2.00	2.00	3.00	3.00	0.00009	0.0005

However, different on-line procedures can not be compared in terms of an instant objective because the value of an instant benefit does not effectively show which one achieve better overall results. Hence, the mean objective function (MOF) of equation 3.2 proposed by Sequeira et al. (2002) has been employed in order to take into account the accumulative profit produced along the whole interval of the transition. This MOF expression has been used in this case study to qualitatively compare an idealistic instantaneous response of the real time evolution (RTE) algorithm, the proposed supervised real time evolution (SRTE) system, the classical RTO approach and a worst case when no action is taken.

$$Mean\ Objective\ Function(t) = \frac{\int_{t_o}^{t} IOF(t)dt}{t - t_o} \tag{3.2}$$

3.5 Results

The goal for this plant is to maximize the profit by adjusting two decision variables that impact on the performance and economics of the process. These variables are the set-point of one of the controllers (temperature of the 6th stage of the column) and the reflux rate of the column. A thorough tuning of the RTE parameters led to select a maximum allowed change in set-points of 0.3% around the old value and 50 seconds between consecutive executions.

As it was commented in previous sections of this chapter, a correct fault detection and diagnosis must be carried out before RTE is activated. In this case study, an abnormal behavior is detected when four consecutive monitoring samples are above the detection threshold. This threshold is calculated from MSPCA model and represents the percentile 99 of the SPE normal data distribution. Once the incidence is detected, abnormal data gathered from the plant are processed by the artificial neural network in order to isolate the current situation.

Through this methodology, all incidences shown in table 3.1 are identified after detection, ensuring a robust system response. Discussion on the use of artificial neural networks for fault diagnosis or MSPCA for fault detection is out of the scope of this thesis. Further reading in detection may refer to Bakshi (1998) and Kourti and MacGregor (1995), as well as Leonard and Kramer (1993) and Holcomb and Morari (1991) for radial basis function networks.

Furthermore, the classical real time optimization (RTO) algorithm has been implemented for a comparative purpose. Assuming that data are reconciled, the steady state detection has been implemented based on the variance of a time-window of the last collected data. The optimization of the model has been carried out solving a successive quadratic programming (SQP) problem. This method reproduces Newton's method for constrained non-linear optimization. At each iteration, an approximation is made of the Hessian of the Lagrangian function using a quasi-Newton updating method. This solution is used to form a search direction for a line search procedure.

Next, different incidences are tested to demonstrate the performance of the supervisory framework proposed in this chapter.

3.5.1 Incidence 1. A step fall in Feed1 flowrate

In a first situation, a step fall (-20%) in the mass flowrate of Feed1 is diagnosed. This incidence may be caused by an upstream problem and does not have any immediate solution. These facts encourage the application of an on-line optimization until normal conditions are restored. Then, once the disturbance is detected and diagnosed as a non-critical fault, SRTE automatically activates the optimization algorithm. As shown in figure 3.3, while SRTE reacts almost immediately against this disturbance, RTO has to wait until steady state is reached to actuate. This faster response of SRTE is translated to a better value of the mean objective function compared to the response obtained by using RTO. In fact, RTO reacts later (at a simulation time of 13200 seconds approximately) losing benefits during the transition to the new steady state. Focusing on the bottom part of figure 3.3, the horizontal solid line defines the detection threshold. Dashed line corresponds to the percentile 95 of the normal operation data. It must be noted that the value of SPE sharply increases and passes its threshold value when the disturbance occurs in the plant. Then, RTE is correctly activated after the fourth consecutive sample upper the detection threshold.

Monitoring after fault detection, reveals a decreasing trend in the value of

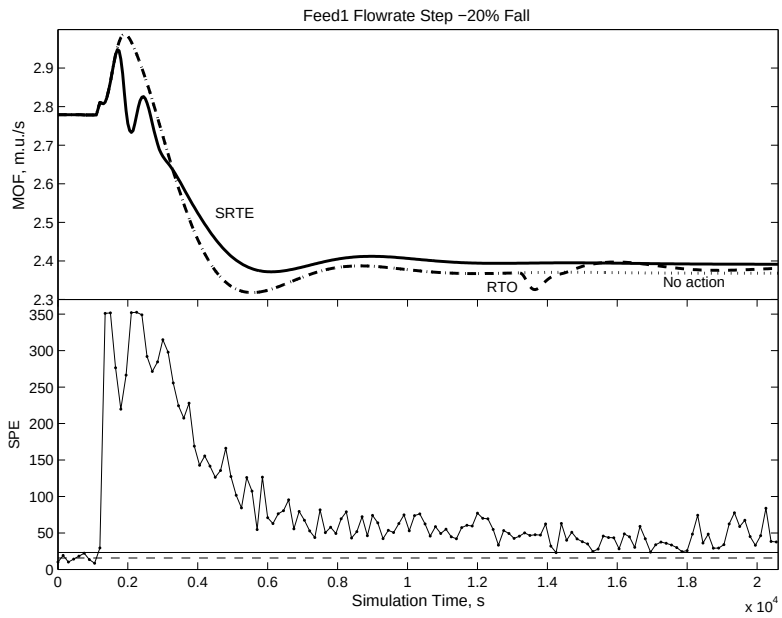

Figure 3.3: Incidence 1 - MOF and SPE representation.

SPE up to getting to a new steady state. This is caused by the combined effect of the on-line SRTE actions and the dynamic characteristics of the plant.

Two possible RTO responses are shown in figure 3.4. In the first response, a trajectory for the optimal set-points implementation is not provided, so the new set-points are directly applied to the controllers. This direct application of these set-points represents how real time optimizers would calculate the new optimal set points neglecting the way as they should be implemented. This sudden change causes a high disturbance in the plant which also alters the value of the instant objective function. This effect is shown looking at the peaks of this graph in figure 3.4. On the other RTO response, marked as "smooth RTO", the optimal set-points have been progressively implemented by a continuous ramp. This is automatically translated into a softer rise of the instant objective function which alters less the stability of the plant.

3.5.2 Incidence 2. A step rise in Feed1 flowrate

In this second case, a step rise (+20%) in the Feed1 flowrate is detected and diagnosed by the FDS. In contrast to Incidence 1, this disturbance produced

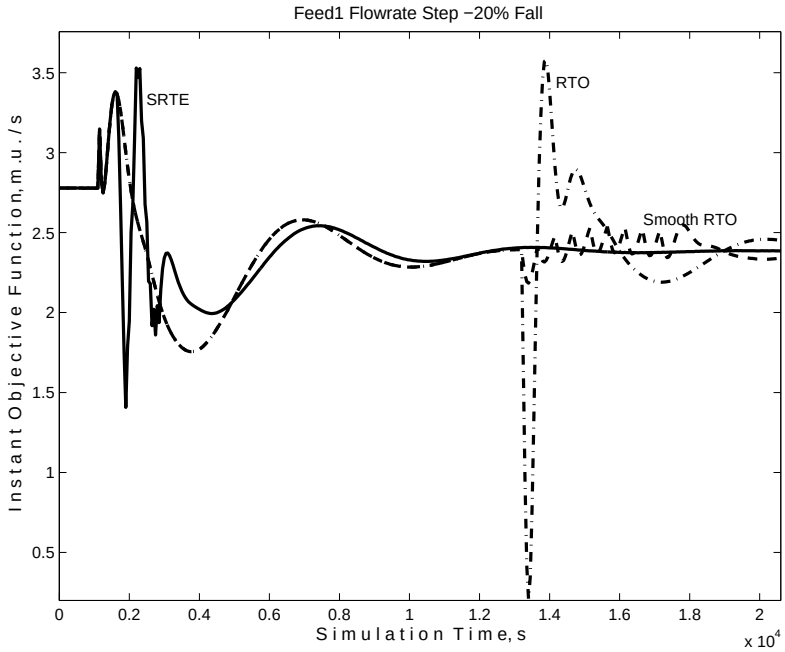

Figure 3.4: Incidence 1 - RTO's comparison (IOF).

by itself an improvement in the objective function. Figure 3.5 shows the value of MOF and SPE for this incidence.

The response of SRTE against this disturbance was similar to the previous disturbance, preparing the system for the future even losing benefits right after the incidence occurs. This trajectory of the operating conditions of the plant is due to the real time evolution algorithm is supported by a model which provides the best search direction looking at a steady-state horizon.

On the other hand, waiting time for the steady-state took more than 8500 seconds before RTO could be triggered. The implementation of the RTO has been accomplished by optimizing the process as soon as the steady state is reached and applying this calculated set-points directly to the plant. The graph of the mean objective function in figure 3.5 demonstrates that SRTE obtains better results along the whole transition.

Figure 3.6 shows how SRTE progressively modifies the two decision variables compared to the performance of RTO. As can be seen in this graph, inertia effects are more noticeable in the tracking of the set-point for the temperature of the 6th stage.

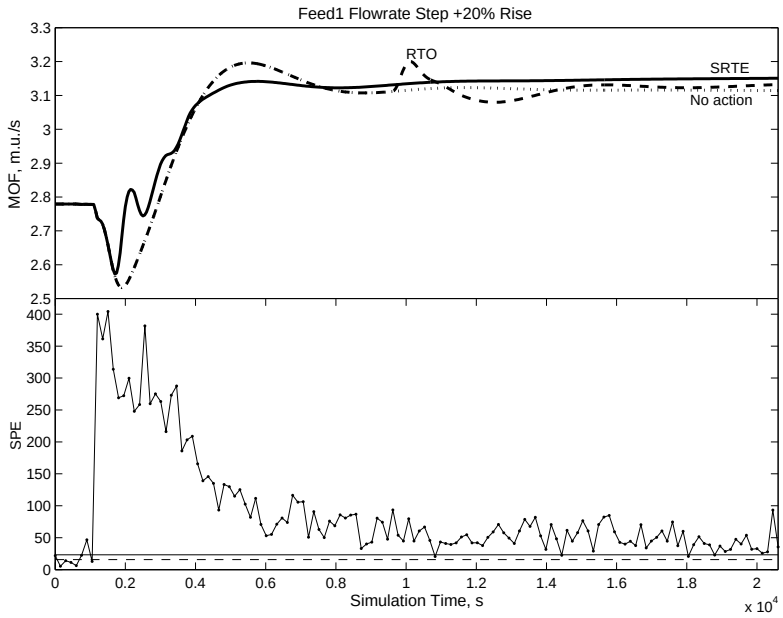

Figure 3.5: Incidence 2 - MOF and SPE representation

3.5.3 Incidence 3. A step rise in Feed1 temperature

The nature of this disturbance allows a smaller chance for optimizing because the variation in the objective function is less noticeable. This fact is because the temperature of the feed has a smaller impact on the value of the objective function. Figure 3.7 presents the representation of the value of MOF using SRTE, an hypothetical instant but idealistic RTE and a passive response when no action is taken. As in the second incidence, the implementation of the RTO has been done by optimizing just after the steady state is reached and applying directly calculated set-points to the plant. This direct application of the new set-points explains the peaks in the RTO representation.

3.5.4 Incidence 4. A continuous ramp fall in Feed1 flowrate

In case of a continuous disturbance arises in the mass flowrate of Feed1 (-0.4%/min), a real time optimization by RTO is not possible since the plant does not reach steady state, at least during the studied period of time. This case is considered as a weak point of the RTO in which RTE shows better performance. Therefore, the obtained results using SRTE can be only compared to those

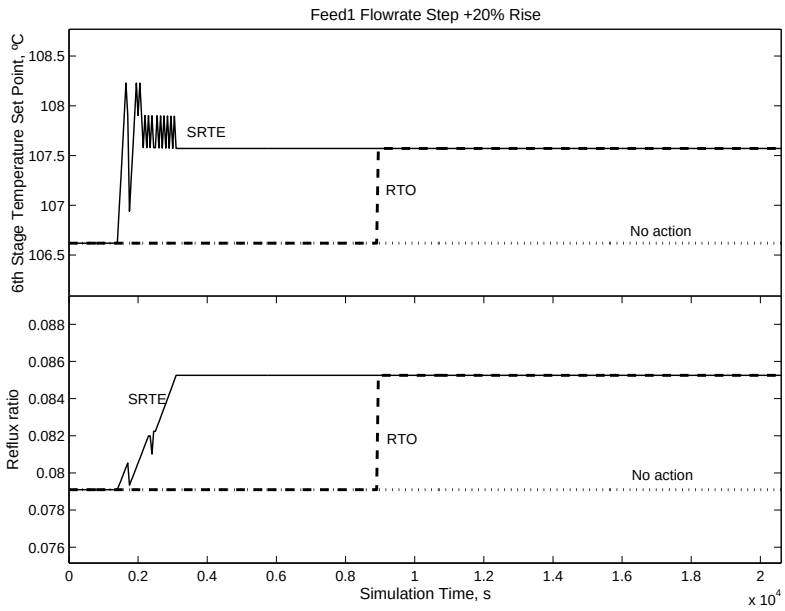

Figure 3.6: Incidence 2 - Decision variables representation.

obtained when no action is carried out.

Figure 3.8 represents the value of the mean objective function (MOF) during the simulation time for this incidence. RTE is triggered when the diagnosis is completely confirmed by the fourth consecutive point over the detection threshold. Figure 3.9 represents the value of the instant objective function for the SRTE and the no action behavior. Despite the instant objective function decreases in both cases, the value for SRTE is almost permanently over the no action. This effect is due to RTE continuously rectifies its trajectory increasing the difference between the mean objective function for both cases at every sample time. Figure 3.10 reflects how SRTE tracks the decision variables in order to accommodate the plant to the changing conditions.

3.5.5 Incidence 5. A step rise in Feed2 flowrate

In this distillation column, a rise of 20% in the second feed stream (Feed2) is identified as a fault. When the fault diagnosis system isolates this incidence, the supervisory control system performs a protocol consisting of a by-pass of the feed flows and an operational change to total reflux. Finally, a message ad-

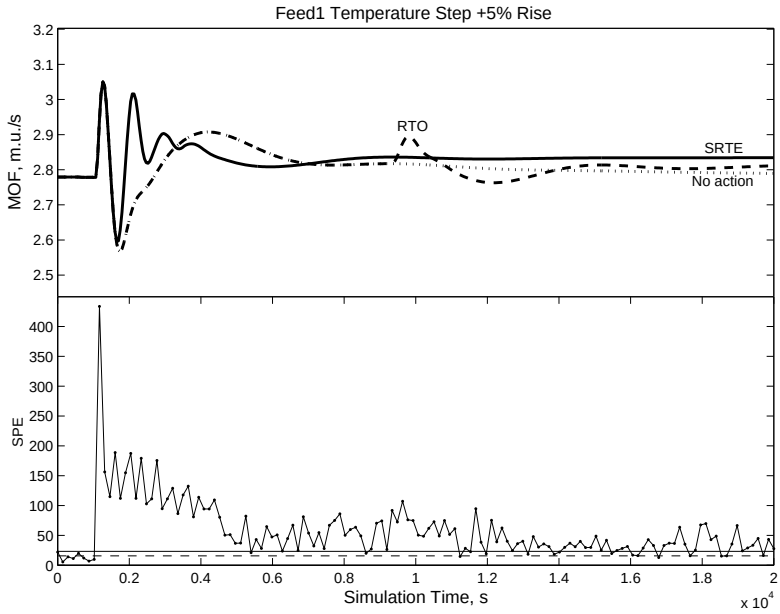

Figure 3.7: Incidence 3 - MOF and SPE representation.

vises the operator to check the equipment involved maintaining the corrective actions until the breakdown is repaired.

Figure 3.11 shows the evolution of the flowrates of the outlet and inlet streams in the column for this incidence. As soon as the increase in the flowrate of Feed2 is detected and diagnosed, corrective actions are automatically executed. Feeds are by-passed to another section of the plant and total reflux is imposed to the column. The effects of these actions in the value of the instant objective function are shown in figure 3.12.

Before the fault occurs, the monitoring of the plant shows normal conditions. During the elapsed time between the occurrence of the fault and its detection, the value of the instant objective function suffers the fluctuations shown in figure 3.12. After the value of SPE passes four consecutive times above the detection threshold, the fault is detected and this information is sent to the supervisory control system. Next, SRTE relates this incidence to its corresponding corrective actions, which are then executed. These corrective actions consist of closing the feed flows and preparing the plant to reach the total reflux operation. Total reflux leads the plant to a stationary conditions fairly far from the normal plant running which makes the value of SPE surpass-

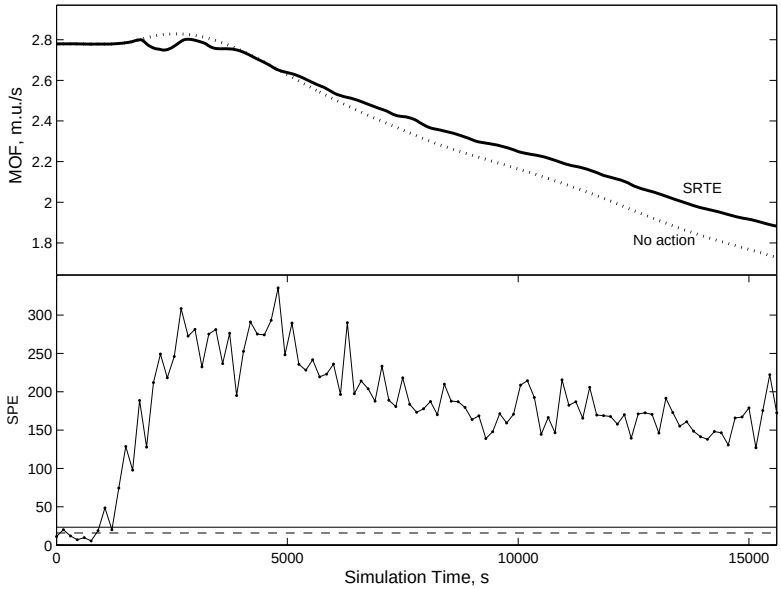

Figure 3.8: Incidence 4 - MOF and SPE representation for a continuous disturbance.

ing the scale of the representation in figure 3.12. Past the repairing time, 5000 seconds approximately after the incidence has been isolated, feeds are connected again to the column and normal operating conditions are progressively restored.

As can be seen from the value of the instant objective function in figure 3.12, the increase of benefits observed right after the incidence, is immediately reduced because of the total reflux conditions imposed to the plant. These apparent initial benefits are explained taking into account that the control system increases gradually the reflux ratio until it becomes total. Thus, some decreasing quantity of product at the first moments is obtained because the cost for raw material are considered as zero since these streams are sent to another section of the plant.

3.6 Conclusions

A supervisory module integrating a fault diagnosis system and the real time evolution algorithm has been developed in this chapter. This supervised real

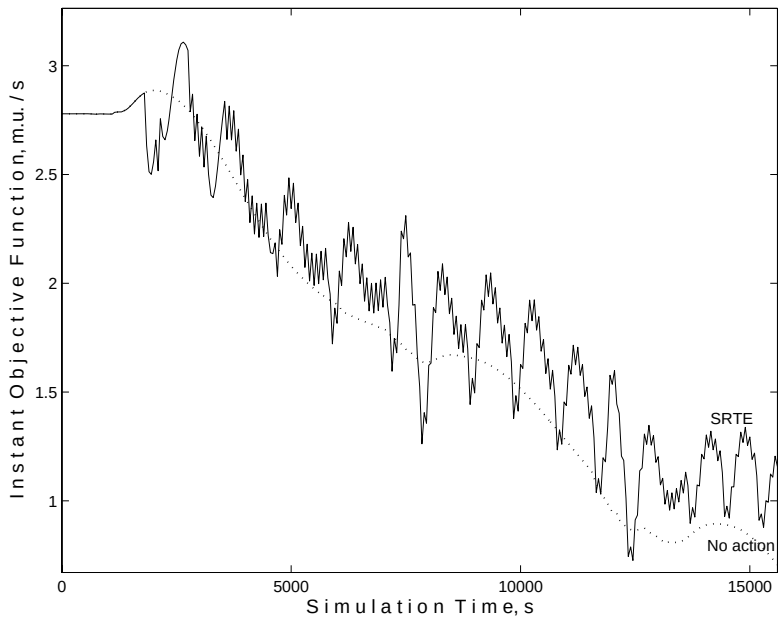

Figure 3.9: Incidence 4 - IOF for a continuous disturbance.

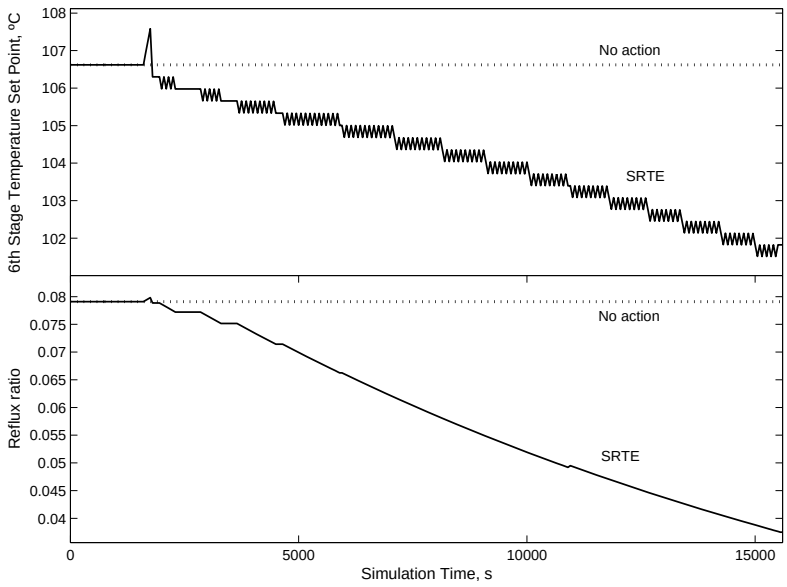

Figure 3.10: Incidence 4 - Decision variables representation.

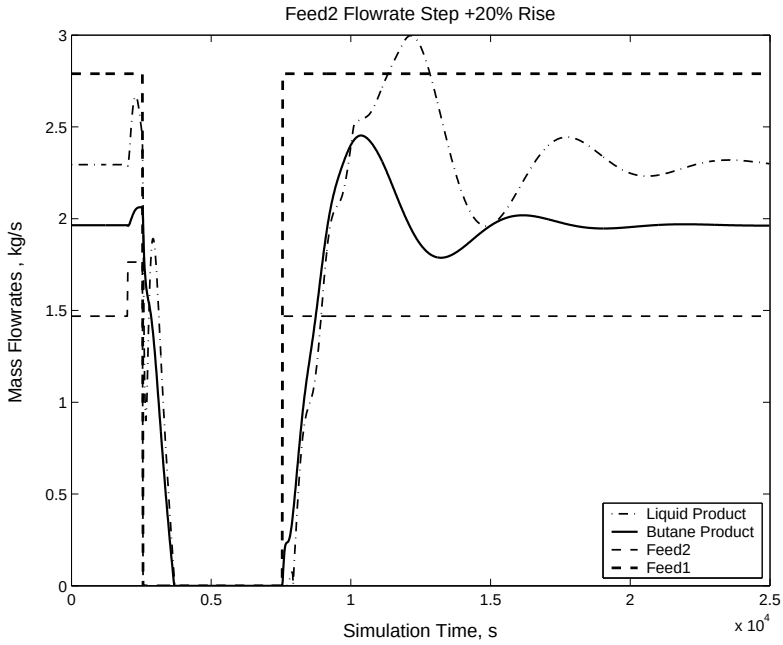

Figure 3.11: Incidence 5 - Evolution of the flowrate for the main streams.

time optimization system takes advantage of the qualities of RTE for creating a more robust and flexible framework. Flexibility of continuous process is managed by tracking its adjustable variables in order to faster and more efficiently optimize the plant in face of disturbances or faults.

Furthermore, decision making is enhanced by the information provided to the plant operators resulting in a better understanding of what is happening in the plant at every moment. From this point of view, this chapter also proposes a most transparent and interactive framework for continuous processes which can lead to a greater confidence of the operator on the automatic system.

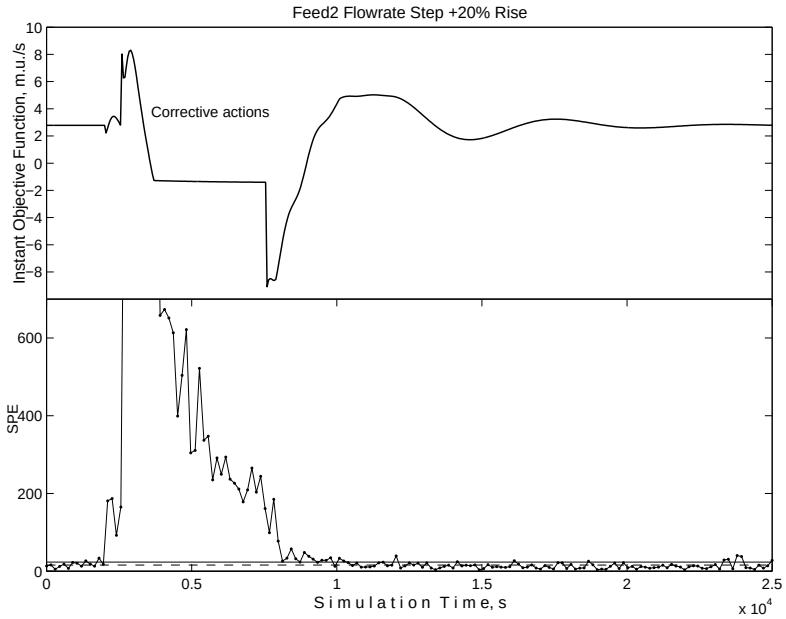

Figure 3.12: Incidence 6 - IOF and SPE representation.

On semicontinuous processes

Chapter 4

An effective tool for the flexible short-term scheduling of multiproduct semicontinuous processes

Chemical plants are moving towards more flexible environments in order to adapt faster to market changes. For the sake of this flexibility, batch and continuous units might be then integrated and operated along the same processing route in a semicontinuous mode featured by production campaigns of finite duration. However, scheduling literature has usually considered constant production rates for the whole operation and so production adjustments could be just achieved through storage resources, production line stops and multiple product campaigns.

The aim of this work is to improve the production schedules by developing a new concept for flexible manufacturing which allows programming production rate profiles within each semicontinuous operation campaign. The extended formulation presented in this chapter allows addressing the trade-off between the opportunities that may arise when adjusting processing rates and the storage needs. Finally, several case studies show that the proposed adjustment of the processing rates lead to less storage consumption and better production efficiency.

4.1 Introduction

Although the recent trends on globalization have opened new markets for the chemical industry, they have also brought about an increasing competitiveness giving a boost to the development of more efficient and highly integrated plants. Therefore, the flexibility becomes an important feature to be considered and exploited to respond to new market demands effectively. Flexible batch processes reemerged, in contrast to the more widespread and rigid continuous production mode addressed in chapter 3 and more typical of mass production, as a more customized operation for very dynamic and uncertain environments.

In a continuous process, the products are consumed at constant rates during long production times and the number of different products is very limited. In batch plants though, much smaller amounts of a larger diversity of products are usually manufactured during shorter production times. A semicontinuous operation mode lies on between a batch and a continuous operation. Its importance, further than an intermediate stage between a pilot plant and a continuous process, has been highlighted as a more flexible equipment utilization for simultaneous processing of medium size amounts of several products. Semicontinuous operations run continuously with periodic start-ups and shutdowns in order to accommodate their rather frequent product transitions. Production is organized in campaigns of finite length, each dedicated to produce a single product. However, an excessive number of too short campaigns might lead the plant to non-economic and non-realistic operation due to the changeover costs when switching the production from one product to another (Méndez and Cerdá, 2002).

Another complexity for the operational management of this type of plants comes out in the storage of intermediate products which becomes of special relevance. Intermediate storages debottleneck different upstream and downstream production rates. A satisfactory storage policy strongly determines the efficiency and flexibility of these plants working in semicontinuous mode.

Using semicontinuous processes aimed to the enhancement of the production flexibility is extensively discussed in this thesis. From this point of view, the consideration of multiple campaigns is undoubtedly a manner to increase flexibility in spite of its unrealistic practice or unfeasibility. However, still other rigid features can be found in most of the problem formulations found in the literature (Ierapetritou and Floudas, 1998b; Giannelos and Georgiadis, 2002; Méndez and Cerdá, 2002; Castro et al., 2004; Shaik and Floudas, 2007). Constant and invariable production rates are usually implicitly assumed for

the whole operation of the campaigns. Transitions in the production rate for a campaign are neglected and so, production adjustments are achieved through production lines stops, storage resources and multiple product campaigns.

The work in this chapter aims to improve the production schedules by developing a new concept of flexible fabrication, *in-campaign* production rate transitions. An MILP model based on this concept is proposed.

This chapter is organized as follows. Firstly, the mathematical formulation of the flexible manufacturing concept for semicontinuous processes is proposed in section 4.2. Next, two different case studies are tested and results are compared to those reported in the literature. Finally, in section 4.4, some conclusions are derived from this study regarding the opportunities that the generalized developed model allows by introducing scheduling flexibility through adjustable processing rates.

4.2 Mathematical formulation

The proposed MILP formulation is aimed at finding the optimal sequence of campaigns that maximizes the production, satisfying a minimum demand for each product. It is based on the model introduced by Méndez and Cerdá (2002), which has one major assumption in considering single campaigns for the production of each product. Although this assumption may lead to sub-optimal solutions, these authors (Méndez and Cerdá, 2002) argued that an excessive number of campaigns entails a much higher demand of manpower and costly equipment idle time which are non-economical in industry.

The proposed MILP formulation is next described in detail. Constraint 4.1 allows the assignment of at most a single production line j to every processing campaign i.

$$\sum_{j \in J_s} Y_{ij} \leq 1 \qquad \forall i \in I_s^+, s \in S \tag{4.1}$$

Constraint 4.2 ensures that each campaign must finish before a pre-specified time horizon (H), while equation 4.3 constrains the duration of a campaign (L_{ij}) to a minimum value (l_{sj}^{min}).

$$Tf_i \leq H \qquad \forall i \in I \tag{4.2}$$

$$l_{sj}^{\min} Y_{ij} \leq L_{ij} \leq Y_{ij} H \qquad \forall i \in I_s^+, j \in J_s, s \in S \tag{4.3}$$

Product release (ro_s) and unit ready (ru_j) times prevent the corresponding campaigns from starting before their values by means of constraint 4.4.

$$Tf_i - \sum_{j \in J_s} L_{ij} \geq \sum_{j \in J_s} \max\left[ru_j, ro_s\right] Y_{ij} \qquad \forall i \in I_s^+, s \in S \tag{4.4}$$

In constraint 4.5, the total production of a campaign (Q_i) is bounded to a minimum and maximum limits, depending on its duration and the maximum production rate of the equipment unit in which it is manufactured. Moreover, at the end of the time horizon, the amount of every final product by any campaigns must satisfy its minimum demand (d_s) in constraint 4.6.

$$\sum_{j \in J_s} r_{sj}^{\min} L_{ij} \leq Q_i \leq \sum_{j \in J_s} r_{sj}^{\max} L_{ij} \qquad \forall i \in I_s^+, s \in S \tag{4.5}$$

$$d_s \leq \sum_{i \in I_s^+} Q_i \qquad \forall s \in S^P \tag{4.6}$$

Constraints 4.7 and 4.8 sequence pairs of campaigns i and i' assigned to the same semicontinuous line. Constraint 4.7 is only active if campaign i precedes i', and constraint 4.8 otherwise. As this model considers also a possible changeover time between different products ($uch_{ii'j}$), the value of M_1 should be chosen equal to $H + max\{uch_{ii'j}\}$, in order to obtain the tightest relaxation.

$$\begin{aligned} &Tf_{i'} - L_{i'j} \geq Tf_i + uch_{ii'j} - M_1 \left(1 - X_{ii'}\right) - M_1 \left(2 - Y_{ij} - Y_{i'j}\right) \\ &\forall i, i' \in I, i < i', j \in J_i \cap J_{i'} \end{aligned} \tag{4.7}$$

$$\begin{aligned} &Tf_i - L_{ij} \geq Tf_{i'} + uch_{i'ij} - M_1 \cdot X_{ii'} - M_1 \left(2 - Y_{ij} - Y_{i'j}\right) \\ &\forall i, i' \in I, i < i', j \in J_i \cap J_{i'} \end{aligned} \tag{4.8}$$

Equation 4.9 establishes the mass balances between a campaign i producing and a campaign i' consuming intermediate state s. In this equation, $F_{ii'}$ is a continuous variable which stands for the amount of material transferred

between both campaigns. Similarly, equation 4.10 adjusts the material consumed by a campaign i which receives material from i'. The amount of material consumed by this campaign is translated to terms of campaign production by means of the coefficient ρ_{is}.

$$Q_i = \sum_{i' \in I_s^-} F_{ii'} \qquad \forall i \in I_s^+, s \in S^I \tag{4.9}$$

$$\rho_{is} Q_i = \sum_{i' \in I_s^+} F_{i'i} \qquad \forall i \in I_s^-, s \in S \tag{4.10}$$

Equation 4.11 introduces a binary variable ($U_{ii'}$) equal to one if $F_{ii'}$ is greater than zero, i.e. campaign i supplies material to campaign i'. In this case, M_2 must be carefully chosen greater than any value of $F_{ii'}$. Variable $U_{ii'}$ is employed to enforce that campaign i supplying material can never start later than campaign i' receiving this material (constraint 4.12).

$$F_{ii'} \leq M_2 \cdot U_{ii'} \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{4.11}$$

$$Tf_i - \sum_{j \in J_i} L_{ij} \leq Tf_{i'} - \sum_{j \in J_{i'}} L_{i'j} + H\left(1 - U_{ii'}\right) \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{4.12}$$

Previous constraints (equation 4.1 to 4.12) represent the least restrictive storage policy, i.e. unlimited intermediate storage (UIS) for which enough storage resources are assumed to be available at any time. In the opposite case, a non-intermediate storage (NIS) policy corresponds to the case in which no storage is available and materials have to be directly transferred between production lines. Then, to account for this NIS case constraint 4.13 does not allow a campaign i' consuming material in stock produced by i'.

$$Tf_i \geq Tf_{i'} - H\left(1 - U_{ii'}\right) \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{4.13}$$

Besides the UIS and NIS policies, this model can account for different finite intermediate storage (FIS) policies. In this case, material might be either stored in a limited number of tanks with restricted capacity, or directly bypassed between production lines. Then, in addition to the former constraints

4.1 to 4.12 for the UIS case, constraints 4.14 to 4.22 are applicable for the cases of scarce or limited storage resources.

Constraint 4.14 assumes that the beginning of a storage task storing intermediate material supplied by campaign i (IT_i) must coincide with the beginning of that campaign.

$$IT_i \geq Tf_i - \sum_{j \in J_s} L_{ij} \qquad \forall i \in I_s^+, s \in S^I \tag{4.14}$$

Constraint 4.15 enforces the storage of an intermediate material to finish (CT_i) not earlier than the completion of all the production campaigns consuming this material.

$$CT_i \geq Tf_{i'} - H\,(1 - U_{ii'}) \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{4.15}$$

Sequencing constraints for the storage tasks, equations 4.16 and 4.17, are considered in a similar way as for the sequencing production campaigns. In this case, an additional binary variable (W_{it}) indicates whether intermediate material produced by campaign i is transferred to tank t. Parameter $tch_{ii't}$ stands for the changeover time between different products in the same tank, thus, in order to have the tightest relaxation of this model, M_3 should be equal to $H + max\{tch_{ii't}\}$.

$$\begin{aligned} &IT_{i'} \geq CT_i + tch_{ii't} - M_3\,(1 - X_{ii'}) - M_3\,(2 - W_{it} - W_{i't}) \\ &\forall i, i' \in I, i < i', t \in T_i \cap T_{i'} \end{aligned} \tag{4.16}$$

$$\begin{aligned} &IT_i \geq CT_{i'} + tch_{i'it} - M_3 \cdot X_{ii'} - M_3\,(2 - W_{it} - W_{i't}) \\ &\forall i, i' \in I, i < i', t \in T_i \cap T_{i'} \end{aligned} \tag{4.17}$$

Constraint 4.18 assigns the value of 1 to the binary variable $Z_{ii'}$ whether a campaign i producing an intermediate consumed by campaign i', finishes before i' starts.

$$Tf_{i'} - \sum_{j \in J_{i'}} L_{i'j} - Tf_i \leq H \cdot Z_{ii'} \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{4.18}$$

Constraint 4.19 introduces a continuous variable $V_{ii'}$ which represents the amount of intermediate material consumed by i' at the end of campaign i

producing that material. Therefore, if $Z_{ii'} = 1$, that is, i and i' do not coincide in time, then $V_{ii'}$ must consequently be equal to zero.

$$V_{ii'} \leq M_2 \left(1 - Z_{ii'}\right) \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{4.19}$$

Equations 4.20 and 4.21 both constrain the value of $V_{ii'}$. Firstly, equation 4.20 forces $V_{ii'}$ to be at most as large as $F_{ii'}$. And secondly, constraint 4.21 establishes an upper bound for the value of $V_{ii'}$ assuming that during the time that i and i' run simultaneously, campaign i' consumes material from i at its maximum rate capacity.

$$V_{ii'} \leq F_{ii'} \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{4.20}$$

$$\begin{aligned} V_{ii'} \leq\ & \rho_{i's} \cdot \min\left(r_{ij}^{\max}, r_{i'j'}^{\max}\right) \cdot \left(Tf_i - Tf_{i'} + L_{i'j}\right) + M_2 Z_{ii'} + M_2 \left(1 - U_{ii'}\right) \\ & + M\left(1 - Y_{ij}\right) + M\left(1 - Y_{i'j'}\right) \\ & \forall i \in I_s^+, \forall i' \in I_s^-, s \in S, j \in J_i, j' \in J_{i'} \end{aligned} \tag{4.21}$$

Finally, constraint 4.22 restricts the utilization of a storage tank to its maximum volumetric capacity (v_t).

$$Q_i - \sum_{i' \in I_s^-} V_{ii'} \leq \sum_{t \in T_s} v_t W_{it} \qquad \forall i \in I_s^+, s \in S \tag{4.22}$$

To sum up, the objective function of this problem is the maximization of the production of final products:

$$\max \sum_{i \in I_s^+, s \in S^P} Q_i \tag{4.23}$$

subject to:

- constraints 4.1 to 4.12, for the UIS case.
- constraints 4.1 to 4.13, for the NIS case.
- constraints 4.1 to 4.12 and 4.14 to 4.22, for the FIS case.

4.3 Case Study

The proposed case study is aimed to illustrate the capabilities of this formulation. It refers to an industrial case of a fast moving consumer goods manufacturing plant. This is a classic case repeatedly addressed in the literature of scheduling of semicontinuous facilities (Zhang and Sargent, 1996; Ierapetritou and Floudas, 1998b; Méndez and Cerdá, 2002; Giannelos and Georgiadis, 2002; Castro et al., 2004; Shaik and Floudas, 2007).

The plant has the basic structure shown in figure 4.1. It consists of three parallel mixers sending material to five packing lines, both working in a semicontinuous mode, and a set of three storage tanks to buffer the production stocks. Three raw materials with non-restricted availability are blended in their corresponding mixers producing seven intermediates (I1-I7). Then, these intermediates are combined in a series of packing lines to produce fifteen final products (P1-P15). Figure 4.2 shows the STN representation for this plant. Table 4.1 presents the maximum production rate for each product and the availability of the semicontinuous units to process them. Sequence-dependent changeover times between different products are stated in table 4.2.

Table 4.1: Maximum rate capacities and units suitability.

Product	Available units	r_{max}, Tn/h
I_1, I_2	M1	17.0000
I_3, I_4	M2,M3	17.0000
I_5, I_6, I_7	M2,M4	12.2400
P_1	L3	5.5714
P_2, P_3	L1	5.8333
P_4, P_5	L2	2.7083
P_6	L3	5.5714
P_7	L1	5.8333
P_8, P_9	L2	2.7083
P_{10}, P_{11}	L5	5.3571
P_{12}, P_{13}	L4	2.2410
P_{14}, P_{15}	L4	3.3333

The objective function consists of the maximization of the production ensuring the fulfillment of a minimum final product demand, stated in table 4.3, over a time horizon of 5 working days (120 hours). Different storage policies have been considered, namely unlimited intermediate storage (UIS), no inter-

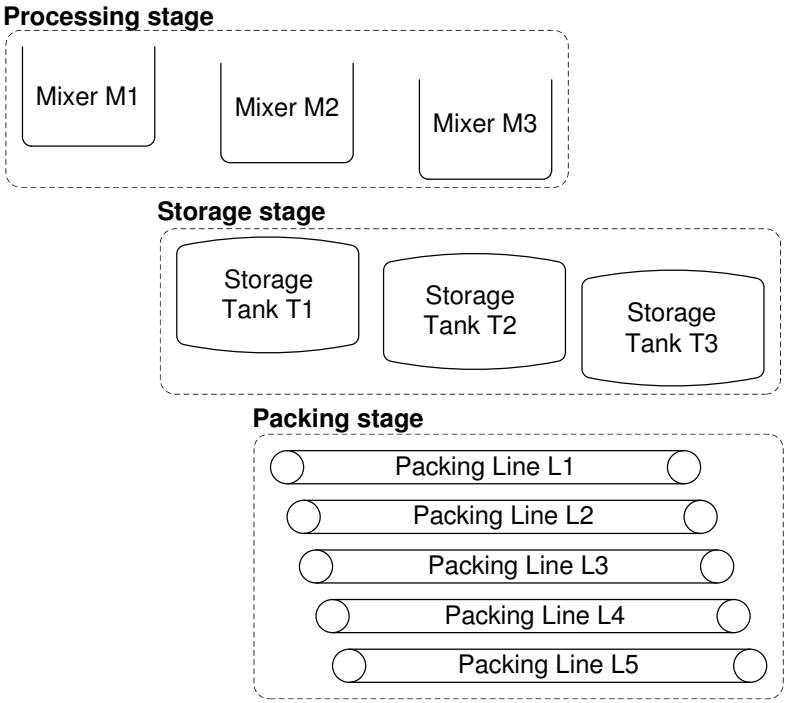

Figure 4.1: Schematic representation.

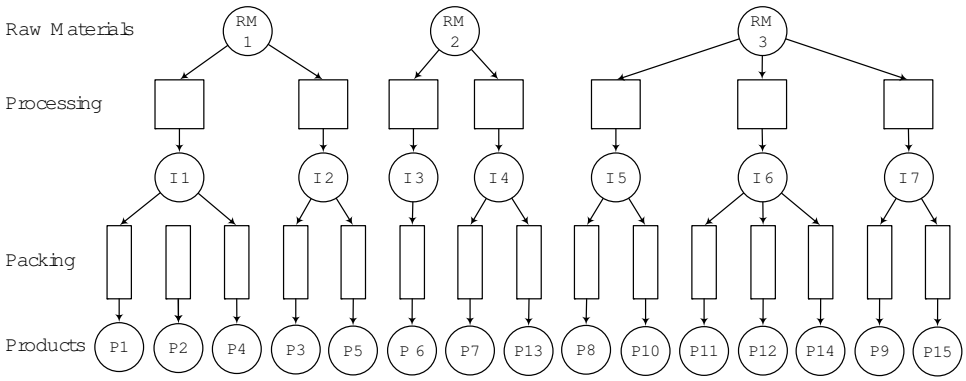

Figure 4.2: STN representation of the plant.

mediate storage (NIS) and finite intermediate storage (FIS). These cases have been implemented in GAMS, solved using CPLEX 10.0 in a 3 GHz computer and compared to the results previously reported in the literature.

From / To	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{12}	P_{13}	P_{14}	P_{15}
P_1						1							
P_2							1						
P_3							1						
P_4								4	4				
P_5								4	4				
P_6	1												
P_7		1	1										
P_8				4	4								
P_9				4	4								
P_{12}												2	2
P_{13}												2	2
P_{14}										2	2		
P_{15}										2	2		

Table 4.2: Changeover requirements in hours

Table 4.3: Minimum demand requirements.

Product	Demand, Tn
P_1	220.0
P_2	251.0
P_3	15.0
P_4	116.0
P_5	7.0
P_6	47.0
P_7	144.0
P_8	42.5
P_9	13.5
P_{10}	114.5
P_{11}	53.0
P_{12}	16.5
P_{13}	8.5
P_{14}	2.5
P_{15}	17.5

4.3.1 Unlimited Intermediate Storage

The optimal schedule obtained under an UIS policy is presented in figure 4.3. This solution represents the best achievable profit given the demand requirements and an unlimited storage capacity for the intermediate products. Due to the unconstrained storage capacity, all mixers in the processing stage can work at their maximum rate and so, reducing their idle times. Alternatively, in the packing stage, units are busy during the whole time horizon. This is due to the fact that the processing rates of the packing lines are much lower than the processing rates of the mixers. Therefore, in this case study, it should be pointed out that, whereas mixers are partially idle, production is limited by the processing rates of the packing units.

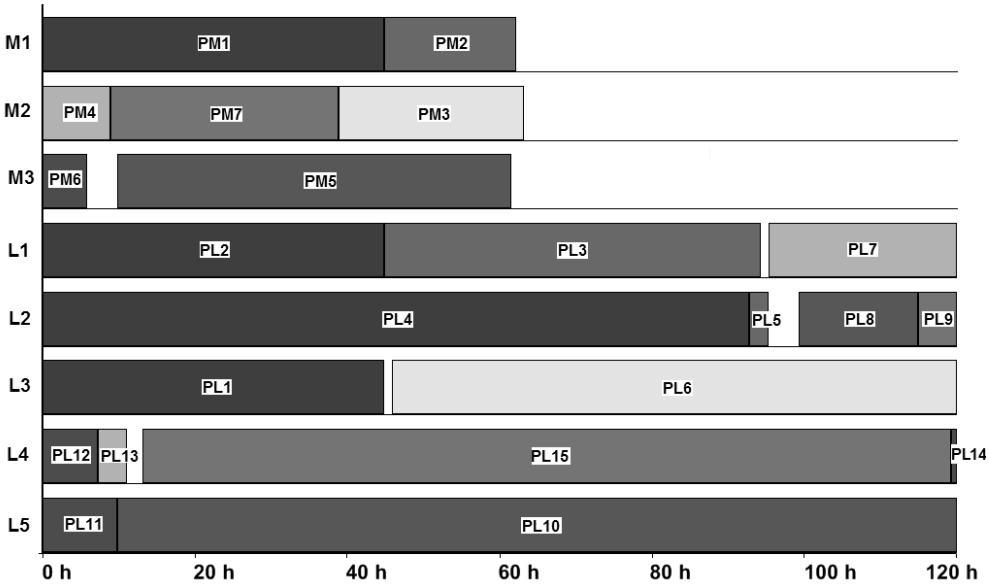

Figure 4.3: Optimal schedule for the UIS policy in CS1

Table 4.4 summarizes the results for the proposed and previous formulations. The same optimal profit value is achieved in all formulations. In the solutions reported by Castro et al. (2004) and Shaik and Floudas (2007), intermediates are produced in more than one campaign. In contrast, Méndez and Cerdá (2002) and the proposed solution of this work contain a single campaign for each intermediate and final product, which accommodates better to industrial practice, since the less number of campaigns the more economically efficient the plant will be.

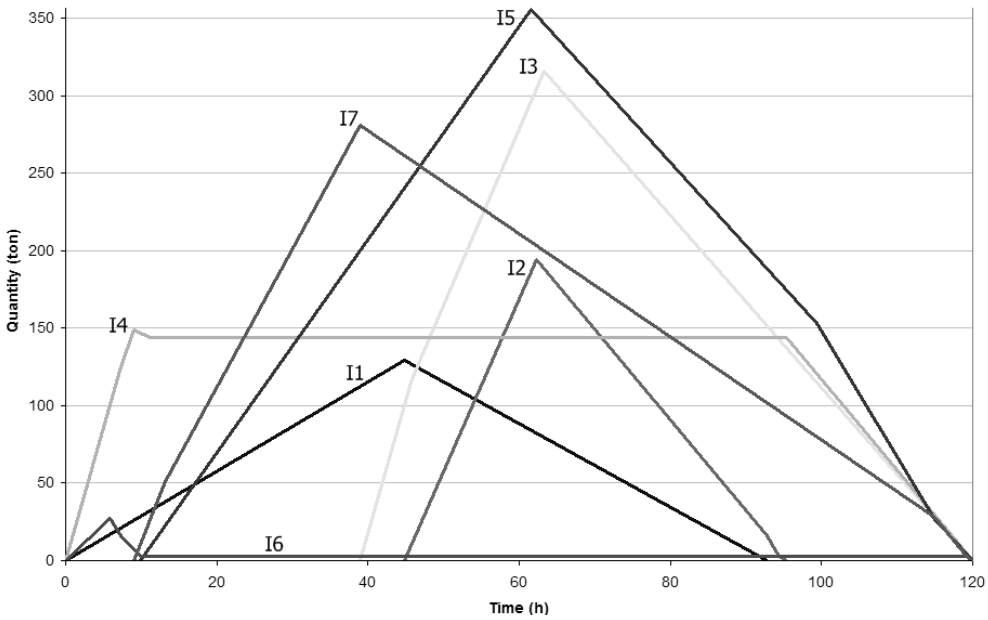

Figure 4.4: Profile of intermediate exceeding material for the UIS in CS1.

Table 4.4: Results for UIS policy for case study 1.

Model	Profit,m.u.	Campaigns/ Product	Bin.,cont.,rows	CPU,s
Shaik and Floudas (2007)	2695.32	> 1	108, 356, 1040	1.03
Castro et al. (2004)	2695.32	> 1	236, 762, 894	58.5
Méndez and Cerdá (2002)	2695.32	1	38, 44, 140	4.77
This work	2695.32	1	38, 85, 140	2.84

In the proposed solution (figure 4.3), as there is not any constraint on the storage, most intermediate products are stored. Figure 4.4 shows the stock profiles for each product which entails as many storage units as intermediates. As it can be seen, the necessary maximum tank capacity is more than 350 tons, an inefficient solution which would be translated into high operational and fixed costs.

This situation is unrealistic, since industries usually have storage space limitations that should have been taken into account by the production scheduler. However, this case is specially relevant to this study because it symbolizes an

upper bound for more restrictive cases. Furthermore, UIS policy represents the best solution in terms of production if the cost of storage is disregarded. More realistic approaches are presented in the following sections.

4.3.2 No Intermediate Storage

In this case, no available storage exists to buffer the production mismatches between the upstream and downstream facilities. Table 4.5 contains the results reported by the different authors. The model by Shaik and Floudas (2007) obtains a profit of 2689.75 m.u., whereas Castro et al. (2004) can only reach a profit of 2672.50 m.u. The formulation presented in this chapter obtains an optimal solution of 2688.31 m.u., which is better than the one reported by Castro et al. (2004), and slightly worse than the optimal reported value (0.05%). However, the proposed solution contains a single campaign per intermediate product, which has already been stated as a more efficient solution in industrial practice.

The solution obtained in this work is presented in figure 4.5. This solution is achieved by regulating the processing rate of the mixers according to the needs of the packing lines. For example, when mixer M1 processes intermediate I1, its processing rate changes from 8.280 to 14.113 ton/h and from 14.113 to 11.407 ton/h at times 34.12h and 95.32h, respectively. These transitions allow the downstream facilities to process the material generated by the mixers without needing intermediate storage.

As previously discussed, the bottleneck of the production resides in the packing stage due to the much lower processing rates of the packing lines compared to those of the mixers. Results from NIS policy may not be realistic in most cases, since intermediate storage tanks are usually available in semi-continuous plants. Nevertheless, on one hand, it proves the generality of the scheduling approach presented. And on the other hand, the solution of this case represents a lower bound for the solution for less restrictive cases. As a result, the solution to the scheduling problem is bounded by the solutions of the UIS case (upper bound) and NIS case (lower bound).

4.3.3 Finite Intermediate Storage

Previous works considered the availability of three storage tanks of 60 tons each. Results within this scenario are summarized in table 4.6. Shaik and Floudas (2007) and Castro et al. (2004) reported an optimal profit of 2695.32

Table 4.5: Results for NIS policy for case study 1.

Model	Profit,m.u.	Campaigns/ Product	Bin.,cont.,rows	CPU,s
Shaik and Floudas (2007)	2689.75	>1	108, 328, 1240	157.9
Castro et al. (2004)	2672.50	>1	228, 762, 894	2701
This work	2688.31	1	38, 119, 202	10.7

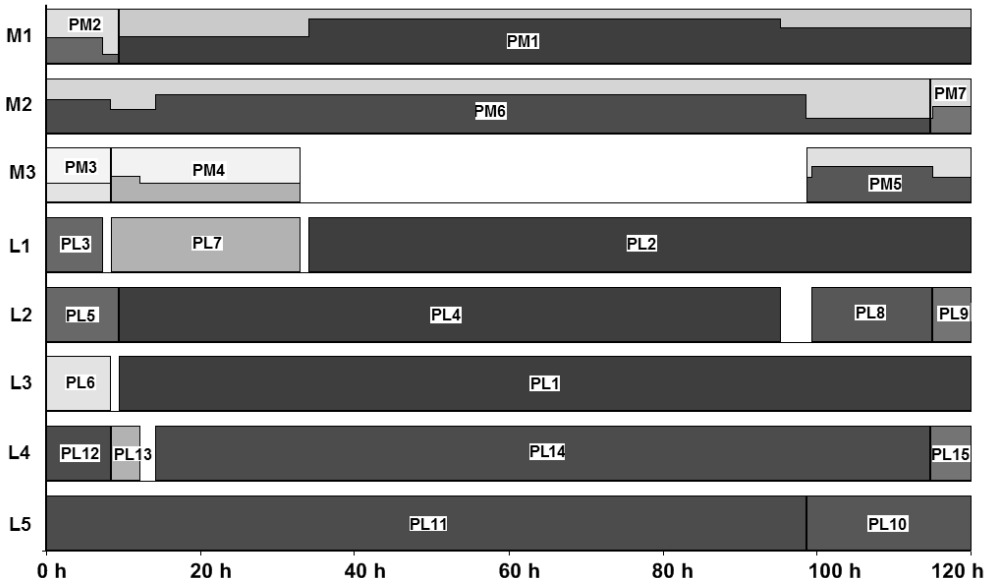

Figure 4.5: Optimal schedule for the NIS policy in CS1

m.u., which is the same as that obtained in the case of unlimited intermediate storage. Therefore, it can be stated that with three 60 ton tanks, an optimal profit is reached which can not be improved using additional tanks. Méndez and Cerdá (2002) yielded a lower profit of 2670.28 m.u. because their formulation only considers a single campaign per product. Alternatively, the model proposed in this work yields the optimal profit of 2695.32 m.u. with a single campaign per product because it can adjust the campaign production rates to the plant requirements.

Therefore, a storage capacity of three tanks of 60 tons each is high enough to be considered equivalent to the unlimited storage case. Reducing storage capacity to just two tanks of 60 ton each is considered next. In this scenario,

Table 4.6: Results for FIS policy with three tanks for case study 1.

Model	Profit,m.u.	Campaigns/ Product	Bin.,cont.,rows	CPU,s
Shaik and Floudas (2007)	2695.32	>1	276, 580, 4267	465.6
Castro et al. (2004)	2695.32	>1	330, 927, 1127	162
Méndez and Cerdá (2002)	2670.28	1	60, 87, 361	398.9
This work	2695.32	1	84, 148, 402	5.72

an optimal profit of 2695.32 m.u. is also achieved assuming a single campaign per product. Results using the proposed formulation are shown in table 4.7. The Gantt chart of the solution obtained is represented in figure 4.6.

Table 4.7: Results for FIS policy with two tanks for case study 1.

Model	Profit,m.u.	Campaigns/ Product	Bin.,cont.,rows	CPU,s
This work	2695.32	1	77, 148, 360	12.9

This optimal schedule has been obtained by adjusting the semicontinuous equipment processing rates. From this production plan, it can be observed, for example, that when mixer M3 processes intermediate I6, its processing rate changes from 7.598 ton/h to 5.357 ton/h and from this to 8.690 ton/h at times 7.36 h and 13.16 h, respectively.

4.4 Conclusions

This chapter considers production rate transitions within semicontinuous campaigns in order to improve production flexibility and storage management. Traditionally, constant production rates are considered for the whole operation, and so production adjustments are achieved through storage resources, production lines stops and multiple product campaigns. In contrast, this chapter proposes production adjustments by adapting processing rates inside a given campaign to production requirements. Certainly, the flexibility presented includes also fixed operation just by tightening up model parameters and bounds. As a result, the proposed model obtains as good results as those previously reported, with the advantage of less storage resources consumption and single

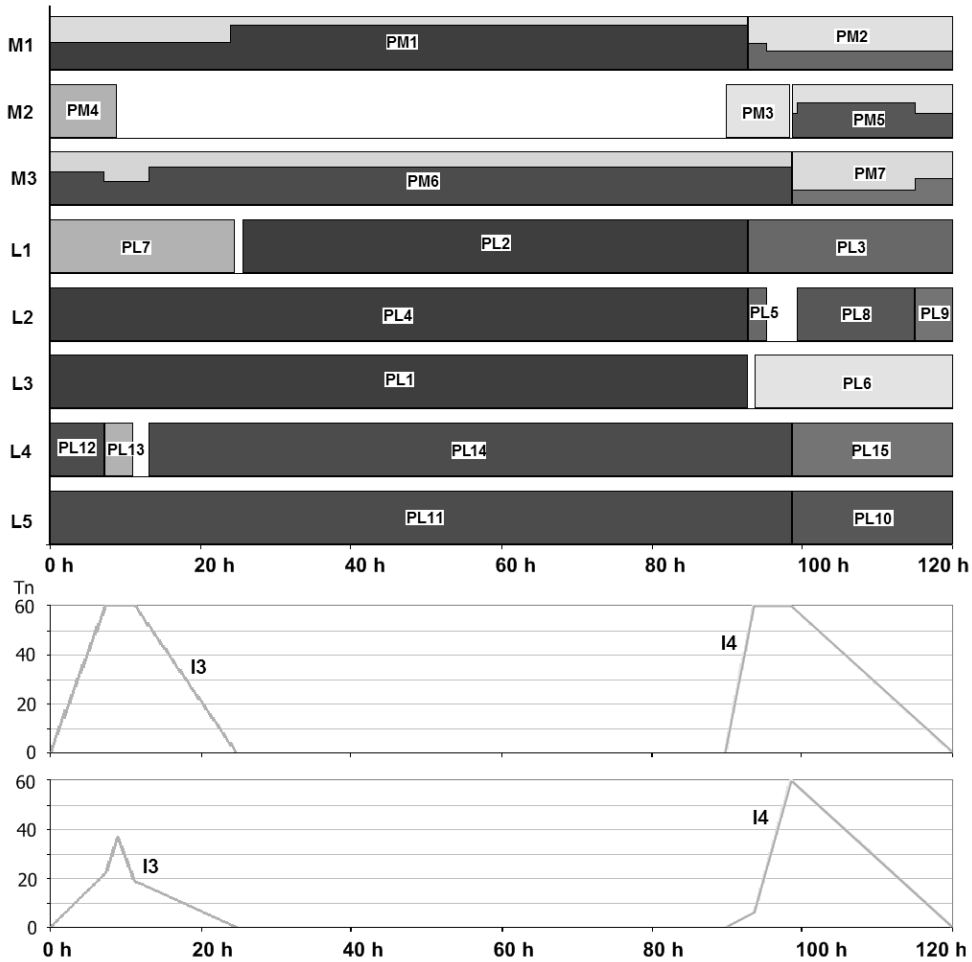

Figure 4.6: Optimal schedule for the FIS policy in CS1 with two storage tanks

product campaigns, which increases the system flexibility and cost efficiency. This allows reducing the capital cost of the plant by means of a flexible management of the storage facilities. Furthermore, this scheduling approach produces schedules which are more robust and tolerant against operation incidences.

On batch processes

Chapter 5

Transfer times in batch scheduling: a critical modeling issue

An effective short-term scheduling formulation must simultaneously deal with several problem difficulties commonly arising in batch processes. A critical feature to be considered when dealing with multi-stage batch plants is the movement of material through the process. A non-zero time as well as certain conditions are always required to transfer the material from one processing stage to the next one according to the specified product recipe. The transfer task consumes a period of time during which a proper synchronization of the equipment units supplying and receiving the material is enforced. Synchronization implies that during the execution of the transfer task, one unit will be supplying the material whereas the other one will be receiving it and consequently, no other task can be simultaneously performed in both units. Furthermore, synchronization is a critical limitation in batch scheduling that reduces the flexibility of production facilities. Hence, it is of paramount importance the accurate models of this essential issue.

Most of the existing mixed-integer linear programming (MILP) optimization approaches have traditionally dealt with the batch scheduling problem assuming zero transfer times, and consequently no synchronization, between consecutive processing stages. The problem simplification relying on negligible transfer times may work properly for the scheduling of multiproduct batch

plants with similar product recipes. However, this thesis demonstrates that ignoring the important role of transfer times may seriously compromise the feasibility of the scheduling whenever shared units and storage tanks, material recycles or bidirectional flows of products are to be considered. In order to overcome this serious limitations of current MILP-based scheduling approaches, a general precedence based framework accounting for non-zero transfer times is introduced. Also, two alternative methods that avoid generating unfeasible schedules are proposed and tested in different case studies involving zero transfer times.

5.1 Introduction

Scheduling, not only for manufacturing operations, is a common requirement for the industry, management and services. Scheduling problems can be posed in so assorted areas as personnel planning, computer design and production management. Operations research has focused in a general way on the research of new solution techniques for this kind of problems since late 50's in the past century (Baker, 1974; French, 1974; Baudin, 1998). Nevertheless, operations research particularly lacks in the modeling of the chemical industry operations becoming an active field of research for process systems engineers.

An exact optimal solution of the scheduling problem can be obtained by formulating it as a mathematical programming model. However, due to the highly combinatorial complexity of the scheduling problems, models usually need to be simplified in order to reduce the exponential growth of solution times with problem size. A wide range of different assumptions can be formulated within the modeling according to the intrinsic characteristics of the problem. For example, if a given activity can only be performed in a specific resource, a non-alternative resource policy formulation is adopted. Moreover, model parameters, such as processing times and demands, are assumed deterministic, if uncertainty is not explicitly taken into account.

Another typical simplification is to consider an unlimited intermediate storage (UIS) policy. This approach assumes that, if necessary, intermediate products are immediately stored after processing until the next task starts, i.e. unlimited storage is available between every pair of consecutive batch tasks. The aforementioned case entails an unrestrictive policy for intermediate storage management typical in mechanical industrial. Instead, the chemical industry is commonly characterized by shared tanks as well as zero-wait, non-intermediate or finite storage policies. Batch processes generally comprise multiple process-

ing stages, complex process layouts and topological implications which usually have a significant influence on the short-term scheduling problem complexity. The UIS policy is the most flexible for batch processing but caution is required to ponder to which extent this model accurately describes the actual plant operation

An additional example of model simplification is the symbolic workshop problem in the operations research which ignores the transfer times of the pieces between different tasks. However, chemical processes cannot stand this simplification. Fluids, contrarily to mechanical pieces, need proper containers to guarantee their handling and require specific units for transfer operations (pumps, piping, vessels, tanks, etc.).

However, in the literature of scheduling of chemical plants, the assumption of negligible transfer times is generally accepted. Authors usually avoid the complexity of managing these transfer operations by arguing that the overall transfer time represents a very small percentage compared to process operation times. Transfer of material between consecutive stages usually needs to take into account a larger number of possible combinations that may lead to an intractable number of constraints. Thus, transfer time modeling has commonly been assumed as an irrelevant feature within the mathematical optimization frameworks and consequently its importance has been scarcely addressed in the literature.

Most of the works explicitly addressing transfer times are only focused on the multiproduct batch plant case. This is the case of the work of Kim et al. (1996) who proposed a mixed-integer nonlinear programming formulation (MINLP) accounting for non-zero transfer times and various storage policies. Ha et al. (2000) considered non-zero transfer times using a sequence-based MILP model for a multiproduct plant. Castro and Grossmann (2005) proposed a multiple-time-grid resource task network (RTN) formulation for multiproduct batch plants and dealt with transfer times using an additional continuous variable.

For the more general and highly combinatorial multipurpose case, transfer time management has been an issue hardly treated in the literature. Transfer times are usually neglected or assumed to be lumped into the batch processing time (Sundaramoorthy and Karimi, 2005). Models relying on the concept of the batch precedence allow a straightforward treatment of the synchronization between consecutive stages and are able to easily deal with the transfer times. Several works have been reported in which the precedence-based scheduling model considers nonzero transfer times (Méndez et al., 2001; Heo et al., 2003;

Ferrer-Nadal et al., 2007).

This chapter focuses on the critical role of transfer times in the batch scheduling problem and points out that the assumptions introduced during the modeling process must be carefully analyzed to avoid generating schedules with unfeasible operational sequences. Accordingly, although simplifications are necessary to reduce problem complexity, they must be carefully considered in order to prevent unrealistic solutions.

This chapter is organized as follows. Firstly, an illustrative example is presented in section 5.2 in order to clearly expose the generation of unfeasible scheduling solutions and conclude with a series of claims in section 5.3. Next, four of the most well-known mathematical programming scheduling formulations are reviewed and analyzed in order to identify the sources of unfeasible schedules. In section 5.5, two approaches to avoid these shortcomings are proposed. The first one is based on a pre-processing algorithm which identifies unfeasible combinations arising from the problem formulation, whereas the second one solves the problem in a two-stage approach. Both strategies are aimed to a MILP problem formulation. Then, two benchmark case studies are addressed, which are used to test the major limitations of existing scheduling formulations. In addition, these results are compared with the solutions provided by the two proposed approaches and their performance is discussed. Finally, some remarks derived from this chapter are presented in the conclusions section 5.7.

5.2 An illustrative example

Consider a multipurpose plant produces different products, A and B, under the assumption of non-intermediate storage policy. Regarding the production recipe, raw material for processing product A is first treated in unit U1 for 3 hours, and then transferred to unit U2 and processed for 3 additional hours. As commonly happens in multipurpose batch plants, product B shares the same equipment units with product A and it is manufactured first in unit U2 for 2 hours and then in unit U1 for 4 hours. Transfer times for discharging and loading intermediate materials between both stages are neglected since they are in the order of a few minutes. This assumption may be derived from a zero-wait policy in which intermediate products have to be consumed immediately after production and transfers are usually very fast. Optimal solution requires the minimization of makespan as a criterion to increase the utilization of the resources and the plant efficiency.

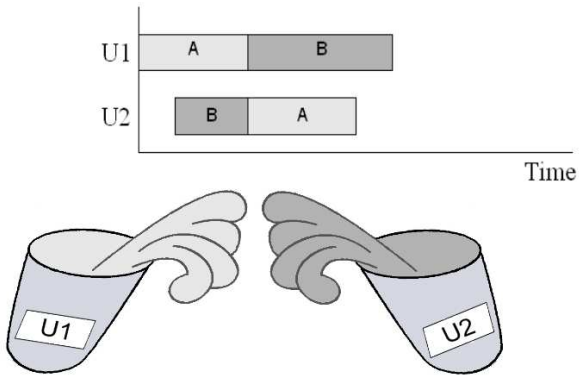

Figure 5.1: Unfeasible situation for the illustrative example.

For this straightforward example, the solution depicted in the Gantt chart of figure 5.1 will be obtained, if most of the MILP formulations available in the literature are applied. The value of the makespan is 7 hours. However, this solution is unfeasible in practice because it is impossible to transfer the material from unit U1 to unit U2 and simultaneously also material from U2 to U1. When the first stage of product A is finished, unit U2 need to be emptied to receive material from unit U1, at least for a while. However, at the same time unit U2 is trying to discharge product B to Unit U1, which generates an unfeasible situation if there is not any intermediate storage to hold its material.This problem cannot be solved unless an intermediate storage tank is available for transferring one of the intermediates to this storage while the other intermediate is transferred to its next processing unit.

Figure 5.2 shows the feasible optimal solution producing a makespan of 12 hours, almost 86% greater than the unfeasible one shown in figure 5.1. Here it is worth remarking that the unfeasible schedule cannot be transformed in the feasible one by only performing left or right shifting. New sequencing decisions were also required.

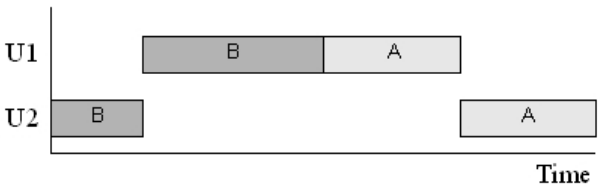

Figure 5.2: Feasible schedule for the illustrative example.

However, an intermediate storage tank does not guarantee feasibility in all instances. For example, deriving from the previous case, assume there is a shared storage tank but three batch transfers are being carried out simultaneously. In this case, two storage facilities would be necessary to make this situation feasible. Therefore, in general, n units form an unfeasible sequence (figure 5.3) if the materials are simultaneously transferred between them and there are less than $n-1$ additional intermediate storage units.

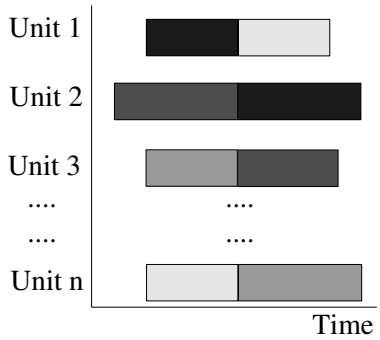

Figure 5.3: Generalization to n products.

5.3 Problem statement

Looking at the situation exposed in the illustrative example, the following claims are stated for the problem of unfeasible transfer:

- ***Claim 1.*** *The unfeasible situations may only appear in multipurpose plant configurations, where multidirectional flow can be generated.*
  In multipurpose plants, routes of different products may undergo equipment units in multiple directions following diverse paths. In contrast, multiproduct batch plants always operate following unidirectional flow that cannot generate unfeasible sequences. In this particular case, nonzero transfer times can be easily incorporated to the scheduling by simply extending task durations and performing right shifting.

- ***Claim 2.*** *Unfeasible solutions may appear under a non-UIS policy.*
  Storage policy assumption is an important issue. Under the simple consideration of unlimited intermediate storage (UIS), unfeasible solutions do not occur since additional space is always available when needed in

transfer operations. However, under more restrictive storage policies, such as common intermediate storage (CIS), finite intermediate storage (FIS) or even non-intermediate storage (NIS), unfeasible situations may arise.

- ***Claim 3.*** *Transfer times are an important matter for tasks synchronization.*
  The main source of these unfeasible solutions has its roots in the usual assumption of negligible transfer times. Since transfer times often represent only a small percentage of the whole task duration, in mathematical formulations they are often neglected or just summed up to the processing times in order to reduce the problem complexity. However, in scheduling problems, transfer time plays a key role in terms of tasks synchronization. Transfer entails that, simultaneously to the emptying of a given product from a unit, the receiving unit is being filled for a transfer time period. When transfer times are neglected or assumed equal to zero, tasks synchronization among units is ignored, leading to the unfeasible solutions in practice as shown in the previous examples. Hence, the modeling of multipurpose batch plants with limited storage policies must consider transfer times. Otherwise, synchronization among tasks regarding transfer times is completely omitted, and unfeasible solutions can be reached. This fact has not been taken into account in most of the existing mathematical formulations for short-term scheduling. Also, the explicit modeling of this feature can be awkward to address by using most of the existing optimization frameworks.

5.4 Mathematical programming formulations for the batch scheduling problem

The scheduling problem has received a great attention over the recent decades as a manner to improve the efficiency of batch chemical processes usually aimed at the production of high-added value products. In this chapter, four of the most popular scheduling continuous-time formulations available in the literature have been selected and implemented. The aim is to analyze these formulations in order to identify the lack of consideration of transfer times which is leading to incorrect task synchronization.

Next, main features of MILP models based on the STN and RTN global time points, unit specific time events and general precedence are described.

The general precedence model is completely developed in next section because an extension to the previously published formulation dealing with intermediate storage (Méndez and Cerdá, 2003a) is also included in this chapter.

5.4.1 State-Task-Network based continuous formulation

There have been many efforts to develop a continuous-time formulation (Giannelos and Georgiadis, 2002; Maravelias and Grossmann, 2003) based on the original STN representation proposed by Kondili et al. (1993a). For the testing purpose, we have selected the work by Maravelias and Grossmann (2003) because it is able to handle general batch process concepts such as variable batch sizes and processing times, various storage policies or sequence-dependent changeover times. This approach is based on the definition of a common time grid that is variable and valid for all shared resources. This definition involves time points occurring at unknown time. To guarantee the feasibility of the material balances at any time during the time horizon of interest, the model imposes that all tasks starting at a time point must occur at the same time. However, the ending time does not necessarily have to coincide with the occurrence of a time point, except for those tasks that need to transfer the material with a zero-wait time policy. For other storage policies, it is assumed that the equipment can be used to store the material until the occurrence of next time point. The model adopts two binary variables, to denote at which time point a given task starts and finishes. A continuous variable represents the quantity of each resource available at each event point. The number of time intervals is a critical issue for all continuous-time models. The selected approach increases the number of time intervals from a relative small number until no improvement in the objective function is achieved. Further details of this formulation can be found in appendix B.1.

None of the proposed STN-based continuous-time formulations available in the literature considers transfer times in their formulation. Therefore, it is predictable that this simplification will have a negative effect on the synchronization of tasks and unfeasible optimal results may appear. This fact is explicitly proven in the case studies section.

5.4.2 Resource-Task-Network based continuous formulation

The RTN-representation was firstly introduced by Pantelides (1994). Further improvements were achieved by Castro et al. (2001) and Castro et al. (2004).

The improved model version (Castro et al., 2004) is tested in this work. This approach adopts a common time grid for all resources. As other continuous time formulations the length of each time interval is unknown and is to be determined. In addition, a timing parameter is used to define the number of event points allowed between the beginning and ending of a batch task, in order to reduce the number of event points considered and so, the problem complexity. However, an exceedingly small value might prevent the formulation from reaching the global optimum or turn the model unfeasible. The use of a fixed value is a quite reasonable assumption in cases where task processing times are of the same order of magnitude, where it is expected that few events exist between the starting and finishing of a task.

The RTN representation considers two types of items: tasks and resources. A task defines an operation that transforms a certain set of resources into another set at the end of its duration. A resource includes all entities that are involved in process steps, such as materials (raw materials, intermediates and products), processing and storage equipment (tanks, reactors, etc) and utilities (operators, steam, etc). All equipment resources, with the exception of storage tanks, are considered individually, moreover only one task can be executed in any given equipment resource at a certain time. The starting and finishing time points for a given task are defined through only one set of binary variables. It makes the model simpler and more compact, but on the other hand, it increases the number of constraints and variables to be defined. The process resource variable represents the excess amount of a given resource at each time point. Further details of this formulation can be found in appendix B.2.

RTN continuous models for multipurpose plants reported in the literature do not consider transfer time in their formulation. Therefore, in cases where neglecting transfer times influences the synchronization of tasks, unfeasible optimal results may appear, as proven in the test case studies.

5.4.3 Unit-Specific Time Event formulation

The original idea of unit-specific events was firstly presented by Ierapetritou and Floudas (1998a) and then improved by Vin and Ierapetritou (2000), Lin et al. (2002) and Janak et al. (2004). This is a flexible representation of the scheduling problem which is able to account for different intermediate storage policies and other resource constraints. The global time point representation is efficiently reformulated in these models: a) by considering as an event just the starting of a task, and b) by allowing event points to take place at different

times in each different unit. Then, the number of event points and associated binary variables are reduced compared to the global time point representation. Although this representation is mainly oriented to batch network processes, it can easily deal with sequential processes. This formulation requires the definition of the number of event points, especially critical when dealing with resource constraints and inventories. Probably the most functional strategy is starting with a small number of event points and to increase this number iteratively until there is no improvement in the objective function value. This formulation does not account for transfer times between tasks assuming them as negligible compared to the processing times.

The formulation proposed by Janak et al. (2004) has been used for the testing purpose of this work. As it can be seen, this formulation also neglects transfer times and then, the entailed synchronization of tasks. Hence, the optimal solutions obtained with this model may be unfeasible, as it will be proved in the case studies addressed. Further details of this formulation can be found in appendix B.3.

5.4.4 General precedence formulation

A very convenient approach for dealing with sequential processes is based on the concept of immediate batch precedence which was initially presented by Cerdá et al. (1997). Subsequent works (Méndez et al., 2001; Méndez and Cerdá, 2003a) developed a more efficient continuous-time MILP formulation that relies on the notion of general precedence. This generalized precedence notion extended the immediate predecessor concept to consider all batches belonging to the same processing sequence. As it can be seen in figure 5.4, this model defines a couple of sets of binary variables in order to sequence ($X_{pisp'i's'}$) and allocate (Y_{pisu}) processing tasks.

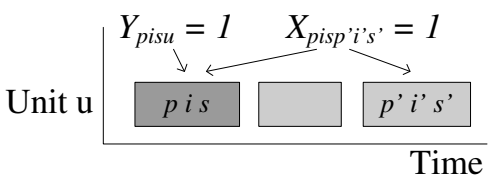

Figure 5.4: General precedence representation.

Y_{pisu} is a binary variable equal to 1 whenever task (p, i, s), that is the stage s for manufacturing the batch i of product p, is allocated to equipment unit u.

Regarding the sequencing decisions, $X_{pisp'i's'}$ is a binary variable which establishes the general precedence relationship between a pair of tasks (p, i, s) and (p', i', s') executed at the same processing unit (otherwise $X_{pisp'i's'}$ is meaningless). If $X_{pisp'i's'}$ is equal to 1, task (p, i, s) is a direct or non-direct predecessor of task (p', i', s') on the waiting line for the allocated unit. Alternatively, in case of task (p', i', s') is processed before than task (p, i, s) in the same unit, $X_{pisp'i's'}$ takes zero value. It is worth noting that the six subindices defined for sequencing variables are needed to deal with the general scheduling problem arising in multipurpose batch plants, in which the same equipment unit can perform several operations related to the same or different products. Consequently, the sequencing variable can distinguish not only the batches and the products involved but also the stages that are being sequenced. Although the number of binary variables seems to be very large at first sight, it should be noted that sequencing variables are only defined for every pair of tasks (p, i, s) and (p', i', s') that can be performed in the same unit, which is an intrinsic characteristic of multipurpose equipment. If the general proposed scheduling method is applied to a multiproduct batch plant, the subindices related to the stages in the sequencing variables are not longer required.

Processing unit constraints. Constraints 5.1 state that a single processing unit must be assigned to every required processing task.

$$\sum_{u} Y_{pisu} = 1 \qquad \forall p, i, s, u \tag{5.1}$$

Constraint 5.2 expresses the duration of a task, starting (Ts_{pis}) and finishing time (Tf_{pis}), by considering:

1. the overall time required to perform the loading of material (transfer time) from the previous stage ($tt_{pu'}$),
2. the batch processing operation itself (pt_{ps}),
3. the batch processing operation itself (pt_{ps}),
4. a possible waiting time in the processing unit (Tw_{pis}) and,
5. the unloading of the material to either next stage or to a suitable intermediate storage tank (tt_{pu}).

An illustrative representation of model variables is depicted in figure 5.5.

$$Tf_{pis} = Ts_{pis} + \sum_{u' \in U_{p(s-1)}} tt_{pu'} + \sum_{u \in U_{ps}} (pt_{psu} + tt_{pu}) Y_{pisu} + Tw_{pis} \qquad \forall p, i, s \tag{5.2}$$

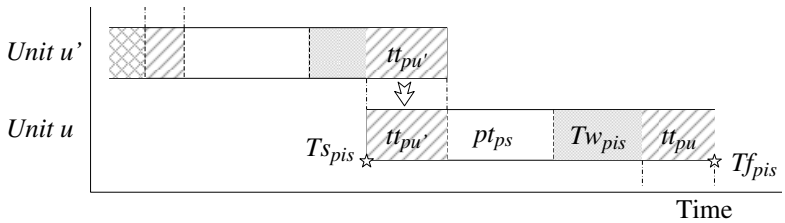

Figure 5.5: Task duration.

Constraints 5.3 and 5.4 sequence two batches of two different products processed in the same unit. Constraint 5.3 is active if task (p, i, s) precedes task (p', i', s') while constraint 5.4 is active in the opposite case. In order to reduce the number of binary variables $X_{pisp's'i'}$ constraints 5.3 and 5.4 are only used in case of product p appears before than p' $(p < p')$ or if $p = p'$ for $s < s'$.

$$\begin{array}{l} Ts_{p'i's'} \geq Tf_{pis} - M\left(1 - X_{pisp'i's'}\right) - M\left(2 - Y_{pisu} - Y_{p'i's'u}\right) \\ \forall (p, i, s), (p', i', s'), u \in \left(U_{ps} \cap U_{p's'}\right) : (p < p') \ or \ (p = p' \ and \ s < s') \end{array} \tag{5.3}$$

$$\begin{array}{l} Ts_{pis} \geq Tf_{p'i's'} - M \cdot X_{pisp'i's'} - M\left(2 - Y_{pisu} - Y_{p'i's'u}\right) \\ \forall (p, i, s), (p', i', s'), u \in \left(U_{ps} \cap U_{p's'}\right) : (p < p') \ or \ (p = p' \ and \ s < s') \end{array} \tag{5.4}$$

Constraint 5.5 sequences two batches i and i' of the same product p at the same stage s. Substantial savings in the number of this constraint is achieved considering that batch i is processed before batch i' $(i < i')$.

$$Ts_{pi's} \geq Tf_{pis} \qquad \forall p, i < i', s \tag{5.5}$$

The task precedence constraint 5.6 is defined for every pair of consecutive tasks that must be sequentially performed for a particular product. A task can never begin before the material from the preceding task starts being transferred to the unit assigned. Transfer times enforce that unloading and loading

operations from/to units involving consecutive tasks must be synchronized, unless the material is previously stored in an intermediate storage tank.

$$Tf_{pis} - \sum_{u \in U_{ps}} tt_{pu} Y_{pisu} \leq Ts_{pi(s+1)} \qquad \forall p, i, s \tag{5.6}$$

Storage constraints. One of the major advantages of the general precedence notion which strongly influences its efficiency is the fact that the same sequencing variables used for a pair of processing tasks can be utilized for their related storage tasks. However, the formulation presented by Méndez and Cerdá (2003a) should be further generalized by allowing selective interconnection between processing units and storage tanks facilities. A new binary variable AT_{pist} denotes whether task (p, i, s) is sent to storage tank t. Then, constraint 5.7 expresses that the material from task (p, i, s) may be stored in a tank t only if processing unit u is connected to storage tank t.

$$AT_{pist} \leq \sum_{u \in T_u} Y_{pisu} \qquad \forall p, i, s, t \tag{5.7}$$

Constraint 5.8 works together with 5.6 in order to sequence two stages of the processing of a batch. In case of intermediate storage is not used, both constraints are the same with different type of inequalities, becoming both in a single equality constraint. Otherwise, constraint 5.8 is relaxed.

$$Tf_{pis} - \sum_{u} tt_{pu} Y_{pisu} \geq Ts_{pi(s+1)} - M \cdot \sum_{t} AT_{pist} \qquad \forall p, i, s, u \tag{5.8}$$

Storage task sequencing constraints 5.9 and 5.10 (figure 5.6) define the order of storage tasks (p, i, s) and (p', i', s') assigned to the same tank. Constraint 5.9 is only active when task (p', i', s') precedes the task (p, i, s) while constraint 5.10 is active in the opposite case.

$$\begin{aligned} Tf_{p'i's'} - \sum_{u'} tt_{p'u'} Y_{p'i's'u'} \geq Ts_{pi(s+1)} + \sum_{u} tt_{pu} Y_{pisu} \\ - M\left(1 - X_{pisp'i's'}\right) - M\left(2 - AT_{pist} - AT_{p'i's't}\right) \\ \forall (p, i, s), (p, i, s), t, u \in U_{p,s}, u'' \in U_{p,s+1}, u' \in U_{p',s'} \end{aligned} \tag{5.9}$$

$$
\begin{aligned}
Tf_{pis} - \sum_{u'} tt_{pu} Y_{pisu} \geq Ts_{p'i'(s+1)} + \sum_{u'} tt_{p'u'} Y_{p'i's'u'} \\
- M \cdot X_{pisp'i's'} - M \left(2 - AT_{pist} - AT_{p'i's't}\right) \\
\forall (p,i,s),(p,i,s),t,u \in U_{p,s}, u'' \in U_{p,s+1}, u' \in U_{p',s'}
\end{aligned}
\tag{5.10}
$$

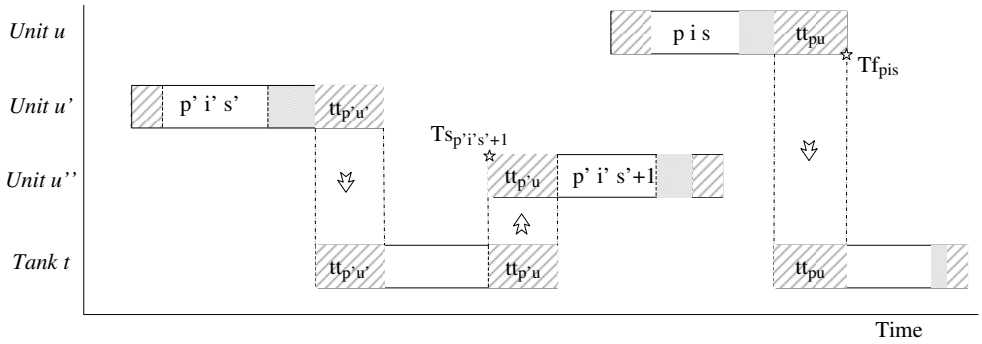

Figure 5.6: Illustrative representation of storage constraint 5.10.

In turn, constraint 5.11 sequences a pair of tasks of two different batches i and i' of the same product p sharing the same intermediate storage t (figure 5.7).

$$
\begin{aligned}
Tf_{pi's} - \sum_{u} tt_{pu} Y_{pi'su} \geq Ts_{pi(s+1)} + \sum_{u} tt_{pu} Y_{pisu} \; - M \cdot \left(2 - AT_{pist} - AT_{pi'st}\right) \\
\forall p,i,i',s,t
\end{aligned}
\tag{5.11}
$$

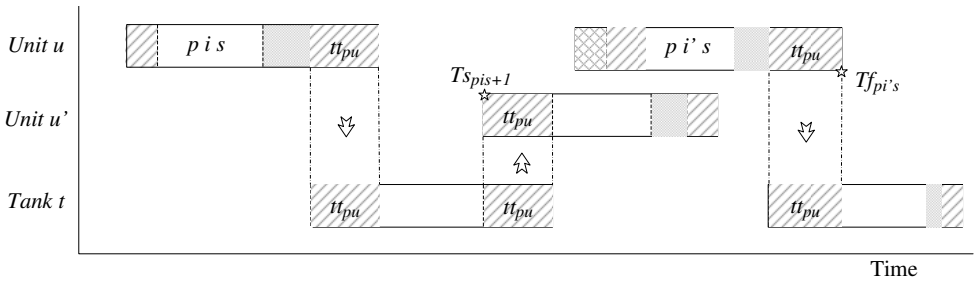

Figure 5.7: Illustrative representation of storage constraint 5.11.

Objective function. Equation 5.12 expresses the objective function in terms of makespan minimization.

$$\min \quad MK \geq Tf_{pis} \quad \forall p, i, s \tag{5.12}$$

The general precedence model allows including transfer times in tasks. But, if negligible transfer times are set to zero in the formulation, this model may fail to observe the synchronization of tasks, thus leading to unfeasible schedules. This will be proved in the testing case studies.

5.5 Solution approaches

This kind of unfeasible schedules was firstly identified under NIS conditions by Sanmartí et al. (2002) using S-graph representation for the scheduling problem (see section 2.2.2.3 in chapter 2). In S-graph, unfeasible solutions can be easily pre-detected beforehand in the bounding procedure. They appear as directed cycles in the graph which are identified and excluded using the algorithm described by Cormen et al. (1997). Romero et al. (2004) extended the use of the S-graph framework to include the common intermediate storage policy and applied a similar algorithm to cycle detection and thus discarding unfeasible solutions. Ferrer-Nadal et al. (2006) carried out a comparative study between the S-graph and a MILP formulation highlighting advantages and inconveniences for both representations. Although S-graph has proved to be a very efficient framework for solving the scheduling problem, it is still a limited framework under development which is not able to include neither some modeling aspects in complex plant configurations nor other kind of resource constraints than the processing units themselves.

Two different strategies are presented to deal with unfeasible sequences if the problem specifically requires zero transfer times in order to ensure the correct synchronization between tasks. The first solution method applies a pretreatment algorithm to generate a series of integer cuts that are introduced in the overall MILP formulation as new constraints. The second approach is based on a two stage algorithm which solves sequentially a MILP with very small transfer times and next a linear programming (LP) problem. Both of them are aimed to be embedded in a mathematical programming formulation, more specifically, they can be directly used in the general precedence formulation.

5.5.1 Pre-treatment algorithm & integer cuts

The first method proposed to avoid unfeasible sequences consists of firstly identifying those groups of tasks that can generate unfeasibilities with a search algorithm, then adding new constraints into the mathematical formulation to avoid them, and finally solving the resulting MILP problem (figure 5.8).

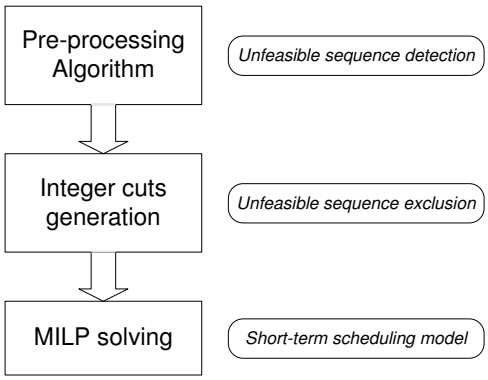

Figure 5.8: Pre-processing and integer cuts strategy.

For limited storage policies, a potentially unfeasible generated sequence is characterized by groups of n pairs of consecutive tasks such that the equipment unit of the second tasks of all given pairs in the group is also used as equipment unit for the first task of the pairs of tasks in the group (figure 5.9).

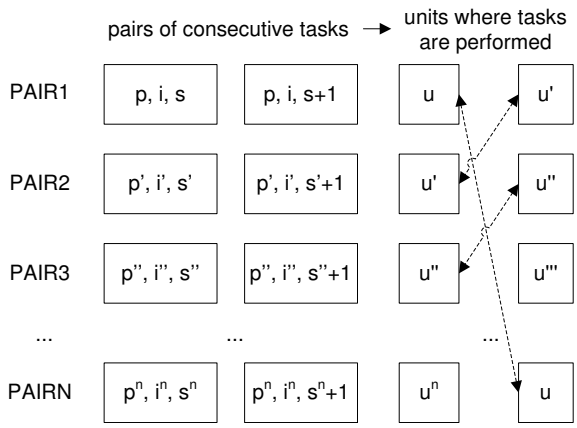

Figure 5.9: General scheme for a group of tasks that generate an unfeasible sequence.

The proposed algorithm (figure 5.10) detects all those groups of tasks that may result in an unfeasible sequence. The algorithm considers a set of products p, their batches i, and each product is processed in consecutive stages s; so, a task is defined by the product, its batch and stage: (p, i, s). $CPAR$ is a set that contains a list of pairs $PAIR$ ($CPAR = PAIR1...PAIRN$). Each pair $PAIR$ in the list is defined by two consecutive tasks (p, i, s) and $(p, i, s + 1)$ as well as their corresponding units u and u', respectively. The set of pairs $CPAR$ must contain as many different units as pairs of tasks. If a given group of pairs of tasks stored in $CPAR$ constitutes an unfeasible sequence, then it is stored in the set $SUNF$. An iterative search algorithm that looks through all the pairs of consecutive tasks $(PAIR(npair), npair = 1...lastpair)$ identifies the potential pairs that can create an unfeasible sequence, $CPAR$, and finally it determines whether a given pair of tasks closes the unfeasible sequence, $SUNF$. This algorithm can also be classified as recursive since it is repeated as many times as necessary in order to increase the group with an additional pair of tasks until an unfeasible sequence is found, and so the number of orders included in a given sequence is represented by $nlevel2$.

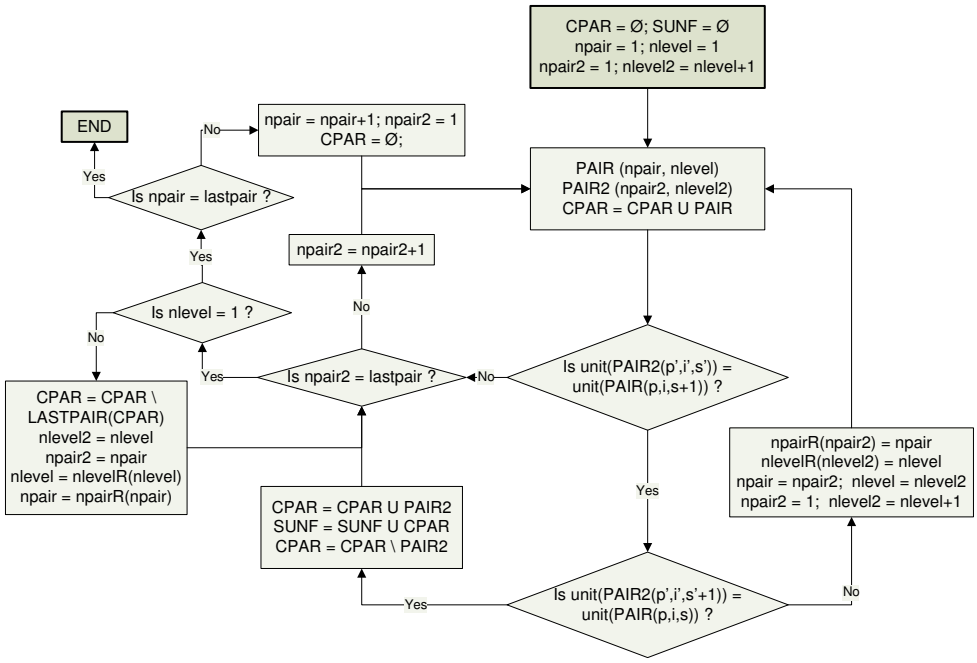

Figure 5.10: Pre-processing algorithm.

Given the information obtained from this algorithm, a number of integer

cuts are added as additional constraints to the original formulation in order to exclude these potential unfeasible configurations of assignments and sequences. Equation 5.13 expresses these integer cuts where $ZERO$ and ONE are the sets of tasks (p, i, s) for which $X_{pisp'i's'} = 1$ and $X_{pisp'i's'} = 0$, respectively.

$$\sum_{\substack{(pisp'i's') \in ONE \\ \forall CPAR \in SUNF}} X_{pisp'i's'} - \sum_{(pisp'i's') \in ZERO} X_{pisp'i's'} \leq |ONE| - 1 \tag{5.13}$$

Computational effort for solving the mathematical problem is not increased since no new variables are created and even the search space is slightly reduced. However, this pre-processing algorithm is restricted to sequential product recipes, with no alternative equipment units working in parallel, and no revisiting of equipment is allowed.

5.5.2 Two-stage formulation

The second alternative consists of a two-stage algorithm (figure 5.11) for unfeasible schedule removal when zero transfer times are requested. In the first step, very small transfer times are specified to achieve a synchronization which automatically discards unfeasible configurations. Then, from the previous solution, allocation and sequencing variables are fixed, and as a second step, the problem is again solved specifying zero transfer times. In this second step, original MILP problem becomes a LP problem in which units can be synchronized by performing left or right shifting. Furthermore, computational effort is almost negligible because all the decision binary variables are already fixed. This makes this strategy very suitable for large problems without extra CPU time.

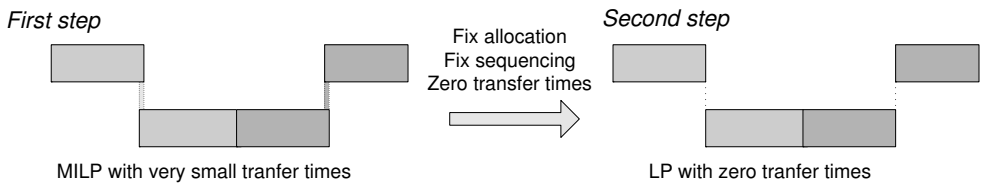

Figure 5.11: Two-stage algorithm.

5.6 Case studies

Two testing case studies have been posed to be solved with the four aforementioned mathematical models for batch plant scheduling. They consist of two multipurpose batch plants with different products under different storage policies and under the assumption of zero transfer times. Although the general precedence models explicitly include transfer times, they are set to zero in order to analyze the results obtained under this assumption. For the sake of simplicity, the objective function consists of the minimization of the makespan. Since optimal unfeasible solutions appear with the previous models, the two proposed solution approaches are applied and the new results are compared to the previous ones. The mathematical models and algorithm have been implemented in GAMS and solved using the MILP solver CPLEX 9.0.

5.6.1 Case study 1

This case study contemplates a multipurpose batch plant which manufactures four products (A, B, C and D) in three units (U1, U2 and U3) through three processing stages. The production recipe and suitable equipment units are shown in table 5.1 and figure 5.12. The case study consists of producing two batches of product A, and one batch of each of the other three products. Four instances of the same problem have been solved using different storage policies, namely, unlimited intermediate storage (UIS), non intermediate storage (NIS), common intermediate storage (CIS), and zero-wait time (ZW).

Table 5.1: Production data for case study 1.

	Stage 1		Stage 2		Stage 3		
Prods.	Unit	Time,h	Unit	Time,h	Unit	Time,h	Batches
A	U1	6	U3	9	U4	7	2
B	U2	9	U3	15	U4	17	1
C	U4	8	U1	14	U2	16	1
D	U2	7	U3	11	U1	4	1

Optimal schedules for this case study are presented in figure 5.13 along with their corresponding Gantt charts. The feasible solutions found using the two proposed solution approaches have a higher makespan value compared to the solutions given by the four MILP formulations. However, the latter solutions include unfeasible operation sequences, which are avoided in the solutions

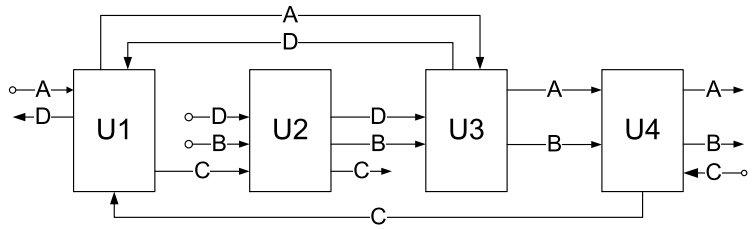

Figure 5.12: Product recipes for case study 1.

obtained by applying any of the two solution approaches in this chapter.

For the UIS scenario, all the formulations produce feasible solutions because enough storage capacity is available to transfer the materials between different units. However, these formulations result in unfeasible situations for the more restrictive cases as NIS, CIS or ZW. When considering a CIS policy with one available storage tank (figure 5.13b), an unfeasible situation is identified when product A is transferred from unit U3 to the storage tank T1 at time 39 h and simultaneously product D is transferred from T1 to U3. Looking at the Gantt chart of the NIS solution in figure 5.13d, three simultaneous transfers take place at time 15 hours. The second batch of product A must be transferred from unit U1 to unit U3, the first batch of A from unit U3 to unit U4, and product C from unit U4 to unit U1. These simultaneous transfers can not actually be performed unless two additional storage units would be available at this point. Similar reasoning can be applied to the solution shown in figure 5.13f using the most restrictive ZW policy. In the solution given by the four mathematical formulations, product D and the first batch of product A must be transferred simultaneously at time 35h, product D from unit U3 to unit U1, and the product A from unit U1 to U3.

These unfeasible situations are avoided applying either the proposed preprocessing algorithm plus integer cuts approach or the two-stage algorithm. In terms of makespan, this case study is giving a maximum of 10% difference between both solutions (6 hours) for the case of NIS. Here, it is worth remarking that this difference is quite substantial in industrial practice since it would cause a bottleneck of 6 hours or an immediate full-rescheduling task when unfeasible situations are observed. Significant differences in sequencing and timing decisions can easily be observed when comparing the Gantt charts corresponding to the feasible schedules and the unfeasible ones.

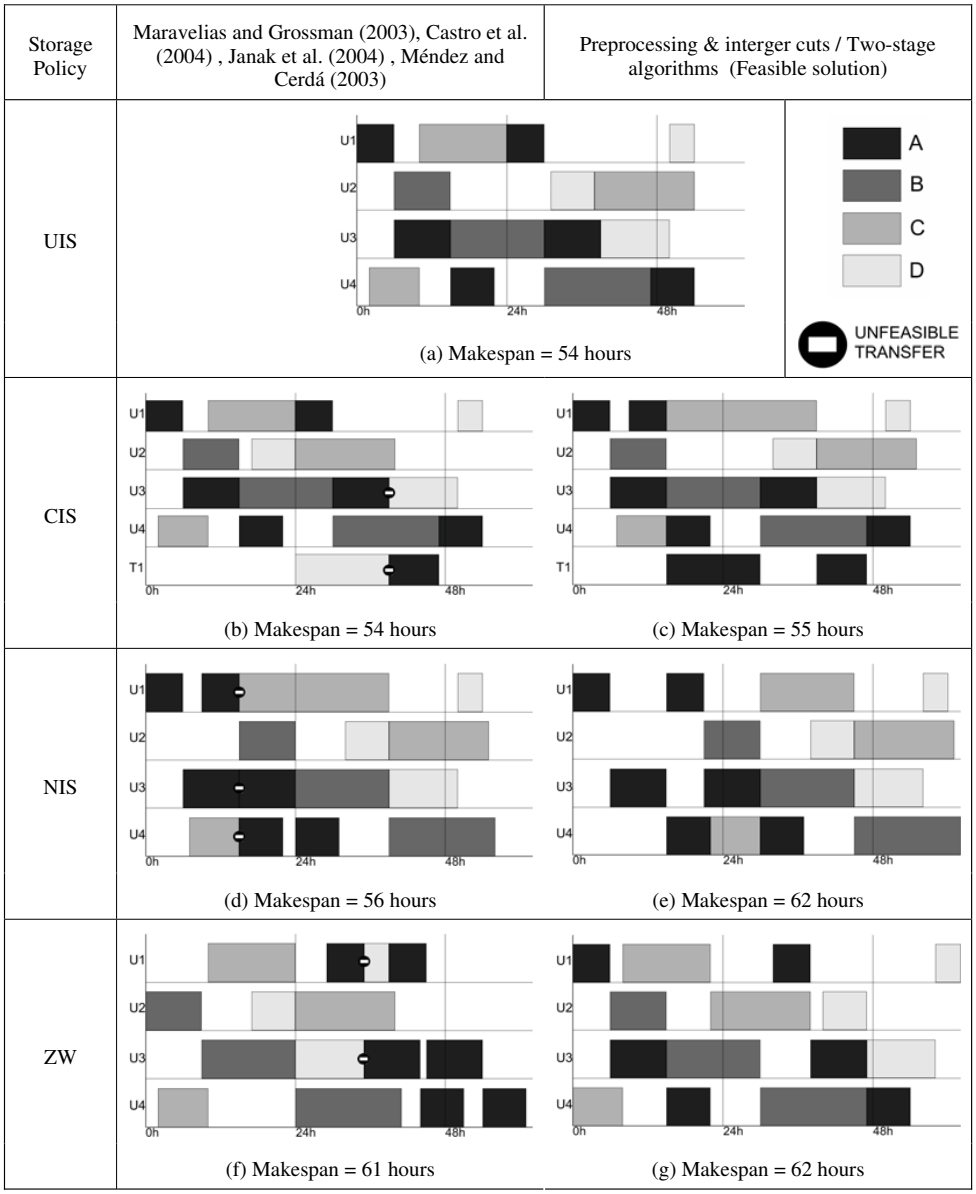

Figure 5.13: Optimal schedules for case study 1.

5.6.2 Case study 2

The scheduling problem presented in case study 2 was originally proposed by Kim et al. (2000), and later solved by Méndez and Cerdá (2003a). This

multipurpose batch plant comprises four products which have to sequentially undergo several processing stages (figure 5.14). One single batch of each product is assumed to be manufactured. In the original definition of the problem transfer times were neglected compared to the much longer processing times. The production sequences and processing times for the recipe of each product are stated in table 5.2. Although in the problem solved by Kim et al. (2000), a single intermediate storage tank was available for receiving material only from unit U3, in this work we have included alternative scenarios to the same problem in order to evaluate the performance of the different models depending on the adopted intermediate storage policy. The alternatives contemplated are unlimited intermediate storage (UIS), non intermediate storage (NIS), common intermediate storage tank (CIS), one common intermediate storage only available after unit U3 (CIS-Kim) and zero-wait time (ZW).

Table 5.2: Production data for case study 2.

	Stage 1		Stage 2		Stage 3		Stage 4		
Prods.	Unit	Time,h	Unit	Time,h	Unit	Time,h	Unit	Time,h	Batches
A	U1	15	U3	8	U4	12			1
B	U1	10	U2	20	U3	5	U4	13	1
C	U3	9	U2	7	U1	20			1
D	U4	5	U3	17	U2	7			1

Figure 5.15 presents the comparison between the results obtained after applying the four MILP formulations and the results using any of the two approaches proposed in this work. MILP formulations obtain lower makespan value because, as expected, unfeasible transfer operations within these solutions are encountered.

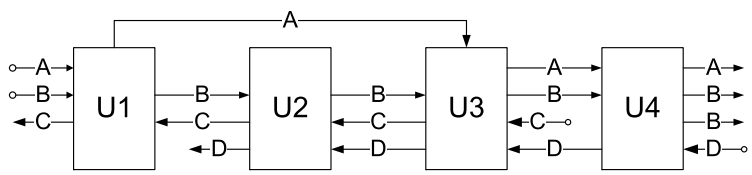

Figure 5.14: Product recipes for case study 2.

Only in the case of UIS policy, MILP formulations lead to an optimum feasible schedule (figure 5.15a). For the case in which one tank can be shared

by all units (CIS), two unfeasible sequences appear in the Gantt chart of the solution shown in figure 5.15b. The first unfeasible transfer takes place at time 16 h with two products involved, products C and B, which are simultaneously transferred from unit U2 to the tank T1 and vice versa. At time 26 h, a second unfeasible transfer occurs with three products involved. Product A is simultaneously transferred (from unit U1 to unit U3) with product C (from tank T1 to unit U1) and product D (from unit U3 to tank T1). Figure 5.15c shows the feasible solution obtained by applying either of the two approaches proposed in this work. For the case described by Kim et al. (2000) with one tank only available after unit U3, an unfeasible situation appears involving three transfers of products at time 30 hours (figure 5.15d). The difference between the values of the makespan of this unfeasible solution (60 hours) and the feasible one shown in figure 5.15e (71 hours) is almost of 20%. A similar situation corresponds to the NIS policy, were unfeasible sequences take place at times 23 h, 25 h and 45 h (figure 5.15f). For this case study, this configuration presents the greatest discrepancy between the makespan values of the unfeasible and the feasible solutions, that is, 24 hours. Finally, two unfeasible sequences arise when adopting a ZW policy. Products B and C are transferred simultaneously between units U1 and U2 at time 16 hours while products B and D are transferred between units U3 and U2 at time 36 hours (figure 5.15h).

5.7 Conclusions

Although transfer times may represent a very small percentage of time regarding the whole duration of tasks in the batch processing, loading and unloading play a crucial role in the synchronization of tasks. Most of the mathematical formulations available in the literature neglect their importance lumping transfer times into the processing times or just assuming them as zero. These formulations usually focus on ensuring that the material balances are feasible between consecutive stages. Therefore, by omitting the reality of transfer times and their corresponding effect on the task synchronization, optimal but actually unfeasible solutions may be reached. To avoid this situation, two alternative solution approaches have been proposed for the general precedence formulation. The first approach consists of identifying those groups of sequenced tasks that can lead to unfeasible sequences, and then adding the corresponding integer-cuts to the mathematical model, so that the unfeasible sequences are avoided. A second approach consists of a two-stage algorithm, which first assigns a small value to the transfer times, in order to force a proper

synchronization, and secondly solves the problem fixing the previous schedule, but assigning zero transfer-times, and assessing the new starting and finish times of tasks. Finally, the appearance of unfeasible solutions has been proved along a series of case studies accounting for different intermediate storage policies. These unfeasible solution schedules have been compared to the feasible ones obtained by using either of the two solution approaches proposed in this chapter.

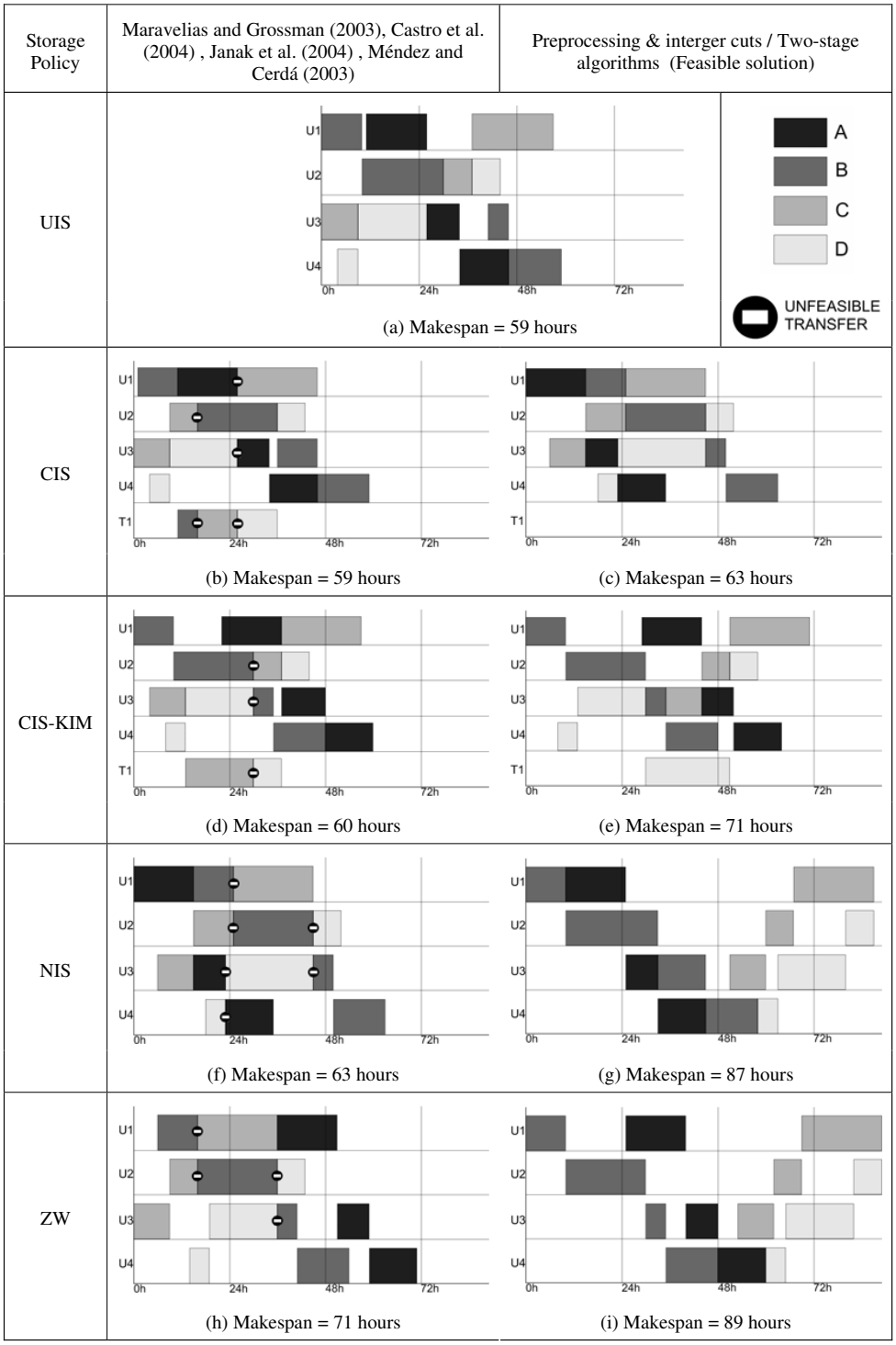

Figure 5.15: Optimal schedules for case study 2.

Chapter 6

Rescheduling using a flexible recipe framework

The next two chapters (6 and 7) take advantage of the flexibility of batch processes to address the management of scheduling under uncertainty. Both chapters demonstrate how the performance of these processes can be enhanced through flexible management of the production recipes.

In particular, this chapter is aimed at developing a reactive strategy that is implemented when an uncertainty is actually unveiled. Dynamic industrial environments often need not only the cyclic revision of scheduling decisions, but also an efficient adjustment of the production recipe to the current process conditions. Therefore, the concept of flexible recipes becomes an important part of the rescheduling framework that allows greater exploitation of the process flexibility in batch plants.

This chapter introduces a rigorous mathematical approach that uses recipe flexibility to address plant-wide batch rescheduling operations. The proposed MILP-based approach is able to address the rescheduling problem of multi-stage multipurpose batch plants involving different storage policies, non-zero transfer times and flexible batch production recipes. The model relies on the concept of general precedence which reduces the number of binary variables and therefore the computational effort. Flexible recipe constraints are incorporated in this model that account for the possibility of changing the processing

time of some tasks by adjusting the parameters of a recipe. The cost of modifying process variables from pre-determined optimal economic conditions is taken into account to represent how productivity is increased in rescheduling situations despite the cost of altering the nominal plant conditions.

Different incidences such as insertion of new orders, equipment failures, due-date changes, maintenance tasks, delay in arrivals, variations in the cost and quality of the raw materials or products taking place throughout the scheduling horizon are considered to evaluate the effectiveness of this approach.

6.1 Introduction

In manufacturing environments, scheduling and rescheduling activities acquire a special importance due to the dynamic nature of highly competitive and continuously changing industrial scenarios. The aim of this chapter is to introduce an optimization tool for the reactive scheduling of multipurpose batch plants. This framework incorporates the concept of recipe flexibility as an additional rescheduling action to fully exploit the inherent flexibility of batch processes. To address this problem, a mixed-integer linear programming (MILP) model is proposed.

This chapter is organized as follows. First, the flexible recipe concept is introduced and discussed in section 6.2. The general problem definition and the rescheduling strategy are next outlined. Section 6.4 describes the mathematical model along with the major assumptions considered. The effectiveness of the proposed approach as a decision-making tool is demonstrated through its application to several scenarios of a scheduling case study presented in section 6.5. Finally, the main contributions of this chapter are summarized in section 6.6.

6.2 Recipe flexibility

The flexible recipe concept was originally introduced by Rijnsdorp (1991) as a set of recipe items that controls the process output and can be modified to adapt to deviations from the nominal conditions.

Theoretically, the nominal recipe for a given product represents the optimal compromise between quality and costs for a given batch regardless of the production environment where this batch will be produced. However, in reality

recipes should be changed or modified to some extent according to the specific features of the production scenario. For instance, in the case of a catalytic process, deviations in the feed quality could be compensated by means of an increase in the amount of catalyst in order to obtain a constant output. Alternatively, in some reactions, an increase of the heating rate might be useful when the production needs to be accelerated by shortening the reaction times. A simple example can be found in the bakery industry in which the temperature of the ovens can be modified to control the required residence time of the bread inside. Currently in industry, recipes are adapted in practice, but usually in a rather unsystematic way that relies on the experience and intuition of the operators.

Verwater-Lukszo (1998) developed this basic idea, that recipes can be modified, and extended the concept to begin to develop a systematic way of adjusting control variables during the execution of production tasks. This methodology was applied to several experiments where small modifications in the nominal conditions were considered. The aim was to observe the process performance under different operating conditions. Since then, several authors have applied the same approach to optimize other chemical processes (Sel et al., 1999; Rutten and Bertrand, 1998). However, most of the scheduling approaches assume that batch processes are operated at nominal conditions following predefined fixed production recipes. Such ideal conditions are very rare in practice and chemical plants often operate under conditions quite different from those considered at the design stage. This traditional way of operation, called in this chapter fixed recipe operation, does not permit adjustments in plant resources availability and variations either in the quality of raw materials or in the actual process conditions.

Romero et al. (2003) carried out one of the first attempts to extend the flexible recipe approach to a plant-wide scheduling problem. This work proposed to optimize the production scheduling of a batch plant where the recipes had some kind of flexibility. These authors integrated a linear flexible recipe model into a multipurpose batch process scheduling formulation based on graph theory (S-graph). In their work, this flexible recipe model allowed the integration of a recipe optimization procedure at the control level and an overall batch plant optimization strategy.

6.3 Problem definition

Typical scheduling methods generate a priori production schedules assuming known, stationary operating conditions and demand along the entire time horizon. However, uncertain production environments force scheduling systems to frequently revise and update the schedule while the process is in progress. This mechanism may be executed periodically (daily or hourly) or under extraordinary conditions due to the occurrence of unforeseen events or external factors.

The main goal here is to optimize the schedule in progress by creating the modified production schedule that most accurately accounts for the current state of the plant. For practical reasons, every time that a rescheduling is carried out, the system should generate updated schedules efficiently and relatively quickly. This feature is highly desirable in industrial practice because shorter reaction times saves costs. In addition, rescheduling solutions with minimal changes to the original schedule are desired to maintain smooth plant operation. Therefore, rather than re-optimizing all the tasks, making only local and partial changes is allowed to reduce the impact upon the schedule in progress. As a result, full-scale rescheduling is avoided in most cases.

This work schematically classifies different types of tasks at the point of rescheduling in order to clearly identify the local actions that will be permitted for each type of batch task. These different sets are as follows:

a. Executed tasks (T^{exec}) are already processed at the rescheduling point and are not included in the formulation since they are past events with no influence on the remaining schedule.

b. Non-directly affected tasks (T^{nda}) which can be either processed when originally were scheduled or delayed.

c. Directly affected tasks (T^{da}) include batch tasks running at the rescheduling point which are rejected due to the event and need to be transferred to an alternative unit in order to be re-processed. This group also includes successive stages of these rejected tasks in the processing sequence of the product.

d. New tasks (T^{new}) associated with late order arrivals that need to be inserted into the current schedule.

Alternatively, processing tasks have also been classified according to the type of recipe as:

e. Fixed recipe tasks (T^{fix}) which can only be performed at nominal operating conditions.

f. Flexible recipe tasks (T^{flex}) which can tolerate the modification to the recipe parameters. Flexibility in the recipe is represented in this work as a mathematical model which relates the process outputs with the recipe items. Batch tasks related to flexible stages may vary their processing times or other recipe parameters in order to optimize the global performance of the plant.

All these different tasks are depicted in the symbolic Gantt chart shown in figure 6.1.

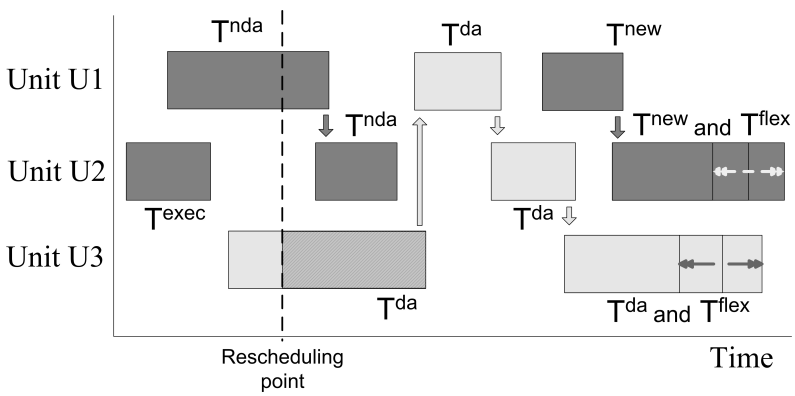

Figure 6.1: Basic representation of task types

This figure illustrates a production scenario in which several orders of two different products (dark grey and light grey) are manufactured. At the rescheduling point, due to the occurrence of an unforeseen event, the first stage of the light grey product has to wait to be transferred and reprocessed in an alternative unit ($U1$). In addition, a late rush order of dark grey product has to be manufactured.

The partial rescheduling actions allowed in each group of tasks are summarized in table 6.1. Directly affected tasks (T^{da}) may require performing re-allocation, re-sequencing and/or re-timing actions until the originally allocated processing unit is again available. Moreover, batch tasks related to new

orders (T^{new}) also need to be simultaneously inserted into the updated schedule. Tasks non-directly affected (T^{nda}) by the incidence are allowed to modify their starting and finishing times (re-timing) or change their position in the unit processing sequence (re-sequencing). In this case, reallocation actions are not allowed for this group of tasks. Tasks following a flexible recipe (T^{flex}) are allowed to modify the current product recipe around a predefined flexibility region as an additional rescheduling action. In fact, the main difference between tasks following fixed (T^{fix}) and flexible (T^{flex}) recipes is that the flexibility region in fixed recipes does not exist or is too narrow to allow changes.

Table 6.1: Allowed rescheduling actions (X: No ; ✓: Yes)

Task type	(Re)Alloc.	(Re)Seq.	(Re)Timing	Recipe adjustment (only T^{flex})
T^{exec}	X	X	X	X
T^{nda}	X	✓	✓	✓
T^{da}	✓	✓	✓	✓
T^{new}	✓	✓	✓	✓

In contrast to the classic strategy that optimizes individual unit operation conditions, here, apparently sub-optimal operating conditions may be implemented in the individual operations to improve the global plant performance because a fixed nominal recipe neglects its production environment. The idea of this flexible recipe framework is to discard the use of traditional fixed recipes which specifies nominal operating conditions and are optimized only for particular situations. Therefore, the approach presented in this chapter is aimed to achieve better performance in the global plant-wide operation in spite of working under non-nominal conditions of individual unit operations.

A detailed description of the mathematical formulation used for this novel reactive scheduling framework using flexible recipes is presented in the next section.

6.4 Mathematical formulation

The underlying structure of the proposed mathematical model relies on both a continuous-time representation and the notion of general precedence. This model considers the following general assumptions:

- Model parameters all are deterministic.
- Batch splitting or mixing is not allowed.
- Allowable rescheduling actions are predefined for each group of tasks as reported in table 6.1.
- No resource constraints except equipment are considered.
- Linear flexible recipe models.
- Predefined flexibility region around nominal operating conditions.

The model is composed of the following constraints:

Processing unit allocation constraints. Constraint 6.1 enforces the assignment of a single processing unit $u \in U_{ps}$ to every task (p, i, s).

$$\sum_{u \in U_{ps}} Y_{pisu} = 1 \qquad \forall (p, i, s) \in (T^{new} \cup T^{da}) \tag{6.1}$$

This constraint allocates a processing unit u to: a) tasks of new orders (T^{new}) that need to be inserted into the current schedule, and b) directly affected tasks (T^{da}) that need to be reallocated to another unit after the occurrence of an unforeseen event. Directly affected tasks which were being processed at the rescheduling point need to be transferred to an alternative unit in order to be reprocessed.

It should be noted the generality of this formulation, which can be used not only for rescheduling purposes, but also for solving short-term scheduling arising at the beginning of the time horizon of interest. This can be done by including all tasks into the set of new tasks (T^{new}).

Flexible recipe model. A linear flexible recipe model relates the deviation of process outputs to deviations of the main flexible recipe items. Let δ_{pisf} be the deviation of the recipe item f in a task (p, i, s) following a flexible recipe. The linear model presented in equation 6.2 is valid for a certain region around nominal process conditions. Additional auxiliary models may be required for operating conditions out of this neighborhood.

$$\sum_{f \in FP_{ps}} lfmod_{psf} \delta_{pisf} = 0 \quad \forall (p,i,s) \in (T^{new} \cup T^{da} \cup T^{nda}) \cap T^{flex} \tag{6.2}$$

Recipe flexibility region. The above linear flexible model is valid around the given nominal conditions representing the recipe flexibility region. Constraint 6.3 establishes the maximum negative ($fplb_{psf}$) and positive ($fpub_{psf}$) deviations allowed for each recipe item.

$$fplb_{psf} \leq \delta_{pisf} \leq fpub_{psf} \quad \forall (p,i,s) \in (T^{new} \cup T^{da} \cup T^{nda}) \cap T^{flex}, f \in FP_{ps} \tag{6.3}$$

As shown in these equations, the conceptual notion of flexible recipes is translated into a parametric meaning within this MILP formulation by means of the allowed flexibility region between $fplb_{psf}$ and $fpub_{psf}$. In the case of fixed recipes, $fplb_{psf}$ and $fpub_{psf}$ are zero.

Associated cost to deviations from nominal conditions. Constraint 6.4 computes the deviation cost for every recipe item in each task (p,i,s) following a flexible recipe. Deviations from nominal operating conditions are penalized by a cost factor ($dcost_{psf}$). This constraint is always active because the deviation cost (DC_{pis}) is minimized in the objective function. If the task is performed at the nominal operating conditions, the deviation cost is set to zero.

$$DC_{pis} = \sum_{f \in FP} dcost_{psf} \delta_{pisf} \quad \forall (p,i,s) \in (T^{new} \cup T^{da} \cup T^{nda}) \cap T^{flex} \tag{6.4}$$

Lower bound on the starting time of the tasks. Constraint 6.5 states that a task (p,i,s) can only be processed if both, the corresponding unit u is ready and the task is also prepared to be executed. For the rescheduling problem, the ready time of a unit (ru_u) is the completion time of the non-directly affected tasks (T^{nda}) that were running at the rescheduling point or simply the rescheduling point if that unit was idle. The ready time (ro_{pis}) of a task can also be employed to express when a unit is not available for any

reason. Ready time for new tasks, $(p, i, s) \in T^{new}$, is the arrival time of these late orders.

$$Ts_{pis} \geq \sum_{u \in U_{ps}} \max(ru_u, ro_{pis}) Y_{pisu} \quad \forall (p, i, s) \in (T^{new} \cup T^{da} \cup T^{nda}) \tag{6.5}$$

Starting time Ts_{pis} for running tasks, $(p, i, s) \in T^{nda}$, is not a variable in the rescheduling formulation because it corresponds to a past event.

Duration of tasks. The following constraints 6.6 and 6.7 establish the duration of a task, taking into account its processing, unit setup and transfer time. These constraints are not applicable for the first stage of products. In order to consider the duration of a task, a distinction is made between stages with and without flexible recipes.

a. *For stages with fixed recipe.* These constraints express that the finishing time of a task (p, i, s) can be computed based on: a) its starting time (Ts_{pis}), b) its unit setup time (ust_{pu}), c) its transfer time from the previous stage ($tt_{pu'}$), d) its nominal processing time (npt_{psu}), e) a possible waiting time (Tw_{pis}), and f) its transfer time to the next stage (tt_{pu}). Here, the transfer operation of a rejected batch and its re-processing operation in an alternative unit are also considered.

$$\begin{aligned} Tf_{pis} = Ts_{pis} + Tw_{pis} + \sum_{u' \in U_{ps'}} tt_{pu'} Y_{pis'u'} + \\ + \sum_{u \in U_{ps}} (npt_{psu} + ust_{pu} + tt_{pu}) Y_{pisu} \\ \forall (p, i, s) \in (T^{new} \cup T^{da} \cup T^{nda}) \cap T^{fix} : s = s' + 1, s \neq \{s^f_{pi}\} \end{aligned} \tag{6.6}$$

b. *For stages with flexible recipe.* These constraints are defined for tasks (p, i, s) following a flexible recipe. They take into account that the predefined optimal processing time may be modified by a deviation ($\delta_{pisDTOP}$) of the nominal recipe processing time (npt_{psu}).

$$\begin{aligned} Tf_{pis} = Ts_{pis} + Tw_{pis} + \sum_{u' \in U_{ps'}} tt_{pu'} Y_{pis'u'} + \\ + \sum_{u \in U_{ps}} (npt_{psu} + \delta_{pisDTOP} + ust_{pu} + tt_{pu}) Y_{pisu} \\ \forall (p, i, s) \in (T^{new} \cup T^{da} \cup T^{nda}) \cap T^{fix} : s = s' + 1, s \neq \{s^f_{pi}\} \end{aligned} \tag{6.7}$$

Duration of the first tasks. Specific constraints (6.9 and 6.9) are necessary for the first processing stage in order to account for the initial transfer time (itt_{pu}).

a. *For stages with fixed recipe.*

$$\begin{aligned} &Tf_{pis} = Ts_{pis} + Tw_{pis} + \sum_{u \in U_{ps}} (npt_{psu} + ust_{pu} + tt_{pu} + itt_{pu}) Y_{pisu} \\ &\forall (p,i,s) \in (T^{new} \cup T^{da} \cup T^{nda}) : s = \{s^f_{pi}\} \end{aligned} \tag{6.8}$$

b. *For stages with flexible recipe.*

$$\begin{aligned} &Tf_{pis} = Ts_{pis} + Tw_{pis} + \\ &\qquad + \sum_{u \in U_{ps}} (npt_{psu} + \delta_{pis,DTOP} + ust_{pu} + tt_{pu} + itt_{pu}) Y_{pisu} \\ &\forall (p,i,s) \in (T^{new} \cup T^{da} \cup T^{nda}) \cap T^{flex} : s = \{s^f_{pi}\} \end{aligned} \tag{6.9}$$

Sequencing constraints. These constraints are used to sequence pairs of tasks that are allocated to the same unit.

a. *If task (p,i,s) precedes task (p',i',s').* Constraint 6.10 ensures that if task (p,i,s) precedes task (p',i',s'), i.e. $X_{pisp'i's'}$ is 1, and both tasks are processed in the same unit u, i.e. $Y_{pisu} = 1$ and $Y_{p'i's'u} = 1$, task (p',i',s') can not start until task (p,i,s) is finished and the corresponding unit changeover time between products p and p' is completed.

$$\begin{aligned} &Ts_{p'i's'} \geq Tf_{pis} + uch_{pp'u} - M(1 - X_{pisp'i's'}) - M(2 - Y_{pisu} - Y_{p'i's'u}) \\ &\forall (p,i,s),(p',i',s') \in (T^{new} \cup T^{da} \cup T^{nda}), u \in (U_{ps} \cap U_{p's'}) \\ &(p < p') \cup (p = p', s < s') \end{aligned} \tag{6.10}$$

b. *If task (p',i',s') precedes task (p,i,s).* Constraint 6.11 considers the opposite case to constraint 6.10. If task (p,i,s) is performed after task (p',i',s'), $X_{pisp'i's'}$ is 0, and both tasks are processed in the same unit ($Y_{pisu} = 1$ and $Y_{p'i's'u} = 1$), task (p,i,s) can not be processed until task (p',i',s') is finished and the changeover time between product p' and p in unit u is completed.

$$\begin{aligned}&Ts_{pis} \geq Tf_{p'i's'} + uch_{p'pu} - M \cdot X_{pisp'i's'} - M(2 - Y_{pisu} - Y_{p'i's'u})\\&\forall (p,i,s),(p',i',s') \in (T^{new} \cup T^{da} \cup T^{nda}), u \in (U_{ps} \cap U_{p's'})\\&(p < p') \cup (p = p', s < s')\end{aligned} \tag{6.11}$$

c. *Different batches of the same product at the same processing stage.* This constraint 6.12 sequences pairs of batches of the same product at the same processing stage, establishing that batch $i' > i$ is always processed after batch i and its corresponding unit-dependent changeover time. In other words, this constraint arranges batches of the same product aiming at reducing the inherent combinatorial complexity of the problem.

$$Ts_{pi's} \geq Tf_{pis} + \sum_{u \in U_{ps}} uch_{ppu} Y_{pisu} \quad \forall (p,i,i',s) \in (T^{new} \cup T^{da} \cup T^{nda}) : i < i' \tag{6.12}$$

d. *Consecutive processing stages of the same batch of the same product.* This constraint 6.13 synchronizes two consecutive stages, taking in account transfer and unit setup times.

$$\begin{aligned}&Tf_{pis} - \sum_{u \in U_{ps}} tt_{pu} Y_{pisu} = Ts_{pis'} + \sum_{u' \in U_{ps'}} ust_{pu'} Y_{pis'u'}\\&\forall (p,i,s) \in (T^{new} \cup T^{da} \cup T^{nda}) : s' = s + 1, s \neq \{s^l_{pi}\}\end{aligned} \tag{6.13}$$

Tardiness and earliness. Tardiness (constraint 6.14) is defined here as the delay in the fulfillment of the order i of product p with respect to its due date (dd_{pi}).

$$TA_{pi} \geq Tf_{pis} - dd_{pi} \quad \forall (p,i,s) \in (T^{new} \cup T^{da} \cup T^{nda}) : s = \{s^l_{pi}\} \tag{6.14}$$

Alternatively, earliness (constraint 6.15) is the amount of time after the completion of order (p,i) but before its due date.

$$EA_{pi} \geq dd_{pi} - Tf_{pis} \quad \forall (p,i,s) \in (T^{new} \cup T^{da} \cup T^{nda}) : s = \{s^l_{pi}\} \tag{6.15}$$

Problem objective function. The objective of the proposed formulation is to minimize a weighted cost function composed of earliness and tardiness penalty costs as well as the deviation from nominal recipe costs. The level of importance of tardiness and earliness penalties is estimated by two cost factors *tcost* and *ecost*, respectively. In this way, orders are forced to be processed near their respective due dates, i.e. in a just-in-time manner, minimizing inventory cost, while taking into account allowable recipe changes. Keeping inventory at a minimum level reduces cost, spoilage (i.e. chemicals with low stability over time) and investment in storage tanks. This objective function can be represented by the following expression (equation 6.16):

$$\min \quad TCOST = \sum_{p \in P} \sum_{i \in I_p} (tcost \cdot TA_{pi} + ecost \cdot EA_{pi}) + \sum_{(p,i,s) \in T^{flex}} DC_{pis} \tag{6.16}$$

Moreover, other performance criteria can also be incorporated, i.e. makespan, maximum throughput or minimum number of tardy orders, together with the cost associated with recipe modifications.

6.5 Case study and Results

The proposed rescheduling strategy under a flexible recipe framework will be illustrated by solving a modified version of the case study proposed by Romero et al. (2003). In this example, five products are manufactured in four processing stages with available alternative equipment units. A flexible recipe for the production of benzyl alcohol is introduced within this production scenario for the second stage of product $P1$ at unit $U2$. Transfer times equal to 5% of the processing time of each processing task have been also considered. For each product, a set of production orders comprising a single batch of each product with a specific due date is defined. Nominal batch processing times, available processing units, order due dates and penalty factors for tardy orders are tabulated in table 6.2.

The crossed-Cannizaro reaction for the batch-wise production of benzyl alcohol from the reduction of benzaldehyde has been studied by Keesman (1993). This author proposed a quadratic model to predict the variation of the yield of the reaction (δ_{DPS}) for a priori known deviations in the process inputs: temperature (δ_{DTEMP}), processing time δ_{DTOP}, amount of catalyst (δ_{DKOH}) and amount of the excess reactant formaldehyde (δ_{DFOR}). Equation 6.17 shows a

Table 6.2: Process data for the case study.

Stage	Product P1		Product P2		Product P3		Product P4		Product P5	
	Unit	npt, h	Unit	npt, h	Unit	npt, h	Unit	npt, h	Unit	npt, h
S1	U1	0.5	U1	1	U7	2	U2	1.5	U8	2
	(Flexible stage)		U2	0.75	U9	2.5	U8	2		
			U8	2						
S2	U2	1.75	U3	2	U3	2	U3	1	U5	1
	(Flexible stage)		U9	2	U4	1	U9	2.5		
S3	U3	2	U4	1.5	U6	1	U7	2	U2	1.25
	U9	1.75	U7	2			U4	2		
S4	U4	0.5	U6	1	U5	1	U5	1.5	U1	1
	U7	0.75								
Due dates, h										
Order I1	10		10		9		10		15	
Order I2	10		20*		9		22*		15	
Order I3	10								15	
Order I4	15								20*	
Order I5	19*									
Order I6	19*							(*) New orders		
pto, m.u	5		6		7		2		5	

linearization of this model proposed by Romero et al. (2003), which has been incorporated into the proposed MILP rescheduling formulation. This linear model has been used in order to illustrate and validate the potential use of flexible recipe models, therefore, non-linear models should be taken into account as a future work to obtain more accurate solutions.

$$\begin{aligned}\delta_{P1,i,S2,DPS} = {} & 4.4\delta_{P1,i,S2,DTEMP} + 4\delta_{P1,i,S2,DTOP} + 95\delta_{P1,i,S2,DKOH} \\ & +95\delta_{P1,i,S2,DFOR} \\ \forall i \in I_{P1} & \end{aligned} \tag{6.17}$$

Table 6.3 shows the flexible recipe variables and the maximum negative and positive deviations allowable for the flexible items around the nominal operating conditions.

Additional flexibility has been considered in the first stage of product $P1$ at

Table 6.3: Recipe parameters, flexibility region and cost for deviation from nominal conditions.

Recipe item		Flexibility Region		
		$fplb_{psf}$	$fpub_{psf}$	$dcost_{psf}$
δ_{DPS}	reaction yield	0	0	-
δ_{DTEMP}	reaction temperature	-0.7 °C	0.5 °C	3 m.u./°C
δ_{DTOP}	reaction duration	-0.3 h	0.1 h	2 m.u./h
δ_{DKOH}	amount of KOH	-27 g	8.5 g	5 m.u./g
δ_{DFOR}	amount of Formaldehyde	-30 g	7.5 g	4 m.u./g

unit $U1$ (equation 6.18). This is a pre-heating stage in which the temperature reached is directly proportional to the duration of the task. The final temperature achieved by this task corresponds to the temperature for the reaction in next processing stage, as shown in constraint 6.19.

$$\delta_{P1,i,S1,DTEMP} = 10\delta_{P1,i,S1,DTOP} \qquad \forall i \in I_{P1} \tag{6.18}$$

$$\delta_{P1,i,S2,DTEMP} = \delta_{P1,i,S1,DPS} \qquad \forall i \in I_{P1} \tag{6.19}$$

Three different scenarios are addressed to demonstrate the applicability of the proposed approach, which is implemented within the modeling language GAMS using CPLEX version 7.5.

6.5.1 Scenario 1

Figure 6.2 shows the original schedule in progress for this scenario which has to be updated at time 3h in order to face the unexpected breakdown of unit $U7$ with requires a repair time of 17 hours. In addition, new batches corresponding to the arrival of late orders of products $P2$, $P4$ and $P1$ must be also inserted in the on-going schedule (marked with stars in table 6.2). Therefore, two unexpected events; a unit break-drown and late order arrivals, occur in the plant simultaneously.

The selected objective function to be minimized is the total value cost associated to the order tardiness ($tcost$ = 5 m.u./h) and earliness ($ecost$ = 1 m.u./h), as well as the corresponding cost for manipulating the process conditions shown in table 6.3.

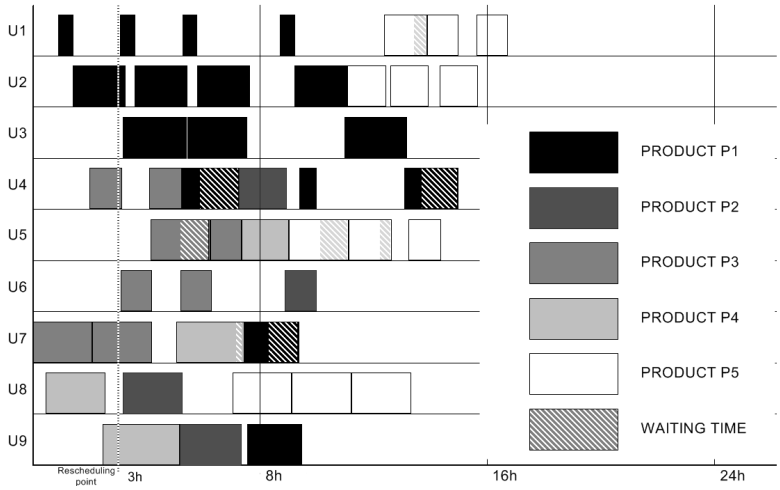

Figure 6.2: Schedule in progress at the rescheduling point for scenario 1.

The proposed formulation for this scenario minimizes 6.16 subject to constraints 6.1 and 6.15.

Figure 6.3 shows a new production plan obtained by using a conventional rescheduling strategy that does not consider recipe flexibility. This approach allows limited re-sequencing, re-allocation and re-timing of tasks as shown in table 6.1. Figure 6.4 depicts a rescheduled production plan that, in addition classic rescheduling actions, has modified the production recipe of some products. Despite the recipe modification cost, this production plan results in a better solution in terms of the proposed objective function.

This improvement, a 12% reduction of the objective function value, comes not only from the recipe changes but also from several modifications of sequencing decisions, which can be easily observed by comparing figures 6.3 and 6.4. Although the reduction of processing times allowed by flexible recipes involves an additional cost, it produces new opportunities and gaps permitting re-allocating or re-sequencing tasks that would be otherwise disallowed. This solution takes advantage of additional opportunities to reduce tardiness and earliness and as a result allows obtaining better objective function values. This situation is illustrated by figure 6.4, in which, despite the increase of the makespan, completion times are tightened to their due-dates by a set of starting time adjustments not allowed in the rigid schedule in figure 6.3. For example, the first batch of product $P5$ has been rescheduled and brought

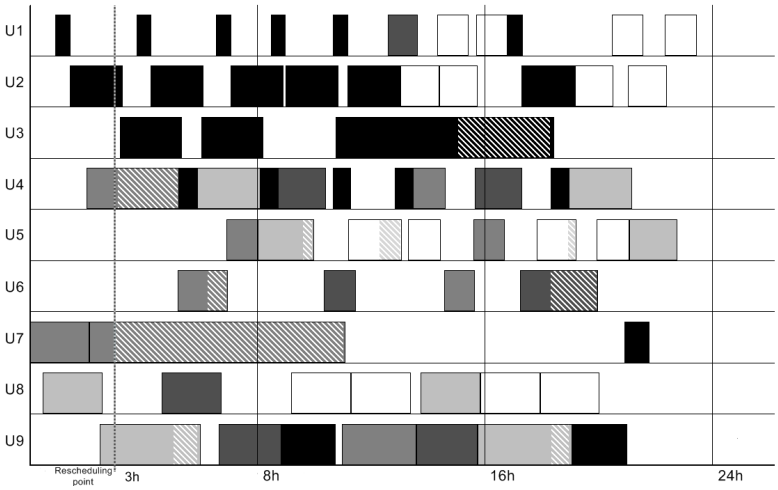

Figure 6.3: Optimal rescheduling without considering recipe flexibility for scenario 1 (TCOST = 139.74 m.u.).

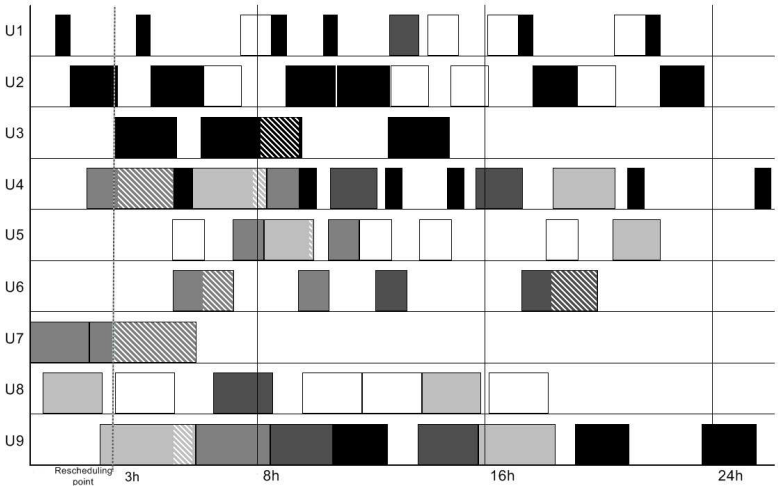

Figure 6.4: Optimal rescheduling considering recipe flexibility for scenario 1 (TCOST = 129.93 m.u.).

forward to start just after the rescheduling point.

In this scenario, the proposed recipe modifications are shown in table 6.4.

The second stage of the first, third, fifth and sixth batch of product $P1$ (flexible reaction tasks) have reduced processing times at the expense of increasing the amount of formaldehyde. However, flexibility is not exploited in any of the batches of $P1$ in the pre-heating stage (first stage). This is because flexibility always has an associated cost and since this stage does not cause a bottleneck in the process, no modification in this stage will improve the performance of the plant.

Table 6.4: Recipe modifications as rescheduling actions for scenario 1.

Product, batch, stage	δ_{DTEMP}, °C	δ_{DTOP}, h	δ_{DKOH}, g	δ_{DFOR}, g
P1, I1, S2	0	-0.175	0	0.007
P1, I3, S2	0	-0.125	0	0.005
P1, (I5-I6), S2	0	-0.3	0	0.013

This example clearly reflects the great importance of recipe flexibility in the rescheduling process of critical and hard-constrained batch operations. It is worth noting that the computational effort for updating the current schedule remains very low, being short reaction times highly important in real industrial environments. Table 6.5 summarizes the main features of the schedules generated with and without recipe flexibility.

Table 6.5: Comparison between fixed and flexible rescheduling for scenario 1.

	Fixed recipe (Fig. 6.3)	Flexible recipe (Fig. 6.4)
Tardiness cost, m.u.	131.81	116.44
Earliness cost, m.u.	7.93	11.54
Recipe modification cost, m.u.	0	1.95
Objective function (TCOST), m.u.	139.74	129.93
Binary vars., cont. vars., rows	283, 555, 974	283, 676, 1131
CPU time, s (AMD 2.6 GHz)	78	207

6.5.2 Scenario 2

In this second scenario, rescheduling actions are required to deal with the breakdown of unit $U3$ at time 4 h with a repairing time of 20 hours. Economical factors have changed from the previous scenario and now, the same cost (3

m.u./g) is assumed for deviations in KOH and formaldehyde. The rest of the deviation costs remain the same as stated in table 6.3.

Furthermore, in this second scenario an alternative objective function is evaluated. Now, the number of tardy orders, i.e. orders ending after their respective due dates, in the schedule is penalized. In order to calculate the number of tardy orders in the schedule, a new binary variable must be defined (TO_{pis}) which denotes whether or not the last stage of the batch i of a product p is completed after the promised due date ($TO_{pis} = 1$) as determined by equation 6.20.

$$TA_{pis} - TO_{pis}M \leq 0 \qquad \forall (p,i,s) \in (T^{new} \cup T^{da} \cup T^{nda}) : s = \{s^l_{pi}\} \quad (6.20)$$

The objective function in equation 6.21 includes the deviation cost from the nominal recipe and a parameter pto which defines the product-dependent penalty cost for tardy orders (see table 6.1).

$$\min \quad TCOST = \sum_{p \in P} \sum_{i \in I_p} \sum_{s = s^l_{pi}} (TO_{pis} \cdot pto_p) + \sum_{(p,i,s) \in T^{flex}} DC_{pis} \qquad (6.21)$$

In this scenario, the objective function (6.21) is then minimized subject to constraints 6.1, 6.15 and 6.20.

Next, figure 6.5 illustrates the original production plan, figure 6.6 the rescheduling solution considering fixed recipes and finally, figure 6.7 the rescheduling proposed solution using flexible recipes.

Similar to the previous scenario, table 6.6 shows the main results for these approaches and the computational effort which has radically decreased due to the changed nature of the objective function. Different assignments and sequencing of tasks are obtained by using flexible recipes, which leads to an improved objective function.

Table 6.7 summarizes the recipe modifications executed as rescheduling actions, i.e. the processing times of the second stage ($S2$) of batches $I4$ and $I5$ of product $P1$ are reduced by increasing the amount of formaldehyde.

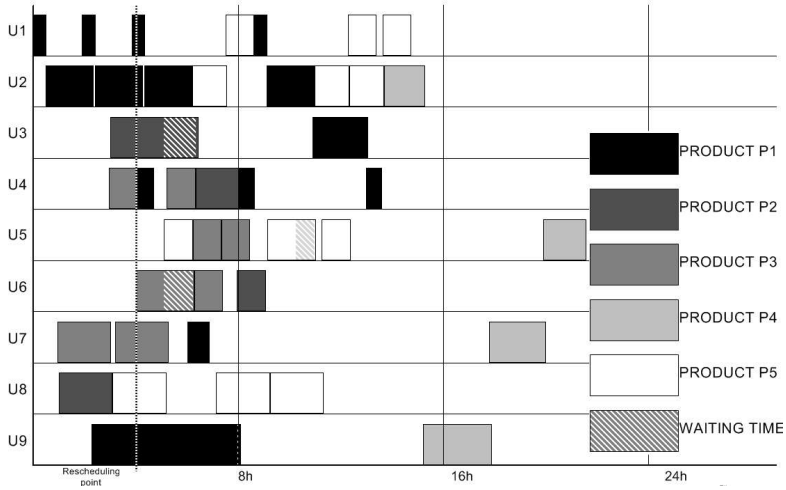

Figure 6.5: Schedule in progress at the rescheduling point for scenario 2.

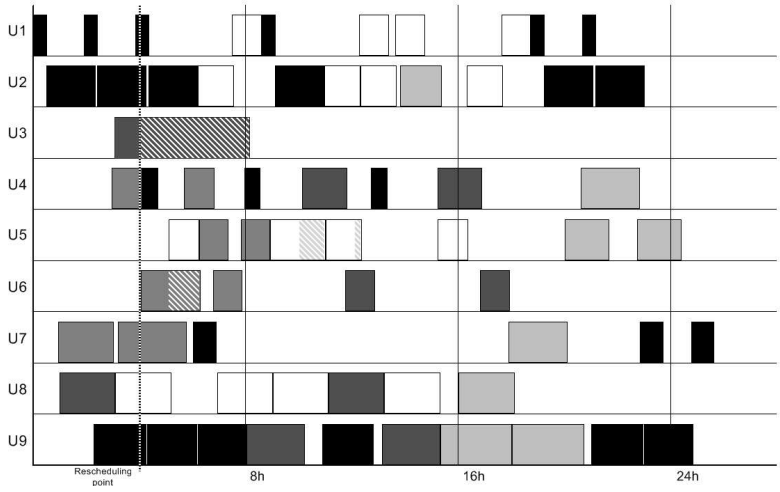

Figure 6.6: Optimal rescheduling without considering recipe flexibility for scenario 2 (TCOST = 20.00 m.u.).

6.5.3 Scenario 3

In this scenario, fortunately no incidence occurs, but a series of maintenance tasks have to be performed in various equipment units. The aim of these

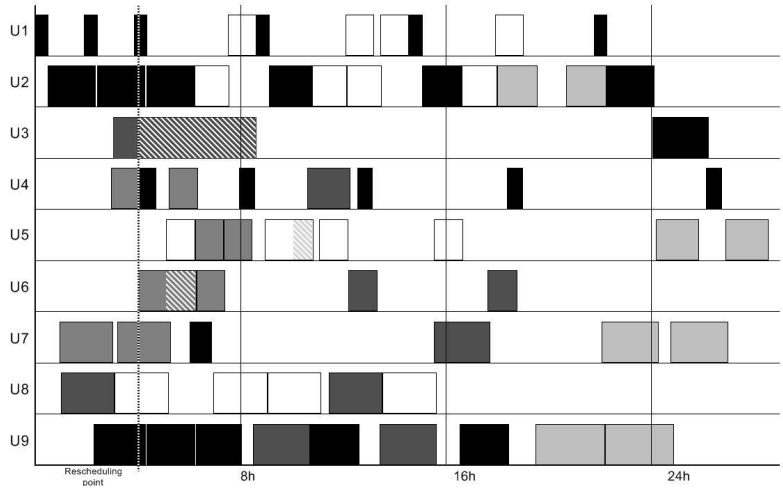

Figure 6.7: Optimal rescheduling considering recipe flexibility for scenario 2 (TCOST = 16.04 m.u.).

Table 6.6: Comparison between fixed and flexible rescheduling for scenario 2.

	Fixed recipe (Fig. 6.6)	Flexible recipe (Fig. 6.7)
Number of tardy orders	5	4
Tardy orders penalty, m.u.	20	15
Recipe modification cost, m.u.	0	1.04
Objective function (TCOST), m.u.	20	16.04
Binary vars., cont. vars., rows	276, 527, 902	276, 626, 1022
CPU time, s (AMD 2.6 GHz)	14.7	4.7

Table 6.7: Recipe modifications as rescheduling actions for scenario 2.

Product, batch, stage	δ_{DTEMP}, °C	δ_{DTOP}, h	δ_{DKOH}, g	δ_{DFOR}, g
P1, I4, S2	0	-0.300	0	0.013
P1, I5, S2	0	0.187	0	0.008

maintenance tasks is precisely to avoid the occurrence of incidences shown in the previous scenario which led to longer unavailability intervals. This scenario proposes a scheduling problem of maintenance tasks that have to be carried

out within a restricted period of time.

In this situation, all the tasks previously scheduled are identified as non-directly affected tasks (T^{nda}), so the rescheduling actions allowed are just re-sequencing and recipe modifications. A new type of task is defined here as a maintenance task (T^{maint}) which is not included in the recipe of any product but must be considered a requirement in the schedule. These maintenance tasks can be included in the model and treated in the same way as the new tasks (T^{new}). Hence, the above mathematical formulation can be adapted to also address maintenance management problem.

Three maintenance tasks of 2 hours are required to be executed in units $U2$, $U3$ and $U7$. Additionally, it is necessary that these tasks are arranged to be completed between hour 1 and hour 12 from the rescheduling point, because there is a technical service that will perform the maintenance work available within this time frame. Uniform sequence-dependent changeover times ($uch_{pp'u}$) of 0.5 h for transition between different products are also taken into account. The objective function to be minimized is the same as the one previously presented in scenario 1.

Figure 6.8 shows the initial on-going schedule that does not consider any maintenance tasks. Figure 6.9 illustrates a production plan which includes the maintenance tasks. Figure 6.10 presents the proposed scheduling accounting for maintenance tasks and taking advantage of recipe modifications as an additional rescheduling action. These figures point out how maintenance tasks are sequenced differently by these two approaches.

Table 6.8 compares rescheduling using fixed and flexible recipes while table 6.9 presents the rescheduling recipe modification actions. The CPU time is shorter than in the previous scenarios since all assignment decisions have already been taken and only sequencing decisions must be made to insert the maintenance tasks.

6.6 Conclusions

This chapter has presented an efficient MILP-based rescheduling framework that incorporates recipe flexibility as an additional opportunity to optimize rescheduling. The approach is based on a continuous-time domain representation and the generalized notion of precedence. This rescheduling strategy considers the on-going production schedule and the current process conditions in order to simultaneously adapt the production recipe to the new scenario

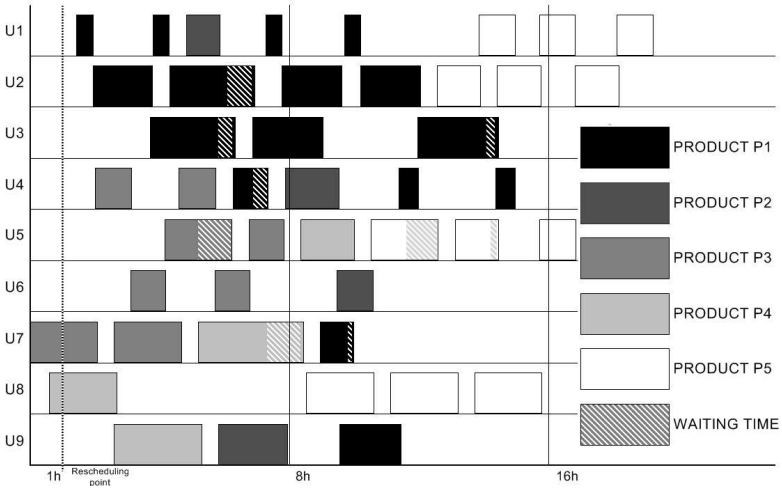

Figure 6.8: Schedule in progress at the rescheduling point for scenario 3.

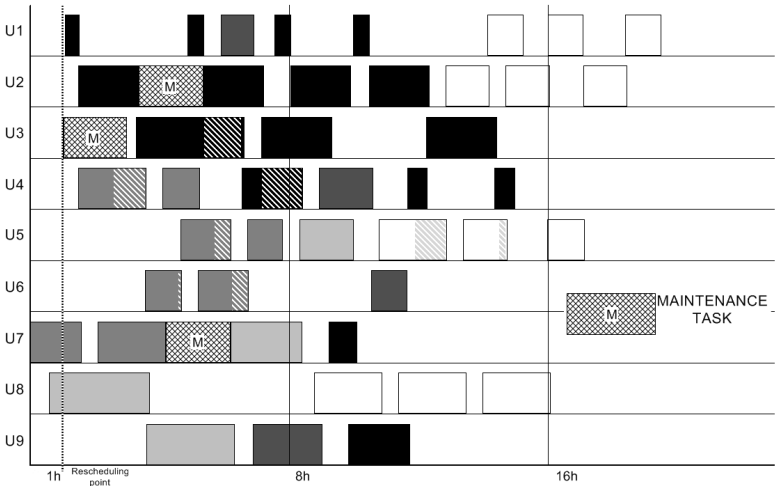

Figure 6.9: Optimal rescheduling without considering recipe flexibility and maintenance operations for scenario 3 (TCOST = 60.08 m.u.).

and re-optimize the schedule with regard to the batches still to be processed. Different objective functions can be employed to regain feasibility or optimality at minimum cost. Efficiency and applicability of the proposed strategy is demonstrated by the successful solution of a complex rescheduling problem in

a multipurpose batch plant with reasonable computational effort.

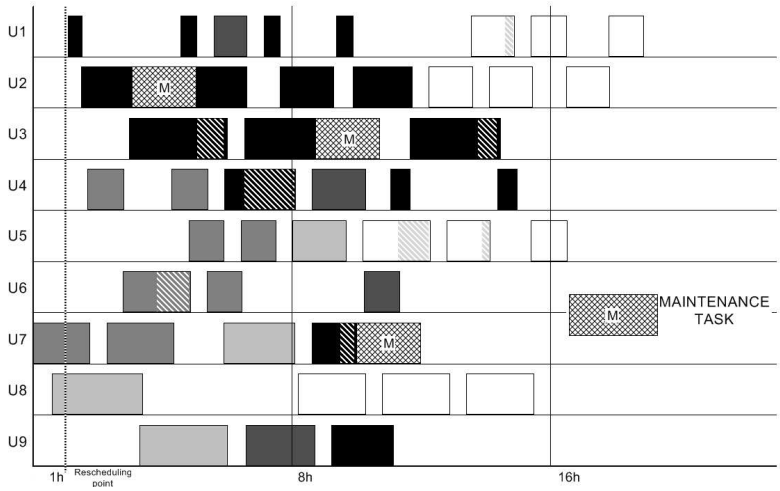

Figure 6.10: Optimal rescheduling considering recipe flexibility and maintenance operations for scenario 3 (TCOST = 50.89 m.u.).

Table 6.8: Comparison between fixed and flexible rescheduling for scenario 3.

	Fixed recipe (Fig. 6.3)	Flexible recipe (Fig. 6.4)
Tardiness cost, m.u.	54.56	42.44
Earliness cost, m.u.	5.51	6.68
Recipe modification cost, m.u.	0	1.65
Objective function (TCOST), m.u.	60.08	50.89
Binary vars., cont. vars., rows	156, 472, 726	156, 555, 974
CPU time, s (AMD 2.6 GHz)	0.3	0.3

Table 6.9: Recipe modifications as rescheduling actions for scenario 3.

Product, batch, stage	δ_{DTEMP}, °C	δ_{DTOP}, h	δ_{DKOH}, g	δ_{DFOR}, g
P1, I1, S2	0	-0.300	0	0.013
P1, I2, S2	0	-0.287	0	0.012
P1, I3, S2	0	-0.175	0	0.007

Chapter 7

Managing risk through a flexible recipe framework

This chapter presents a proactive strategy to deal with the management of uncertainty in batch scheduling. In the previous chapter 6, a reactive scheduling was presented which may be mainly used when the level of disturbances is important or uncertain data are known very late. However, a proactive scheduling can be a more convenient approach when uncertainty is known or some suspicion exits for a future trend in order to construct the production plan.

This chapter proposes a novel approach that exploits the flexibility of the recipes as a better way to handle the risk associated with the proactive scheduling under uncertainty of batch chemical plants. The proposed solution strategy relies on a novel two-stage stochastic mathematical formulation that explicitly includes the trade-off between risk and profit at the decision-making level. Management of risk is addressed by including a control measure (i.e. the profit in the worst scenario), as an additional objective to be considered, thus leading to a multi-objective optimization problem. To overcome the numerical difficulties associated with such mathematical formulation, a decomposition strategy based on the Sample Average Approximation (SAA) is introduced.

The main advantages of this approach are illustrated through a case study, in which a set of solutions appealing to decision makers with different attitudes

toward risk are obtained. The potential benefits of considering the flexibility of the recipes as a way of managing the risk associated with the plant operation under demand uncertainty are highlighted through the comparison with the conventional approach that considers nominal operating conditions. Numerical results corroborate the advantages of exploiting the capabilities of the proposed flexible recipe framework for risk management purposes.

7.1 Introduction

The aim of this work is to balance the trade-off between a high demand satisfaction level, achieved through large stocks, and low inventory costs, which may imply leaving part of the demand unsatisfied. In this context, the flexible recipe framework is used to increase the production rate and thus the final inventories. This is done through variations in the parameters of the flexible stages of the recipe.

To address this problem, a large scale multi-objective stochastic MILP model is presented. This model is based on the general precedence model and explicitly includes the trade-off between risk and profit. The outcome of such formulation consists of a set of Pareto-optimal solutions from which the decision-maker may choose the best one according to his/her preferences. Thus, this approach exploits the concept of flexible recipe within a risk management framework. In the resulting strategy, the detailed decisions associated with the control recipe are calculated in conjunction with those dealing with the scheduling tasks. This integrated way of operation increases the plant flexibility and its capability of giving a quick response to the market changes.

This chapter is organized as follows. First, the problem definition is outlined together with the assumptions considered, in section 7.2. Then, the proposed stochastic formulation accounting for the maximization of the expected profit is given in section 7.3. The inclusion of an explicit control measure to manage risk is addressed in section 7.4. The effectiveness of the proposed approach as a decision-making tool is next illustrated through its application to a scheduling case study in section 7.5. Finally, some concluding remarks are given at the end of the chapter.

7.2 Problem statement

Several products that are given to be manufactured on a multistage multipurpose batch plant within a scheduling time horizon. The following data are assumed to be known in advance:

- Set of raw materials, intermediate and final products to be manufactured, and their costs.
- Set of production recipes with fixed batch sizes.
- Set of plant facilities, i.e. number of equipment units, their capacities and suitabilities for the labor tasks.
- Cost functions associated with raw materials and utilities consumption, holding inventory over a horizon and penalties for unsatisfied demand.

Sales of products are assumed to be executed at the end of the time horizon. The demand associated with each product cannot be perfectly forecasted and its uncertainty is represented by a set of scenarios with given probability of occurrence. Decisions regarding scheduling tasks are made prior to the realization of the uncertain parameter. Therefore, while the scheduling decisions (number of tasks to be performed, batch sizes, assignment and sequencing decisions) must be taken at the beginning of the time horizon, that is to say, prior to the demand realization, sales are computed once the random events take place at the end of it.

Therefore, the problem contemplated involves two types of decisions. Those at the first stage, i.e. scheduling decisions and purchases of raw materials, must be calculated bearing in mind the demand uncertainty. Those at the second stage, i.e. sales of final products, are taken in order to react against the uncertainty, i.e. demand realization.

The stochastic problem consists of finding the scheduling decisions that maximize the total expected profit, which is computed over a set of demand scenarios. The profit term is calculated from the sales revenues, operating costs, inventory costs and unsatisfied demand costs. This overall two-stage stochastic framework strategy is illustrated in figure 7.1.

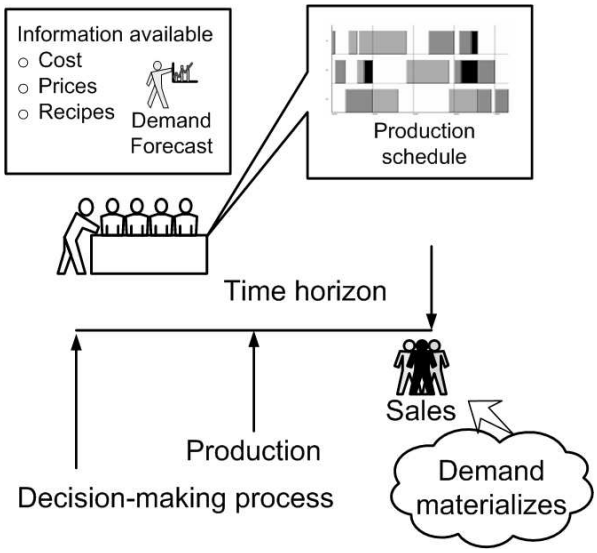

Figure 7.1: Two-stage stochastic framework.

7.3 Mathematical formulation

The MILP formulation is next described in detail. Constraints 7.1 and 7.2 describe the flexible recipe model and its valid region, respectively. Constraint 7.3 enforces the assignment of tasks to units.

$$\sum_{f \in FP_{ps}} lfmod_{psf} \delta_{pisf} = 0 \qquad \forall p \in P, i \in I_P, s \in S_P, (p,s) \in FL_{ps} \tag{7.1}$$

$$fplb_{psf} \leq \delta_{pisf} \leq fpub_{psf} \quad \forall p \in P, i \in I_P, s \in S_P, (p,s) \in FL_{ps}, f \in FP_{ps} \tag{7.2}$$

$$\sum_{u \in U_{ps}} Y_{pisu} \leq 1 \qquad \forall p \in P, i \in I_p, s \in S_p \tag{7.3}$$

Equation 7.4 forces the condition for which the non-produced batches are located at the end of the schedule, i.e. it ensures that the decision to manufacture a batch of a product is only taken if its preceding batch is also produced.

Although this constraint is not formally needed, it helps computations. Indeed, by fixing the position of the non-produced batches, it produces smaller branch-and-bound trees and shorter computational times. The specific position of these non-produced batches is absolutely meaningless and arbitrary.

$$\sum_{u \in U_{ps}} Y_{pisu} \geq \sum_{u' \in U_{ps'}} Y_{pi's'u'} \quad \forall p \in P, i, i' \in I_P, s, s' \in S_P : i' > i \tag{7.4}$$

Constraint 7.4 states that every production task must be completed within a specified scheduling horizon of length H. Constraint 7.6 establish the duration of a task taking into account the processing times of the recipes. Constraints 7.7 and 7.8 sequence pairs of task allocated to the same unit. Constraint 7.9 sequences pairs of batches of the same product, provided that both batches are manufactured.

$$Tf_{pis} \leq H \qquad \forall p \in P, i \in I_P, s \in S_P \tag{7.5}$$

$$Tf_{pis} \geq Ts_{pis} + npt_{ps} + \delta_{pisDTOP} + tt_{ps} \qquad \forall p \in P, i \in I_P, s \in S_P \tag{7.6}$$

$$\begin{aligned} &Ts_{p'i's'} \geq Tf_{pis} - M(1 - X_{pisp'i's'}) - M(2 - Y_{pisu} - Y_{p'i's'u}) \\ &\forall p, p' \in P, i, i' \in I_P, s, s' \in S_P, u \in (U_{ps} \cap U_{p's'}) : (p < p') \cup (p = p', s < s') \end{aligned} \tag{7.7}$$

$$\begin{aligned} &Ts_{pis} \geq Tf_{p'i's'} - M(1 - X_{pisp'i's'}) - M(2 - Y_{pisu} - Y_{p'i's'u}) \\ &\forall p, p' \in P, i, i' \in I_P, s, s' \in S_P, u \in (U_{ps} \cap U_{p's'}) : (p < p') \cup (p = p', s < s') \end{aligned} \tag{7.8}$$

$$Ts_{pi's} \geq Tf_{pis} - M(1 - Y_{pi'su}) \quad \forall p \in P, i, i' \in I_P, s \in S_P, u \in U_{ps} : i < i' \tag{7.9}$$

Constraint 7.10 is defined for every pair of consecutive processing stages that must be sequentially performed for a particular product. This constraint is applied in conjunction with constraint 7.6 in order to describe a

non-intermediate storage policy (NIS). Then, an equipment unit is not free until it finishes its processing and the material is transferred to the equipment unit assigned to the next task. In this type of general precedence models, different storage policies may be easily adopted by changing the signs of the corresponding inequalities.

$$Tf_{pis} - tt_{ps} = Ts_{pis'} \qquad \forall p \in P, i \in I_P, s, s' \in S_P : s' = s + 1 \tag{7.10}$$

Equation 7.11 represents the demand constraints and states that sales can be lower or equal to the demand. This model assumes that some of the demand can be left unsatisfied because of limited production capacity.

$$SALES_{pe} \leq dem_{pe} \qquad \forall p \in P, e \in E \tag{7.11}$$

Equation 7.12 constraints the sales to be lower or equal to the amount produced plus the available inventory from the previous period, which is computed through equation 7.13. Here, the amount of each product manufactured in the plant is calculated from the batch sizes of the products and the binary variables representing the existence of these batches.

$$SALES_{pe} \leq QP_p + IN_p^{ini} \qquad \forall p \in P, e \in E \tag{7.12}$$

$$QP_p = \sum_{i \in I_P} \sum_{s=\{s_p^l\}} \sum_{u \in U_{ps}} bsz_p Y_{pisu} \qquad \forall p \in P \tag{7.13}$$

Constraint 7.14 allows calculating the average inventory of each product.

$$IN_p = \sum_{i \in I_P} \sum_{s=\{s_p^l\}} bsz_p \frac{(H - Tf_{pis})}{H} \quad \forall p \in P \tag{7.14}$$

The model presented here accounts for the maximization of the total expected profit (equation 7.15), which is computed by calculating the average of profits over the entire range of scenarios.

$$max \quad E[PFS] = \sum_{e \in E} prob_e \cdot PFS_e \tag{7.15}$$

The profit values in each scenario are computed through equation 7.16. This equation assumes that revenues are obtained through sales of final products, while costs are due to holding inventories, consumption of utilities and raw materials and the under-production, i.e. leaving part of the demand unsatisfied. An additional term accounting the cost associated to the deviation of the recipe from its nominal parameters has been also considered in this expression.

$$\begin{aligned} PFS_e = & \sum_{p \in P} (SALES_{pe} \cdot sp_p - IN_p \cdot pinv_p - QP_p \cdot pc_p \\ & -udc_p \cdot (dem_{pe} - SALES_{pe})) - \sum_{p \in P} \sum_{i \in I_p} \sum_{s \in S_p} \sum_{f \in FP_{ps}} dcost_{psf} \delta_{pisf} \\ \forall e \in E & \end{aligned} \tag{7.16}$$

Finally, the overall stochastic model without risk management considerations can be expressed as (so-STOC) the maximization of the objective function equation 7.15 subject to constraints 7.1 to 7.16:

$$\begin{array}{ll} \max & E\left[PFS\right] \\ s.t. & \\ constraints & 7.1 - 7.16 \end{array}$$

7.4 Risk Management

The previously proposed stochastic MILP model attempts to account for the uncertainty by optimizing the expected profit without reflecting and controlling the variability of performances associated with each specific scenario. Although the schedules obtained could be considered more robust than the deterministic ones based on nominal parameter values by taking a purely expected profit maximization perspective, the model assumes that the decision maker is risk-neutral or indifferent to profit variability. Therefore, there is no guarantee that the process will perform better at a certain level considering the whole uncertain parameters space. The only guarantee is that the average is optimized (Samsatli et al., 1998; Suh and Lee, 2001).

The underlying idea of risk management is the incorporation of a trade-off between economic risk and profit within the decision-making process. This trade-off leads to a multi-objective optimization problem in which the expected performance and a specific risk measure are the objectives considered. In this

work, the probability of meeting unfavorable scenarios is controlled by adding the worst-case profit as an additional objective to be maximized. A major difference with respect to other probabilistic metrics such as the financial risk (Barbaro and Bagajewicz, 2004) or the downside risk (Eppen et al., 1989) is that the probability information of the problem cannot be explicitly used when manipulating the worst case. Nevertheless, this metrics has been shown to be very effective in identifying robust schedules. Furthermore, it is easy to implement and it leads to a good numerical performance in two-stage stochastic models (Bonfill et al., 2004). Both issues have motivated its application to this specific problem.

The worst case profit can be computed through equation 7.17:

$$WC \leq PFS_e \qquad \forall e \in E \tag{7.17}$$

The inclusion of this term as an alternative objective to be maximized along with the expected profit leads to the following multi-objective formulation (mo-STOC):

$$\begin{array}{ll} \max & E[PFS], WC \\ s.t. & \\ constraints & 7.1 - 7.17 \end{array} \tag{7.18}$$

The above presented multi-objective problem can be solved by standard algorithms for multi-objective optimization, such as aggregation methods, the ε-constraint or goal programming algorithms (Steuer, 1986). Moreover, the problem can be reformulated as a multi-parametric mixed integer programming problem (Papalexandri and Dimkou, 1998), which can then be solved using any of the recently developed algorithms for parametric optimization (Dua and Pistikopoulos, 2000). Specifically, this work applies an enumeration-based approach based on the ε-constraint method (Hugo and Pistikopoulos, 2005) to generate the efficient solutions for the multi-objective problem that accounts for the maximization of the expected profit and the worst case profit. The multi-parametric problem (mp-STOC) that has to be solved can be expressed as follows (equation 7.19):

$$\begin{array}{ll} \max & E[PFS] \\ s.t. & \\ constraints & 7.1 - 7.17 \\ WC \geq \theta & \\ \theta \in [\theta^L, \theta^U] & \end{array} \tag{7.19}$$

where θ^L is given by the value of the worst case profit in the maximum expected profit solution, that is:

$$\theta^L = \max[PFS_e] \tag{7.20}$$

Here, PFS_e represents the profit value attained by the maximum expected profit solution in each scenario. It is worth noting that this solution is obtained by maximizing the expected profit and without considering any financial risk control measure. Similarly, the value of θ^U is the optimal solution of the following single-objective problem:

$$\begin{array}{ll} \max & WC \\ s.t. & \\ constraints & 7.1 - 7.17 \end{array} \tag{7.21}$$

After computing the lower and upper bounds of θ, the resulting interval is discretized into NQ sufficiently small sub-intervals and the ε-constraint method is then applied at each parameter interval realization (Steuer, 1986). In this case, NQ single optimization problems as the one presented above (equation 7.19) are solved. The specific values of θ are calculated from equation 7.22.

$$\theta^1 = \theta^L; \quad \theta^q = \theta^L + \frac{\theta^U - \theta^L}{NQ}; \quad \theta^{NQ} = \theta^U \tag{7.22}$$

Once all the NQ solutions have been computed, an additional post-processing step is required for detecting efficiency based on the concept of dominance (Hugo and Pistikopoulos, 2005). A solution A is said to be dominated by another solution B, if the expected profit and worst case values associated with solution B are both greater or equal to those associated with solution A, and at least one of them is strictly greater. Thus, this last step of the algorithm has to be included as part of the overall multi-objective optimization algorithm and focuses on discarding those solutions which are dominated by at least one of

the others. A solution is considered to be Pareto-optimal if it is not dominated by any other solution.

7.4.1 Decomposition strategy

The approach presented above, leads to a large-scale mo-MILP. This model may become computationally expensive in real industrial scenarios with complex plant configurations and large number of different products and production recipes. The complexity of this formulation may be further increased by the high number of scenarios required to capture the demand uncertainty.

Thus, in this section a decomposition strategy is introduced aiming at the objective of overcoming the numerical difficulties associated with the approach described in the previous section. The proposed strategy is based on the Sample Average Approximation (SAA) algorithm (Verweij et al., 2001). The underlying idea consists of using a variation of this technique to approximate the solution of the mo-MILP that accounts for the maximization of the expected profit and worst case. Here, the SAA is used as a way of generating a set of candidate solutions that exhibit different risk performances and whose Pareto optimality, in terms of expected profit and worst case, must be checked based on the concept of dominance. The proposed method to manage risk has similar features to the one suggested by Aseeri and Bagajewicz (2004).

In general terms, the SAA is an approach for solving single objective stochastic optimization problems by using Monte Carlo simulation (Verweij et al., 2001). In the SAA algorithm, the expected second-stage profit (recourse function) in the objective function is approximated by an average estimate of NS independent random samples of the uncertain parameters, and the resulting problem is called approximation problem. Here, each sample corresponds to a possible scenario. Then, the resulting approximation problem is solved repeatedly for R different independent samples (each of size NS) as a stochastic optimization problem of size NS. The average of the objective function of the approximation problems provides an estimate of the stochastic problem objective. Notice that this procedure may generate up to R different candidate solutions. To determine which of these R (or possibly less) candidates is optimal in the original problem, the values of the first-stage variables corresponding to each candidate solution are fixed and the problem is solved again using a larger number of scenarios $NS' >> NS$ in order to better distinguish between the candidates. After solving these new problems, the optimal solution of the original problem ($\hat{x}^e$) is determined. Therefore, $\hat{x}^e$ is given by the

solution of the approximate problems that yields the highest objective value for the approximation problem with NS' samples.

In this chapter, we apply a variation of the SAA that aims to calculate the set of efficient solutions to the multi-objective formulation that simultaneously accounts for the maximization of expected profit and worst case (mo-STOC). In this approach the problem is solved deterministically for each scenario, in other words, $NS = 1$; $R = |E|$ and $NS' = |E|$. This provides a set of first-stage decision variables (i.e. schedules) that must be evaluated in the uncertain parameters space. Thus, the second stage decision variables (sales and inventory profiles) associated with each of these deterministic solutions are calculated by fixing the deterministic schedules in the stochastic problem and solving it. The stochastic problem is the original single-objective two-stage model with $|E|$ scenarios that maximizes the expected profit (so-STOC). In such model, the combinatorial complexity is eliminated by fixing the scheduling decisions. This provides, for each schedule being assessed, the expected profit and the worst case attained under the uncertain environment. Finally, the solutions are filtered in terms of expected profit and worst case by applying the dominance concept. Those solutions that are dominated in terms of the predefined criteria are discarded from the original set, whereas the set of non-dominated schedules are stored. The latter ones constitute the approximated solution to the original problem. For a stochastic problem of type mo-STOC with $|E|$ scenarios, the proposed algorithm can be summarized in figure 7.2.

This algorithm provides an approximation (lower bound) to the original Pareto frontier of the problem. Let us note that the computational resources associated with the proposed algorithm can be decreased by reducing the optimality gap as well as the maximum CPU time imposed to the so-MILPs to be solved.

7.5 Case study and Results

The advantages of the proposed framework are illustrated by a new scenario for the benzyl alcohol (Romero et al., 2003). Table 7.1 shows the flexible recipe items considered for the flexible recipe model of this reaction (equation 6.17) and the new valid flexibility region around the nominal operating conditions. Data for this production scenario are shown in table 7.2. Transfer times have been considered equal to the 5% of the processing times.

This case study consists of four products, $P1$ to $P4$, which must be pro-

```
begin
    for e=1 to |E| do
        solve the deterministic problem for scenario k. Let x̂^e be the optimal
        solution of the problem;
        store the first stage decisions (i.e., schedules) associated with x̂^e;
        fix the first stage decisions in the stochastic problem with |E| scenarios
        (so-STOC) and solve it;
        store the expected profit and worst case associated with x̂^e;
    end for
    filter the solutions x̂^e by applying the dominance concept;
end
```

Figure 7.2: Sample Average Algorithm.

Table 7.1: Recipe parameters, flexibility region and cost for deviation from nominal conditions.

Recipe item		Flexibility Region		$dcost_{psf}$
		$fplb_{psf}$	$fpub_{psf}$	
δ_{DPS}	reaction yield	0	0	-
δ_{DTEMP}	reaction temperature	-0.7 °C	0.5 °C	0.03 m.u./°C
δ_{DTOP}	reaction duration	-0.3 h	0.1 h	0.02 m.u./ h
δ_{DKOH}	amount of KOH	-27 g	8.5 g	0.05 m.u./g
δ_{DFOR}	amount of Formaldehyde	-30 g	7.5 g	0.04 m.u./g

cessed over a 112-hour horizon in a multistage multipurpose batch plant. Initial inventory of final products is assumed to be inexistent at the period of time considered. It is assumed that the demand cannot be perfectly forecasted and its uncertainty is represented by 100 equiprobable and independent scenarios. These scenarios are generated by applying a Monte Carlo sampling over a set of probability functions. Specifically, the scenarios are calculated assuming that the demand of each product follows a normal probability distribution. The

production rates (bsz_p), maximum plant capacity (mpc_p), selling prices (sp_p) and production costs (pc_p) are also given in table 7.2, as well as the market information, namely the mean demands ($mdem_p$), their standard deviations ($demdev_p$) and the unsatisfied demand costs (udc_p).

Second stage of products $P1$ and $P2$ and first stage of product $P4$ are flexible tasks that follow the flexible recipe model in equation 6.17. Furthermore, first stage of product P1 is a preheating stage in which end-point temperature ($DTEMP$) is assumed to increase linearly with time ($DTOP$) as shown in equation 6.18 (chapter 6).

Table 7.2: Problem data.

Product	P1		P2		P3		P4	
Stages	Unit	npt, h	Unit	npt, h	Unit	npt, h	Unit	npt, h
1	U1	2	U3	6	U4	4	U2	7
2	U2	7	U1	6	U2	8	U1	2
3	U3	8	U2	8	U3	6	U4	6
4	U4	4	U4	8	U1	4	U3	4
bsz_p, kg/batch	40		55		40		35	
mpc_p, kg	160		110		120		105	
$mdem_p$, kg	280		110		120		120	
$demdev_p$, %	30		35		40		40	
sp_p, m.u./batch	70		55		40		40	
pc_p, m.u./batch	8		7		5		8	
$pinv_p$, m.u./h	1		2		1.1		0.5	
udc_p, m.u./kg	4		3		2		3	

The stochastic approach that maximizes expected profit and neglects risk management is solved in first place. The model is implemented within the modeling language GAMS and solved with the MILP solver of CPLEX version 9.0. The mathematical formulation involves 1569 constraints, 653 continuous variables and 224 binary variables. Specifically, two models are solved, the first one for the fixed recipe case and the second one assuming a flexible recipe framework. The fixed recipe model leads to an expected profit of 15348.4 m.u., while the flexible one yields 15858.7 m.u. The schedules associated with both solutions are shown in figure 7.3.

As can be observed, the improvement in the objective function value using flexible recipes is achieved by producing one more batch of product $P3$. To do so, the flexible recipe framework takes advantage of its capability of reducing

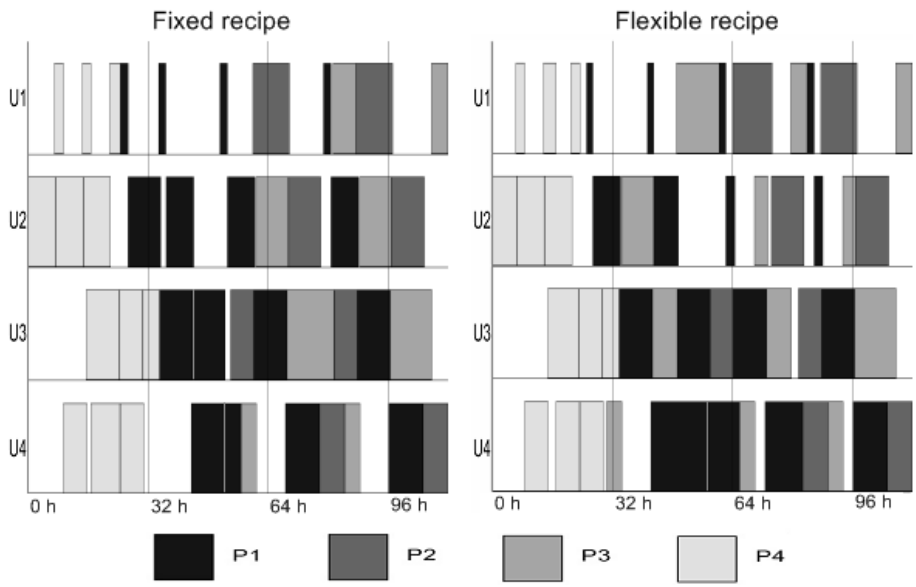

Figure 7.3: Gantt-charts of the stochastic solutions.

the processing times. This results in an extra recipe modification cost which is compensated by the increase in the sales revenues, thus leading to an overall better solution in terms of the proposed objective function. Furthermore, comparing both schedules, several sequencing decisions have been changed in order to accommodate this additional batch while using the flexible recipe framework.

The trade-off between risk, which is assessed by using the worst-case and expected profits, is next investigated by solving the multi-objective MILP model that accounts for the maximization of both criteria. The problem is solved in two different ways. The first one is the enumeration-based approach based on the ε-constraint method (Hugo and Pistikopoulos, 2005), which consists of parametrically varying the value of the lower bound imposed to the worst-case metric. The second one is the strategy based on the SAA algorithm. With regard to the latter, let us note that it decomposes the original two-stage stochastic problem with 100 scenarios into 100 deterministic problems that are solved for every scenario e in the original formulation. To increase the speed of the algorithm, a limit of 100 CPU seconds and an optimality gap of 5 % are imposed for every deterministic sub-problem. The results obtained in both cases are illustrated in figure 7.4. This figure also depicts the efficient solutions

associated with the fixed recipe mode of operation.

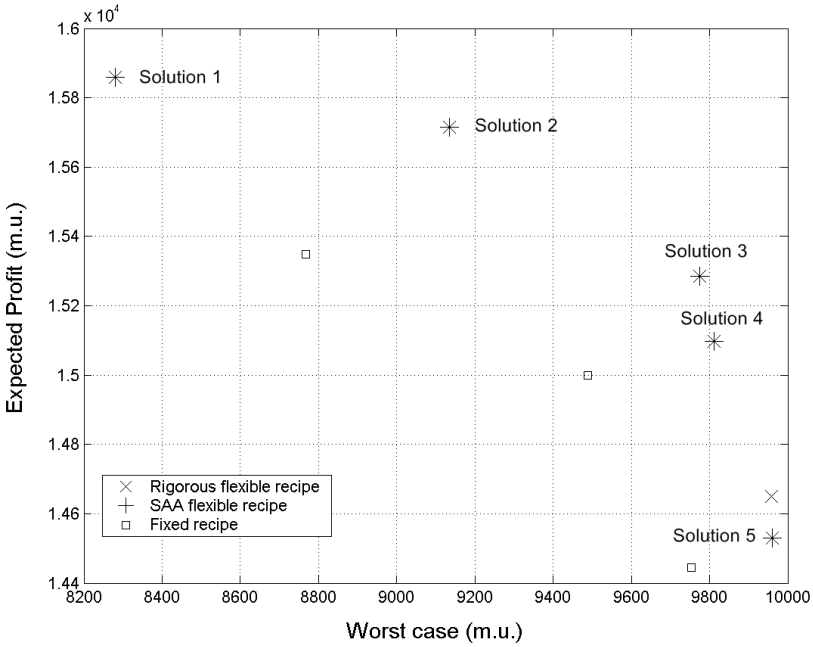

Figure 7.4: Optimal-pareto solutions for fixed and flexible recipe frameworks.

From the set of efficient solutions of the problem, it is clear that in both cases (i.e. fixed and flexible recipe modes of operation) a conflict exists between both objectives (i.e. maximum expected profit and maximum worst case). Thus, results indicate that an improvement in the worst case is only possible if the decision-maker is willing to compromise the expected profit. Certainly, schedules with better worst-case values will reduce the risk but at the expense of a reduction in the value of the expected profit.

The effectiveness of introducing flexible recipes is justified by the fact that the curve obtained using flexible recipe lies entirely above the one associated with the fixed recipe. This implies that for any worst-case value the flexible recipe framework always leads to a higher expected profit. Specifically, the difference between both expected profits ranges from 2 to 6% depending on the specific worst case value being analyzed.

In addition, the use of the flexible recipe allows attaining solutions with higher worst-case values, which are impossible to achieve otherwise by using

fixed recipes. In fact, the maximum worst-case solution computed with the flexible recipe formulation is equal to 9774 m.u., whereas the fixed recipe model is not able to provide solutions with a worst case higher than 9960 m.u.

This analysis provides valuable information to the decision maker and indicates that the inclusion of flexible recipes in a batch plant manufacturing environment leads to a more effective way of managing the associated risk.

Table 7.3: Optimal-pareto solution applying SAA for the flexible recipe framework.

Solutions	WC, m.u.	Expected profit, m.u.	Number of batches			
			P1	P2	P3	P4
1	8280.18	15858.74	3	3	3	3
2	9133.99	15715.05	3	3	2	3
3	9774.14	15285.79	3	3	2	2
4	9811.44	15097.96	3	3	1	3
5	9960.29	14528.82	3	3	2	1

With regard to the proposed decomposition strategy, it shows a good numerical performance, as it is able to detect five out of the six Pareto-optimal solutions identified by the rigorous approach. These solutions are summarized in table 7.3. Figure 7.5 includes the corresponding Gantt charts which shows how all the batches of $P1$ and $P2$ are always produced. Furthermore, in order to satisfy the constraint of a minimum profit in the worst scenario, as the value of the worst case is more demanding, solutions prefer producing fewer amount of $P3$ and $P4$ because products $P1$ and $P2$ have a less uncertain demand than $P3$ and $P4$. The whole problem was solved in approximately 3.8 h CPU time, which implies a great time reduction compared to the rigorous solution procedure by using the ϵ-constraint method discretizing the space into 12 sub-intervals. This comparison can be observed in table 7.4.

Next, for each Pareto solution of the problem a cumulative probability curve can be plotted (i.e. each efficient schedule in terms of expected profit and risk has an associated cumulative probability function). Cumulative risk curves provide further insights for better understanding and assessment of the trade-off posed between risk and expected profit (Barbaro and Bagajewicz, 2004). Figure 7.6 represents the cumulative risk curves of the solutions provided by the fixed and flexible recipe strategies for a worst case equal to 9600 m.u.

The aforementioned curves show the level of financial risk incurred at each

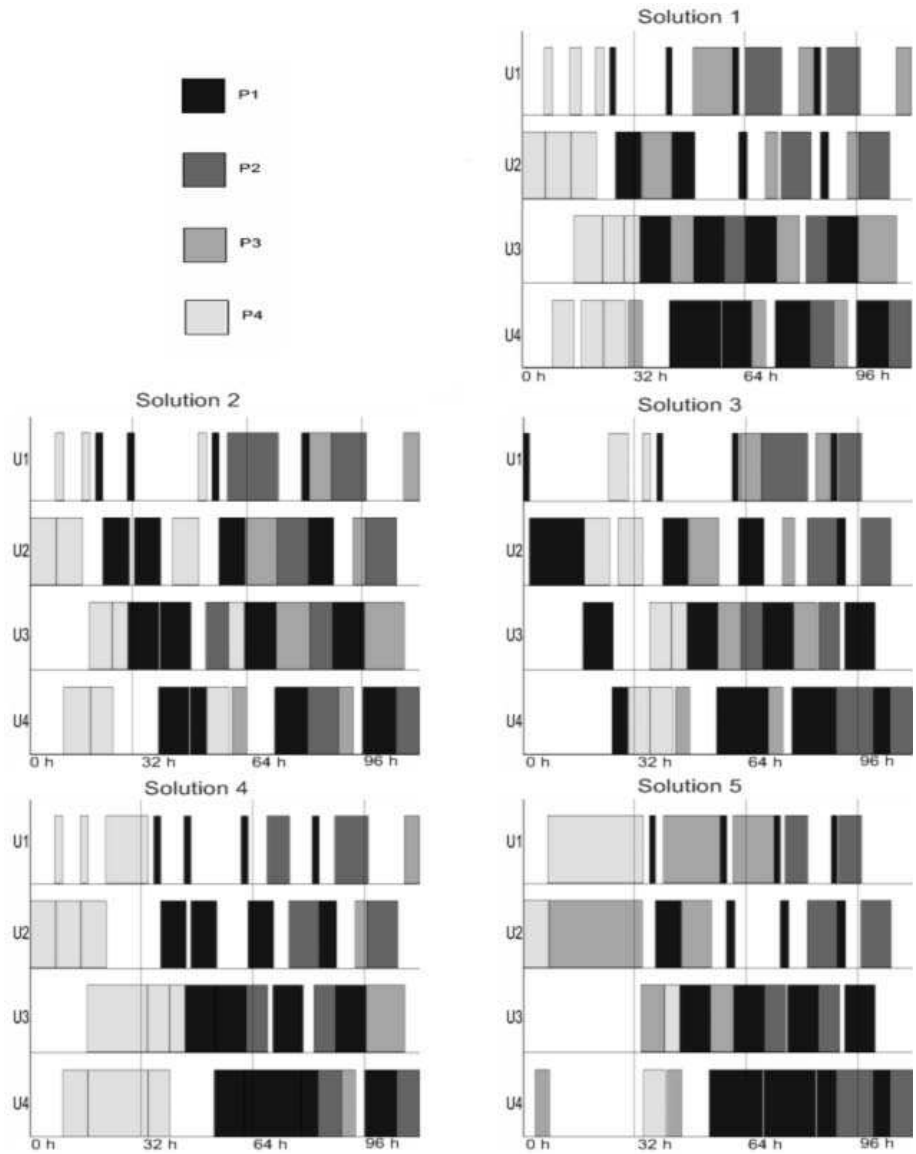

Figure 7.5: Associated schedules for each solution.

profit level. Comparative analysis of the curves obtained for the fixed and the flexible recipe frameworks shows how the flexible recipe curve lies entirely above the fixed recipe one. Thus, the flexible recipe framework not only leads

Table 7.4: Computational results.

Problem (recipe)	Bin., cont., rows	Average CPU, s	Number of realizations	Total CPU, h
Full scale (fixed)	224, 654, 1669	2600	12	8.7
Full scale (flexible)	224, 808, 1853	4300	12	14.3
SAA algorithm (flexible)	224, 311, 1653	70	100	3.8
	212, 819, 1753	69	100	

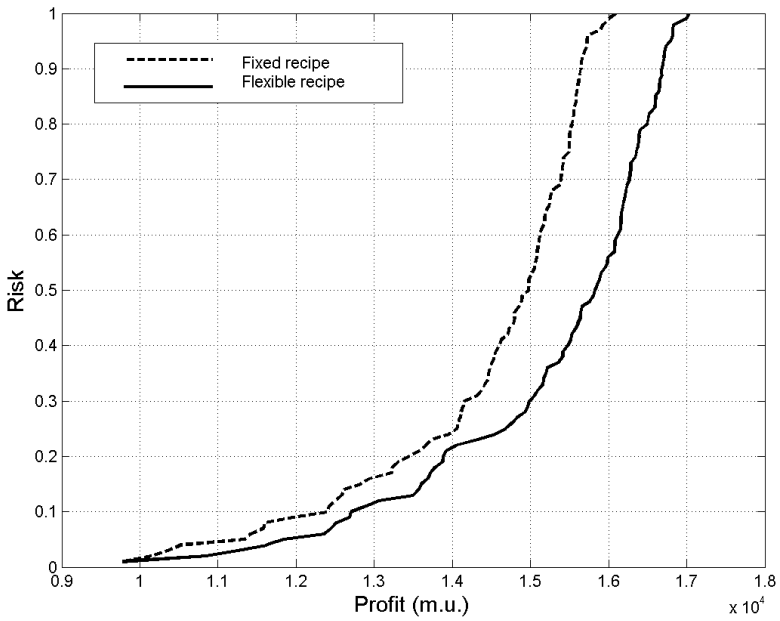

Figure 7.6: Cumulative risk curve for a worst-case value of 9600 m.u.

to lower probabilities of poor profits but also to higher chances of large benefits, which makes the overall production less risky in economic terms. Hence, the flexible recipe solution performs better over the entire uncertain space as can be seen in the shift to the right of the risk curve. For instance, a 5% probability of scenarios with earnings below 12000 m.u. is achieved using flexible recipes, while this probability increases up to 10% using traditional fixed recipes. Similarly, the use of flexible recipes yields higher probabilities of larger benefits.

For instance, a 45% probability of earnings above 16000 m.u. is reported by this solution compared to the very scarce probability, less than 1% using fixed recipes.

Consequently, these results shows that considering recipe flexibility allows an enhanced management of the financial risks.

7.6 Conclusions

This chapter has proposed a novel stochastic mathematical formulation to address in a proactive way risk management in the scheduling problem associated with batch chemical plants operating under demand uncertainty. The main novelty of this work lies in the application of a flexible recipe framework as a way of reducing the probability of meeting unfavorable scenarios in an uncertain market environment. The problem has been mathematically posed as a multi-objective multi-scenario two-stage stochastic formulation accounting for the maximization of the profit and the worst case value. The resulting formulation is based on a continuous-time domain representation and the generalized notion of precedence. As such, the proposed model explicitly incorporates the trade-off between risk and profit at the decision-making level. Furthermore, a decomposition strategy based on the sampling average approximation (SAA) has been studied as a way of overcoming the numerical difficulties associated with the application of the proposed strategy to large scale industrial scenarios. The decomposition method provides near optimal solutions and incurs in much less CPU time than the monolithic formulation.

Finally, the effectiveness of the proposed approach as a decision-making tool has been highlighted through a scheduling case study. Results indicate that for a given level of risk, higher benefits can be obtained by using flexible recipes. Furthermore, using flexible recipes allows attaining conservative solutions that would be impossible to obtain by applying the conventional approach.

Therefore, the consideration of the flexibility in the batch recipes has been shown to be a highly valuable risk management instrument to support the decision making process in an uncertain market environment.

On pipeless plants

Chapter 8

Short-term scheduling of pipeless batch plants

Significant increase in flexibility of a batch production environment can be achieved by using the pipeless plant concept. In a pipeless plant, interconnected vessels by fixed network of pipes no longer exist. Instead, moveable units are used to transport material from processing station to station.

This chapter presents a contribution for the optimal management of pipeless plants presenting an alternative MILP mathematical formulation for solving the short-term scheduling problem. In order to show the applicability and performance of the proposed approach, an example taken from the literature is examined and compared to others approaches of other authors.

8.1 Introduction

Batch pipeless plants are a very attractive alternative to manufacture specialty, low volume and high added value pharmaceuticals, cosmetics, polymers or biochemical products. Its inherent flexibility is capable of easily accommodating a diverse range of products in order to quickly react against highly dynamic market conditions. However, as a consequence of this greater flexibility, the scheduling of pipeless batch plants present also a higher combinatorial complexity.

The scheduling model presented in this chapter for the scheduling of pipeless plants is based on a continuous-time representation that relies on the general precedence notion. Besides of the intrinsic characteristics of a pipeless plant, this model considers the moveable vessels as an additional resource that has to be allocated and sequenced. This fact allows proposing a strategy consisting of a sequential treatment of the resources in order to reduce the complexity of the problem. First, the most critical resources are identified between the processing units and the available moveable vessels. Therefore, the problem is solved assuming that infinite amount of resources exists and only the most critical resources are constrained. Afterwards, the characteristics of the general precedence model allow setting this first-step decision variables associated to the critical resource, in order to next solve again the model just estimating the rest of decision variables related to the other resources. This strategy dramatically reduces the computational burden of the optimization problem and is able to obtain satisfactory solutions for a fast industrial environment.

This chapter is structured as follows. After this introduction, the problem is briefly defined in section 8.2 and next, the mathematical formulation is described in section 8.3. A motivating example taken from the literature is described in section 8.4 and solved in section 8.5. Later on, the problem size of this example is increased for discussing how to address this problem in order to obtain efficient solutions in reasonable time. Finally, this chapter ends up with some conclusions in section 8.6.

8.2 Problem statement

Given are a set of products p and a set of batches i of each product that have to be manufactured in a series of s consecutive processing stations. A suitable moveable vessel k carries the material between every station where at least one processing unit u exists.

The following list contains the main assumptions of the proposed model:

- The process layout for processing units is fixed.
- The number of moving vessels is fixed.
- Model parameters all are deterministic.
- Batch splitting or mixing is not allowed.

- Sufficient raw materials are always available.
- One batch is required of each product.

The goal of this problem is the minimization of makespan.

8.3 Mathematical model

Constraint 8.1 enforces the allocation of a suitable processing unit u to every task (p, i, s). Constraint 8.2 guarantees that if task (p, i, s) precedes task (p', i', s') and both tasks are processed in the same unit u, task (p', i', s') can not start until task (p, i, s) is finished. Alternatively, constraint 8.3 considers the opposite case of the aforementioned constraint 8.2, that is, task (p', i', s') precedes task (p, i, s). Constraint 8.4 synchronizes pair of tasks performed in two consecutive stages and constraint 8.5 sequences two batches of the same product executed in the same processing unit.

$$\sum_{u \in U_{ps}} Y_{pisu} = 1 \qquad \forall p \in P, i \in I_p, s \in S_p \tag{8.1}$$

$$\begin{aligned} &Ts_{p'i's'} \geq Tf_{pis} - M \cdot (1 - X_{pisp'i's'}) - M\left(2 - Y_{pisu} - Y_{p'i's'u}\right) \\ &\forall p, p' \in P, i, i' \in I_p, s, s' \in S_p, u \in (U_{ps} \cap U_{p's'}) : p < p' \cap (p = p', s < s') \end{aligned} \tag{8.2}$$

$$\begin{aligned} &Ts_{pis} \geq Tf_{p'i's'} - M \cdot X_{pisp'i's'} - M\left(2 - Y_{pisu} - Y_{p'i's'u}\right) \\ &\forall p, p' \in P, i, i' \in I_p, s, s' \in S_p, u \in (U_{ps} \cap U_{p's'}) : p < p' \cap (p = p', s < s') \end{aligned} \tag{8.3}$$

$$Ts_{pis'} \geq Tf_{pis} + tt_{ps'} \qquad \forall p \in P, i \in I_p, s, s' \in S_p : s' = s + 1 \tag{8.4}$$

$$Ts_{pi's} \geq Tf_{pis} - M(2 - Y_{pisu} - Y_{pi'su}) \quad \forall p \in P, i, i' \in I_p, s \in S_p, u \in U_{ps} : i' > i \tag{8.5}$$

The set of available moveable vessels is an additional limiting resource that has to be considered in this model. Likewise, as it was done for the processing

units, constraint 8.6 assigns a suitable moveable vessel ($k \in K_p$) by using a decision variable, Z_{pik}, equal to 1 if that batch is assigned to that moveable vessel. Special vessels might be needed to transport a particular product in case of, for example, corrosive products or products with special preservation requirements as perishable foods.

$$\sum_{k \in K_p} Z_{pik} = 1 \qquad \forall p \in P, i \in I_p \tag{8.6}$$

Similarly to constraints 8.2 and 8.3, constraints 8.7 and 8.8 sequence two tasks of different products but assigned to the same moveable vessel.

$$\begin{aligned} &Ts_{p'i's'} \geq Tf_{pis} + tt_{p's'} - M(1 - X_{pis'p'i's'}) - M(2 - Z_{pik} - Z_{p'i'k} \\ &\forall p, p' \in P, i, i' \in I_p, s, s' \in S_p, k \in (K_p \cap K_{p'}) : p < p', s = \left\{s^f_{pi}\right\}, s' = \left\{s^l_{p'i'}\right\} \end{aligned} \tag{8.7}$$

$$\begin{aligned} &Ts_{pis} \geq Tf_{p'i's'} + tt_{ps} - M \cdot (1 - X_{pisp'i's'}) - M\left(2 - Z_{pik} - Z_{p'i'k}\right) \\ &\forall p, p' \in P, i, i' \in I_p, s, s' \in S_p, k \in (K_p \cap K_{p'}) : p < p', s = \left\{s^f_{pi}\right\}, s' = \left\{s^l_{p'i'}\right\} \end{aligned} \tag{8.8}$$

Constraint 8.9 sequences tasks sharing the same moveable vessel to make them not simultaneous in time.

$$\begin{aligned} &Ts_{pi's'} \geq Tf_{pis} + tt_{ps'} - M\left(2 - Z_{pik} - Z_{pi'k}\right) \\ &\forall p \in P, i, i' \in I_p, s, s' \in S_p, k \in K_p : i' < i, s = \left\{s^l_{pi}\right\}, s' = \left\{s^f_{p'i'}\right\} \end{aligned} \tag{8.9}$$

Constraint 8.10 establishes the initial transfer time for the first task of the batch manufactured.

$$Ts_{pis} \geq tt_{ps} \qquad \forall p \in P, i \in I_p, s \in S_p : s = \left\{s^f_{pi}\right\} \tag{8.10}$$

Constraint 8.11 establishes the duration of a task.

$$Tf_{pis} \geq Ts_{pis} + \sum_{u \in U_{ps}} pt_{pu} Y_{pisu} \qquad \forall p \in P, i \in I_p, s \in S_p \tag{8.11}$$

Alternative objective functions may be evaluated using this formulation. In this case makespan has been considered for simplicity (equation 8.12).

$$\min \quad MK \geq Tf_{pis} \qquad \forall p \in P, i \in I_p, s \in S_p : s = \left\{ s^l_{pi} \right\} \tag{8.12}$$

8.4 Case study

The case study addressed was firstly introduced by Bok and Park (1998). Figure 8.1 depicts the layout of this multistage pipeless plant with parallel available units for each station (table 8.1).

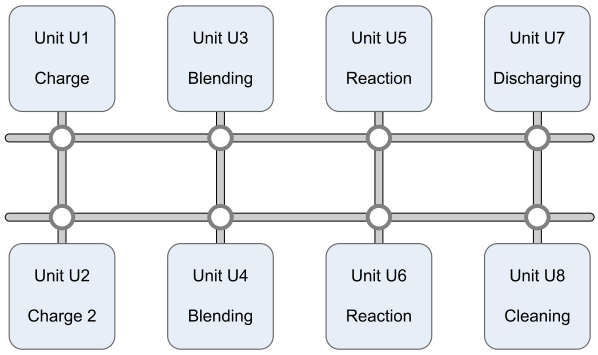

Figure 8.1: Layout of the pipeless plant.

Table 8.1: Processing stations and parallel units.

Station	Task	Processing units
S1	Charging A	U1
S2	Charging B	U2
S3	Blending	U3 , U4
S4	Reaction	U5 , U6
S5	Blending	U3 , U4
S6	Discharging	U7
S7	Cleaning	U8

This plant is designed to manufacture three batches of different products following the same production sequence. The vehicle to which a vessel is at-

tached moves on the rail and drops by each station to carry out unit operations. Figure 8.2 represents the state-task network for this plant. Table 8.2 shows the seven stations that the products have to undergo and the available processing units at every station. Transfer times from/to the stations are listed in table 8.3. Setup times are included within the transfer times.

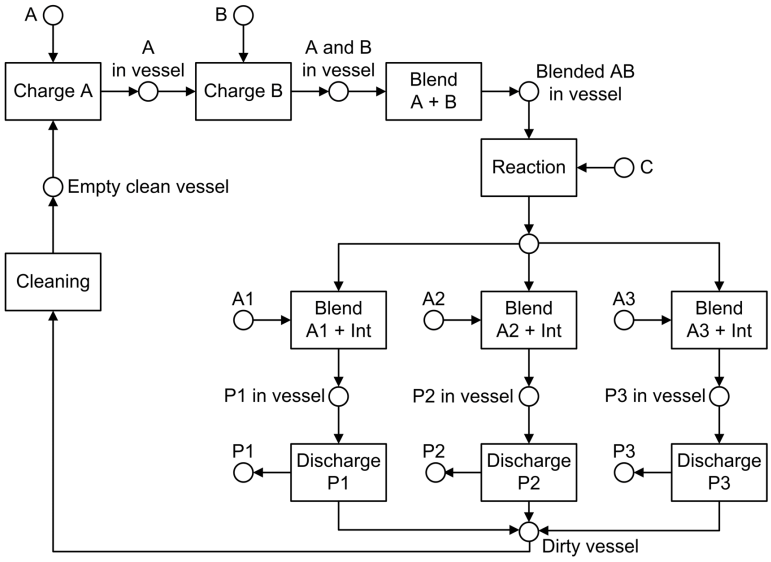

Figure 8.2: STN representation of the case study.

Table 8.2: Processing times in hours.

Products	Units					
	U1	U2	U3	U4	U5	U6
P1	0.6	0.5	0.5	0.85	0.85	0.6
P2	0.5	0.5	0.7	0.75	0.75	0.5
P3	0.5	0.6	0.5	0.65	0.65	0.6

Table 8.3: Transfer times in hours.

Products	Stations						
	S1	S2	S3	S4	S5	S6	S7
P1, P2, P3	0.05	0.06	0.05	0.06	0.07	0.05	0.1

This formulation has been implemented within the modeling language GAMS and solved using CPLEX version 7.5.

8.5 Results

Table 8.4 summarizes the results obtained by means of the direct application of the proposed formulation. This table also presents a comparison with the results reported by Bok and Park (1998).

Table 8.4: Comparative results.

K	Bok and Park, 1998		This model		MK, h
	Bin., cont., rows	Iters.	Bin., cont., rows	Iters.	
2	207, 340, 442	6375	66, 43, 177	495	8.28
Unconst.	207, 340, 441	100751	60, 43, 162	200	5.54

Looking at the model size, it is remarkable the significant saving of binary variables, continuous variables and constraints achieved by this model. This reduction in the number of variables is directly translated into the number of solver iterations in order to reach global optimality. An addition advantage of the proposed formulation can be exploit when the number of moveable vessels is equal or larger than the number of batches (unconstrained case). In this case, by handling both resources (processing units and moveable vessels) through different sets of constraints, the moveable vessels constraints can be discarded, and thus reducing significantly the magnitude of the problem.

Left part of table 8.5 shows a direct application of this model to the same problem but with an increased demand. As it was expected, the model complexity increases rapidly with the number of batches to be scheduled in combination with the number of available moveable vessels. In this case with two batches of each product, six batches in total, the proposed formulation needs an extremely high CPU time.

8.5.1 Decomposition strategy

In order to overcome the calculation complexity of this model for large-sized problems, this formulation allows a sequential treatment of the resources by decomposing the scheduling problem into two sub-problems. By means of

this decomposition strategy, every type of resource is sequenced and allocated separately. The priority order for solving these scheduling sub-problems is given by those resources considered as more critical. The underlying idea here is trying to provide efficient solutions, but not necessarily optimal, when the problem size makes the solution unaffordable in a reasonable calculation time.

Therefore, the unconstrained problem is solved first discarding the variables and constraints related to the moveable vessels. Next, the binary allocation and sequencing variables for the processing unit are fixed and the model is solved again just working only in the decision variables related to the moveable vessels. Right par of table 8.5 shows also the results obtained using this sequential approach and its CPU times.

Table 8.5: Comparative performance using the sequential approach for large-sized problems.

K	Full problem using this model				Sequential approach	
	Bin.,cont.,rows	Iters.	CPU, s	MK, h	CPU, s	MK, h
3	198, 85, 627	57624851	36635	9.75	27.7 + 0.1	10.07
4	204, 85, 654	12087918	9850	8.94	27.7 + 0.1	9.37
5	210, 85, 681	$7.3 \cdot 10^7$	46633	8.28	27.7 + 0.1	8.53
Unconst.	192, 85, 546	135619	27.7	7.59	27.7	7.59

Although the optimal solution may not be found, good quality solutions are encountered in very short computational times. This situation poses a trade-off between obtaining optimal solutions at the expense of huge calculation time and acceptable solutions in very small times. Figure 8.3 shows the Gantt chart of the solution using four moveable vessels.

8.6 Conclusions

This last chapter has presented pipeless plants as the most more flexible manufacturing plant configuration. An important operational issue in these plants is the scheduling problem which involves an additional complexity compared to the conventional batch plants.

In this chapter, a continuous-time MILP formulation is proposed to address the scheduling of pipeless plants that achieves an important saving of binary variables and computational effort. The use of a sequential treatment

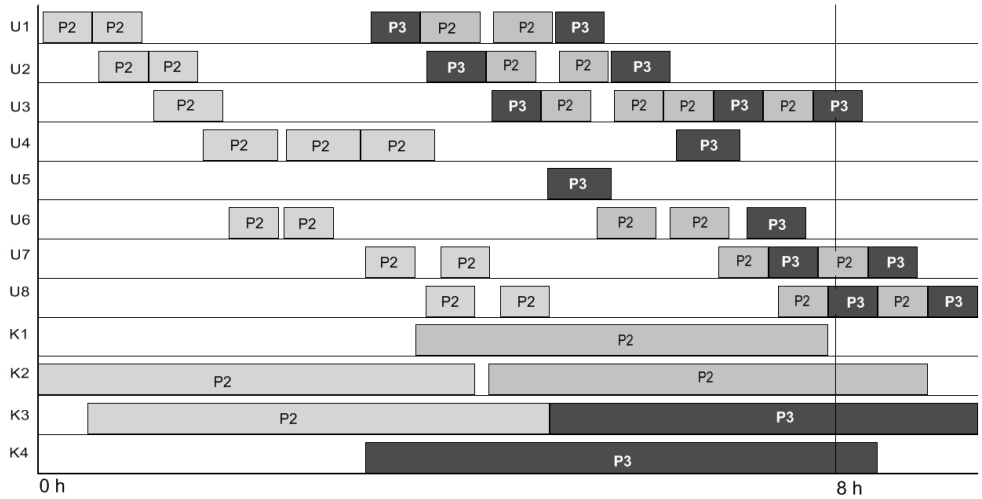

Figure 8.3: Gantt chart of the solution with four moveable vessels.

approach to find effective solutions for large-sized scheduling problems with modest computational time has been also illustrated.

Chapter 9

Conclusions and future work

This thesis has presented new methodologies and solution approaches aimed at providing decision support for the flexible operation of chemical plants. The contribution of this thesis has not been providing more flexible processes, because in fact processes already are flexible, but to identify this inherent flexibility and to develop a set of tools to exploit it in the most convenient way. Since flexibility in chemical processes is a very general concept, thus a very large field of research, this thesis has focused on the operational management of plants at the supervisory control, production planning and scheduling levels. Furthermore, the work presented in this thesis has improved current models to manage plant flexibility by allowing interaction between different decision levels of the operational hierarchy.

In order to give a general overview of the research in this thesis, the introductory Chapter 1 exposed the current state of the chemical industry and emphasized the necessary role that the PSE community must play in developing novel methods and tools to help industry remain competitive. This chapter included a comprehensive review of the different types of operation modes, with the aim of demonstrating the intrinsic flexibility in each mode. In order of increasing flexibility, the analyzed operation modes were continuous, semicontinuous, batch, and batch pipeless.

Chapter 2 laid the stepping stones for the development of this thesis. This chapter briefly reviewed some of the approaches reported in the literature that are oriented towards the efficient management of chemical processes by making use of the process flexibility. A study for each individual type of operation mode revealed a series of challenges and opportunities for improvement that were posed as the main objectives of this thesis at the end of the chapter.

Flexibility in continuous processes has been addressed in Chapter 3 by presenting a supervisory control system able to correctly manage the occurrence of abnormal events and optimization in real time. The system proposed in this chapter interacts with the control system by tracking its set-points in order to adapt the plant faster to changes in operating conditions. The excellent on-line performance showed by this system can be only achieved through the adjustment of the operating conditions, which is a flexibility allowed by continuous processes.

The scheduling of semicontinuous processes, that is continuous processes with frequent product transition, was addressed in Chapter 4. In the scheduling literature, semicontinuous processes have been considered to be operated through production campaigns of finite duration and constant rates. The mathematical formulation presented in this chapter allowed transitions of the processing rates within a campaign, resulting in more flexible operation. The case study used to validate the applicability of the proposed formulation led to the conclusion that more efficient production can be achieved by an optimized management of the storage and the rates of production. In this context, intermediate storage was handled to buffer different upstream and downstream production rates.

The following three chapters were devoted to the optimal management of batch processes, which have been traditionally considered the most flexible mode of operation.

Chapter 5 analyzed the problem of finding optimal schedules in batch plants. The study carried out in this chapter identified the lack of consideration for transfer operations of most of the MILP scheduling formulations reported in the literature, which can lead to unfeasible solutions in multipurpose plants. Therefore, the main aim of this chapter was to highlight the importance of transfer operations for the synchronization between tasks in batch scheduling. In addition, two algorithmic approaches were proposed which remove the unfeasible transfers before the MILP model is solved.

The works presented in chapters 6 and 7 have considered uncertainty is-

sues in batch scheduling. Both chapters relied on the use of flexible recipes that integrate batch optimal control with short-term production scheduling optimization as a way to deal with uncertainty. Linear models were used to illustrate the potential use of the flexible recipe concept.

On one hand, Chapter 6 presented a rescheduling strategy to be applied once the uncertainty is unveiled. It uses the flexibility of the batch recipes as an extra rescheduling action. On the other hand, Chapter 7 promoted the generation of proactive schedules which take into account information about uncertainty before the production plan is executed in the plant. In this work, the probability of producing unfavorable scenarios has been controlled by using flexible recipes and a risk measurement that assesses the probability of meeting unfavorable scenarios.

These results indicate that the concept of nominal recipes should no longer be used to represent an isolated entity. Production recipes must allow some flexibility in order to be considered viable in a global context, which dictates a non-steady production environment in which nominal conditions are rarely found.

Finally, Chapter 8 addressed the highest degree of flexibility in manufacturing, that is the operational management a special type of batch plants which do not require the use of piping to transfer materials between different processing stages. The scheduling problem of these pipeless plants was addressed in this chapter applying the concept of general precedence. The MILP formulation presented in this chapter outperforms the previous approaches reported in the literature, drastically reducing the calculation times.

9.1 Future work

The work presented in this thesis represents several substantial contributions that can be used as the basis for future work. In order to build and enhance this body of work, a great deal of extended research may be addressed to the following topics:

- On *continuous processes*, and regarding the issues that have been treated in this thesis, much work should be done to promote the use of Real Time Optimization in industrial scenarios. Applications of this technology are still scarce, and therefore robust systems, such as the one proposed in this thesis, should be marketed to the chemical industry as an attractive way

to gain a quantitative leap in profit. Parallelly, academic research should continue working on the development of new algorithms and methodologies to more efficiently address the on-line optimization of plants using dynamic models, which is the future for RTO.

- On *semicontinuous processes*, few publications have paid attention to its scheduling problem in comparison to batch processes. Therefore, future work may consider more aspects in the scheduling, such as limited resources, more complex plant configurations or the assessment of the convenience of using single or multiple campaigns for the productions of a sole product. Other interesting contributions could be the development of strategies for dealing with the uncertainty in semicontinuous processes.

- On *batch processes*, this thesis has pointed out the necessity of taking into account the synchronization between transfer operations in the scheduling problem. Hence, the modeling of transfer operations constitutes a very interesting and challenging work with the aim of adapting the current formulations while keeping reasonable computational times. Another important point for discussion is the use of linear flexible recipe models. As it was previously mentioned in the corresponding chapter, linear flexible recipe models were used in this thesis to serve only for demonstrative purposes. Therefore, the consideration of non-linear models should be taken into account in future works in order to obtain more accurate solutions.

- On *batch pipeless plants*, further work should be done, starting from the formulation developed in this thesis to consider specific plant layouts.

Nomenclature

Although most of the acronyms and symbols are explained in the place where they appear, this section is a quick reference to the reader for the notation used along this work.

Acronyms

AGV	Automated Guided Vehicle.
$B\delta B$	Branch and Bound.
BPN	Back-Propagation Networks.
CIS	Common Intermediate Storage.
CP	Constraint Programming.
CPU	Central Processor Unit.
CST	Constraint Satisfaction Techniques.
DAE	Differential Algebraic Equation.
DR	Data Reconciliation.
EDD	Earliest due date.
ERD	Earliest Release Date.
$FCFS$	First Come First Served.

FDS	Fault Diagnosis System.
FIS	Finite Intermediate Storage.
GA	Genetic Algorithms.
GBD	Generalized Benders Decomposition.
GDP	Gross Domestic Product.
GED	Gross Error Detection.
IOF	Instant Objective Function.
LP	Linear Programming.
LPT	Longest Processing Time.
$MILP$	Mixed Integer Linear Programming.
$MINLP$	Mixed Integer Non-Linear Programming.
mo	Multi-objective.
MOF	Mean Objective Function.
mp	Multi-parametric.
$MSPCA$	Multi-Scale Principal Component Analysis.
$m.u.$	Monetary units.
NIS	Non-Intermediate Storage.
NLP	Non-Linear Programming.
OA	Outer Approximation.
PCA	Principal Component Analysis.
PID	Proportional Integral Derivative.
PSE	Process System Engineering.
$RBFN$	Radial Basis Function Networks.
RTE	Real Time Evolution.
RTN	Resource-Task Network.
RTO	Real Time Optimization.
SA	Simulated Annealing.
SAA	Sample Average Approximation.
SQP	Successive Quadratic Programming.
SPE	Squared Prediction Error.
SPT	Shortest Processing Time.
$SRTE$	Supervised Real Time Evolution.

STN	State-Task Network.
$TAHPN$	Timed Arc Hybrid Petri Net.
TS	Tabu Search.
UIS	Unlimited Intermediate Storage.
ZW	Zero Wait.

Subscripts

e, e'	Scenario.
$DTOP$	Time duration in the flexible recipe model.
DPS	Reaction yield in the flexible recipe model.
$DTEMP$	Temperature in the flexible recipe model.
$DKOH$	Amount of KOH in the flexible recipe model.
$DFOR$	Amount of Formaldehyde in the flexible recipe model.
f, f'	Flexible recipe item.
i, i'	Batch or production campaign (semicontinuous processes).
j, j'	Production line.
k, k'	Moveable vessel.
p, p'	Product.
s, s'	Stage or states (semicontinuous processes).
p, i, s	Production task.
t, t'	Storage tank.

Sets

E	Scenarios.
FL_{ps}	Stages s of product p with a flexible recipe.
FP_{ps}	Recipe items of a linear recipe model for stage s of product p.
I	Campaigns.
I_p	Batches of product p.
I_s^-	Campaigns that consume state s.
I_s^+	Campaigns that supply state s.

J_i	Available production lines for campaign i.
J_s	Available production lines for manufacturing state s.
K_p	Suitable moveable vessels for product p.
P	Products.
S	States.
S^I	Intermediate states.
S^P	Final states.
S_p	Stages for producing product p.
T_I	Available tanks to store state from campaign i.
T_S	Available tanks to store state s.
T_u	Available storage tanks after unit u.
T^{exec}	Already executed tasks.
T^{nda}	Non directly affected tasks by the expected event.
T^{da}	Directly affected tasks by an unexpected event.
T^{new}	Tasks of new orders to be inserted into the schedule in progress.
T^{fix}	Tasks following a fixed recipe.
T^{flex}	Tasks following a flexible recipe.
T^{maint}	Maintenance tasks.
U_{ps}	Set of available units for processing product p at stage s.

Parameters

ρ_{is}	Amount of state s required per unit size of supplying campaign i.
bsz_p	Batch size of every batch of product p.
d_s	Minimum demand for state s.
$dcost_{psf}$	Unitary deviation cost for recipe item f.
dd_{pi}	Due date for order i of product p.
dem_{pe}	Demand of product p for scenario e.
$demdev_p$	Demand deviation for product p.
$ecost$	Weighting coefficient for earliness cost penalty.
$fplb_{psf}$	Maximum negative deviation for recipe item f.
$fpub_{psf}$	Maximum positive deviation for recipe item f.

H	Time horizon.
itt_{pu}	Transfer time of product p for the first processing stage.
l_{sj}^{min}	Minimum allowed length campaign at production line j producing state s.
$lfmod_{psf}$	Model coefficient for recipe item f.
M	A very large number.
M_1	Equals to $H + max\{uch_{ii'j}\}$.
M_2	A very large number.
M_3	Equals to $H + max\{tch_{ii't}\}$.
mpc_p	Maximum plant capacity for producing product p.
npt_{psu}	Nominal processing time.
pc_p	Production costs of a batch of product p.
$pinv_p$	Inventory cost for product p.
$prob_e$	Associated probability for scenario e.
pt_{psu}	Processing time.
r_{sj}^{min}	Minimum production rate at line j generating state s.
r_{sj}^{max}	Maximum production rate at line j generating state s.
ro_i	Release time of processing campaign i.
ro_{pis}	Release time of a task.
ru_j	Ready time of production line j.
ru_u	Ready time of unit u.
s_{pi}^f	First processing stage for batch i of product p.
s_{pi}^l	Last processing stage for batch i of product p.
sp_p	Selling price of every batch of product p.
$tch_{ii't}$	Changeover time between campaigns i and i' at storage tank t.
$tcost$	Weighting coefficient for tardiness cost penalty.
tt_{pu}	Transfer time of product p from unit u.
$uch_{ii'j}$	Changeover time between campaigns i and i' at production line j.
$uch_{pp'u}$	Product-dependent changeover time for unit u.
ust_{pu}	Setup time for unit u for processing product p.
udc_p	Unsatisfied demand cost for product p per unit time.
v_t	Volumetric capacity of storage tank t.

Continuous variables

δ_{pisf}	Deviation of the flexible parameter f in task (p, i, s).
CT_i	Completion time of a storage task receiving material from i.
DC_{pis}	Cost of a task due to the deviation from nominal recipe parameters.
$E[PFS]$	Expect profit for a set of scenarios.
EA_{pi}	Earliness of batch i of product p.
$F_{ii'}$	Amount of material supplied by i and consumed by i'.
IN_p	Inventory of product p.
IN_p^{ini}	Initial inventory of product p.
IT_i	Starting time of stage task receiving material from i.
L_{ij}	Length of campaign i in production line j.
MK	Makespan.
PFS_e	Deterministic profit for scenario e.
Q_i	Overall production of campaign i.
QP_p	Produced amount of product p.
$SALES_{pe}$	Sales of product p in scenario e.
TA_{pi}	Tardiness of batch i of product p.
Tf_i	Completion time of campaign i.
Tf_{pis}	Completion time of task (p, i, s).
Ts_{pis}	Starting time of task (p, i, s)..
$TCOST$	Weighted objective function.
Tw_{pis}	Waiting time.
$V_{ii'}$	Accumulated material consumed by i' at the completion time of its supplying campaign i.
WC	Worst case profit.

Binary variables

AT_{pis}	Equals to 1 if material from task (p, i, s) is transferred to storage tank t, and 0 otherwise.
TO_{pis}	Equals to 1 if task (p,i,s) is a tardy order, and 0 otherwise.
$U_{ii'}$	Equals to 1 if campaign i supplies material to i', and 0 otherwise.
W_{it}	Equals to 1 if material from campaign i is sent to storage tank t, and 0 otherwise.
$X_{ii'}$	Equals to 1 if campaign i is run or stored before i', and 0 otherwise.
$X_{pisp'i's'}$	Equals to 1 if task (p, i, s) is processed before another task (p', i', s'), and 0 otherwise.
Y_{ij}	Equals to 1 if campaign i is run in production line j, and 0 otherwise.
Y_{pisu}	Equals to 1 if task (p, i, s) is allocated to equipment unit u, and 0 otherwise.
$Z_{ii'}$	Equals to 1 if campaign i supplying material to i' starts after i has finished, and 0 otherwise.
Z_{pik}	Equals to 1 if batch (p, i) is assigned to moveable vessel k, and 0 otherwise.

Bibliography

A. Aseeri and M. Bagajewicz. New measures and procedures to manage financial risk with applications to the planning of gas commercialization in asia. *Comput. Chem. Eng.*, 28:2791– 2821, 2004.

K.R. Baker. *Introduction to Sequencing and Scheduling.* New York: Wiley and Sons, 1974.

B.R. Bakshi. Multiscale PCA with application to multivariate statistical process monitoring. *AIChE J.*, 44:1596–1610, 1998.

J. Balasubramanian and I.E. Grossmann. A novel branch and bound algorithm for scheduling flowshop plants with uncertain processing times. *Comput. Chem. Eng.*, 26:41–57, 2002.

J. Balasubramanian and I.E. Grossmann. Scheduling optimization under uncertainty - an alternative approach. *Comput. Chem. Eng.*, 27:469–490, 2003.

J. Balasubramanian and I.E. Grossmann. Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty. *Ind. Eng. Chem. Res.*, 43:3695–3713, 2004.

V. Bansal, J.D. Perkins, and E.N. Pistikopoulos. Flexibility analysis and design using parametric programming framework. *AIChE J.*, 48:2851–2868, 2002.

A.F. Barbaro and M.J. Bagajewicz. Managing financial risk in planning under uncertainty. *AIChE J.*, 50:963–989, 2004.

M.H. Bassett, J.F. Pekny, and G.V. Reklaitis. Decomposition techniques for the solution of large-scale scheduling problems. *AIChE J.*, 42:3373–3387, 1996.

M.H. Bassett, J.F. Pekny, and G.V. Reklaitis. Using detailed scheduling to obtain realistic operating policies for a batch processing facility. *Ind. Eng. Chem. Res.*, 36:1717–1726, 1997.

M. Baudin. *Manufacturing Systems Analysis with Applications to Production Scheduling.* Englewood Cliffs: Yourdon Press, 1998.

R.M. Bethea and R.R. Rhinehart. *Applied Engineering Statistics.* New York: Marcel Dekker, 1991.

L.T. Biegler and I.E. Grossmann. Retrospective on optimization. *Comput. Chem. Eng.*, 28:1169–1192, 2004.

L.T. Biegler, I.E. Grossmann, and A.W. Westerberg. *Systematic Methods of Chemical Process Design.* New Jersey: Prentice-Hall, 1997.

J.K. Bok and S. Park. Continuous-time modeling for short-term scheduling of multipurpose pipeless plants. *Ind. Eng. Chem. Res.*, 37:3652–3659, 1998.

A. Bonfill, M. Bagajewicz, A. Espuña, and L. Puigjaner. Risk management in scheduling of batch plants under uncertain market demand. *Ind. Eng. Chem. Res.*, 43:741–750, 2004.

A. Bonfill, A. Espuña, and L. Puigjaner. Addressing robustness in scheduling batch processes with uncertain operation times. *Ind. Eng. Chem. Res.*, 44: 1524–1534, 2005.

H. Britt and R. Luecke. The estimation of parameters in nonlinear, implicit models. *Technometrics*, 15:233–247, 1973.

P. Castro, A.P.F.D. Barbosa-Povoa, and H. Matos. An improved rtn continuous-time formulation for the short-term scheduling of multipurpose batch plants. *Ind. Eng. Chem. Res.*, 40:2059–2068, 2001.

P. Castro, A.P.F.D. Barbosa-Povoa, H. Matos, and A.Q. Novais. Simple continuous-time formulation for short-term scheduling of batch and continuous processes. *Ind. Eng. Chem. Res.*, 43:105–118, 2004.

P.M. Castro and I.E. Grossmann. New continuous-time milp model for the short-term scheduling of multistage batch plants. *Ind. Eng. Chem. Res.*, 44: 9175–9190, 2005.

L. Cavin, U. Fischer, F. Glover, and K. Hungerbühler. Multi-objective process design in multi-purpose batch plants using a tabu search optimization algorithm. *Comput. Chem. Eng.*, 28:459–478, 2004.

J. Cerdá, P. Henning, and I.E. Grossmann. A mixed integer linear programming model for short-term scheduling of single-stage multiproduct batch plants with parallel lines. *Ind. Eng. Chem. Res.*, 36:1695–1707, 1997.

X. Chen, R. Pike, T. Hertwig, and J. Hopper. Optimal implementation of on-line optimization. *Comput. Chem. Eng.*, 22:435–442, 1998.

T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. Cambridge: MIT Press, 1997.

B.J. Cott and S. Macchietto. Minimizing the effects of batch process variability using on-line schedule modification. *Comput. Chem. Eng.*, 13:105–113, 1989.

C.M. Crowe. Data reconciliation - progress and challenges. *J. Process Control*, 6:89–98, 1996.

C.R. Cutler and R.T. Perry. Real time optimization with multivariable control is required to maximize profits. *Comput. Chem. Eng.*, 7:663–667, 1983.

I.T. Dedopoulos and N. Shah. Optimal short-term scheduling of maintenance and production for multipurpose plants. *Ind. Eng. Chem. Res.*, 34:192–201, 1995.

V. Dua and E.N. Pistikopoulos. A parametric mixed-integer optimization algorithm for multiobjective engineering problems involving discrete decisions. *Annals of Operations Research*, 99:123–139, 2000.

G.D. Eppen, R.K. Martin, and L. Schrage. A scenario approach to capacity planning. *Operations Research*, 37:517–527, 1989.

S. Ferrer-Nadal, T. Holczinger, C.A. Méndez, F. Friedler, and L. Puigjaner. Rigorous scheduling resolution of complex multipurpose batch plants: S-graph vs. milp. In *Computer Aided Chemical Engineering*, pages 2033–2038. Elsevier, 2006.

S. Ferrer-Nadal, C.A. Méndez, M. Graells, and L. Puigjaner. Optimal reactive scheduling of manufacturing plants with flexible batch recipes. *Ind. Eng. Chem. Res.*, 46:6273–6283, 2007.

C.A. Floudas and X. Lin. Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review. *Comput. Chem. Eng.*, 28: 2109–2129, 2004.

C.A. Floudas, Z.H. Gumu, and M.R. Ierapetritou. Global optimization in design under uncertainty: feasibility test and feasibility index problems. *Ind. Eng. Chem. Res.*, 40:4267–4282, 2001.

S. French. *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop.* England: Ellis Horwood Limited, 1974.

C.E. Garcia and M. Morari. Optimal operation of integrated processing systems. part i: Open-loop online optimizing control. *AIChE J.*, 27:960–968, 1981.

N.F. Giannelos and M.C. Georgiadis. A novel event-driven formulation for short-term scheduling of multipurpose continuous processes. *Ind. Eng. Chem. Res.*, 41:2431 –2439, 2002.

R. Gonzalez and M.J. Realff. Operation of pipeless batch plants - i. milp schedules. *Comput. Chem. Eng.*, 22:841–855, 1998a.

R. Gonzalez and M.J. Realff. Operation of pipeless batch plants - ii. vessel dispatch rules. *Comput. Chem. Eng.*, 22:857–866, 1998b.

I.E. Grossmann. Challenges in the new millennium: product discovery and design, enterprise and supply chain optimization, global life cycle assessment. *Comput. Chem. Eng.*, 29:29–39, 2004.

I.E. Grossmann and C.A. Floudas. Active constraints strategy for flexibility analysis in chemical processes. *Comput. Chem. Eng.*, 11:675–693, 1987.

J.H. Ha, H.K. Chang, E.S. Lee, I.B. Lee, B.S. Lee, and G. Yi. Intermediate storage tank operation strategies in the production scheduling of multi-product batch processes. *Comput. Chem. Eng.*, 24:1633–1640, 2000.

K.P. Halemane and I.E. Grossmann. Optimal process design under uncertainty. *AIChE J.*, 29:836–846, 1983.

S.P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22:297–309, 1977.

I. Harjunkoski and I.E. Grossmann. Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. *Comput. Chem. Eng.*, 26:1533–1552, 2002.

S. Hasebe, I. Hashimoto, and A. Ishikawa. General reordering algorithm for scheduling of batch process. *Journal of Chemical Engineering of Japan*, 24: 483–489, 1991.

Y.H. He and C.W. Hui. Automatic rule combination approach for single-stage process scheduling problems. *AIChE J.*, 53:2026–2047, 2007.

P.V. Hentenryck. *Constraint satisfaction in logic programming.* Cambridge: MIT Press, 1989.

S.K. Heo, K.H. Lee, H.K. Lee, I.B. Lee, and J.H. Park. A new algorithm for cyclic scheduling and design of multipurpose batch plants. *Ind. Eng. Chem. Res.*, 42:836–846, 2003.

T. Holcomb and M. Morari. Local training of radial basis function networks: Towards solving the hidden unit problem. In *American Control Conference*, pages 2331–2336, 1991.

S.J. Honkomp, L. Mockus, and G.V. Reklaitis. A framework for schedule evaluation with processing uncertainty. *Comput. Chem. Eng.*, 23:595–609, 1999.

J. Hooker. *Logic based methods for optimization: Combining optimization and constraint satisfaction.* New York: John Wiley and Sons, Inc, 2000.

W. Huang and P.W.H. Chung. Scheduling of pipeless batch plants using constraint satisfaction techniques. *Comput. Chem. Eng.*, 24:377–383, 2000.

W. Huang and P.W.H. Chung. Integrating routing and scheduling for pipeless plants in different layouts. *Comput. Chem. Eng.*, 29:1069–1081, 2005.

A. Huercio, A. Espuña, and L. Puigjaner. Incorporating on-line scheduling strategies in integrated batch production control. *Comput. Chem. Eng.*, 19: S609–S614, 1995.

A. Hugo and E.N. Pistikopoulos. Environmentally conscious long-range planning and design of supply chain networks. *Journal of Cleaner Production*, 13:1471–1491, 2005.

C.W. Hui and A. Gupta. A bi-index continuous time milp model for short-term scheduling of single-stage multi-product batch plants with parallel line. *Ind. Eng. Chem. Res.*, 40:5960–5967, 2001.

M.G. Ierapetritou and C.A. Floudas. Effective continuous-time formulation for short-term scheduling: 1. multipurpose batch processes. *Ind. Eng. Chem. Res.*, 37:4341–4359, 1998a.

M.G. Ierapetritou and C.A. Floudas. Effective continuous-time formulation for short-term scheduling. 2. continuous and semicontinuous processes. *Ind. Eng. Chem. Res.*, 37:4360–4374, 1998b.

M.G. Ierapetritou and E.N. Pistikopoulos. Global optimization for stochastic planning, scheduling and design problems. *Global optimization in engineering design*, pages 231–287, 1996.

V. Jain and I.E. Grossmann. Algorithms for hybrid milp/cp models for a class of optimization problems. *INFORMS Journal on Computing*, 13:258–276, 2001.

S. Janak, X. Lin, and C.A. Floudas. Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: Resource constraints and mixed storage policies. *Ind. Eng. Chem. Res.*, 43:2516 –2533, 2004.

S. Janak, X. Lin, and C.A. Floudas. A new robust optimization approach for scheduling under uncertainty. ii. uncertainty with known probability distribution. *Comput. Chem. Eng.*, 43:171–195, 2007.

K.B. Kanakamedala, G.V. Reklaitis, and V. Venkatasubramanian. Reactive schedule modification in multipurpose batch chemical-plants. *Ind. Eng. Chem. Res.*, 33:77–90, 1994.

I.A. Karimi and C.M. McDonald. Planning and scheduling of parallel semicontinuous processes: 2. short-term scheduling. *Ind. Eng. Chem. Res.*, 36: 2701–2714, 1997.

K.J. Keesman. Application of flexible recipes for model building, batch process optimization and control. *AIChE J.*, 28:581–588, 1993.

M.S. Kim, J.H. Jung, and I.B. Lee. Optimal scheduling of multiproduct batch processes for various inter-stage storage policies. *Ind. Eng. Chem. Res.*, 35: 4058–4066, 1996.

S.B. Kim, H. Lee, I. Lee, E.S. Lee, and B. Lee. Scheduling of non-sequential multipurpose batch process under finite intermediate storage policy. *Comput. Chem. Eng.*, 24:1603–1610, 2000.

E. Kondili, C.C. Pantelides, and R.W.H. Sangent. A general algorithm for short-term scheduling of batch operations. i. milp formulation. *Comput. Chem. Eng.*, 17:211–227, 1993a.

E. Kondili, N. Shah, and C.C. Pantelides. Production planning for the rational use of energy in multiproduct continuous plants. *Comput. Chem. Eng.*, 17: S123–S128, 1993b.

T. Kourti and J.F. MacGregor. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28:3–21, 1995.

H. Ku and I. Karimi. Scheduling in serial multi-product batch processes with finite interstage storage: A mixed integer linear program formulation. *Ind. Eng. Chem. Res.*, 27:1840–1848, 1988.

H.M. Ku and I. Karimi. An evaluation of simulated annealing for batch process scheduling. *Ind. Eng. Chem. Res.*, 30:163–169, 1991.

Y.G. Lee and M.F. Malone. Batch process schedule optimization under parameter volatility. *International Journal of Production Research*, 39:603–623, 2001.

J.A. Leonard and M.A. Kramer. Diagnosing dynamic faults using modular neural nets. In *IEEE Expert 8 (2)*, pages 44–53, 1993.

M. Lim and I.A. Karimi. Resource-constrained scheduling of parallel production lines using asynchronous slots. *Ind. Eng. Chem. Res.*, 42:6832–6842, 2003.

X. Lin, C.A. Floudas, S. Modi, and N.M. Juhasz. Continuous-time optimization approach for medium-range production scheduling of a multiproduct batch plant. *Ind. Eng. Chem. Res.*, 41:3884–3906, 2002.

X. Lin, S. Janak, and C.A. Floudas. New robust optimization approach for scheduling under uncertainty: I. bounded uncertainty. *Comput. Chem. Eng.*, 28:1069–1085, 2004.

R. Liu and C. Mcgreavy. A framework for operation strategy of pipeless plants. *Comput. Chem. Eng.*, 20:S1161–S1166, 1996.

A. Malcolm, J. Polan, L. Zhang, B.A. Ogunnaike, and AA. Linninger. Integrating systems design and control using dynamic flexibility analysis. *AIChE J.*, 53:2048–2061, 2007.

C.T. Maravelias and I.E. Grossmann. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Ind. Eng. Chem. Res.*, 42:3056–3074, 2003.

C.T. Maravelias and I.E. Grossmann. A hybrid milp/cp decomposition approach for the continuous time scheduling of multipurpose batch plants. *Ind. Eng. Chem. Res.*, 28:1921–1949, 2004.

T. Marlin and A.N. Hrymak. Real-time operations optimization of continuous processes. In *AIChE Symposium Series Fifth International Conference on Chemical Process Control*, volume 93, pages 156–164, 1997.

R.C. McFarlane and D.W. Bacon. Adaptative optimizing control of multivariable constrained chemical processes. *Ind. Eng. Chem. Res.*, 28:1828–1834, 1989.

I.P. Miletic and T.E. Marlin. On-line statistical results analysis in real-time operations optimization. *Ind. Eng. Chem. Res.*, 37:3670–3684, 1998.

C.A. Méndez and J. Cerdá. An efficient milp continuous-time formulation for short-term scheduling of multiproduct continuous facilities. *Comput. Chem. Eng.*, 26:687–695, 2002.

C.A. Méndez and J. Cerdá. An milp continuous-time framework for short-term scheduling of multipurpose batch processes under different operation strategies. *Optim. Eng.*, 4:7–22, 2003a.

C.A. Méndez and J. Cerdá. Dynamic scheduling in multiproduct batch plants. *Comput. Chem. Eng.*, 27:1247–1259, 2003b.

C.A. Méndez and J. Cerdá. Milp framework for batch reactive scheduling with limited discrete resources. *Comput. Chem. Eng.*, 28:1059–1068, 2004.

C.A. Méndez, G.P. Henning, and J. Cerdá. An milp continuous time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Comput. Chem. Eng.*, 25:701–711, 2001.

C.A. Méndez, J. Cerdá, I.E. Grossmann, I. Harjunkoski, and M. Fahl. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chem. Eng.*, 30:913–946, 2006.

L. Mockus and G.V. Reklaitis. Continuous time representation approach to batch and continuous process scheduling. 1. minlp formulation. *Ind. Eng. Chem. Res.*, 38:197–203, 1999.

S.A. Munawar, M. Bhushan, R.D. Gudi, and A.M. Belliappa. Cyclic scheduling of continuous multi-product plants in a hybrid flowshop facility. *Ind. Eng. Chem. Res.*, 38:5861–5882, 2003.

T. Niwa. Batch plants become a pipeless dream. *Chemical Engineering*, 100: 102, 1993.

S. Orçun, K. Altinel, and Ö. Hortacsu. Scheduling of batch processes with operational uncertainties. *Comput. Chem. Eng.*, 20:S1191–S1196, 1996.

G.M. Ostrovsky and Y.M. Volin. Flexibility analysis: Taking into account fullness and accuracy of plant data. *AIChE J.*, 52:3173–3188, 2006.

G.M. Ostrovsky, Y.M. Volin, and M.M. Senyavin. An approach to solving a two-stage optimization problem under uncertainty. *Comput. Chem. Eng.*, 21:317–325, 1997.

C.C. Pantelides. Unified frameworks for the optimal process planning and scheduling. In *Second International Conference on Foundations of Computer-Aided Process Operations*, pages 253–274, 1994.

C.C. Pantelides, M. Realff, and N. Shah. Short-term scheduling of pipeless batch plants. *Chem. Eng. Res. Des*, 73:431–444, 1995.

S.S. Panwalkar and W.A. Iskander. Survey of scheduling rules. *Operations Research*, 25:45–61, 1997.

L.G. Papageorgiou and C.C. Pantelides. Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 1. mathematical formulation. *Ind. Eng. Chem. Res.*, 35:488–509, 1996.

K. Papalexandri and T. Dimkou. A parametric mixed-integer optimization algorithm for multiobjective engineering problems involving discrete decisions. *Ind. Eng. Chem. Res.*, 37:1866–1882, 1998.

D.I. Patsiatzis, G. Xu, and L.G. Papageorgiou. Layout aspects of pipeless batch plants. *Ind. Eng. Chem. Res.*, 44:5672–5679, 2005.

J.F. Pekny. Algorithm architectures to support large-scale process systems engineering applications involving combinatorics, uncertainty, and risk management. *Comput. Chem. Eng.*, 26:239–267, 2002.

J.F. Pekny and G.V. Reklaitis. Towards the convergence of theory and practice: A technology guide for scheduling/planning methodology. In *Proceedings of*

the third international conference on foundations of computer-aided process operations, pages 75–90, 1998.

S.B. Petkov and C.D. Maranas. Multiperiod planning and scheduling of multipurpose batch plants under demand uncertainty. *Ind. Eng. Chem. Res.*, 36: 4864–4881, 1997.

M. Pinedo. *Scheduling theory, alogrithms, and systems.* Englewood Cliffs: Prentice Hall, 1995.

J. Pinto and I. Grossmann. Optimal cyclic scheduling of multistage continuous multiproduct plants. *Comput. Chem. Eng.*, 18:797–816, 1994.

J.M. Pinto and I.E. Grossmann. A continuous time milp model for short term scheduling of batch plants with pre-ordering constraints. *Comput. Chem. Eng.*, 20:1197–1202, 1996.

E.N. Pistikopoulos and I.E. Grossmann. Optimal retrofit design for improving. flexibility in nonlinear systems. i. fixed degree of flexibility. *Comput. Chem. Eng.*, 13:1003–1016, 1989a.

E.N. Pistikopoulos and I.E. Grossmann. Optimal retrofit design for improving. flexibility in nonlinear systems. ii. optimal level of flexibility. *Comput. Chem. Eng.*, 13:1087–1096, 1989b.

E.N. Pistikopoulos and M.G. Ierapetritou. Novel approach for optimal process design under uncertainty. *Comput. Chem. Eng.*, 19:1089–1110, 1995.

A. Ponsich, C. Azzaro-Pantel, S. Domenech, and L. Pibouleau L. Mixed-integer nonlinear programming optimization strategies for batch plant design problems. *Ind. Eng. Chem. Res.*, 46:854–863, 2007.

M.J. Realff, N. Shah, and C.C. Pantelides. Simultaneous design, layout and scheduling of pipeless batch plants. *Comput. Chem. Eng.*, 20:869–883, 1996.

G.V. Reklaitis. *Overview of scheduling and planning of batch process operations.* NATO advanced study institute - batch process systems engineering. Turkey, 1992.

J.E. Rijnsdorp. *Integrated Process Control and Automation.* Amsterdam: Elsevier, 1991.

M.T.M. Rodrigues, L. Gimeno, C.A.S. Passos, and M.D. Campos. Reactive scheduling approach for multipurpose chemical batch plants. *Comput. Chem. Eng.*, 20:S1215–S1220, 1996.

J. Romero, A. Espuña, F. Friedler, and L. Puigjaner. A new framework for batch process optimization using the flexible recipe. *Ind. Eng. Chem. Res.*, 42:370–379, 2003.

J. Romero, L. Puigjaner, T. Holczinger, and F. Friedler. Scheduling intermediate storage multipurpose batch plants using the s-graph. *AIChE J.*, 50: 403–417, 2004.

J. Roslöf, I. Harjunkoski, J. Bjorkqvist, S. Karlsson, and T. Westerlund. An milp-based reordering algorithm for complex industrial scheduling and rescheduling. *Comput. Chem. Eng.*, 25:821–828, 2001.

D. Ruiz, J. Cantón, N.J. Mara, A. Espuna, and L. Puigjaner. On-line fault diagnosis system support for reactive scheduling in multipurpose batch chemical plants. *Comput. Chem. Eng.*, 25:829–837, 2001.

W.G.M. Rutten and J.W.M. Bertrand. Balancing stocks, flexible recipe costs and high service level requirements in a batch process industry: A study of a small scale model. *Eur J Oper Res.*, 110:626–642, 1998.

S.A. Sadrieh, M. Ghaeli, P.A. Bahri, and P.L. Leed. An integrated petri net and ga based approach for scheduling of hybrid plants. *Computers in Industry*, 15:519–530, 2007.

N.V. Sahinidis and I.E. Grossmann. Minlp model for cyclic multiproduct scheduling on continuous parallel lines. *Comput. Chem. Eng.*, 15:85–103, 1991.

N.J. Samsatli, L.G. Papageorgiou, and N. Shah. Robustness metrics for dynamic optimization models under parameter uncertainty. *AIChE J.*, 44: 1993–2006, 1998.

E. Sanmartí, A. Huercio, A. Espuña, and L. Puigjaner. A combined scheduling/reactive scheduling strategy to minimize the effect of process operations uncertainty in batch plants. *Comput. Chem. Eng.*, 20:1263–1268, 1996.

E. Sanmartí, A. Espuña, and L. Puigjaner. Batch production and preventive maintenance scheduling under equipment failure uncertainty. *Comput. Chem. Eng.*, 21:1157–1168, 1997.

E. Sanmartí, T. Holczinger, L. Puigjaner, and F. Friedler. Combinatorial framework for effective scheduling of multipurpose batch plants. *AIChE J.*, 48:2557–2570, 2002.

R. Sargent. Process systems engineering: A retrospective view with questions to the future. *Comput. Chem. Eng.*, 29:1237–1241, 2005.

R. Sargent. The development of sequential quadratic programming. *Large Scale Optimization with Applications, IMA Volumes in Mathematics and Applications*, 93, 1997.

G. Schilling and C.C. Pantelides. A simple continuous-time process scheduling formulation and a novel solution algorithm. *Comput. Chem. Eng.*, 20:S1221–S1226, 1996.

D. Sel, N. Hvala, S. Strmcnik, S. Milanic, and B. Suk-Lubej. Experimental testing of flexible recipe control based on a hybrid model. *Control Eng. Pract.*, 7:1191–1208, 1999.

S. Sequeira, M. Graells, and L. Puigjaner. Real-time evolution for on-line optimization of continuous processes. *Ind. Eng. Chem. Res.*, 41:1815–1825, 2002.

S. Sequeira, M. Herrera, M. Graells, and L. Puigjaner. On-line process optimization: Parameter tuning for the real time evolution (RTE) approach. *Comput. Chem. Eng.*, 28:661–672, 2004.

N. Shah. Single and multisite planning and scheduling: Current status and future challenges. In *Third International Conference on Foundations of Computer-Aided Process Operations*, pages 75–90, 1998.

M.A. Shaik and C.A. Floudas. Improved unit-specific event-based continuous-time model for short-term scheduling of continuous processes: Rigorous treatment of storage requirements. *Ind. Eng. Chem. Res.*, 46:1764 –1779, 2007.

R.E. Steuer. *Multiple criteria optimization: theory, computation and application.* New York: John Wiley and Sons, 1986.

D.A. Straub and I.E. Grossmann. Design optimization of stochastic flexibility. *Comput. Chem. Eng.*, 24:339–354, 1993.

M. Suh and T. Lee. Robust optimization method for the economic term in chemical process design and planning. *Ind. Eng. Chem. Res.*, 40:5950–5959, 2001.

A. Sundaramoorthy and I.A. Karimi. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chem. Eng. Sci.*, 60:2679–2702, 2005.

R.E. Swaney and I.E. Grossmann. An index for operational flexibility in chemical process design. *AIChE J.*, 31:621–630, 1985.

Z. Verwater-Lukszo. A practical approach to recipe improvement and optimization in the batch processing industry. *Comput. Ind.*, 36:279–300, 1998.

B. Verweij, S. Ahmed, A.J. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: A computational study. *Comput. Appl. Opt.*, 24:289–333, 2001.

J.P. Vin and M.G. Ierapetritou. A new approach for efficient rescheduling of multiproduct batch plants. *Ind. Eng. Chem. Res.*, 39:4228–4238, 2000.

J.P. Vin and M.G. Ierapetritou. Robust short-term scheduling of multiproduct batch plants under demand uncertainty. *Ind. Eng. Chem. Res.*, 40:4543–4554, 2001.

Y. Yang, R. Ten, and L. Jao. A study of gross error detection and data reconciliation in process industries. *Comput. Chem. Eng.*, 19:217–222, 1995.

W.S. Yip and T.E. Marlin. The effect of model fidelity on real-time optimization performance. *Comput. Chem. Eng.*, 28:267–280, 2004.

L.J. Zeballos and G.P. Henning. A cp approach to the scheduling of resource-constrained multiproduct continuous facilities. *Lat. Am. Appl. Res.*, 36: 205–212, 2006.

M.G. Zentner, J.F. Pekny, and G.V. Reklaitis. Practical considerations in using model-based optimization for the scheduling and planning of batch/semicontinuous processes. *Journal of Process Control*, 4:259–280, 1994.

X. Zhang and R.W.H. Sargent. Optimal operation of mixed production facilities-a general formulation and some approaches for the solution. *Comput. Chem. Eng.*, 20:897–904, 1996.

J. Zhao, B. Chen, and J. Shen. Multidimensional non-orthogonal wavelet-sigmoid basis function neural network for dynamic process fault diagnosis. *Comput. Chem. Eng.*, 23:83–92, 1998.

Bibliography

Appendix A

Publications

Research work developed in the scope of this thesis has resulted in several publications either articles in scientific journals, articles in conference proceedings and communications in congresses. Besides, there has been the opportunity to participate in different research projects. All these contributions are detailed in this appendix.

A.1 Journal Articles

Ferrer-Nadal, S., Yélamos-Ruiz, I., Graells, M. and Puigjaner, L. (2007). An integrated framework for on-line supervised optimization. *Computers and Chemical Engineering*, 31 (5-6), 401-409. ISSN: 0098-1354.

Yélamos, I., **Ferrer, S.**, Graells, M. and Puigjaner, L. (2007). Sistema de optimización en línea y diagnosis de fallos para procesos químicos. *Información tecnológica*, 18 (2), 87-92. ISSN: 0718-0764.

Ferrer-Nadal, S., Méndez, C. A., Graells, M. and Puigjaner, L. (2007). Optimal reactive scheduling of manufacturing plants with flexible batch recipes. *Industrial and Engineering Chemistry Research*, 46 (19), 6273-6283. ISSN: 0888-5885.

Ferrer-Nadal, S., Puigjaner, L. and Guillén-Gosálbez, G. (2008). Managing risk through a flexible recipe framework. *AIChE Journal*, 54 (3), 728-740. ISSN: 0001-1541.

A.1.1 Manuscripts submitted

Ferrer-Nadal, S., Capón-García, E. and Puigjaner, L. (2008). Transfer times in batch scheduling: a critical modeling issue. Submitted to *Industrial and Engineering Chemistry Research*.

Capón-García, E., Ferrer-Nadal, S., Graells, M. and Puigjaner, L. (2008). An extended formulation for the flexible short-term scheduling of multi-product semicontinuous plants. Submitted to *Industrial and Engineering Chemistry Research*.

A.2 Publications in conference proceedings

Ferrer-Nadal, S., Yélamos-Ruiz, I., Graells, M. and Puigjaner, L. (2005). On-line fault diagnosis support for real time evolution applied to multi-component distillation. In 15th *European Symposium on Computer Aided Process Engineering* (Eds. L. Puigjaner and A. Espuña), Elsevier, 20B, 961-966. ISBN: 0-444-51991-2.

Ferrer-Nadal, S., Holczinger, T., Méndez, C. A., Friedler, F. and Puigjaner, L. (2006). Rigorous scheduling resolution of complex multipurpose batch plants: S-Graph vs. MILP. In 16th *European Symposium on Computer Aided Process Engineering and* 9th *International Symposium on Process Systems Engineering* (Eds. W. Marquardt and C. Pantelides), Elsevier, 22B, 2033-2038. ISBN: 978-0-444-52970-1.

Mitta, N., **Ferrer-Nadal, S.**, Lazovic, A. M., Perales, J. F., Velo, E. and Puigjaner, L. (2006). Modelling and simulation of a tyre gasification plant for synthesis gas production. In 16th *European Symposium on Computer Aided Process Engineering and* 9th *International Symposium on Process Systems Engineering* (Eds. W. Marquardt and C. Pantelides), Elsevier, 22B, 1771-1776. ISBN: 978-0-444-52970-1.

Ferrer-Nadal, S., Méndez, C. A., Graells, M. and Puigjaner, L. (2006). A mathematical programming approach including flexible recipes to batch operation rescheduling. In 16th *European Symposium on Computer Aided*

Process Engineering and 9th *International Symposium on Process Systems Engineering* (Eds. W. Marquardt and C. Pantelides), Elsevier, 22B, 1377-1382. ISBN: 978-0-444-52970-1.

Ferrer-Nadal, S., Méndez, C. A., Graells, M. and Puigjaner, L. (2007). A novel continuous-time MILP approach for short term scheduling of multipurpose pipeless batch plants. In 17th *European Symposium on Computer Aided Process Engineering* (Eds. V. Plesu and P.S. Agachi), Elsevier, 24, 595-600. ISBN: 978-0-444-53158-2.

Guillén-Gosálbez, G., **Ferrer-Nadal, S.** and Puigjaner, L. (2007). Exploiting the use of a flexible recipe framework to manage financial risk. In 17th *European Symposium on Computer Aided Process Engineering* (Eds. V. Plesu and P.S. Agachi), Elsevier, 24, 643-648. ISBN: 978-0-444-53158-2.

Ferrer-Nadal, S., Guillén-Gosálbez, G. and Puigjaner, L. (2007). A MILP decomposition approach for the risk management within a flexible recipe framework. In 6th *European Congress of Chemical Engineering* (Eds. R. Gani and K. Dam-Johansen), Norhaven Books, Vol. 1, 485-486. ISBN: 978-87-91435-56-0.

Graells, M., Yélamos, I., **Ferrer-Nadal, S.** and Pérez-Moya, M. (2007). Gestión de recursos docentes para la adaptación al espacio europeo de educación superior. In *15º Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas* (Eds. M. A. Martín Bravo and J. M. García Terán). Escuela Universitaria Politécnica de Valladolid, 1811-1817. ISBN: 978-84-690-7547-0.

A.3 Abstracts in conference proceedings

Méndez, C. A., **Ferrer-Nadal, S.**, Friedler, F. and Puigjaner, L. (2005). Scheduling of multipurpose batch plants with different storage policies: a comparative study between S-Graph and MILP methods. In *10º Congreso Mediterráneo de Ingeniería Química*, Sociedad Española de Química Industrial e Ingeniería Química (SEQUI), 576.

Ferrer-Nadal, S., Lazovic, A. M., Mitta, N., Perales, J. F., Velo, E. and Puigjaner, L. (2005). Clean fuel gas production using tyre gasification plant. In *10º Congreso Mediterráneo de Ingeniería Química*, Sociedad Española de Química Industrial e Ingeniería Química (SEQUI), 569.

Ferrer-Nadal, S., Méndez, C. A., Graells, M. and Puigjaner, L. (2005). An MILP-based framework for reactive scheduling of batch processes with flexible recipes. In *10º Congreso Mediterráneo de Ingeniería Química*, Sociedad Española de Química Industrial e Ingeniería Química (SEQUI), 559.

Ferrer-Nadal, S., López-Castro, J., Graells, M. and Puigjaner, L. (2005). An on-line approach for optimal maintenance of multistage continuous plants. In *10º Congreso Mediterráneo de Ingeniería Química*, Sociedad Española de Química Industrial e Ingeniería Química (SEQUI), 558.

Ferrer-Nadal, S., Méndez, C.A., Graells, M., Puigjaner, L. and Angelini, R. (2006). A new strategy for exploiting recipe flexibility in batch processes with decaying performance. In 17th *International Congress of Chemical and Process Engineering*, Process Engineering Publisher, Summaries 6, 1861. ISBN: 80-86059-45-6.

Pérez-Fortes, M., Bojarski, A., **Ferrer-Nadal, S.**, Kopanos, G., Mitta, N., Pinilla, C. A., Nougés, J.M., Velo, E. and Puigjaner, L. (2007) Enhanced model for integrated simulation of an entrained bed gasifier implemented as Aspen Hysys extension. In *2007 International Conference on Coal Science and Technology*, The IEA Clean Coal Centre, 118. ISBN: 92-9029-437-X.

A.4 Communications to congresses

Ferrer-Nadal, S., Yélamos-Ruiz, I., Graells, M. and Puigjaner, L. (2004). Application of real time evolution to continuous multi-component distillation column. AIChE Annual Meeting, Austin, USA.

Mitta, N., **Ferrer-Nadal, S.**, Perales, J. F., Velo, E. and Puigjaner, L. (2006). Conceptual design and modeling of entrained bed gasifier. AIChE Annual Meeting, San Francisco, USA.

Ferrer-Nadal, S., Guillén-Gosálbez, G. and Puigjaner, L. (2006). Hedging risk through the flexible recipe framework. AIChE Annual Meeting, San Francisco, USA.

Pérez-Fortes, M., Bojarski, A., **Ferrer-Nadal, S.**, Kopanos, G., Nougés, J.M., Velo, E. and Puigjaner, L. (2007). Integrated environment for

detailed modelling and evaluation of gasification processes for energy production. AIChE Annual Meeting, Salt Lake City, USA.

A.5 Manuscript submitted and accepted to congresses

Capón-García, E., **Ferrer-Nadal, S.** and Puigjaner, L. (2008). Uncovering the relevance of modeling transfer times in the short-term multipurpose batch plants scheduling. Accepted in *Foundations of Computer-Aided Process Operations*. Boston, USA.

Ferrer-Nadal, S., Capón-García, E., Graells, M. and Puigjaner, L. (2008). Flexible management for the short-term scheduling of multiproduct semi-continuous plants. Accepted in 18th *International Congress of Chemical and Process Engineering*. Prague, Czech Republic.

Pérez-Fortes, M., Bojarski, A., **Ferrer-Nadal, S.**, Kopanos, G., Nougés, J.M., Velo, E. and Puigjaner, L. (2008). Enhanced modelling and integrated simulation of gasification and purification gas units targeted to clean power production. Accepted in 18th *European Symposium on Computer Aided Process Engineering*, Lyon, France.

Pérez-Fortes, M., Bojarski, A., **Ferrer-Nadal, S.**, Kopanos, G., Nougés, J.M., Velo, E. and Puigjaner, L. (2008). Valorisation of biomass and wastes co-gasification with coal in an IGCC power plant. Accepted in 3rd *edition of World Bioenergy Conference*, Jönköping , Sweden.

Pérez-Fortes, M., Bojarski, A., **Ferrer-Nadal, S.**, Kopanos, G., Nougés, J.M., Velo, E. and Puigjaner, L. (2008). Valorisation of waste in a gasification plant to clean power production. Accepted in *2008 CTSI Clean Technology and Sustainable Industries Conference and Trade Show*, Boston, USA.

Pérez-Fortes, M., Bojarski, A., **Ferrer-Nadal, S.**, Kopanos, G., Nougés, J.M., Velo, E. and Puigjaner, L. (2008). Conceptual model and evaluation of generated power and emissions in an integrated gasification combined cycle power plant. Accepted in 18th *International Congress of Chemical and Process Engineering*, Prague, Czech Republic.

A.6 Participation in research projects

OCCASSION, *Desarrollo e Implementación de un Sistema de Gestión y Optimización de Cadena de Suministro Global en Tiempo Real*, supported by the *Ministerio de Educación y Ciencia* (DPI2002-00856), Spain, 2003-2005.

AGAPUTE, *Advanced gas purification technologies for co-gasification of coal, refinery by-products, biomass & waste, targeted to clean power produced from gas & steam turbine generator sets and fuel cells*, supported by the *European Community* (RFC-CR-04006), 2004-2008.

Appendix B

MILP formulations for batch scheduling

B.1 State-Task Network based continuous formulation (Maravelias and Grossmann, 2003)

Assignment constraints

These constraints ensure that each task i is allocated to a single equipment j. Binary variables denote if a task i starts (Ws_{in}) or finishes (Wf_{in}) at an event point n. Constraints B.1 and B.2 impose that for each event point n and equipment j, at most one task i can take place. Constraint B.3 obliges all started tasks to be finished. Equation B.4 is necessary to ensure that a task can only be started if all other previous tasks in the same equipment have finished. Through constraints B.5 and B.6, no task can finish at $t = 0$, and no task can start at the last event point.

$$\sum_{i \in I_j} Ws_{in} \leq 1 \qquad \forall j, n \tag{B.1}$$

$$\sum_{i \in I_j} Wf_{in} \leq 1 \qquad \forall j, n \tag{B.2}$$

$$\sum_n Ws_{in} = \sum_n Wf_{in} \qquad \forall i \tag{B.3}$$

$$\sum_{i \in I_j} \sum_{n' \leq n} (Ws_{in'} - Wf_{in'}) \leq 1 \qquad \forall j, n \tag{B.4}$$

$$Wf_{i0} = 0 \qquad \forall i \tag{B.5}$$

$$Ws_{in} = 0 \qquad \forall i, n = |N| \tag{B.6}$$

Timing constraints

In one hand, time event ($n = 1$) corresponds to zero time. On the other hand, finishing time corresponds to the last event point ($n = /N/$). In between, time points are ordered increasingly in time. These premises are enforced through equations B.7 to B.9.

$$T_{n=1} = 0 \tag{B.7}$$

$$T_{n=|N|} = MK \tag{B.8}$$

$$T_{n+1} \geq T_n \qquad \forall n < |N| \tag{B.9}$$

This formulation allows fixed and variable time durations of tasks (equation B.10).

$$D_{in} = \alpha_i Ws_{in} + \beta_i Bs_{in} \qquad \forall i, n \tag{B.10}$$

Variables Ts_{in} and Tf_{in} represent respectively starting and finishing time of a task i at time event n. Finishing time of a task starting at event point n is defined through constraints B.11 and B.12.

$$Tf_{in} \leq Ts_{in} + D_{in} + M\,(1 - Ws_{in}) \qquad \forall i, n \tag{B.11}$$

$$Tf_{in} \geq Ts_{in} + D_{in} - M\,(1 - Ws_{in}) \qquad \forall i, n \tag{B.12}$$

Constraints B.13 and B.9 define the finishing time of task i starting at event point n equal to the finishing time at event point $n-1$. Equation B.14 is aimed at reducing the search space by constraining the finishing time of a task i at event point n to be greater or equal to the duration of the task at event point $n-1$.

$$Tf_{in} - Tf_{in-1} \leq M \cdot Ws_{in} \qquad \forall i, n > 1 \tag{B.13}$$

$$Tf_{in} - Tf_{in-1} \geq D_{in} \qquad \forall i, n > 1 \tag{B.14}$$

Variable Ts_{in} can be removed from this formulation because starting times of tasks must fit with an event point (equation B.15).

$$Ts_{in} = T_n \qquad \forall i, n \tag{B.15}$$

Constraints B.16 and B.17 are formulated in order to match the finishing time of a task with its corresponding event point. In general, a task can finish at or before an event point n. A special case corresponds to tasks following a zero-wait time policy that are enforced to finish exactly at an event point n through constraint B.17.

$$Tf_{in-1} \leq T_n + M(1 - Wf_{in}) \qquad \forall i, n > 1 \tag{B.16}$$

$$Tf_{in-1} \geq T_n - M(1 - Wf_{in}) \qquad \forall i \in I^{ZW}, n > 1 \tag{B.17}$$

Batch size constraints

Batch size of a given task must lie between upper and lower limits. This is enforced for starting, ending and processing event points of tasks through equations B.18 to B.20.

$$B_i^{MIN} Ws_{in} \leq Bs_{in} \leq B_i^{MAX} Ws_{in} \qquad \forall i, n \tag{B.18}$$

$$B_i^{MIN} W f_{in} \leq B f_{in} \leq B_i^{MAX} W f_{in} \qquad \forall i, n \tag{B.19}$$

$$\begin{aligned} B_i^{MIN} \left(\sum_{n'<n} W s_{in'} - \sum_{n' \leq n} W f_{in'} \right) \leq B p_{in} \leq \\ \leq B_i^{MAX} \left(\sum_{n'<n} W s_{in'} - \sum_{n' \leq n} W f_{in'} \right) \quad \forall i, n \end{aligned} \tag{B.20}$$

Mass balances for batches are enforced by making the initial amount plus the produced amount of processing task i, at time event $n-1$, equal to the produced and final amount at event point n. This is accomplished by constraint B.21.

$$B s_{in-1} + B p_{in-1} = B p_{in} + B f_{in} \qquad \forall i, n \tag{B.21}$$

Produced and consumed amounts of state s by a processing task i at event point n are computed through constraints B.22 to B.25. Constraints B.22 and B.23 are aimed at input states of the processing tasks, whereas B.24 and B.25 constraint output states. Under no circumstances, processed tasks are allowed to exceed the maximum quantity allowed (B.23 and B.25).

$$B_{isn}^{I} = \rho_{is} B s_{in} \qquad \forall i, n, s \in SI_i \tag{B.22}$$

$$B_{isn}^{I} \leq \rho_{is} B_i^{MAX} W s_{in} \qquad \forall i, n, s \in SI_i \tag{B.23}$$

$$B_{isn}^{O} = \rho_{is} B f_{in} \qquad \forall i, n, s \in SO_i \tag{B.24}$$

$$B_{isn}^{O} \leq \rho_{is} W f_{in} B_i^{MAX} \qquad \forall i, n, s \in SO_i \tag{B.25}$$

Mass balance/storage constraints

Constraint B.26 imposes that available amount of material at time point n plus the amount sold (SS_{sn}) is equal to the available amount from previous period $n-1$ adjusted by the amount produced and consumed during the current

period.

$$S_{sn} + SS_{sn} = S_{sn-1} + \sum_{i \in I_s^p} B_{isn}^O - \sum_{i \in I_s^c} B_{isn}^I \qquad \forall s, n > 1 \tag{B.26}$$

Constraint B.27 ensures that available amount of state s does not exceed its maximum capacity. This constraint may be used in order to define different storage policies.

$$S_{sn} \leq C_s \qquad \forall s, n \tag{B.27}$$

Binary variable V_{jsn} establishes if unit j performs as a shared storage tank for state s. By using this variable, it is possible to enforce that at most one state is stored in that tank at a time and tank capacity is not exceeded (constraints B.28 and B.29).

$$\sum_{s \in S(j)} V_{jsn} \leq 1 \qquad \forall n, j \in JT \tag{B.28}$$

$$S_{sn} \leq C_j \cdot V_{jsn} \qquad \forall n, j \in JT, s \in S_j \tag{B.29}$$

Demand constraints

By means of constraint B.30, demand satisfaction is ensured for every product (final state).

$$\sum_n SS_{sn} \geq d_s \qquad \forall s \tag{B.30}$$

Tightening constraints

In order to avoid weak relaxations in the solution of this formulation, the following tightening constraints B.31, B.32 and B.33 are used.

$$\sum_{i \in I_j} \sum_n D_{in} \leq M \qquad \forall j \tag{B.31}$$

$$\sum_{i \in I_j} \sum_{n' \geq n} D_{in'} \leq M - T_n \qquad \forall j, n \tag{B.32}$$

$$\sum_{i \in I_j} \sum_{n' \geq n} (\alpha_i W f_{in'} + \beta_i B f_{in'}) \leq M - T_n \qquad \forall j, n \tag{B.33}$$

B.2 Resource-Task-Network based continuous formulation (Castro et al., 2004)

Timing constraints

A set of global time points is predefined in which the first time point takes place at the beginning whereas the last one is located at the end of the time horizon. This feature is represented by constraints B.34 and B.35.

$$T_1 = 0 \tag{B.34}$$

$$T_{|N|} = MK \tag{B.35}$$

Timing constraint B.36 enforces time between any two event points (n and n') is either equal or greater than the processing time of all tasks starting and ending at these event points. As only one task can be executed in an equipment unit at a certain time, all tasks taking place in the same equipment resource ($r \in R^{EQ}$) must be considered here.

$$T_{n'} - T_n \geq \sum_{i \in I^b} \overline{\mu}_{r,i} \left(\alpha_i \bar{N}_{i,n,n'} \right) \qquad \forall r \in R^{EQ}, n, n', n < n' \leq \Delta n + n, n \neq |N| \tag{B.36}$$

If a zero-wait policy is adopted, constraint B.37 ensures that the difference between starting and finishing times is exactly equal to the processing time.

$$T_{n'} - T_n \leq H\left(1 - \sum_{i \in I^b, i \in I^{ZW}} \overline{\mu}_{r,i} \bar{N}_{i,n,n'}\right) + \sum_{i \in I^b, i \in I^{ZW}} \overline{\mu}_{r,i} \left(\alpha_i \bar{N}_{i,n,n'}\right)$$
$$\forall r \in R^{EQ}, n, n', n < n' \leq \Delta n + n, n \neq |N| \tag{B.37}$$

Balance constraints

Amount of resource consumed or produced at the beginning and at the end of a task is assumed to be proportional to the extent of that task. Total amount of a resource r consumed at the beginning of a task i at event point n and finishing at event point n' is proportional to $\bar{N}_{inn'}$ by μ_{ri}. In addition, amount produced at its completion is proportional to $\overline{\mu}_{ri}$. Constraint B.38 represents the excess resource balance. Here, amount of resource r at time period n is equal to the amount at the previous period $n-1$, plus the amount produced or consumed by tasks ending or starting at time period n.

$$R_{rn} = R_r^0\big|_{n=1} + R_{rn-1}^0\big|_{n>1} + \sum_{i \in I^b} [\sum_{n \leq n' \leq \Delta n + n} \mu_{ri} \overline{N}_{inn'} + \sum_{n - \Delta n \leq n' \leq n} \overline{\mu_{ri}} \overline{N}_{inn'}] +$$
$$+ \sum_{i \in I^s} [\mu_{ri} \overline{N}_{inn+1} + \mu_{ri} \overline{N}_{in-1n}] \qquad \forall n, r \in R \tag{B.38}$$

Storage constraints

Constraint B.39 ensures that for an excess amount of material resource r, storage task $i \in I^S$ is activated.

$$V_i^{MIN} \bar{N}_{inn+1} \leq \sum_{r \in I_r^s} R_{rn} \leq V_i^{MAX} \bar{N}_{inn+1} \qquad \forall i \in I^s, n \neq |N| \tag{B.39}$$

Capacity constraints

Excess amount for any resource at any time period must lie between its predefined minimum and maximum capacities. This is accomplished through constraint B.40, which is also used to define different storage policies. Namely, a

zero value assigned to material resources corresponds to a NIS policy, whereas a big value corresponds to an UIS policy. A limited intermediate storage may be represented by assigning a finite value to R_r^{MAX}, which stands for the maximum storage capacity for material resource r.

$$R_r^{MIN} \leq R_{rn} \leq R_r^{MAX} \qquad \forall n, r \in R \tag{B.40}$$

Demand constraints

Constraint B.41 states that the amount of final products at the end of the time horizon must be equal to or greater than the demand.

$$\sum_n R_{rn} \geq d_r \qquad \forall r \in R^{final} \tag{B.41}$$

B.3 Unit-Specific Time Event formulation (Janak et al., 2004)

Allocation constraints

This model contemplates a set of continuous variables (W_{in}) in order to establish whether a task i is active at an point event n. Constraint B.42 states that at most one task can be active at each event point.

$$\sum_{i \in I_j} W_{in} \leq 1 \quad \forall j, n \tag{B.42}$$

Binary variables Ws_{in} and Wf_{in} are related to W_{in} by constraint B.43.

$$\sum_{n' \leq n} Ws_{in'} - \sum_{n' < n} Wf_{in'} = W_{in} \qquad \forall i, n \tag{B.43}$$

Constraint B.44 that every task starts and finishes during the time horizon.

$$\sum_n Ws_{in} = \sum_n Wf_{in} \qquad \forall i \tag{B.44}$$

Constraint B.45 guarantees that a task i may start at an event point n if all tasks i beginning earlier have already finished. Alternatively, constraint B.46 expresses that a task i may finish at an event point n if it has started at a previous event point n' and has not finished yet.

$$Ws_{in} \leq 1 - \sum_{n'<n} Ws_{in'} + \sum_{n'<n} Wf_{in'} \qquad \forall i, n \tag{B.45}$$

$$Wf_{in} \leq \sum_{n'<n} Ws_{in'} - \sum_{n'<n} Wf_{in'} \qquad \forall i, n \tag{B.46}$$

Batch size constraints

Equation B.47 establishes maximum and minimum allowable batch sizes, and equation B.48 gives the maximum available storage capacity for the storage tasks.

$$B_i^{MIN} W_{in} \leq B_{in} \leq B_i^{MAX} W_{in} \qquad \forall i, n \tag{B.47}$$

$$B_i^{st} n^{st} \leq V_i^{stMAX} \qquad \forall i^{st}, n \tag{B.48}$$

Constraints B.49 and B.50 define the batch sizes of the tasks that take more than one time period to be accomplished.

$$B_{in} \leq B_{in-1} + B_i^{MAX} \left(1 - W_{in-1} + Wf_{in-1}\right) \qquad \forall i, n > 1 \tag{B.49}$$

$$B_{in} \geq B_{in-1} - B_i^{MAX} \left(1 - W_{in-1} + Wf_{in-1}\right) \qquad \forall i, n > 1 \tag{B.50}$$

Initial and final batch sizes are defined in constraints B.51 to B.56 in order to accurately perform the mass balances.

$$Bs_{in} \leq B_{in} \qquad \forall i, n \tag{B.51}$$

$$Bs_{in} \geq B_{in} - B_i^{MAX}\,(1 - Ws_{in}) \qquad \forall i, n \tag{B.52}$$

$$Bs_{in} \leq B_i^{MAX} Ws_{in} \qquad \forall i, n \tag{B.53}$$

$$Bf_{in} \leq B_{in} \qquad \forall i, n \tag{B.54}$$

$$Bf_{in} \geq B_{in} - B_i^{MAX}\,(1 - Wf_{in}) \qquad \forall i, n \tag{B.55}$$

$$Bf_{in} \leq B_i^{MAX} Wf_{in} \qquad \forall i, n \tag{B.56}$$

Mass balance constraints

Constraint B.57 states that the available amount of state s at event point n is equal to the amount of material corresponding to the previous time point, minus the quantity sold at the current time period, plus the difference in storages, and the quantities produced and consumed for that state at the previous and current time points.

$$S_{sn} = S_{sn-1} - S_{sn} + \sum_{i \in I_s^p} \rho_{is} Bf_{in-1} - \sum_{i \in I_s^c} \rho_{is} Bs_{in} + \sum_{i^{st} \in I_s^{st}} Bf_{in-1}^{st} - \sum_{i^{st} \in I_s^{st}} B_{in}^{st}$$
$$\forall s, n > 1 \tag{B.57}$$

Demand requirements are expressed by constraint B.58.

$$\sum_n SS_{sn} \geq d_s \qquad \forall s \tag{B.58}$$

Timing constraints

Finishing time of a task i has to be greater than or equal to the starting time at time event n (constraint B.59).

$$Tf_{in} \geq Ts_{in} \qquad \forall i, n \tag{B.59}$$

Alternatively, constraint B.60 makes equal the starting and finishing time of a task i, if it does not take place at event point n. Otherwise, this constraint is relaxed.

$$Tf_{in} \leq Ts_{in} + M \cdot W_{in} \qquad \forall i, n \tag{B.60}$$

Combination of constraint B.61 with B.62 and B.63 ensures that finish time at $n-1$ is equal to the starting time at n if task i is active and must be extended to the following event n. Otherwise, these constraints are relaxed.

$$Ts_{in} \leq Tf_{i(n-1)} + M(1 - W_{i(n-1)} + Wf_{i(n-1)}) \qquad \forall i, n > 1 \tag{B.61}$$

Constraint B.62 works together with constraint B.63 relating the starting time of a task i at event point n with the finishing time of this task at event point n'. Equation B.63 just constrains tasks that can not both process and store material.

$$Tf_{in'} - Ts_{in} \geq pt_i Ws_{in} - M\left(1 - Ws_{in}\right) - M\left(1 - Wf_{in'}\right) - M\left(\sum_{n \leq n'' \leq n'} Wf_{in''}\right)$$
$$\forall i, n \leq n' \tag{B.62}$$

$$Tf_{in'} - Ts_{in} \leq pt_i Ws_{in} + M\left(1 - Ws_{in}\right) + M\left(1 - Wf_{in'}\right) + M\left(\sum_{n \leq n'' \leq n'} Wf_{in''}\right)$$
$$\forall i \notin I^{ps}, n \leq n' \tag{B.63}$$

Constraint B.64 is applied to tasks performing storage functions.

$$Tf^{st}_{i^{st}n} \geq Ts^{st}_{i^{st}n} \qquad \forall i^{st}, n \tag{B.64}$$

Appendix B

Sequencing constraint

Constraints B.65, B.66 and B.67 sequence, in this order, the same task in the same unit, different tasks in the same unit and different tasks in different units.

$$Ts_{in} \geq Tf_{in-1} \qquad \forall i, n > 1 \tag{B.65}$$

$$Ts_{in} \geq Tf_{i'n-1} - M(1 - W_{i'n-1}) \qquad \forall i \neq i', j, n > 1 \tag{B.66}$$

$$\begin{array}{l} Ts_{in} \geq Tf_{i'n-1} - M(1 - Wf_{i'n-1}) \\ \forall s, i \in I_s^c, i' \in I_s^p, j \neq j', n > 1 \end{array} \tag{B.67}$$

Consecutive tasks i and i' under zero-wait transfer policy are sequenced by constraint B.68.

$$\begin{array}{l} Ts_{in} \leq Tf_{i'n-1} - M(2 - Wf_{i'n-1} - Ws_{in}) \\ \forall s, i \in I_s^c, i' \in I_s^p, j \neq j', n > 1 \end{array} \tag{B.68}$$

Sequencing constraint: Storage tasks

Constraints B.69 to B.73 sequence storage tasks i^{st}. Specific variables $Ts^{st}_{i^{st}n}$ and $Tf^{st}_{i^{st}n}$ are used to denote the time at which a storage task i^{st} starts and finishes. Shared storage is modeled by specifying the set of storage tasks, so that multiple states s are linked to each storage task.

$$Ts_{in} \geq Tf^{st}_{i^{st}n-1} \qquad \forall s, i \in I_s^c, i^{st}, n > 1 \tag{B.69}$$

$$Ts_{in} \leq Tf^{st}_{i^{st}n-1} + M(1 - Ws_{in}) \qquad \forall s, i \in I_s^c, \forall i^{st} \in I_s^{st}, n > 1 \tag{B.70}$$

$$Ts^{st}_{i^{st}n} \geq Tf_{i'n-1} - M(1 - Wf_{i'n-1}) \qquad \forall s, i' \in I_s^p, i^{st} \in I_s^{st}, n > 1 \tag{B.71}$$

$$Ts^{st}_{i^{st}n} \leq Tf_{i'n-1} + M(1 - Wf_{i'n-1}) \qquad \forall s, i' \in I^p_s, i^{st} \in I^{st}_s, n > 1 \quad \text{(B.72)}$$

$$Ts^{st}_{i^{st}n} = Tf^{st}_{i^{st}n-1} \qquad \forall i^{st} \in I^{st}_s, n > 1 \quad \text{(B.73)}$$

Tightening constraints

Tighter relaxed solutions of this formulation can be obtained by applying the constraints introduced by Maravelias and Grossmann (2003) and previously reported in this appendix (section B.1).

B.4 Nomenclature

Subscripts

i, i'	Tasks
i^{st}	Storage tasks
j, j'	Equipment units
n, n'	Event points
r, r'	Resources
s, s'	States

Sets

I_j	Tasks i that can be allocated in equipment unit j.
I^b	Batch tasks.
I^c_s	Tasks that consume state s.
I^p_s	Tasks that produce state s.
I^{ps}	Processing or storing tasks.
I^s	Storage tasks.
I^{st}_s	Storage tasks for state s.
I^{ZW}	Tasks under zero-wait storage policy.
JT	Shared storage tanks.
S_j	States that can be stored in shared storage tank j.
SI_i	States consumed by task i.
SO_i	States produced by task i.

Δn	Maximum number of event points between the beginning and ending of a task.
J_i	Units that can perform task i.
N	Total number of event points.
R_r^{EQ}	Equipment resources (storage tanks are not included).
R_r^{INT}	Resources r corresponding to intermediate products.
R_r^{final}	Resources r corresponding to final products.

Parameters

α_i	Fixed duration of a task i.
β_i	Variable duration of a task i.
μ_{ri}	Amount consumed of resource r in task i.
$\bar{\mu}_{ri}$	Amount produced of resource r in task i.
ρ_i	Mass balance coefficient for consumption/production ratio of state s in task i.
d_s	Demand of state s at the end of the time horizon.
C_s, C_j	Storage capacity for state s / shared tank j.
pt_i	Processing time of task i.
M	A very large number.
R_r^{MIN}	Minimum availability of resource r.
R_r^{MAX}	Maximum availability of resource r.
B_i^{MIN}	Lower bound on the batch size of task i.
B_i^{MAX}	Upper bound on the batch size of task i.
V_i^{MIN}	Lower bound on storage capacity for task i.
V_i^{MAX}	Upper bound on storage capacity for task i.

Continuous variables

D_{in}	Duration of task i starting at time point n.
T_n	Absolute time of time point n
Ts_{in}	Starting time of task i that starts at time point n.
Tf_{in}	Finishing time of task i that starts at time point n.
T_{ijn}^s	Time at which task i starts in unit j at time point n.
T_{ijn}^f	Time at which task i ends in unit j at time point n.
$Ts_{i^{st}n}^{st}$	Time at which storage task i_{st} starts at time point n.
$Tf_{i^{st}n}^{st}$	Time at which storage task i_{st} ends at time point n.
Bs_{in}	Batch size of task i starting at time point n.

Bf_{in}	Batch size of task i finishing at or before time point n.
Bp_{in}	Batch size of task i being processed at time point n.
B^I_{isn}	Amount of state s used as input for task i at time point n.
Bs_{in}	Batch size of task i starting at time point n.
Bf_{in}	Batch size of task i finishing at or before time point n.
Bp_{in}	Batch size of task i being processed at time point n.
B^I_{isn}	Amount of state s used as input for task i at time point n.
B^O_{isn}	Amount of state s produced as output for task i at time point n.
D_{in}	Duration of task i starting at time point n.
MK	Makespan. .
R_{rn}	Excess amount of resource r at time point n.
R^0_r	Initial amount of resource r.
S_{sn}	Amount of state s available at time point n.
SS_{sn}	Sales of state s at time point n.
T_n	Absolute time of time point n.
Ts_{in}	Starting time of task i that starts at time point n.
Tf_{in}	Finishing time of task i that starts at time point n.
$Ts^{st}_{i^{st}n}$	Time at which storage task i_{st} starts at time point n.
$Tf^{st}_{i^{st}n}$	Time at which storage task i_{st} ends at time point n.
T^s_{ijn}	Time at which task i starts in unit j at time point n.
T^f_{ijn}	Time at which task i ends in unit j at time point n.

Binary variables

$\bar{N}_{inn'}$	Determines if task i starts at time point n and finishes at time point n'.
V_{jsn}	Determines if state s is stored in shared tank j during time point n.
W_{in}	Determines if task i is active at time point n.
Wf_{in}	Determines if task i ends at time point n.
Wp_{in}	Determines if task i is processed at time point n.
Ws_{in}	Determines if task i starts at time point n.

List of Tables

List of Figures