Integrated Scheduling Decision Making in Enterprise Wide Optimization

Integrated Scheduling Decision Making in Enterprise Wide Optimization

Elisabet Capón García
Chemical and Industrial Engineer

A Thesis presented for the degree of
Doctor of Philosophy

Directed by Prof. Dr. Antonio Espuña and Dr. Gonzalo Guillén

Universitat Politècnica de Catalunya. BarcelonaTech
Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

June 2011

A mis padres

Summary

In the current environment of markets globalization and fierce competition, process industries must strive to remain competitive. In this sense, companies pursue decision integration among the different space and time levels of their organizational structure in order to improve their overall results. Indeed, several cases reported in the literature confirm the economic benefits derived from decision integration. Hence, much research is devoted to obtain improved models, integration and information tools as well as optimization algorithms which provide with decision support tools within a coherent framework for the enterprise design.

From a plant level perspective, the short-term scheduling problem deals with the management of production orders in order to optimally fulfill customers' demands by assigning the available resources. This decision level is further related to other decision levels such as planning and control. The need for integration of these decision levels has been reported from the 60s, but the contributions in this area are still scarce. Therefore, further efforts have to be devoted to the integration within the operational level, namely the scheduling and control functions.

This thesis aims at contributing to the *integration* of the short-term scheduling problem of batch process industries from a *structural* and *functional* point of view. The structural perspective refers to other decision levels from the managerial organization, which only comprises the basic process control in this work, but it does not limit the capacity of the proposed strategies to include other decision levels. As for the functional issues, the completion of the objective functions used at scheduling level may lead to integrated decisions from an overall perspective. Therefore, the inclusion of non-economic objectives in the decision making may lead to more concerned solutions from other problem perspectives, such as environmental. Thus, the extension of economic criteria to consider process variables costs may ease the integration approaches.

First of all, an overview of the current global scenario, the relevance of scheduling problem and of its integration in the decision making, as well as the existing solution approaches are presented. The second part of this work is devoted to the description and extension of the immediate and general precedence formulations of the scheduling problem, in order to consider non-trivial problem features such as the batch cleaning operations, equipment transfer operations, variable processing rates, timing synchronization of operations and the introduction of process dynamics.

Precisely, the third part of this thesis is devoted to the introduction of process dynamics at the scheduling level, which can be achieved either (i) indirectly: by considering cost functions of time; or (ii) directly: by combining discretized dynamic equations in the scheduling formulation. This part explores the adequacy of each integration method and assesses the benefits that can be achieved with such integration. Moreover, the consideration of variable processing rates within single campaign semicontinuous batch processes is studied.

The last part of the thesis focuses on extending the traditional economic function of the scheduling problem to consider environmental issues. Specifically, the trade-offs arising between environmental and economic criteria are studied for by means of Pareto frontiers, which provide the decision maker with highly valuable information about production schedule trade-offs. Additionally, the decision maker may reach completely different Pareto frontiers, in terms of number and sequence of product batches, as well as in selected cleaning methods by considering different objective functions. Specifically, depending on the choice of absolute or relative metrics, i.e. time or quantity related, different solutions may be reached. Finally, strategies for dealing with large size scheduling problems are provided based on a hybrid strategy considering multi-objective genetic algorithm with mathematical based local search.

Resumen

En el entorno actual de globalización de los mercados y una competencia feroz, las industrias de proceso deben esforzarse para seguir siendo competitivas. En este sentido, las empresas buscan la integración de decisiones dentro de su estructura organizativa a distintos niveles temporal y espacial con el fin de mejorar sus resultados globales. En la práctica, se han publicado en la literatura varios casos industriales que confirman los beneficios económicos derivados de la integración de decisiones. Por este motivo, se invierten muchos esfuerzos de investigación para la obtención y mejora de modelos, de herramientas de integración y de flujo de información, así como para el desarrollo de algoritmos de optimización que proporcionen las herramientas de soporte a las decisiones dentro de un marco coherente para el diseño y operación de la empresa.

Desde una perspectiva a nivel de planta, el problema de programación de operaciones a corto plazo persigue la gestión óptima de las órdenes de producción mediante la asignación de los recursos disponibles con el fin de cumplir con las demandas de los clientes. Este nivel de decisión está además relacionado con otros niveles como la planificación y control. La necesidad de integración de estos niveles de decisión se ha citado ya desde los años 60, pero las contribuciones en esta área de investigación son todavía escasas. Por eso, es necesario invertir más esfuerzos para la integración de decisiones a nivel operativo, el cual incluye las funciones de programación de operaciones y control.

El principal objetivo de esta tesis consiste en contribuir a la *integración* del problema de programación de operaciones a corto plazo de las industrias batch de proceso desde un punto de vista *estructural* y *funcional*. La perspectiva estructural se refiere a la integración con otros niveles de decisión de la estructura organizativa de la empresa, que se limita en este trabajo al nivel de control básicos, pero en cualquier caso, no limita la capacidad de las estrategias propuestas para incluir otros niveles de decisión. En cuanto a las cuestiones funcionales, la adopción de funciones objetivo que incluyan todos los aspectos del problema de programación de operaciones puede llevar a decisiones integradas desde un punto de vista global. En este sentido, la inclusión de los costes de las variables de proceso en la función objetivo del problema de programación de operaciones puede facilitar la integración de los niveles. Además, la inclusión de objetivos no económicos en la toma de decisiones puede generar soluciones más comprometidas desde otras perspectivas del problema, como por ejemplo la

medioambiental.

En primer lugar, se presentan una visión general del panorama actual de la industria de proces, la relevancia del problema de programación de operaciones y de su integración en la toma de decisiones, así como los enfoques existentes para la solución de dicho problema. La segunda parte de esta tesis está dedicado a la descripción y la extensión de las varias formulaciones del problema de programación de operaciones, a fin de considerar características no triviales del problema, tales como las operaciones de limpieza, las operaciones de transferencia de equipo, velocidades de proceso variables, la sincronización de operaciones y la introducción de la dinámica del proceso.

Precisamente, la tercera parte de esta tesis está dedicada a la introducción de la dinámica del proceso dentro del nivel de programación de operaciones, que se puede lograr ya sea (i) indirectamente: considerando funciones de coste en función del tiempo, o (ii) directamente: mediante la discretización de las ecuaciones dinámicas del modelo de proceso y su incorporación a la formulación del problema de programación de operaciones. En esta parte, se analiza la idoneidad de cada método de integración y se evalúan los beneficios que pueden lograrse con la integración. Además, se estudia la repercusión de velocidades de proceso variables, en procesos semicontinuos con producción de un solo lote por campaña de producto.

La última contribución de esta tesis se centra en la ampliación de la tradicional función objetivo económica del problema de programación de operaciones para examinar cuestiones medioambientales. En concreto, se estudian las soluciones de compromiso que aparecen entre los criterios ambientales y económicos mediante fronteras de Pareto, que proporcionan información muy valiosa al decisor sobre los compromisos existentes. En esta parte se observa que decisor puede llegar a soluciones de la frontera de Pareto completamente diferentes, tanto en términos de número como secuencia de los lotes de productos, así como en los métodos de limpieza, cuando se consideran funciones objetivo diferentes. En concreto, en función de la elección de indicadores absolutos o relativos, ya sea cuanto a tiempo o cantidad, se pueden alcanzar soluciones completamente diferentes. Por último, se presenta una estrategia híbrida de optimización para poder resolver problemas de programación de operaciones de tamaño real, que consiste en una metaheurística, concretamente un algoritmo genético combinado con una búsqueda local matemática rigurosa.

Acknowledgments

First of all, I would like express my gratitude to Antonio and Gonzalo for their mentorship and fruitful discussions. In addition, I would like to thank Professor Puigjaner for his support and attention to the work that I have developed in the group. Financial support received from FPU grant from the Ministerio de Educación is also fully acknowledged.

Moreover, I am deeply grateful to Ignacio Grossmann for the fruitful and pleasant stay in his research group in Carnegie Mellon. Thus, I very much appreciate Sebastian Engell receiving me in his research group in Technische Universität Dortmund.

This thesis has offered me the opportunity to improve professionally, but also and more importantly from a personal point of view. I have learned a lot from my colleagues along our joined path these years. Thus, I would like to thank all of them who have honored me with their valuable friendship.

Finally, I am truly indebted to my parents for their understanding, trust and unconditional love. I also would like to mention my brother Luis and my sister Mireia. I thank you for your patience and your continuous support.

Agradecimientos

En primer lugar, quisiera expresar mi agradecimiento a Antonio y Gonzalo por su supervisión y las provechosas discusiones que hemos mantenido. Además, me gustaría dar las gracias al Profesor Puigjaner por su apoyo y atención a la labor que he desarrollado en el grupo. Sin duda he de reconocer también el apoyo económico recibido por parte del Ministerio de Educación mediante la beca de Formación de Profesorado Universitario.

Estoy sinceramente agradecida a Ignacio Grossmann por la fructífera y agradable estancia que tuve en su grupo de investigación de Carnegie Mellon. Asimismo, le agradezco a Sebastián Engell por recibirme en su grupo de investigación de la Technische Universität Dortmund.

Esta tesis me ha ofrecido la oportunidad de crecer no sólo profesionalmente, sino también y más importante desde el punto de vista personal. He aprendido mucho de mis compañeros del grupo a lo largo del camino compartido estos años. Además, me gustaría dar las gracias a todos aquellos que me han honrado con su valiosa amistad.

Este trabajo merece el reconocimiento a mis padres, cuya comprensión, confianza y amor incondicional siempre me acompañan. También me gustaría mencionar a mis hermanos Luis y Mireia. Les doy a ellos las gracias por su paciencia y continuo apoyo.

Contents

Part I Overview

List of Tables

List of Figures

Part I

Overview

Chapter 1

Introduction

1.1 Introductory perspective

The development of chemical and other process industries, such as food, agriculture or pharmaceutical, has spawned many advances which improve human life conditions, but also poses serious concerns and dangers. Hence, the perspective of the process industry as a whole is of crucial importance for effectively dealing with existing challenges. Precisely, a day-to-day question in process plants consists of optimally fulfilling customers' demands by managing production orders and accommodating them to the available resources. Such function is handled by the scheduling level in the enterprise structure. Therefore, the goal of this chapter is to gain insight into the domain of process scheduling from a global perspective of the process industry.

In general terms, the current landscape of process industries is ruled by the globalization of trade. Such trend has opened new markets and business opportunities and the adoption of worldwide information tools has brought forth greater market efficiencies. Thus, globalization can even help to improve the standard of living throughout the world. However, enterprises must face a higher uncertainty regarding external factors such as demand, product prices or raw materials supply. Moreover, businesses tackle a fiercer competitive environment stemming from the entrance of new competitors, which leads to dwindling margins. From an economic point of view, process industries are very sensitive to both fluctuations of the economic cycle and possible changes in customers' behaviour. Hence, as a result of the serious economic recession at the end of last decade, their activity has contracted and enterprise benefits have globally decreased. Even in some cases, given the important demand reduction, enterprises currently operate far from their optimal capacity and it will take them some years to return to their maximum production levels (Cuchí, 2010).

From a social perspective, companies must deal with increasingly stricter constraints related to safety and environmental regulations, and also cope with other issues to gain a positive corporative image and social acceptance. For example, industry has an increasingly important environmental commitment to produce in a sustainable way, by consuming less resources and energy, promoting renewable energy sources and improving water management (Grossmann, 2004; Council, 2010).

In such scenario, enterprises must strive to remain competitive by improving their functional, technological and operational advantage. Therefore, companies tend to look for economies of scale, production concentration policies and higher plant and business specialization. Thus, quick time-to-market and operational flexibility have become crucial business drivers in many industries for reacting rapidly to the continuously changing market conditions. In order to achieve such production objectives, process management and scheduling play an important role at plant level.

Thus, the current situation increases the inherent complexity of process industries, which can be regarded as highly involved systems. Such industries consist of multiple business and process units ranging from molecule to enterprise level. The organization of the different scales and levels within such complex systems is crucial to analyze and understand their behavior and function, as well as to implement any given requirement over them. The basis for solving a systems problem is the system representation in an adequate model, which captures the features relevant for the observer whose ultimate aim lies on decision making. Precisely, decision making in process industries results in a highly challenging task. In this area, process systems engineering (PSE) is a well established discipline of chemical engineering which covers a set of methods and tools to support decision-making for the creation and operation of the process supply chain constituting the discovery, design, manufacturing and distribution of chemical products and other process goods from a holistic approach. In order to deal with the problem complexity, it is necessary to decouple the system across a hierarchy of appropriately chosen levels.

The supply chain (SC) can be defined as the group of interlinked resources and activities required to create and deliver products and services to customers. Decisions are taken at different stages within the SC at different levels in the management hierarchy and they also differ in business scope, time horizon and resolution, data certainty and accuracy, process detail and optimization mechanism (Lasschuit & Thijssen, 2004). From a functional point of view, the enterprise has been traditionally divided in three basic decision levels: strategic, tactical and operational (Figure 1.1). Long-term strategic level defines the business scope by determining the structure of the supply chain in a time period of years. Medium-term tactical planning is concerned with decisions such as the assignment of production targets to facilities and the transportation from facilities to targets. The operational level is related to short-term planning or scheduling which determines on a daily or weekly basis the assignment of tasks to units and the sequencing of tasks in each unit. Control of production processes is an additional function concerning the operational level which involves the real time manipulation of production variables to deal with process disturbances and hold product qualities and production rates near the target values.

The aforementioned functional decision levels have different space and time scales, but they are directly related to each other since the decisions made at one level directly affect others. According to Shobrys and White (2002), companies pursuing integration among the different decision levels in the production management environment report substantial economic benefits. Hence, it is of utmost importance to coordinate and integrate information and decisions among the various functions that comprise the whole supply chain. Recently, enterprise-wide optimization (EWO) has emerged as a new area which aims at optimizing the operations of supply, production and

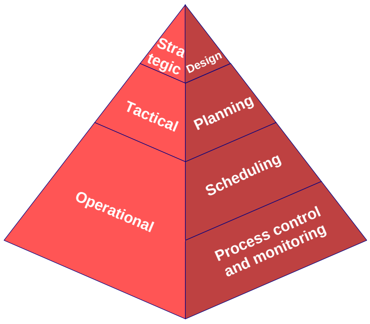


Figure 1.1: Decision levels in enterprise structure.

distribution of process companies to reduce costs and inventories. Specifically, EWO places emphasis on production facilities focusing on their planning, scheduling and control taking into account the domain knowledge in engineering.

In this area, only some modest attempts at integrating a small subset of enterprise-wide decision models exist, since the complex organizational structures underlying integrated process models challenge our understanding of cross-functional coordination and its business impact (Varma *et al.*, 2007). Hence, much work still remains to be done to target computational optimization models and tools that allow a comprehensive application of the enterprise wide optimization throughout the process industry. Therefore, much effort must be devoted to obtain improved models, integration and information tools as well as optimization algorithms, providing decision support tools within a coherent framework which takes into account the available information on actual plant operations and market economics. On the whole, process industry research should focus on multidisciplinary and multiscale methodologies in order to deal with increasing environmental, societal and economic requirements for such complex systems (Ottino, 2005; Edwards, 2006; Charpentier, 2007; Klatt & Marquardt, 2009). Specifically, this thesis aims at tackling the decision integration within the operational levels from a modeling and functional perspective of the scheduling problem.

1.2 Operational decision levels

The operational decision levels comprise the scheduling and control problems. Both levels have different time scale domains, different problem perspective and boundaries and a wide variety of approaches to formulate and solve them (Shobrys & White, 2002). Hence, the scheduling and control functions are optimized sequentially in isolation from each other in order to overcome the complexity of the combined problem. However, the integration of these functions would lead to improve overall plant operability and enterprise economic advantage. Indeed, many companies which are pursing integration report substantial financial incentives for better integration, for example oil companies estimate incentives of up to 1 dollar per product barrel for better integration of planning, scheduling and control for gasoline blending (Shobrys & White, 2002).

On the one hand, the control level addresses real time execution and aims at achieving an efficient, safe, environmentally friendly and reliable operation to execute the production requests calculated at the scheduling level. Therefore, it enables production by means of controlling, coordinating and communicating the process plant. Anyhow, the most important task of the control system consists of ensuring process safety by monitoring and maintaining process conditions within operating limits making up for the impact over the process of external disturbances such as changes in feed flowrate, feed conditions, product specifications, product prices or ambient temperature (Smith, 2005).

On the other hand, the scheduling problem consists of the organization of human and technological resource use in the plant to directly satisfy client demands issued from a production plan prepared by the company planning function. In production scheduling, it may be distinguished between long term-scheduling and short-term scheduling. The former is basically a material resources planning (MRP) which receives orders for products and deals with demand forecasting for inventory management considering plant locations and capacities (Perry *et al.*, 1997). In contrast, short-term scheduling is implemented at plant level and typically involves decisions on the amount of products to be produced, equipment and resources allocation, production sequence and operations timing.

The first works on the need for integration date back to the 1960s (Shobrys & White, 2002). However, relevant work has been published recently, when a true optimization approach which can overcome the problem complexity has become possible through more efficient solution algorithms and increased computing power. Hence, there is a trend in industrial environments toward the management of the process plant as an integral part of the global supply chain. Particularly at operational level, an important open issue is the coupling of scheduling with process models, which are inherently dynamic models (Grossmann *et al.*, 2008). Therefore, the objective of plant operation is moving to maximize plant economics in real-time subject to equipment, safety and product related constraints, based on an integrative approach considering the process and its operational support systems, namely control and scheduling functionalities in a simultaneous manner (Klatt & Marquardt, 2009).

Precisely, this thesis focuses on the integration of decision making at the operational decision level from the short-term scheduling perspective. Anyhow, the presented work can be generalized and extended to include any other decision level in the enterprise structure.

1.2.1 Control level

The process control level aims at achieving an efficient, safe and environmentally friendly operation in order to produce the desired products by means of the process control system and the understanding of the process dynamics. This decision level comprises several functions, which can be classified as follows according to the ANSI/ISA-88 standard (International Society for Measurement and Control, 2001):

- *Procedural control.* Its objective consists of implementing the sequence of control steps in the equipment modules in order to fulfill the desired production. It comprises the unit procedure, operations and phases, as well as the transition logics from stage to stage.

- *Basic control.* This function is dedicated to establishing and maintaining a specific state of equipment and process conditions. Thus, it regulates the value of process variables along time, by means of, for example, PID controllers appropriately situated and tuned, supervises process variables and warns if there is a specific value that exceeds the processing limits. For each step of the procedural control, there is a basic control function developed, which comprises different parameters, such as set points or controller parameters, defined in the control recipe.

- *Coordination control.* It contains the control recipe to be implemented at each time and the transition between control recipes. Therefore, coordination control directs, initiates and modifies the execution of procedural control over the different equipment entities. Thus, it includes allocating resources, propagating modes and arbitration of shared resources and equipment requests at low operational level.

The actual process variables values are controlled by the basic control function. Hence, such control function is the one appropriate to be integrated in the scheduling level, since process variations influence processing time and consequently the production schedule.

1.2.2 Short-term process scheduling

Specialty products are generally produced in batch processes, which provide inherent operational flexibility. In practice, batch processing is widely used to manufacture an extremely broad range of processes and products such as metals, electronic materials, ceramics, polymers, food and agricultural materials, biochemicals and pharmaceuticals, multiphase materials/blends, coatings or composites. In batch plants, several products can be manufactured using the same process equipment units and several batches can be simultaneously processed depending on the equipment configuration of the plant.

Precisely, production scheduling is particularly relevant in the production of specialties, high-margin products, for which a high speed for reacting to market conditions and meeting changing demands is needed. In fact, these products, whose added value, based on their differentiation, usually requires the control of their end-use properties, are normally produced in relatively small quantities according to the customer demands regarding high efficiency and quick time to market conditions.

Therefore, this thesis focuses on the short-term scheduling of batch processes. Along this section the main features regarding the modeling of the scheduling problem and its integration challenges are outlined.

Modeling of process scheduling The starting point when posing a scheduling model consists of defining the problem features, namely the objectives pursued by the scheduling function, the decisions to be made, and finally the elements which represent the system and describe its behavior.

Scheduling objectives. The objective function measures the quality of the decisions to be made. The objectives which have traditionally been considered in scheduling problems are time related, such as the total completion time or makespan, lateness, tardiness and earliness (Hoogeveen, 2005). However, decisions in chemical industry are usually driven by profitability criterion. Hence, it is necessary to adequately quantify economic criteria in order to reach a high quality decision from an integrative perspective (Edgar, 2004). Therefore, global metrics such as profit, cost or profitability itself should be considered (Méndez *et al.*, 2006).

Scheduling decisions. The decisions involved in the scheduling function vary according to the plant management needs and depend on the problem features. Anyways, such decisions are highly connected to each other, and to other decision levels, for instance to planning for demand issues or to control for actual processing times. In general, decisions within the scheduling level can be broadly classified in four types which are listed next.

- *Batching.* It consists of deciding on the number and size of the lots of products that are to be produced (Maravelias & Sung, 2009). Therefore, these decisions are directly related to the mass balances and storage management, and so to the planning level.

- *Allocation.* It involves the assignment along time of tasks to equipments and other plant resources, such as manpower, electricity, water and so on, according to their availability. Resources are usually finite and product specific, and they may be reusable or not.

- *Sequencing.* It determines the order in which batches of the different products are to be produced in the different equipments along the process plant.

- *Timing.* It concludes the initial and final times at which batches are to be performed. Such decisions highly depend on the process features. Therefore, according to existing intermediate storage policies and the relationship between tasks, production timing may be differently performed inside the whole production time horizon. A key point in batch scheduling consists of the time representation, which depends on whether actions may take place at any time or at some predefined time points. Since actual processing times depend on process conditions, which may be influenced by external disturbances, actual timing decisions may be highly changed from the initial estimation. Hence, timing is intimately related to the basic control level.

Scheduling constraints. Production process features and process plant environment determine the formulation of the scheduling problem. The most fundamental element describing the production process is the product recipe, which contains the information about the amount of raw materials as well as the processing steps with their conditions, such as temperature or pressure, and the times over which they take place. Even though product recipes are usually determined and fixed in the design step, they can potentially introduce flexibility in the scheduling problem, if variable process conditions are allowed.

Depending on the equipment configuration of the plant, the number of batches that can be processed simultaneously varies. In multiproduct batch plants, the final products have an identical recipe structure. Therefore, all products require all steps in the process and follow the same sequence of operations. In contrast, in multipurpose batch plants, the production steps are not the same for all products. Such plants are more flexible and effective for a large number of products produced in small volumes, since equipment items can be fully utilized and vessels cleaned easily.

According to the perishable features of the intermediate products, alternative storage policies can be adopted, namely a zero-wait transfer, a no intermediate storage (if intermediates can be hold up in the equipment), a intermediate storage, which may be shared or exclusive, or an unlimited intermediate storage policy.

In general, operation is favored by large batches and long production campaigns, which increases storage costs (Smith, 2005). Thus, equipment cleaning and material transfer are two practical issues that may have a significant effect in the scheduling results (Smith, 2005) because they may cause a reduction in the overall equipment utilization.

Integration challenges in process scheduling The short-term scheduling function is directly related to: i) the company's planning function which determines the production targets; ii) the process control function which implements and monitors the scheduling outputs, and iii) the production process itself by means of the product recipe, which is based on the process stages. Therefore, the integration of process scheduling with other functions is a broad area which poses a large number of challenges (Figure 1.2).

On the one hand, many efforts have been devoted to incorporate scheduling models in the production planning function in order to address the integration between these two levels. In a recent review, Maravelias and Sung (2009) identify several challenges related to such integration, such as the development of computationally effective scheduling formulations for complex process networks, the development of hybrid methods which exploit the strengths of the solution methods or the communication between the master and slave subproblems in iterative schemes. The authors broadly classify the solution methods of the scheduling and planning integration problem in three approaches, namely hierarchical, iterative and full-space methods. Although the integration of scheduling with the planning level is beyond the scope of this thesis, the integration challenges and needs are very similar among the different decision levels.

As for the integration between the control and scheduling functions, the contributions to this field in literature are still scarce. Several problem aspects must be considered, for instance the mathematical approaches to formulate and solve the two levels are widely different and the information flow and data between the decision layers must be enabled. According to Harjunkoski *et al.* (2009), current challenges require well-defined modeling and optimization approaches as well as software architectures for better collaboration, and integration should emerge from a functionality point of view.

From the process perspective, batch processes do not hold steady state conditions, but are dynamic in nature. Hence, the dynamic models of the process units must be

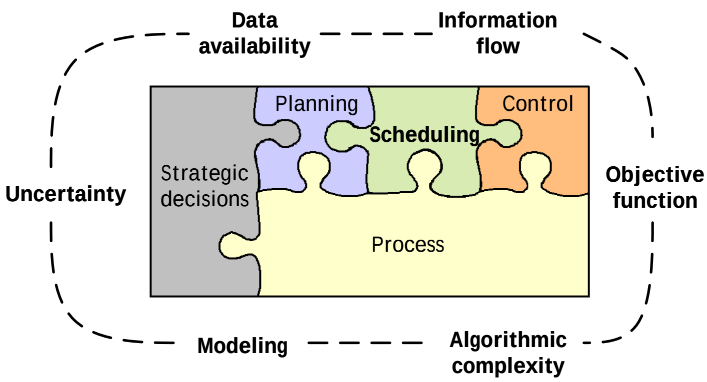


Figure 1.2: Challenges in decisions integration in the enterprise structure.

formulated and the degrees of freedom adjusted, when designing the process recipe, which requires the solution of an optimal control problem. However, even for small problems, the determination of optimal batch times can be very computationally demanding. Therefore, since most processes have recipes with numerous tasks, when preparing a schedule of tasks and equipments, it is common to specify batch times for tasks to be performed in specific equipment, usually with batch sizes (Seider *et al.*, 2004). The integration of the process model in the scheduling problem results in a highly complex mixed integer dynamic optimization problem (Grossmann *et al.*, 2008). However, such integration adds degrees of freedom to the scheduling problem, and consequently provides a higher level of flexibility.

Finally, uncertainty unveils in different forms at all decision levels affecting production schedules. Therefore, an efficient handling of disturbances is a major challenge for the integration, which can be dealt at the scheduling level by means of either effective rescheduling actions (reactive approaches) or through the use of proactive methods (e.g. stochastic programming).

1.3 Research scope and objectives

Current trends in process industry highlight the importance of improving the decision making process at all scales in the company. At plant level, the scheduling function plays a crucial role on daily production decisions. However, owing to the inherent complexity of the scheduling problem, it is a common industrial practice to keep it as simple as possible. Fortunately, modern tools allow to obtain solutions which are able to cope with existing challenges more effectively. Certainly, a large number of challenges in the coordination and integration of the scheduling problem with other decision levels is open. Specifically, such integration may lead to higher plant flexibility to fulfill stricter requirements and meet improved economic goals, as well as to better adaptability against uncertainty.

This thesis aims at providing a set of models and tools for the integration of decisions at scheduling level with the ultimate goal of facilitating the decision making process and improving the overall company's flexibility to respond to uncertainty. The consideration of multiple objective functions and new problem dimensions entails the inclusion of more detailed scheduling models and additional objective functions. Therefore, such general aim can be formalized in four specific objectives as follows:

- To improve existing scheduling models in order to deal effectively with non trivial features of batch processes, such as multiple cleaning methods, transfer operations or variable processing rates.

- To integrate process models within scheduling formulations, considering different levels of detail.

- To propose methodologies that extend the functional scope of the scheduling problem, and allow to assess several objectives, basically economic and environmental, at a time.

- To present reactive strategies for dealing with the scheduling problem under uncertainty.

1.4 Thesis outline

This thesis has been structured in order to introduce progressively the contributions to the integration of the short-term scheduling level from a structural and functional point of view. It consists of five parts as represented in Figure 1.3.

As well as the introduction to the research topic in this chapter, the first part contains a thorough State-of-the-Art in Chapter 2, which leads to the identification of current challenges. Moreover, the methods and tools applied in the work developed along this thesis are outlined in Chapter 3.

Part II introduces the scheduling models which are to be applied in the integration of the scheduling level. Specifically, it improves existing formulations by modeling problem features which are not usually taken into account, namely multiple changeover methods in Chapter 4 and transfer operations in Chapter 5. The two formulations allow for the representation of batch oriented scheduling of sequential processes, the consideration of process control issues as it is shown in Part III and the introduction of alternative objective functions as shown in Part IV.

Precisely, Part III aims at the introduction of process decisions in the scheduling problem. The basic consideration of variable processing rates within single campaign semicontinuous batch processes is studied in Chapter 6. Moreover, a first approach to the actual introduction of process conditions is presented in Chapter 7 by means of functions which relate processing times and cost. Thus, variable batch-to-batch processing times are allowed. Chapter 8 focuses on a more rigorous approach to the simultaneous control and scheduling functions. Specifically, the full process model is included in the scheduling model by discretizing the dynamic equations which represent the process behavior.

Furthermore, Part IV considers the integration of scheduling from a functional point of view. Specifically, Chapter 9 deals with the multiobjective optimization of the scheduling problem considering environmental and economic criteria using several objective functions, and presents an hybrid method to effectively tackle large size scheduling problems.

Finally, Part V summarizes in Chapter 10 the conclusions derived from the research developed in this thesis, and points out the future work lines to be explored.

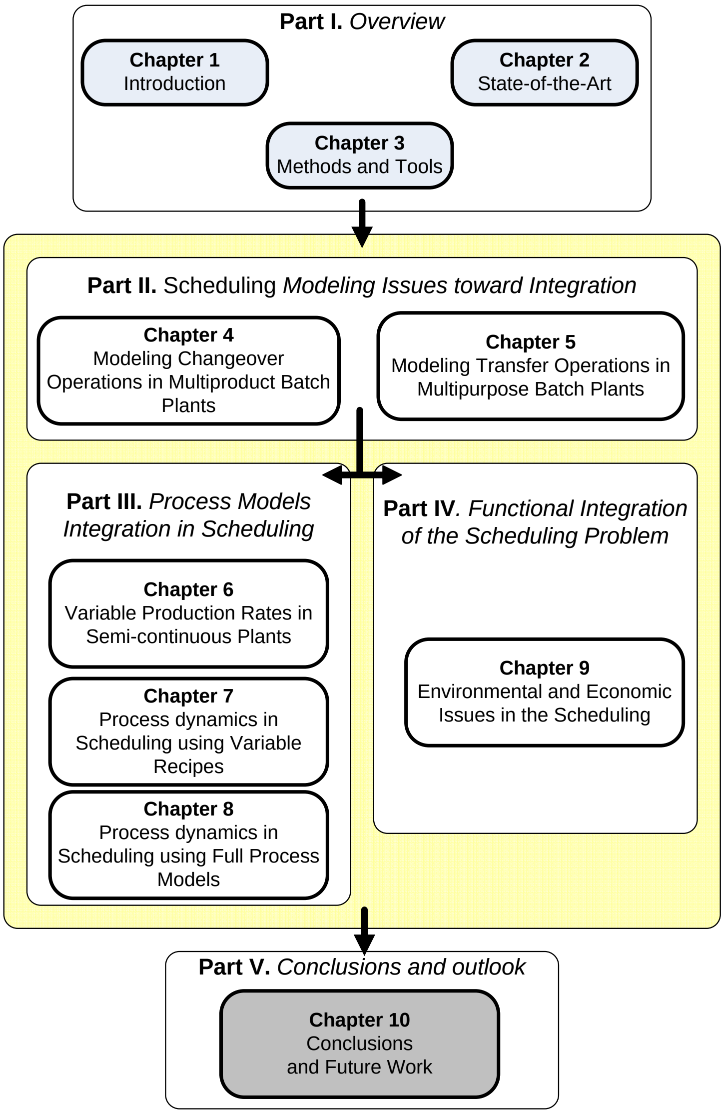


Figure 1.3: Thesis outline

Chapter 2

State-of-the-Art

2.1 Introduction

The intrinsic complexity of the scheduling decision making has brought forth the development of a wide variety of problem models and solution algorithms. The selection and adoption of a given approach depends on the process plant features and the problem definition and size. In fact, there is not a unique optimal approach to solve all scheduling problems, and it is highly unlikely to find a universal one at all (Reklaitis *et al.*, 1996). Therefore, the main aim of this chapter is to describe the current scheduling modeling and solution approaches. It must be noted that the modeling and solution algorithms are highly related, so many references may be cited more than once.

Moreover, the integration objective has been an important issue along the last decades. In contrast to the simplifying trends and problem decoupling principles whose main objective was to succeed in tackling difficult problems, integration strategies offer the possibility to obtain enhanced economic and operational results, as pinpointed in Section 1.2. The integration may be achieved from a structural and functional point of view. From the scheduling perspective, the structural integration is given by linking this decision level to other levels, such as design, planning or process levels, whereas the functional consists of extending the scheduling objectives from the traditional time or economic related metrics, to environmental concerns or reliability issues. Therefore, this chapter also presents specifically current work on integration of the scheduling problem.

Finally, the trends and challenges identified along the state-of-the-art are summarized in order to characterize properly the actual framework for the development of this thesis.

2.2 Scheduling problem features

From a global perspective, batch production scheduling aims at optimizing the resource utilization of batch manufacturing facilities in order to fulfill customer orders within a specific time horizon (Barker & Rawtani, 2005).

The building block of batch process scheduling is the process recipe, which contains the whole information required to produce the product, as well as the set of processing tasks, i.e. the process flow. The recipe is usually obtained in a design stage prior to the scheduling stage, and the process conditions to perform the product are optimized and fixed for all production batches of a given product. In addition, such information must be complemented with the production facility data regarding equipment and resources, such as manpower, inventory or general services availability; production planning information regarding sales, time horizon, order due dates or prices; and actual plant state (Korovessi & Linninger, 2006). As a result, the scheduling function determines the amount of each product to produce, the allocation of equipments and resources to tasks, as well as the sequencing and timing of such tasks, in order to fulfill certain objectives (Figure 2.1).

The complexity and variety of the scheduling problems requires effective organization of the aforementioned information in order to improve communication and process efficiency. For example, Zentner *et al.* (1998) proposed a high level language which aims at expressing process scheduling problems, but lacks of a standard terminology which could be widely adopted in the process industry community by consent. Another approach is given by the ANSI/ISA88 (International Society for Measurement and Control, 2001), which is an IEC standard, approved by the International Society of Automation in 1995. Such standard presents a common, consistent model for design and operation of batch manufacturing processes and batch control systems (Barker & Rawtani, 2005). In such standard, the process recipe is defined with different degree of accuracy according to its scope, for example the site recipe for the production site, the master recipe for the production process and the control recipe for the control procedure. Therefore, the ANSI/ISA88 represents a standardized information structure, which embraces scheduling problem definition and may improve the availability, com-

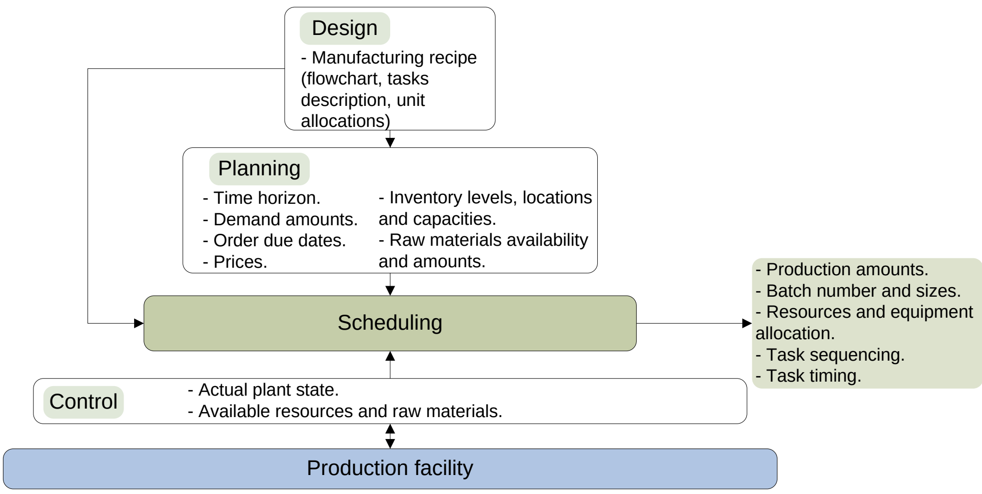


Figure 2.1: Description of the scheduling information and decisions.

munication and coordination of data between different decision levels and the models behind the corresponding decision support tools.

A wide range of criteria can be used to classify the scheduling problems as demonstrated in the following paragraphs. Furthermore, different alternatives are available in the literature for modeling the scheduling problem depending on its features as presented in section 2.3. Next, the most relevant characteristics of the scheduling problem are described (Floudas & Lin, 2004, 2005; Méndez *et al.*, 2006).

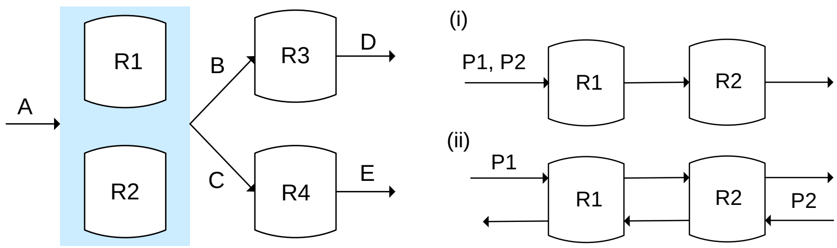
Material routing. One of the most important criteria to classify the scheduling problem consists of the process topology which determines the material routing. Therefore, a rough classification relies on the number of stages that raw materials must fulfill to be transformed into products, namely it may be *single stage* or *multistage* process.

Thus, the process may be classified as *network* (Figure 2.2(a)), in which case the output of multiple batches, or tasks, can be merged to form the input of a subsequent batch or the output of a single batch can be split to be consumed by more than one batch. In contrast, a *sequential* process maintains batch integrity, that is, the output of a batch can only be consumed by a single batch and the input of a batch can only be the output of another batch. Sequential processes may be further classified as *multiproduct* (Figure 2.2(b)(i))(flow-shop), if the sequence of tasks is the same for all products; or *multipurpose* (Figure 2.2(b)(ii)) (job-shop), if the sequence is different among products.

While most chemical processes can be treated as networks, there are cases where batch integrity must be maintained, such as the pharmaceutical, food or biotechnology processing industry. However, the previous traditional division between network and sequential topology is not strictly valid in many cases, because mixed topologies also exist. Hence, this classification should be related to processing tasks rather than the entire facility (Sundaramoorthy & Maravelias, 2011).

In order to represent the production sequences of the chemical processes, two general graph frameworks are available, namely the State-Task Network (STN) and the Resource-Task Network (RTN) (Figure 2.3). The STN representation was proposed by Kondili *et al.* (1993) and consists of a directed graph with two types of nodes: (i) the state node, denoted by a circle representing raw materials, intermediate materials and final products; and (ii) the task node, denoted by a rectangle box representing an operation. For the network processes in which a fraction of a state is consumed or produced by a task, such value is given along with the arch linking the corresponding state and task nodes. Later, Pantelides (1994) extended such framework to the RTN representation in order to describe processing equipment, storage, material transfer and utilities as resources in a unified way. The novelty of this representation is that resources are denoted by ellipses, and those tasks taking place in different units are treated as different tasks.

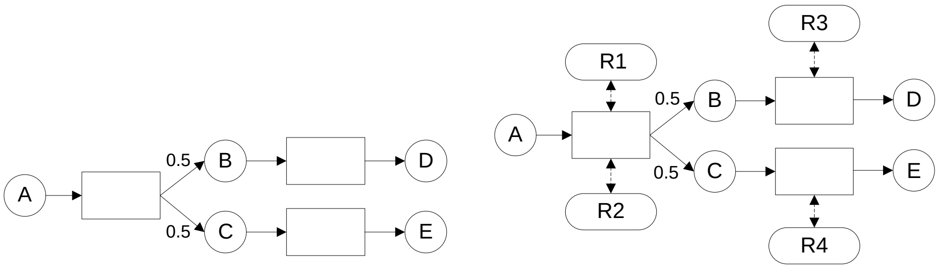
Operation modes. The processing tasks can be classified according to the way input and output products are fed and discharged. Therefore, in a *batch task*, materials are fed at the start of the task; and after a certain time, products are produced at once at the end. In a *continuous task*, materials and products are produced continuously along the time period of the task, and processing rate can be either fixed or within a certain range. Finally, a *semi-continuous task* is a mixture of the other two kinds of



(a) Network process.

(b) Sequential processes: (i) multistage, (ii) multipurpose.

Figure 2.2: Comparison of network and sequential processes.



(a) STN representation.

(b) RTN representation.

Figure 2.3: Graph representations for network process in Figure 2.2(a).

tasks, that is, either the raw materials or the products are fed/discharged continuously; whereas the other is loaded/discharged at a time. A series of lots of the same product run consecutively on a production line is called production campaign. In contrast, one batch each of many products may be also run on line.

Equipment assignment and connectivity. According to equipment availability, its type, capacity and production rate, it may be possible to assign either multiple units or only a single unit to a specific task. Therefore, the assignment may be *fixed* or *flexible*. Thus, the *interconnections* between equipments may impose strict *constraints* on allocation decisions.

Storage policies. They are an important factor which greatly affects the scheduling results. Depending on the product and intermediates nature and stability, they may be either stored in storage vessels, the equipment units or not stored at all. Therefore, different storage policies can be adopted depending on time and space limitations: (i) *zero-wait* (ZW), the intermediate product must be consumed immediately after processing; (ii) *no intermediate storage* (NIS), if there is no storage tank available for the intermediate materials, but they can be temporally held in the processing unit after processing and before being transfered to the next unit; (iii) *finite intermediate storage* (FIS), there is a limited storage capacity available, and storage tanks may be either dedicated to a single product or unit, or shared by several of them, and the storage time may be limited as well; and, (iv) *unlimited intermediate storage* (UIS), the storage capacity and time is unlimited and there is no need to model such feature. From the storage vessel perspective, either single or multiple batches of a given product may be fed or discharged at a time.

Material transfer. In batch process industry, the time for material transfer can be regarded as negligible when compared to operation time. Hence, material transfer operations are often assumed *instantaneous*, but in some cases they may result in resource consuming tasks, such as in pipe-less plants where vessels are used for transfer. In such cases, transfer time *duration* is significant and must be considered in the modeling approaches.

Changeovers and set-up operations. Changeovers and set-up operations needed between two tasks for safety or quality reasons in a given equipment stand for a very important factor in the scheduling results. Changeovers may be classified as *time dependent* or *sequence dependent*. In the former case, after a certain amount of time (or batches), a changeover operation must be performed, whereas in the latter the operation duration depends on the sequence of products and units involved. In contrast, set-up operations do not depend on the sequence but only on the unit and/or product. Alternative changeover cleaning methods with different times and characteristics may be available in the plant. However, the modeling of alternative methods, which may modify the scheduling decisions, has not been specifically dealt in existing formulations as shown in Chapter 4. In that chapter, an existing model has been adequately extended to consider such feature.

Batch size and processing time. An important factor to take into account is the batch size, which may be either fixed or variable. In some cases, such as in pharmaceutical plants, the batch integrity must be maintained and the batch size is fixed to a certain value. However, the use of variable batch sizes provide the production process with higher flexibility. Similarly, processing times may be either fixed and depend on the unit, or variable and depend on the unit, batch size and operating conditions. Thus, as mentioned before, processing times are usually established in a optimization design stage, previous to the scheduling problem.

Demand pattern. This feature highly depends on customers' requirements. Therefore, product demands may be either specified at the end of a given *time horizon* under consideration, or product orders must be fulfilled at specific time instances within the time horizon, which are named as *due dates*. In the former case, the demand fulfillment may be either compulsory or flexible, that is, the quantity of demand to be satisfied is fixed, or ranges between a minimum and maximum amount.

Resource constraints. In addition to equipment and raw materials, tasks may need other resources, such as labor, steam, cooling water or electricity. These resources may be either continuous or discrete, and can be classified as reusable, if they are completely recovered after the task finishes, such as manpower; or non-reusable, for instance energy. Therefore, resources needs and availability is an important feature, since the maximum amounts or use rates can never be exceeded at any time during production.

Time constraints. Such constraints may arise when considering *interruptions* in the time horizon, which may stem from maintenance, unit shifts or non working periods. In addition, it is necessary to consider *synchronization* and precedence regarding task operations, such as the discharge and load of equipment units or the simultaneous operation of two consecutive tasks.

Objective functions. The results of the scheduling problem are highly conditioned by the pursued objective function. Therefore, typical objective functions are time related: (i) *makespan minimization* aims at finding the minimum completion time of the whole process, given a production requirement; (ii) *earliness/tardiness minimization* seek for reducing the deviation from the specified due dates of the demand orders; or may be cost related: (iii) *cost minimization* consists of finding the optimal schedule for a given demand requirement, considering costs, such as equipment, utilities, changeover or inventory costs; or (iv) *profit maximization* pursues the highest economic value in a specified time horizon, given the available equipment and plants resources. Nevertheless, the functionality of scheduling can be widened as presented in Section 2.6.

Uncertainty. The plant data, such as production rates or customers demands, may be fixed and known with certainty, in which case the scheduling problem is classified as *deterministic*. However, such data are often uncertain, specially the demands for long time periods. If such factor is considered, the scheduling problem is *stochastic*.

2.3 Classification of scheduling models

In general, models can be classified according to the way that information is represented. Therefore, a general classification distinguishes between qualitative and quantitative models. The former represent the physical and logic relationships among the elements of the system and describe the reality, such as conceptual or semantic models; whereas the latter allow to make decisions based on actual data regarding the system, such as mathematical or statistical models. This thesis considers the mathematical representation of the scheduling problem in order to make decisions. Next, the main mathematical models proposed in the literature are briefly described next, and other kinds of existing models for scheduling problems are briefly outlined.

2.3.1 Mathematical models

A high number of mathematical models has been proposed in the literature in order to adequately formulate scheduling problems. However, each modeling option is only able to cope with a subset of the features described in section 2.2. The choice of the mathematical model has an important impact on computational performance. Hence, the model capabilities and limitations must be carefully considered for each scheduling problem. Recent reviews describe the most used mathematical formulations and compare their features and performance (Floudas & Lin, 2004, 2005; Méndez *et al.*, 2006; Shaik *et al.*, 2006; Pan *et al.*, 2009). In general, scheduling models may be classified according to the following three criteria:

- **Time representation.** This key feature of scheduling models classifies them as *discrete time* and continuous time formulations. The former category divides the time horizon into a number of time intervals with predetermined duration and the beginning or ending of tasks are allowed to happen only at the boundaries of these time periods. Essentially, this representation is an approximation of time, and scheduling constraints are only monitored at specific and known time points. As a result, the problem complexity is reduced, and the model structure is simpler and easier to solve, particularly when resource and inventory limitations are considered. However, in order to achieve a suitable approximation of the original problem, the time interval must be sufficiently small, such as the greatest common factor (GCF) of the processing times. Such feature usually leads to very large combinatorial problems of intractable size, especially for real-world problems. In general, this representation has proved to be very efficient and convenient for those cases where a small number of time intervals is sufficient. In order to overcome the drawbacks of the discrete-time formulation, researchers have developed *continuous-time* models, in which timing decisions are explicitly modeled as a set of continuous variables whose values define the exact times at which events take place. The exact definition of event varies from one formulation to another and may be associated to a unit or defined globally. Therefore, a reduction of the number of variables of the model may be achieved since a major fraction of the inactive event-time interval assignments are avoided, and more flexible solutions can be generated. However, resource and inventory limitations require more complicated constraints for the continuous-time representation, which increases the complexity of the model structure, and affects the capabilities of the method to obtain the optimal solution. According to the material balances, this latter approach can be further classified in sequential, which is batch oriented and does not require explicit mass balances, and network based.

- **Material balance representation.** The handling of batches and batch sizes gives rise to two kinds of approaches. On the one hand, a monolithic approach, which simultaneously deals with the optimal number and size of batches, allocation and sequencing of resources, as well as the timing of processing tasks. Such models are able to cope with network processes containing complex recipes. However, large models are obtained, whose applicability is typically restricted to a small number of processing tasks and short scheduling horizons. On the other hand, a sequential approach comprises models that assume that the number of batches of each size is known in advance. Therefore, the problem is decomposed in two stages, namely batching and batch scheduling, so larger problems can be addressed. Nevertheless, this approach is still restricted to processes comprising sequential product recipes.

- **Event representation.** Scheduling models are based on different concepts that arrange the events of the scheduling function over time with the main purpose of guaranteeing that the maximum capacity of the shared resources is never exceeded. According to Méndez *et al.* (2006), five different types of event representations are available, which are oriented toward the solution of either arbitrary network processes or sequential batch processes. Global time intervals are defined for discrete time formulations, and consist of a common and fixed time grid valid for all shared resources, in which batch tasks are enforced to begin and finish exactly at a point of the grid. For continuous time and network processes, both global time points and unit-specific time events can be used. The former representation is similar to global time intervals, where the timing of time intervals is a new model variable, so a common and variable time grid is defined for all shared resources. In contrast, unit-specific time events define a different variable time grid for each shared resource. The usefulness and efficiency of the global time points and unit-specific time event formulations depend greatly on the number of time or event points. On the other hand, for continuous time and sequential processes, the concepts of time slots and batch precedence have been developed. The idea of time slots consists of a set of predefined time intervals with unknown durations. As a result, an appropriate number of time slots for each processing unit is obtained in order to allocate them to the batch tasks to be performed. Slot-based representations may be either synchronous, if slots are identical across all units, or asynchronous, if slots differ from one unit to another. As for batch precedence, three main approaches are distinguished, namely the immediate precedence, general precedence and the unit-specific immediate precedence. The aforementioned approaches enforce the sequential use of the shared resources by means of model variables and constraints.

The mathematical models which are more widely applied in the literature are described along the remaining paragraphs of this subsection, and some recent applications are also presented. Table 2.1 contains the summary of the general characteristics of such models along with the critical problem features. For the sake of brevity, the comparison between the formulations is restricted to the aforementioned table.

Table 2.1: Features of the mathematical programming scheduling formulations.

	Time representation		Type of processes				
Event representation	Discrete	Continuous	Network	Sequential	Critical modeling issues	Critical problem features	References
Global time intervals	x		x		Time interval duration and scheduling period	Variable processing times and sequence dependent changeovers	Kondili *et al.* (1993); Shah *et al.* (1993); Pantelides (1994); Castro *et al.* (2002, 2003); Maravelias and Grossmann (2003a); Amaro and Barbosa-Póvoa (2008); Castro *et al.* (2008)
Global time points		x	x		Number of time points estimation	Intermediate due dates and raw material supplies	Pantelides (1994); Mockus and Reklaitis (1999b); Castro *et al.* (2001, 2004); Giannelos and Georgiadis (2002); Maravelias and Grossmann (2003b); Maravelias (2005); Maravelias and Grossmann (2006); Castro and Novais (2009); Castro (2010)
Unit-specific time events		x	x		Number of time events estimation	Intermediate due dates and raw material supplies	Ierapetritou and Floudas (1998); Vin and Ierapetritou (2000); Lin *et al.* (2002); Janak *et al.* (2004); Janak and Floudas (2008); Shaik and Floudas (2008)
Time slots		x	x		Number of time slots	Resource limitations	Pinto and Grossmann (1995); Sundaramoorthy and Karimi (2005); Erdirik-Dogan and Grossmann (2008); Susarla *et al.* (2010)
Unit-specific immediate precedence		x		x	Number of batch tasks sharing units and resources	Resource limitations and inventory	Cerdá *et al.* (1997)
Immediate precedence		x		x	Number of batch tasks sharing units and resources	Resource limitations and inventory	Gupta and Karimi (2003)
General Precedence		x		x	Number of batch tasks sharing resources	Inventory	Mendez *et al.* (2001); Mendez and Cerda (2004); Ferrer-Nadal *et al.* (2007)

STN-based discrete formulation. This formulation was initially proposed by Kondili *et al.* (1993), and later extended by Shah *et al.* (1993). Maravelias and Grossmann (2003a) propose an algorithm to consider makespan as an objective function in such model, whose general features are specified in Table 2.1. A recent application of this formulation can be found in Amaro and Barbosa-Póvoa (2008), whose work improves the capabilities of this formulation in the scheduling domain in order to consider supply chain decisions.

RTN-based discrete formulation. The RTN-representation was first introduced by Pantelides (1994). The model features are given in Table 2.1. This formulation has been widely adopted in a large number of applications. For example, Castro *et al.* (2003) compare the computational performance of the discrete and continuous time RTN-based formulations of a periodic schedule in an acid sulphite pulp mill industrial application, and conclude that the discrete-time based performs better. In a previous work Castro *et al.* (2002), the authors develop an alternative model to the discrete-time based that includes the dynamic behavior of the batch digester operation. Furthermore, Castro *et al.* (2008) apply this formulation to a scheduling problem from a fine chemicals company that synthesizes active pharmaceutical ingredients, in which the main goal is to select from the set of available equipment units, those units that are better-suited for the production of the ingredient being considered. In all the aforementioned applications, the importance of modeling resource limitations is crucial.

STN-based continuous formulation. There have been many approaches to the STN-based continuous time models in the last years, for instance Mockus and Reklaitis (1999b), Giannelos and Georgiadis (2002) and Maravelias and Grossmann (2003b). The latter approach allows to handle general batch process concepts such as variable batch sizes and processing times, various storage policies and sequence-dependent changeover times. Appendix B contains the general constraints of this model. Furthermore, Maravelias and Grossmann (2006) demonstrate that the continuous time STN-based formulation is a generalization of the discrete time model, since the latter can be derived when using a uniform time grid with constant processing times. As a result, a mixed-time representation for STN-based scheduling models was proposed (Maravelias, 2005), in which the time grid is fixed but processing times are allowed to be variable and span an unknown number of time periods. Such formulation can handle holding costs and intermediate due dates at no additional computational cost.

RTN-based continuous formulation. Based on the concept of RTN-based representation introduced by Pantelides (1994), Castro *et al.* (2001) improved earlier attempts towards the RTN continuous time formulation. Their model was later revised by Castro *et al.* (2004), who introduced the handling of continuous tasks, a more efficient set of constraints to deal with zero-wait storage policies, and a set of timing constraints that improved the linear relaxations of the formulation. The major assumptions of this model as well as the main equations proposed by Castro *et al.* (2004) are included in Appendix B. Some instances of applications of this formulation are provided by Castro (2010), whose previous formulation is extended to the scheduling of multiproduct pipeline systems, and can handle multiple discrete due dates as well as continuous demand rates, and the work by Castro and Novais (2009) where a new model considering this formulation based on 4 index binary variables is built, for multistage plants with multiple product batches and sequence-dependent changeovers.

Unit-specific time event. The original idea of unit-specific events was firstly presented by Ierapetritou and Floudas (1998) and then developed by Vin and Ierapetritou (2000), Lin *et al.* (2002) and Janak *et al.* (2004). It consists of a flexible representation of the scheduling problem which is able to account for different intermediate storage policies and other resource constraints. The global time events representation is efficiently reformulated using two approaches: 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. By doing so, the number of event points and associated binary variables are reduced compared to the global time points representation. Appendix B contains a brief description of the formulation introduced by Janak *et al.* (2004). This formulation is further developed by Janak and Floudas (2008), by including tightening constraints and the idea of partial task splitting; whereas Shaik and Floudas (2008) extend the formulation to handle dedicated finite storage without the need for considering storage as a separate task.

Time slots. Time slots consist of a set of predefined time intervals with unknown durations. The first contributions were aimed at sequential batch-oriented processes (Pinto & Grossmann, 1995). However, further attention was devoted to this formulation, for example, Sundaramoorthy and Karimi (2005) presented an extension to deal with network batch processes, Erdirik-Dogan and Grossmann (2008) incorporate slot-based mass balances and account for sequence-dependent changeovers and Susarla *et al.* (2010) modify the earlier model of Sundaramoorthy and Karimi (2005) in order to use unit-slots, handle shared resources and model several storage policies.

Precedence-based formulations. Several formulations based on the precedence concept have been introduced. Depending on the immediate or general batch predecessor basis, three different models are proposed in the literature, namely the unit-specific immediate (Cerdá *et al.*, 1997), the immediate (Gupta & Karimi, 2003) and the global (Mendez *et al.*, 2001; Mendez & Cerda, 2004; Ferrer-Nadal *et al.*, 2007) precedence models. The unit-specific immediate precedence model applies the concept of immediate precedence in each unit to formulate the problem using a binary variable $X_{ii'j}$ that states that a batch i is performed immediately before batch i' in unit j. In contrast to the unit-specific formulation, the immediate batch precedence model further divides the allocation and sequencing decisions in two different sets of binary variables. Finally, the global precedence model considers all batches processed before in the same processing sequence, thereby simplifying the model and reducing the number of sequencing variables. These formulations are mainly used for sequential processes because batch splitting and merging is difficult to be modeled with these formulations.

General framework. Recently, several authors have proposed generalized formulations which aim at encompassing the modeling of all types of scheduling problems. For instance, Westerlund *et al.* (2007) present a mixed-time representation of the problem, using a discrete-time model where a continuous-time representation is also incorporated. Such formulation is useful for modeling multi-stage multi-product production processes using intermediate storages with highly nonlinear profiles for large sized industrial problems, but still is limited in the representation of critical production events such as changeovers and does not include the representation of utilities consumption. Furthermore, Sundaramoorthy and Maravelias (2011) propose a common framework for modeling facilities containing sequential and network subsystems based on a discrete-time representation, using a material-based formalism and developing constraints that enforce batch integrity in sequential subsystems.

Insomuch the previous formulations have been continuously extended to consider additional process scheduling features, some non common features such as multiple changeover methods or non negligible transfer operations, which are highly case dependent, should still be tackled.

2.3.2 Graph-based models

The discrete nature of the scheduling problem allows to introduce the graph representation as a means of modeling. Therefore, a framework named as S-graph was introduced by Sanmartí *et al.* (2002). Its main advantage lies in the capability to exploit the problem structure to reduce the computational complexity. It represents all processing tasks of the recipe in a graph manner, namely a node is assigned to each task and for each product. In addition, the set of available equipment for each task is defined, and the processing of two consecutive tasks are joined by means of weighted arcs, whose value is the processing time of the task. Moreover, an additional arc is established from the node task generating the product and the corresponding product node. The resulting graph is formulated as a highly efficient MILP problem and solved by the branch and bound method. This framework has been applied to the problems of makespan minimization and throughput maximization, but major limitations are still to be addressed, such as the modeling of variable processing times or resource consumption (Hegyháti *et al.*, 2009).

Moreover, timed automata (TA), which are finite state automata (graphs containing finite set of locations and finite set of labeled transitions) extended by the notion of clocks to model discrete event systems with timed behavior, have been proposed to model the scheduling problem. The strength of this modeling approach lies on its graphical representation and modularity to model complex systems with ease and clarity. However, the TA-based approach also suffers from the problem of combinatorics and an important limitation is that only problems with discrete decisions over time in the broad sense, such as resource allocation or sequencing, can be addressed, but not problems containing decision variables with continuous degrees of freedom, such as batch sizing (Subbiah *et al.*, 2009).

2.3.3 Alternative modeling structures

From a more descriptive point of view, artificial intelligence provides with tools for representing the scheduling problem from an abstract conception. For example, chromosome strings of the genetic algorithm method allow to represent scheduling solutions (He & Hui, 2010).

Furthermore, semantic models, such as ontologies, offer the alternative to represent and share the knowledge of the process and engineering domains. In this sense, Munoz *et al.* (2010) develop an ontology-framework called BaPron based on the ANSI/ISA-88 standard which may be used as a straightforward guideline for standardizing batch process management and control.

2.4 Solution methods of the scheduling problem

In order to tackle real world scheduling problems, it is necessary to develop algorithms and computational architectures so that large-scale optimization models can be posed and solved effectively and reliably. Hence, the collaboration among different scientific disciplines, namely process systems engineering, operations research and artificial intelligence, is highly important (Grossmann, 2005; Rardin, 2000).

Grossmann and Biegler (2004) summarize the application areas in process systems engineering of different optimization methods. Thus, there have been several contributions to review optimization methods in general and in the area of process systems engineering in particular (Kallrath, 2002; Biegler & Grossmann, 2004; Grossmann & Biegler, 2004; Kallrath, 2005; Méndez *et al.*, 2006; Li & Ierapetritou, 2007; Caballero & Grossmann, 2007; Barbosa-Povoa, 2007). Particularly for process scheduling optimization, there is a wide variety of methods as shown by Méndez *et al.* (2006). Along the following subsections, a brief review of these methods in the area of scheduling and other decision making related functions is presented.

2.4.1 Mathematical programming

Méndez *et al.* (2006) review the short-term scheduling with mixed integer linear programming methods where the most common solution algorithms are linear programming-based branch and bound methods, which are enumeration methods that solve linear programming subproblems at each node of the search tree. The most used mixed integer linear programming methods correspond to branch-and-cut techniques in which cutting planes are generated at the various nodes of the branch and bound tree in order to tighten the linear programming relaxation. Biegler and Grossmann (2004) provide a general review on optimization for process systems engineering that have been extensively studied and applied, namely, nonlinear programming, mixed-integer nonlinear programming, dynamic optimization and optimization under uncertainty.

The challenges in process system engineering lead to the need to extend and reinvent optimization algorithms for large-scale problems. In particular, the areas with special interest that should be addressed are (Grossmann & Biegler, 2004):

- Problem size: for large-scale problems, many of the conceptual algorithms do not change. However, care is needed to deal with the scale-up of subproblems, particularly with the solution of linear systems. This scale-up is influenced by the size of the process model as well as the number of variables available for optimization (degrees of freedom).

- Growth of combinatorics: the combinatorics of algorithms are affected significantly by increases in problem size. In non linear programming, this is usually observed in finding optimal active constraint sets. In mixed integer non linear programming and global optimization algorithms, this is observed in the exponential growth of the branch and bound tree and the need to enumerate many more alternatives.

- Effects of problem structure: with increases in size, it is imperative to exploit specific problem structures. In this sense, Durand and Bandoni (2010) propose the introduction of integer cuts that exploit the structural characteristics of existing RTN/STN-based formulations in order to improve the computational performance of the MILP approaches. Likewise, the application of decomposition strategies that solve a series of smaller subproblems can lead to better computational results. There is often an opportunity (e.g. with convex problems) to generate bounds on the optimal solution in order to accelerate convergence. Hence, rigorous mathematical decomposition strategies may be used to improve the solution process of the problem (Conejo *et al.*, 2006). For example, approaches based on spatial or temporal decomposition usually rely on Lagrangean decomposition. In the case of spatial decomposition, the idea is to use links between subsystems by dualizing interconnection constraints, whereas in temporal decomposition, inventory constraints are dualized in order to decouple the problem in time periods. The main drawback of decomposition approaches is that there is rarely an indication on the quality of the solutions.

2.4.2 Logic-based methods

One of the emerging areas related to discrete optimization is logic-based optimization. The major motivation in this area lies in developing symbolic representations that can facilitate the modeling of discrete constraints, and motivate more effective solution techniques that can help to reduce the computational complexity of discrete/continuous optimization problems. Recently, the development of different logic-based optimization techniques has arisen (Grossmann & Biegler, 2004).

Constraint programming (CP) is an alternative approach to discrete and continuous problem solving that was developed in the computer science and artificial intelligence communities. It has proved to be successful in several applications, particularly in scheduling and logistics. It does not use relaxations in the way that mathematical programming uses them, but it applies sophisticated methods of logical inference (primarily domain reduction and constraint propagation) to reduce the domain of possible values a discrete or continuous variable may take. Méndez *et al.* (2006) define the main features of this approach, and state that its main application consists of combining it with MILP techniques, resulting in hybrid methods as presented in Subsection 2.4.6. A recent work by Kotecha *et al.* (2010) presents efficient strategies for constraint programming.

2.4.3 Heuristics

There are several heuristics called dispatching rules which are considered as construction heuristics. These rules use certain empirical criteria to prioritize all the batches that are waiting for processing on a unit. For simple scheduling problems, they have demonstrated to have very good performance, although their efficiency is usually evaluated empirically and their applicability is usually very case specific. The usefulness of dispatching rules is still limited to quite a narrow variety of scheduling problems and optimality can be proved only in some special cases since these methods cannot guarantee the quality of the solution. In addition, they are considered very fast and easy to implement. Some relevant dispatching rules are first come first served, earliest due date, shortest processing time, longest processing time, earliest release date or weighted shortest processing time. Often, composite dispatching rules involving a combination of basic rules can perform significantly better. Besides, dispatching rules can be easily embedded in exact models to generate more efficient hybrid approaches for large-scale scheduling problems. An extensive review and a classification of various dispatching rules can be found in Panwalkar and Iskander (1977).

Recent contributions to this field are presented for example by Pan *et al.* (2008) who propose a precedence based formulation and four heuristic rules based on experience of production to reduce the number of binary variables and tackle complex scheduling problems of multipurpose batch plants. Thus, Shafeeq *et al.* (2008b) present an algorithm based on simple mathematical formulas to quickly calculate the makespan for all batch production sequences derived from specified batch process recipes for multiproduct batch plants considering zero wait and no intermediate storage policies, and later the authors extended this work to further consider intermediate storage (Shafeeq *et al.*, 2008a).

2.4.4 Metaheuristics

A metaheuristic is an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space (Blum & Roli, 2003). Methods based on meta-heuristics, also known as local search methods, are often inspired by moves arising in natural phenomena. These methods have the advantage of easy implementation and require little prior knowledge of the optimization problem. In particular, such methods are well suited for fast optimization studies that explore the scope of optimization for new problems, prior to investing effort for more sophisticated modeling and solution strategies (Biegler & Grossmann, 2004). However, these algorithms also have significant drawbacks since they do not provide any guarantee on the quality of the solution obtained, require considerable problem customization and it is often impossible to tell how far the current solution is from optimality. Furthermore, these methods do not require to formulate the problem as a mathematical program since they involve procedural search techniques that in turn require some type of discretization or graph representation of the model variables, and the violation of constraints is usually handled through ad hoc penalty functions. For that reason, the use of meta-heuristics based on local search methods might be problematic for problems involving complex constraints and continuous variables. Several examples include genetic algorithms, hill-climbing, particle swarm optimization (Liu *et al.*, 2010) or ant colony algorithm (Jayaraman *et al.*, 2000).

The genetic algorithm is based on the analogy of improving a population of solutions through modifying their gene pool. Two forms of genetic modification, crossover or mutation, are used and the elements of the optimization vector are represented as binary strings. Crossover deals with random swapping of vector elements (among parents with highest objective function values or other rankings of population) or any linear combinations of two parents. Mutation deals with the addition of a random variable to elements of the vector. Genetic algorithms (GAs) have seen widespread use in process engineering and a wide number of codes are available.

The simulated annealing derives from a class of heuristics with analogies to the motion of molecules in the cooling and solidification of metals. Here, a temperature parameter, can be raised or lowered to influence the probability of accepting points that do not improve the objective function. The method starts with a base point, and objective value. The next point is chosen at random from a distribution. If the objective function improves, the move is accepted with the initial point as the new point. Otherwise, the point is accepted with certain probability.

In general, metaheuristics are widely applied in industrial scheduling (Xhafa & Abraham, 2008) for complex problems. For example, He and Hui (2007) present a heuristic approach based on GA for solving large-size multi-stage multi-product scheduling problem in batch plants. Méndez *et al.* (2006) cite several works related to the application of the aforementioned techniques to the scheduling problem.

2.4.5 Artificial intelligence

Artificial intelligence techniques have also been widely applied to scheduling (Metaxiotis *et al.*, 2002). In order to use more efficiently the process information as well as the essential knowledge provided by human schedulers, artificial intelligence mimics human thought and cognitive processes to solve complex problems automatically.

Different techniques mimic the different ways that people think and reason. For instance, case-based reasoning solves a problem by retrieving the solution to previous similar problems and altering those solutions to meet the current needs. It is based upon previous experiences and patterns of previous experiences. On the other hand, model-based reasoning concentrates on reasoning about a system behavior from an explicit model of the mechanisms underlying that behavior.

Within the artificial intelligence field, agent-based approaches are software programs that are capable of autonomous, flexible, purposeful and reasoning action in pursuit of one or more goals. They are designed to take timely action in response to external stimulus from their environment on behalf of a human.

Moreover, artificial neural networks construct networks in which the weighting parameters are determined from training data. For scheduling applications (Strojny *et al.*, 2006), such training requires the availability of many good schedules, but suffers from poor performance of neural nets as extrapolation models.

2.4.6 Hybrid methods

Significant computational savings may be achieved if the previous methods are combined. Wu and Ierapetritou (2003) use an hybrid approach that merges a number of different heuristic-based decomposition approaches including time-based decomposition, required production method and resource-based decomposition. Lagrangean relaxation and Lagrangean decomposition are then employed to yield an upper bound to the original scheduling problem. Finally, an iterative solution framework is proposed to exploit the lower bound obtained through the heuristic-based approaches and the upper bound based on the Lagrangean relaxation and decomposition, which ultimately provides a refined schedule for large-scale scheduling problems.

Blomer and Gunther (2000) present a mixed-integer linear programming model for scheduling chemical batch processes with a two-stage solution procedure. In the first stage, an initial solution is derived by use of a linear programming-based heuristic. The proposed heuristic relies on a time grid that includes only a limited number of feasible periods in which a processing task is allowed to start. Thus, the size of the original multi-period mixed integer linear programming model is reduced in a controlled manner and optimal solutions to the relaxed model are obtained within reasonable computational time. The second stage consists of an improvement step that aims to compress the initial schedule by left-shifting operations over the time-axis. Burkard and Hatzl (2006) investigate a heuristic for batch processing problems occurring in the chemical industry, aiming at makespan minimization, proposing an iterative construction algorithm which alternates between construction and deconstruction phases.

The combination of CP and MILP has received increased attention for their complementarity. Specifically, important computational savings in many industrial applications have been reported in the literature (Harjunkoski & Grossmann, 2002; Maravelias & Grossmann, 2004; Zeballos *et al.*, 2011).

2.5 Integration of decision making in enterprise structure

Recent trends in process industries are shifting the focus from controlling the process plant as a stand-alone entity toward managing it as an integral part of a larger system (Klatt & Marquardt, 2009). Such approach aims at exploiting the process and environment dynamics in order to maximize the plant economic indicators. Obviously, such understanding of process management entails the integration of the different decision level functions. Therefore, a current important challenge lies on the coordination of the decision making and the optimization of different decision levels, both vertically across a single process plant, and horizontally along the different geographically distributed subsystems of the supply chain in a given time horizon.

One first step toward such integration consists of the sharing of information, which is nowadays being achieved with modern IT tools, such as SAP and Oracle, that allow the instantaneous flow of information along the various organizations in a company (Grossmann *et al.*, 2008). However, a better understanding, structuring and even modeling of the whole process is necessary for an effective transformation of the information into knowledge. In this line, several standards are used in enterprises in order to improve their efficiency and flow of information, such as CAPE-OPEN, the ANSI/ISA 88 or the ANSI/ISA 95. Thus, semantic technologies seem to offer an appealing way to capture knowledge and integrate information, for supporting a smooth integration of information and mathematical modeling in a single modeling framework (Klatt & Marquardt, 2009).

Another major issue consists of the modeling and optimization approaches for integration among decision levels. In fact, the border lines between the decision-making levels of the enterprise structure are often diffuse, and there are even strong overlaps between planning in production, distribution or supply chain management and strategic planning (Kallrath, 2005). Thus, simultaneous optimization approaches for the integration of the entire supply chains naturally lead to the definition of centralized systems. However, operation tends to take place in practice, as if the supply chain was decentralized. Hence, coordinated procedures are needed, that can maintain a certain degree of independence of subsystems, while at the same time aiming at the objectives of integrating optimization of the overall system (Grossmann *et al.*, 2008). In addition, the development of procedures that can effectively work across large spatial and temporal scales, as well as global model-based optimization techniques

(Grossmann, 2005) are crucial for attaining global solutions.

From a process scheduling perspective, recent reviews highlight the actual importance of its integration both upward (Kallrath, 2005; Grossmann *et al.*, 2008; Maravelias & Sung, 2009), that is mid- and long-term planning, and downward (Grossmann *et al.*, 2008; Harjunkoski *et al.*, 2009), that is process and control, within the decision level pyramid. Therefore, the objective of plant operation is moving from controlling the plant at its set-point toward optimizing its performance in real-time subject to process, environmental, and others (such as quality and services) constraints (Klatt & Marquardt, 2009). Since this thesis tackles the integration of the scheduling problem with the control level, the next subsections basically focuses on such integration and the integration with planning is only hollowly described.

2.5.1 Scheduling and control

Recently, Harjunkoski *et al.* (2009) present the issue of scheduling and control integration from an industrial and academic perspective. The authors state that the scheduling and control functions aim at filling the gap between enterprise resources planning (ERP) and operations by ensuring that business targets are correctly transferred to the production level. On the one hand, the scheduling function is directly related to the planning level, since production objectives and time horizon are compulsory data for scheduling. On the other hand, the control level is directly related to product quality, equipment monitoring and other equipment related activities.

In order to manage the previous systems and apply optimization theory, the use of standards can be adequate, since they allow re-usability of the solutions and components, improve the connectivity with vendors and provide with an unified data structure. In the scheduling and control scenario, the standard ANSI/ISA88 (IEC61512) (International Society for Measurement and Control, 2001) was approved in 1995 by the International Society of Automation, a consortium of academics and industrialists; their purpose was to overcome the existing difficulties in batch automation. This standard provides a framework that an engineer can use to specify automation requirements in a modular fashion and can be used for integrating batch-related information and formalizing the description of the scheduling and control decision levels and the whole production plant, including data, information and knowledge required for the decision-making. However, the standards do not specify how the implementation should be done, and leave a large number of unanswered questions.

The integration between control and scheduling should emerge from the functionality point of view (Harjunkoski *et al.*, 2009), and should fulfill the workflow tasks within the production planning environment; in which, scheduling contains the production schedule, the actual production information and the production capacity, whereas the control function is related to the tasks regulated by the process control system.

Three main problems have been identified when aiming at the integration of the control and scheduling levels according to Harjunkoski *et al.* (2009):

- *Methodological aspects.* The amount of details and problem complexities increase towards the control direction, and most of the methods used in the lower level can be applied to handle upper level complexities. Therefore, the time granularity and level of detail are the two main issues that must be tackled when facing integration.

- *Information transfer.* The information flow between the control and scheduling systems must be standardized, and it can be either unidirectional or bidirectional, which is the most common and challenging case.

- *Modeling approaches.* The way the scheduling and control problems are formulated and solved differs widely. In fact the modeling approaches are not straightforward compatible with each other, and it is difficult to capture efficiently all the problem aspects. At lower level, an important question is how to couple scheduling models with process models, and particularly with dynamic models that can rigorously predict the optimal control (Grossmann *et al.*, 2008). With today's knowledge and tools, both levels cannot be fully merged, but need to jointly find better and more natural ways of collaborating. Therefore, it is crucial to focus on the actual needs for optimization, and on the understanding of the system, in order to obtain the trade-offs between shorter solution times and the solution quality.

The solution of the control and scheduling problems in an isolated manner leads to suboptimal results. Hence, the amount of academic contributions to this specific field of integration has increased along the last years, although it is still very scarce. In general, the modeling approaches to the scheduling problem can be either mixed integer linear programming (MILP) or mixed integer non linear programming (MINLP); whereas the control problem consists of the optimization of dynamic algebraic equations (DAE). As a result, the integration problem gives rise to a mixed-integer dynamic optimization model (MIDO). Therefore, depending on the solution approaches to tackle the integration problem, the contributions found in the literature can be classified as: (i) heuristic based methods; (ii) decomposition based methods, either Benders or Lagrangean decomposition; and (iii) transformation of the MIDO into large MILP/MINLP problems.

The heuristic based approach for the integration basically consists of agent-based systems. In this area, Lim and Zhang (2003) present a multi-agent system for integrating scheduling and planning. The agents share data and results, and allow the optimization of the manufacturing resources dynamically. Although the control function is not included, it could be easily accounted, but would increase the problem complexity. In the work of Musulin *et al.* (2005), a multi-agent system is applied to batch process on-line scheduling based on the standard ANSI/ISA 88 for closing the loop for process robustness, fault diagnosis, recipe coordination and exception handling. In addition, Pawlewski *et al.* (2009) present a multi-agent for production planning, scheduling and control in order to overcome the limitations of material resource planning (MRP) and ERP systems regarding the differences in time and scope. Their conceptual framework involves solutions that synchronize all production and material flows.

As for decomposition based strategies, Nystrom *et al.* (2005) present a method for optimizing the scheduling and control problem for the grade changes in polymerization processes which consists of decomposing the problem into a master and primal problem. The former considers scheduling related decisions, whereas the latter performs the dynamic optimization. Such problems are solved iteratively updating the linking key parameters. Their decomposition strategy stands for an alternative to Benders or Lagrange decomposition methods, but it is highly application-dependent. Alle and Pinto (2002) use the outer approximation method to solve the MINLP problem of the simultaneous scheduling and optimization of the operation conditions of continuous multistage multiproduct systems with intermediate storage considering product campaign transitions.

An additional approach to the simultaneous control and scheduling consists of the transformation of the MIDO problem into a MINLP, such as proposed by Flores-Tlacuahuac and Grossmann (2006). In their work, the authors explicitly incorporate the process dynamics into the cyclic scheduling problem of a continuous polymerization plant in an iterative scheme. Their work was further extended by Terrazas-Moreno *et al.* (2007) in order to explicitly formulate transition times. An alternative approach was proposed by Prata *et al.* (2008) who integrate control and scheduling for a continuous polymerization process using General Disjunctive Programming.

In general, works involving integration of optimal control profiles of batch scheduling problems are still very scarce. The main difficulties are related to the high computational requirements to solve the final MIDO problem, the lack of appropriate process models and the large time needed to develop and reduce process models which require both sufficient detail to represent the process and enough simplicity to be solved by the available optimization tools (Bhatia & Biegler, 1997).

One of the very first works to consider the full process dynamics in the batch scheduling problem was presented by Bhatia and Biegler (1996). They include dynamic models of processing tasks within the design and scheduling formulation for flowshop batch plants with unlimited intermediate storage and zero wait transfer policies with one unit per stage. The dynamic process models are discretized through collocation on finite elements; and the authors prove that dynamic process considerations can contribute significantly to increase profitability. Their work was further extended in Bhatia and Biegler (1997) incorporating process model uncertainty, but as a result the problem size increases considerably.

On the other hand, Mishra *et al.* (2005) broadly classify the scheduling problem in two categories, namely the standard recipe approach and the overall optimization approach. The former defines a two-step strategy, in which a recipe is established either empirically or by single batch optimization; and next, the scheduling problem is posed on the basis of these fixed standardized recipes. The latter approach directly includes process dynamics in the scheduling problem restoring degrees of freedom in the problem. The authors compare both approaches for a single product plant and a multiproduct plant. Standard recipes are modeled as polynomials that relate duration and reaction heat to the processed quantities. The better performance of the overall optimization approach in terms of solution quality is demonstrated, but its major drawback is the large size of the resulting problems and the computational difficulty in solving large-scale problems.

Therefore, the complete integration of scheduling and control into large-scale MIDO problems is probably only achievable in certain selected cases, where the complexity of the dynamic model is not high Harjunkoski *et al.* (2009). In this sense, possible solutions to the integration problem may probably remain in the domain of including in the scheduling problem some indicators related to the dynamic part of the problem. Therefore, some work has been presented in the literature, which may be regarded as intermediate strategies, since they define processing times as a function of batch sizes, state variables or approximations to the dynamic model Bhatia and Biegler (1996). Castro *et al.* (2002) solve a scheduling problem based on the discrete-time RTN formulation whose processing times are estimated by the dynamic models of an industrial batch digester cooking system constrained by heating utility. The authors conclude that although the scheduling model cannot consider some effects in the plant, it stands for a powerful tool when combined with the dynamic model for the decision making of the plant. Romero *et al.* (2003) present a framework that includes the possibility of recipe adaptation in the optimization of batch processes. In their work, a linear-based recipe model is integrated into the S-graph model and productivity maximization is established as objective function, considering negligible the cost of modifying process variables. Ferrer-Nadal *et al.* (2008) incorporate the concept of recipe flexibility as an additional rescheduling action in the reactive batch operation of multipurpose batch plants. They assume a linear model in a predefined flexibility region around nominal operating conditions, penalize any deviation from the optimal operating conditions and solve a MILP based on the general precedence model (Mendez *et al.*, 2001). The aforementioned authors assume that there is a single optimal nominal recipe, and models are assumed to be linear around such operating point.

In general terms, the integration of scheduling and control requires intensive involvement of other areas, and their collaboration for improving modeling and optimization approaches and software architectures. Thus, the integration can be tackled by using adaptation through parameters, improving the information flow, the modularity, the data availability, the use of standards, and adopting general objective functions (Harjunkoski *et al.*, 2009).

2.5.2 Scheduling and planning

In the process industry, the planning function focuses on the creation of the production, distribution, sales and inventory plans based on customer and market information while observing all relevant process constraints. In particular, operational plans have to be determined which are aimed to structure future production, distribution and other related activities according to business objectives (Kallrath, 2005; Shah, 2005). According to Kallrath (2002), most of the planning problems in the process industry lead to mixed integer linear programming or mixed integer non linear programming models and contain the following building blocks: tracing the states of plants, modeling production, balance equations for material flows, transportation terms, consumption of utilities, cost terms and special model features. In fact, the planning model is a simplified representation that is used to predict production targets and material flows over several months. At this level, effects of changeovers and daily inventories are usually neglected, which tends to produce optimistic estimates that cannot be realized at the scheduling level (Grossmann *et al.*, 2008).

As stated by Kallrath (2002), the border lines between scheduling and planning are diffuse and there are strong overlaps between them. Furthermore, the integration between scheduling and planning is increasingly demanded so that facilities can respond quickly to demand fluctuations and better utilize the existing resources close to their capacity and actual needs. Hence, several authors highlight the importance of integrating the planning and the scheduling levels (Grossmann *et al.*, 2008; Maravelias & Sung, 2009; Verderame *et al.*, 2010) since economic incentives for such integration are substantial.

2.6 Functionality issues related to scheduling

As mentioned by Harjunkoski *et al.* (2009), the integration between scheduling and other decision levels should stem from the functionality point of view. Along the previous sections, the tasks corresponding to the scheduling problem have been defined along with the several solution approaches. Therefore, the functional issues related to scheduling, namely the different objective function approaches are presented next. In general, the scheduling level, as a main building block of the enterprise structure, pursues the overall company objectives which arise from economic, environmental and social aspects.

Economic criteria Resource consumption stems from process operation and can be expressed in terms of cost. For instance, set-up and changeover operations, inventory levels, raw materials and utilities consumption, such as electricity, steam, cooling needs or process water, entail important operational costs. In fact, scheduling results are heavily influenced by the structure of costs (Mishra *et al.*, 2005).

Nevertheless, economic criteria are often simplified to time related objectives. For instance, if late orders are highly penalized and other economic criteria may be disregarded, instead of the economic function, a time related criterion, namely lateness, could be directly adopted as the minimization objective. In contrast, if inventory costs should be minimized, production earliness minimization could be the selected objective. As a result, a wide number of objective functions Hoogeveen (2005) can be adopted in the scheduling problem, which may be divided in either economic, such as maximization of sales or profit over a fixed scheduling horizon or minimization of processing costs of given orders; or time-related objective functions, such as makespan, tardiness or earliness, for a given demand with release and due times.

Environmental concerns Environmental aspects are usually considered in the design of chemical processes due to pressure from regulation policies and a global trend toward sustainability in businesses (Clift & Azapagic, 1999). Hence, as a result of the increasing environmental concerns in chemical industry, more accurate approaches to assess process sustainability are required. Several authors highlight the importance of considering life-cycle assessment of production processes at process synthesis, product design and its integration with processing (Grossmann, 2004; Barbosa-Povoa, 2007). However, environmental considerations, such as waste minimization, material recovery or utilities consumption, in process scheduling have received little attention, and have been mainly dealt as an integrated part of the design phase of batch plants (Yao & Yuan, 2000; Melnyk *et al.*, 2001; Stefanis *et al.*, 1997; Al-Mutairi & El-Halwagi, 2010).

According to Diwekar and Shastri (2011), the most critical environmental issues in batch process scheduling are energy and waste management. Hence, significant effort has been initially devoted to energy management within batch processes. Two approaches, namely simultaneous and sequential, can be distinguished to integrate heat issues in the scheduling problem. The former consists of mathematical programming approaches which account for the optimal scheduling while considering environmental issues. For example, Majozi (2009) addresses the problem of simultaneously maximizing the profit and the intra-process heat transfer of multipurpose and multiproduct batch plants using mathematical programming. Two scenarios are presented, namely a situation in which energy requirement is dependent on the batch size resulting in a nonconvex MINLP, which is linearized to obtain a MILP problem, and a second scenario considering fixed batch sizes which is directly a MILP problem.

On the other hand, sequential approaches often deal with more complex problems where the simultaneous approaches are not possible. For instance, Halim and Srinivasan (2009) present a two stage approach based on the time slot continuous formulation of Sundaramoorthy and Karimi (2005), in which the optimal scheduling according to either economic or time criteria is optimized first; and next, additional schedules are generated using a stochastic integer cut procedure to the scheduling formulation. The obtained schedules are analyzed from a heat integration perspective to establish the minimum utility targets.

Nevertheless, environmental concerns should be integrally considered in the scheduling problem beyond energy and waste. In this sense, a wide range of process design frameworks have been proposed including environmental considerations, such as the methodology for obtaining minimum environmental impact processes (MEI, or MEI methodology) (Stefanis *et al.*, 1997) or the waste reduction algorithm (WAR) Cabezas *et al.* (1999) proposed by the United States Environmental Protection Agency (US-EPA) which uses the pollution balance concept and the environmental fate and risk assessment tool (EFRAT) (Chen & Shonnard, 2004). Song *et al.* (2002) consider the scheduling problem, modeled by a MILP formulation of a refinery process taking into account the environmental impact. The ϵ-constraint method is used to obtain a set of Pareto solutions for the multiobjective optimization which considers global environmental impacts by means of the critical surface-time 95 (CST95) assessment methodology. Berlin *et al.* (2007) consider a case study of the dairy industry, where the production sequencing affects the environmental impact from a life-cycle perspective. They developed a heuristic method to minimize production waste based on production rules. Their methodology is further applied by Berlin and Sonesson (2008) to a case study with two dairy products. The authors conclude that the environmental impact of processing cultured milk products can be greatly reduced by adopting sequences with fewer changes of product.

The aforementioned methodologies embed the concepts of Life Cycle Assessment (LCA) (ISO14001, 2004). Within LCA, the overall life cycle of a process or product is analyzed, taking into account upstream and downstream flow from the process along the whole life cycle, from cradle to grave. LCA is hence a holistic approach that avoids shifting environmental burdens from one part of the process supply chain to another. For this reason, it has been selected in this thesis for the environmental assessment of scheduling problems. A range of LCA software packages, such as PEMS or SimaPro, is available and they include reliable databases on materials, energy, transport and waste management options (Azapagic *et al.*, 2003).

2.7 Thesis objectives review

Along the previous sections, the most recent solution methods and features of the scheduling problem as well as the current trends toward its integration with other decision levels have been described. This literature review has revealed that there is a clear need for extending the scheduling function boundaries. Thus, tools and approaches that extend the capabilities of the scheduling problem are needed to improve the operation of process industries. Specifically in this thesis, research efforts have been devoted to the following issues:

- *Realistic modeling of scheduling problems.* Scheduling models should represent all those problem features which influence the results, including solution feasibility, such as transfer or set-up operations or time related constraints. Some of the existing scheduling formulations for multipurpose and multiproduct batch plants scheduling should be revised and adequately extended to consider alternative changeovers and transfer operations, as well as variable batch-to-batch processing times, variable processing times and rates, and multiple alternative units per stage, which would improve the scheduling flexibility, widen the potential decision making results and may lead to important trade-offs at the scheduling level.

- *Integration of process control and scheduling.* One goal of this thesis is to develop tools to achieve the integration between these areas. The inclusion of variables related to process control at the scheduling decision level may lead to the actual integration of both decision levels. According to Mishra *et al.* (2005), two strategies could be adopted to introduce process dynamics, an indirect manner or a direct one. The former approach disregards the thorough description of the operated process, and an approximation to the actual conditions should be introduced. The latter rigorously introduces control variables at the scheduling level, resulting in highly complex problems. The assessment of the benefits of the integration between the control and short-term batch scheduling using both strategies is be performed in this thesis, since only partial attempts have been reported in the literature.

- *Enlarging the scope of scheduling formulations.* According to Harjunkoski *et al.* (2009), the integration need should emerge from the functionality point of view. One key aspect in this regard is the inclusion of additional criteria, such as environmental, quality, safety or reliability, in the decision-making process. Particularly, in this work the optimization of environmental metrics along with economic ones in scheduling problems will be explored. In this sense, multiobjective problems extend the traditional functionalities of the scheduling problem. Thus, performance metrics for decision making should be proposed for a multiobjective framework.

- Since problems increase both in complexity and size, novel optimization strategies should emerge in order to deal with real sized problems. Therefore, methods which may improve the computational performance of the solution process and facilitate the integration function should be studied.

On the whole, this thesis aims at widening the scope of the scheduling level by tackling four main issues: i) revising existing models to improve their capabilities, ii) providing strategies which are able to include other decision levels variables in the scheduling problem, iii) enlarge the scope of traditional scheduling models in order to account for objectives others than economic performance; and iv) proposing algorithms and optimization strategies which can cope with large sized problems.

Chapter 3

Optimization tools

3.1 Introduction

In this thesis, the decision making process for scheduling integration is tackled by means of optimization, also termed as mathematical programming. Indeed, optimization is a wide discipline which aims at systematically finding the best solution of a problem, represented as variable values, according to specified criteria, expressed in terms of objective functions, by fulfilling, if necessary, a given set of requirements, i.e. constraints. Therefore, the problem representation must be firstly formalized, specifically as a mathematical model in mathematical programming, and next optimization strategies can be applied. This chapter presents the basic principles of the optimization techniques considered along this thesis.

Regarding mathematical models, they can be classified according to different features. For example, deterministic models are those whose parameter values are assumed to be known with certainty, whereas stochastic models involve quantities known only in probability. Additionally, models may be either lineal or non-lineal, in the former case the model equations are algebraic expressions which may contain constants and the product of a constant and a single variable, whereas in the latter, non-linear functions are also included. Moreover, they may be classified as dynamic or static, depending on whether the variables change over time or not, respectively.

In general, optimization techniques may be classified in deterministic and stochastic. The former methods ensure the global optimality of the solution found within a specific tolerance even though for large problems the solution may not be found for computational complexity reasons. In contrast, the latter techniques do not guarantee optimal solutions, but provide with approximations to such solutions and can be applied to large problems. In this thesis, both deterministic and stochastic methods are used. Firstly, the deterministic methods, that is mathematical programming techniques such as linear programming, mixed-integer linear programming, non-linear programming, mixed-integer non-linear programming and dynamic optimization, are presented. Stochastic methods based evolutionary

Table 3.1: Classification of the mathematical programming problems (Y stands for a problem feature, N stands for those features not included in the problem and O is an optional feature of the problem).

	Variables		Equations		Number OF		Time dependent variables
	Discrete	Continuous	Linear	Non linear	1	>1	
Linear programming (LP)	N	Y	Y	N	Y	N	N
Mixed integer linear programming (MILP)	Y	Y	Y	N	Y	N	N
Non linear programming (NLP)	N	Y	O	Y	Y	N	N
Mixed integer non linear programming (MINLP)	Y	Y	O	Y	Y	N	N
Dynamic optimization (DO)	N	Y	O	O	Y	N	Y
Mixed integer dynamic optimization (MIDO)	Y	Y	O	O	Y	N	Y
Multiobjective optimization (MOO)	O	O	O	O	N	Y	O
Genetic Algorithm (GA)	Y	Y	O	O	Y	O	O

algorithms, specifically the genetic algorithm, are also introduced. Thus, multiobjective techniques, based on the aforementioned deterministic methods are then presented. Finally, some of the most widespread software packages which may be used to solve the posed optimization problems are pointed out within each section.

3.2 Features of Mathematical Programming

The general expression of a mathematical programming problem is given by Equation 3.1. Depending on the nature of the variables, x, the objective function, $f(x)$, and the constraints, $h(x)$ and $g(x)$, different kinds of mathematical programming problems may be posed.

$$\left.\begin{array}{lrcl} & \underset{\boldsymbol{x}}{\text{minimize}} & f(\boldsymbol{x}) & \\ \text{subject to} & & & \\ & \boldsymbol{h}(\boldsymbol{x}) & = & \mathbf{0} \\ & \boldsymbol{g}(\boldsymbol{x}) & \leq & \mathbf{0} \end{array}\right\} \tag{3.1}$$

In general, three basic steps may be identified when formulating a mathematical problem: (i) identifying and defining integer, continuous, state and control variables; (ii) identifying all restrictions and formulating all corresponding constraints in terms of linear, nonlinear or dynamic equations or inequalities; and (iii) identifying and formulating the objective(s) as a function of the decision variables to be optimized (either minimized or maximized). Table 3.1 classifies the different problems according to their features, namely linearity of the constraints, decision variables continuity, time dependence and number of objective functions.

Thus, it is necessary to define some terms related to mathematical programming in order to understand its procedures. A feasible solution is a choice of values of the decision variables that satisfies all problem constraints. An optimal solution is a feasible solution that achieves a objective function value that is at least as good as (if not better) any other feasible solution. In this sense, it is necessary to distinguish between local and global optima. Depending on the objective function and feasible region, there may be more than a single optimum. As a result, a given function may have multiple local optima, one of which will be in turn the global optimum (if the problem is not degenerated). A necessary but not sufficient condition for optimality is reaching a point of zero gradient. A local optimum is directly a global optimum as well, if both the feasible region and the objective function are convex, that is, for given any two points x_1 and x_2 of the domain, Equation 3.2 is valid.

$$f\left(tx_1 + (1-t)\,x_2\right) \leq t \cdot f\left(x_1\right) + (1-t) \cdot f\left(x_2\right) \quad t \in [0,1] \tag{3.2}$$

In contrast, the area of global optimization addresses the computation and characterization of global solutions to non convex continuous, mixed-integer, differential-algebraic, and non-factorable problems. In practice, process models are highly non-linear and non-convex, so their integration with the scheduling problem leads to mixed-integer non convex problems.

3.3 Linear Programming

Linear programming (LP) deals with techniques for solving systems of linear equations. An LP model is suitable for modeling decision making of real world problems, if all decision variables are continuous and the objective function and constraints of the problem are linear functions of the decision variables. Regarding scheduling, LP problems are only applicable if all discrete decisions are fixed beforehand.

When the LP is appropriate to model a given problem, some useful information can be derived from the model. For example, solving the model gives an optimum solution, if it exists; and the algorithms may even identify the set of all optimal solutions, if there are several optima. The model may be infeasible, i.e. it has no feasible solution, if there is a subset of constraints that are mutually contradictory in the model. Additionally, the values of the slack variables may provide useful information regarding unused resources related to the inequality constraints. Moreover, each constraint in an LP model may be regarded as the material balance of some item. The marginal value of that item (marginal value of that constraint) is defined as the rate of change in the optimum objective value of the LP per unit change in the right hand side constant of the constraint. This marginal value associated with a constraint is called the dual variable corresponding to that constraint. The analysis of the marginal values is called marginal analysis, and helps to identify the most critical resources and requirements to achieve better results. Finally, if the LP problem is not formulated properly, it may not have a unique solution, or even any solution at all.

Several algorithms have been developed and implemented in order to solve LP models. The first computationally viable method for solving LPs was the simplex method, presented in 1947. Such method is based on recognizing that the optimum of the problem occurs at an extreme point of the convex feasible region defined by the linear constraints. Along time, the technology for implementing the simplex method has gone through many refinements, and even nowadays it is the most important algorithm for solving LP problems in the software systems. Other methods for solving LP problem are based on the interior point method, such as the primal-dual path following the interior point method or the gravitational interior point method. The most common commercial solvers which contain solution methods for LP are CPLEX, OSL, MATLAB, LINDO and EXCEL. Further references to LP problems in operations research and management science can be found in Ravindran (2008).

3.4 Mixed Integer Linear Programming

In mixed integer linear programming, some of the decision variables are discrete in nature and take integer values. In fact, many decisions in engineering can be posed as discrete decision variables. As for the scheduling problem, allocation and sequencing variables are the most common source of discrete decisions. A MILP is generally written as in Equation 3.3, where x represents the continuous variables and y the integer variables. If all integer variables of a MILP are restricted to binary, i.e. 0 or 1, the MILP is referred to as binary MILP (BMILP).

$$\left.\begin{array}{rccc} & \min cx + dy & & \\ \text{subject to} & & & \\ & Ax + By & = & 0 \\ & x \in \mathbb{R}^n & y \in \mathbb{Z}^m & \end{array}\right\} \tag{3.3}$$

None of the solution methods developed for MILP is totally reliable from a computationally efficiency point of view, specially as the number of integer variables increases. In contrast to LP, where problems with hundreds of thousands of variables and thousands of constraints can be solved in a reasonable time, MILP problems are computationally expensive.

When integer conditions on all variables are omitted, an LP relaxation of the MILP is obtained. The feasible region of the MILP is a subset of the feasible region of its LP relaxation. The optimal objective value of an MILP is no better than that of its LP relaxation. Hence, the optimal objective value of the LP relaxation is a lower bound of the MILP optimal objective. Therefore, the LP relaxations are often used in developing solution techniques for solving MILP problems.

In general, MILP are more difficult to solve than LPs. While the latter are polynomially solvable via the interior point methods, MILP are NP-hard, that is, they cannot be solved in polynomial time. The integer nature of the variables makes it difficult to devise an efficient algorithm that searches among the integer points of the feasible region. As a result, solution procedures are based on exploiting the success in solving LPs.

Two powerful solution procedures for MILP are the Branch and Bound (B& B), and the Cutting Plane methods. Specifically, the B& B method consists of an implicit enumeration approach and it is the most effective and widely used technique for solving MILP. The B& B method starts with solving the LP relaxation. If the optimal solution to the relaxed LP is integer-valued, the optimal solution to the LP relaxation is also optimal to the MILP. However, such condition is mostly unlikely and the MILP is partitioned into a number of subproblems that are generally smaller in size or easier to solve than the original problem. In contrast, the basic idea of the Cutting Plane method consists of changing the boundaries of the convex set of the relaxed LP feasible region by adding cuts, i.e. additional linear constraints, so that the optimal extreme point becomes all-integer when all such cuts are added. Therefore, when enough such cuts are added, the new optimal extreme point of the sliced feasible region becomes all-integer, and is optimal to the MILP.

CPLEX is one of the most sophisticated existing packages for integer programming. For this reason, it has been used in this thesis.

3.5 Non Linear Programming

Non linear programming studies the problem where a nonlinear function has to be minimized or maximized over a set of values delimited by several nonlinear equalities and inequalities. The presence of nonlinearities is very frequent in science and engineering.

There are three major groups of optimization techniques in nonlinear optimization, namely deterministic, stochastic and heuristic techniques. The former are commonly used in convex optimization because the global optima may be achieved, and the most important method is the descend algorithm, but the steepest descend method, the Newton method, the penalty and barrier methods, and the feasible direction methods are also within this group. The previous techniques may be categorized depending on whether they require the use of derivatives or not. Stochastic techniques are based on probabilistic meta-algorithms, that explore the feasibility region by moving from feasible solutions to feasible solutions in directions that minimize the objective value. Simulated annealing or tabu search are two examples of stochastic techniques. Finally, the heuristic strategies are methods based on heuristics for finding good feasible solutions to very complicated optimization problems. Such techniques are used whenever the first two groups of techniques fail to find solutions or perform poorly..

Within constrained nonlinear optimization programs, three main numerical algorithms can be distinguished:

- Sequential quadratic programming (SQP). It is one of the most popular NLP algorithm because it has fast convergence properties and can be tailored to a wide variety of problem structures. Some examples of commercial codes which apply the SQP method are fmincon in Matlab, or SNOPT.

- Interior point methods. This method relaxes the complementarity conditions and solves a set of relaxed problems. Some commercial codes are IPOPT or KNITRO.

- Nested projection methods. These methods are useful for NLPs with nonlinear objectives and constraints where it is important for the solver to remain close to feasible over the course of iterations. MINOS, CONOPT or LANCELOT are available codes based on nested and gradient projection.

A recent review about non linear programming concepts and algorithms can be found in Biegler (2010).

3.6 Mixed Integer Non Linear Programming

Mixed-Integer Nonlinear Programming (MINLP) addresses optimization problems in which the variables are constrained to take integer values and the objective function or feasible region are described by nonlinear functions. There is a high interest in solving such kind of problems, since they have a large number of real-world applications. Integer variables are related to logical relationships, whereas non linear functions are required to model physical properties and complex phenomena. The general form of a MINLP is given as follows:

$$\left.\begin{array}{lllll} & \min f(x,y) & & & \\ \text{subject to} & & & & \\ & h(x,y) & = & 0 & \\ & g(x,y) & \leq & 0 & \\ & x \in \mathbb{R}^n & y \in \mathbb{Z}^m & & \end{array}\right\} \tag{3.4}$$

In fact, MINLP are challenging optimization problems, since they combine the difficulty of optimizing with integer variables while handling nonlinear functions.

The computational tractability depends greatly on the convexity of the feasible region and the objective function. If both the objective function and the constraints are convex over the domain of x and y, then the MINLP is convex. Otherwise, the MINLP is said to be nonconvex. Although significant progress has been made in the past years for solving convex MINLP problems, the treatment of nonconvex MINLP poses significant challenges. The convergence of nonconvex problems to a global optimum cannot be guaranteed with local optimization methods. Therefore, only global optimization methods can ensure the convergence to a global optimum in those cases.

Methods for solving MINLP problems rely on the generation and refining of bounds on its optimal solution value. Lower bounds are usually generated by solving a relaxation of MINLP, whereas upper bounds are usually provided by the value of a feasible solution. Algorithms differ in the manner in which bounds are generated and share many general characteristics with the branch-and-bound methods for solving MILPs.

Several algorithms have been proposed based on the aforementioned elements to solve convex MINLP, namely the NLP-Based Branch and Bound, the Outer Approximation, the Generalized Benders Decomposition, Extended Cutting Plane and LP/NLP-Based Branch and Bound. A recent review on algorithms for solving convex MINLP problems can be found in Bonami *et al.* (2009).

Several solvers are available for MINLP problems. In this thesis, both DICOPT and SBB have been used in the modeling system GAMS. DICOPT ensures global optimal solutions for convex MINLPs and alternates between solving MIP outer approximations and NLP subproblems of the primal. It also accommodates nonconvex MINLPs, by relaxing equality constraints through the use of slack variables and penalty parameters. SBB also ensures global optimal solutions of convex MINLPs and implements a branch-and-bound algorithm using nonlinear relaxations for the bounding step. There are two general-purpose solvers for nonconvex MINLP, namely BARON and αBB, both of which rely on solving iteratively a lower bounding problem, constructed using convex relaxations of the nonconvexities of the model, and the original problem in a reduced space (upper bounding problem). A recent review on MINLP solver software is given by Bussieck and Vigerske (2010).

Mixed-Integer Linear Fractional Programming Those MINLP problems in which the non-linearity is not associated with the problem constraints, but with the objective function being the ratio of two linear functions, may be further classified as a special type of MINLP, namely mixed-integer linear fractional program (MILFP) (Problem 3.5).

$$\left.\begin{array}{lrcl} & \min \frac{N(x,y)}{D(x,y)} & & \\ \text{subject to} & & & \\ & h\,(x,y) & = & 0 \\ & g\,(x,y) & \leq & 0 \\ & x \in \Re^n; y \in \{0,1\} & & \end{array}\right\} \tag{3.5}$$

To solve such problems, in addition to the aforementioned MINLP methods, an alternative consists of using the Dinkelbach's algorithm that successively calculates a number of mixed-integer linear programming (MILP) problems. Dinkelbach's algorithm was initially developed in 1967 to solve nonlinear fractional programing (NLFP) problems, which do not contain discrete variables, by exploiting the relationship between NLFP and nonlinear parametric programing. Recently, You *et al.* (2009) demonstrate the convergence and optimality conditions of the Dinkelbach's algorithm to obtain the global optimal solution of MILFP problems by solving a sequence of MILP subproblems. A major advantage of this algorithm (Algorithm 3.1) is that no NLP solver is required to solve the problem, and the computational complexity tends to be smaller compared with standard MINLP methods.

Algorithm 3.1: Dinkelbach's algorithm for MINLFP proposed by You *et al.* (2009).

Data: A MILFP problem (Problem 3.5), whose constraints are all linear, the objective function is a fraction of linear functions $N\,(x,y)$ and $D\,(x,y)$ and the variables are both continuous and discrete, and an optimality tolerance:tol.
Result: The global optimal solution of a MILFP problem
begin
- define $q = \frac{N(x,y)}{D(x,y)}$;
- assign arbitrary values to x and y and compute q or set $q \longleftarrow 0$;
- $k \longleftarrow 2$;
- solve the MILP problem with constraints of Problem 3.5 and the following objective function: $\min F(q_k) = \min N\,(x,y) - q_k \cdot D\,(x,y)$;
- **while** $F(q_k) \geq tol$ **do**
  - $k \longleftarrow k+1$;
  - $q_k \longleftarrow \frac{N(x_{k-1},y_{k-1})}{D(x_{k-1},y_{k-1})}$;
  - solve the MILP problem with constraints of Problem 3.5 and the following objective function: $\min F(q_k) = \min N\,(x,y) - q_k \cdot D\,(x,y)$;

3.7 Dynamic Optimization

The dynamic nature of chemical processes can be described by means of mass, energy and momentum balances, which ensure the chemical, physical and thermodynamic consistency of the system. Such relationships are usually modeled by means of ordinary differential equations (ODEs), differential/algebraic equations (DAEs), or partial differential/algebraic equations (PDAEs). Dynamic optimization aims at automating the decision process regarding such dynamic systems, by determining the values of the input/control variables, which may be time variable, that optimize the system performance according to a desired criterion. In general, the dynamic optimization problem can be formulated as follows:

$$\left.\begin{array}{ll} & \min\limits_{u(t),t_f} J = \Phi\left(x\left(t_f\right)\right) + \int\limits_0^{t_f} L\left(x,u\right)dt \\ \text{subject to} & \\ & \dot{x}\left(t\right) = F[t, x\left(t\right), u\left(t\right)] \\ & x\left(0\right) = x_0 \\ & h\left[t, x\left(t\right), u\left(t\right)\right] = 0 \\ & g\left[t, x\left(t\right), u\left(t\right)\right] \leq 0 \\ & x\left(t\right)^L \leq x(t) \leq x\left(t\right)^U \\ & u\left(t\right)^L \leq u\left(t\right) \leq u\left(t\right)^U \end{array}\right\} \tag{3.6}$$

where J is the scalar performance index to be minimized; x, the n-dimensional vector of state variables with known initial conditions x_0; u, the m-dimensional vector of control variables; h and g are the equality and inequality constraints respectively (which may include state, path and boundary constraints), and $x\left(t\right)^L$, $x\left(t\right)^U$, $u\left(t\right)^L$ and $u\left(t\right)^U$, the bounds over the state and control variables. Additionally, F represents the dynamic relationships of the state and control variables, $\Phi\left(x\left(t_f\right)\right)$, a smooth scalar function related to the terminal cost, and L a smooth scalar function representing the integral cost. Thus, t_f, the final time, can be either fixed or free.

There are several solution strategies to solve dynamic optimization problems which may be classified in direct (sequential, simultaneous, analytic parametrization), indirect (shooting-method, gradient method) and dynamic programing methods. However, the most applied techniques are the direct sequential and simultaneous dynamic optimization methods. Since decision variables, u, may depend on time and so have infinite dimensions, they must be parameterized to a finite number of parameters in order to use numerical techniques. According to explicit or implicit integration of the dynamic equations, the methods may be either sequential or simultaneous. Further details regarding dynamic optimization methods can be found in Srinivasan *et al.* (2003); Chachuat *et al.* (2006); Biegler and Grossmann (2004).

3.7.1 Sequential approximation or partial discretization

In this case, the optimization is applied only to the space of control (input) variables. After the parametrization, that is, discretization, of u, the differential equations are integrated by means of standard integration algorithms, and the objective function is evaluated. This problem is known as feasible path problem, because the differential equations are satisfied at every step of the optimization process. A piecewise or polynomial approximation of the input variables may be used. The basic procedure consists of: i) parameterizing the input variables in a finite number of decision variables; ii) choosing an initial estimation of the decision variables; iii) integrating the systems state until the final time and evaluating the objective function and the constraints; and iv) using an optimization algorithm, such as maximum descending or quasi-Newton, to update the values of the decision variables. Finally, steps iii and iv should be repeated in order to minimize the objective function.

If a constant piecewise approximation over equally spaced time intervals is used, the method is known as control vector parametrization (CVP). It is a method which tends to be slow, specially when there are inequality constraints. Additionally, the solution quality strongly depends on the parametrization of the control profile.

3.7.2 Simultaneous approximation or total discretization

This method consists of introducing an approximation to the differential equations system in order to avoid the explicit integration of each input profile. The optimization is done in a discretized space of control and space variables. Therefore, the differential equations are only satisfied in the solution point of the optimization problem. The general procedure consists of: i) parameterizing the input and state variables using a finite number of decision variables; ii) discretizing differential equations, such that they are only satisfied in a finite number of time points (typically orthogonal collocation); iii) choosing an initial estimation of the decision variables; and iv) solving iteratively the group of variables using a NLP solver.

Steps i and ii transform the dynamic optimization problem in a nonlinear problem. Since the previous procedure results in large NLP problems, efficient numerical methods are needed to solve them. It is important to be conscious of the trade-off between approximation and optimization, because low discretized problems can lead to very good values of the objective function, but inaccurate solutions, whereas a fine discretization can lead to large NLP problems. The methods described in Section (NLP) can be used to solve these problems.

In this thesis, the simultaneous approach based on orthogonal collocation over finite elements (OCFE) has been applied. Next, the main features of such technique are described.

Orthogonal collocation over finite elements In the orthogonal collocation method, the time domain is discretized in a specific number of finite elements (N_{FE}), and the state and control variable values approximated at the collocation points (r_{NCP}) (Figure 3.1). The OCFE keystone consists of transforming the differential equations into algebraic equations using Lagrange based polynomial approximations to the solution variables over the finite elements. Specifically, the polynomial order depends on the number of collocation points (N_{CP}). Therefore, the state and control variables can be generally written as follows:

$$x_{K+1}(t) = \sum_{j=0}^{N_{CP}} x_{ij}\varphi_j(t); \qquad \varphi_j(t) = \prod_{k=0}^{N_{CP}} \frac{t - t_{ik}}{t_{ij} - t_{ik}} \qquad \forall i \subset 1..N_{FE} \tag{3.7}$$

$$u_K(t) = \sum_{j=1}^{N_{CP}} u_{ij}\theta_j(t); \qquad \theta_j(t) = \prod_{k=1}^{N_{CP}} \frac{t - t_{ik}}{t_{ij} - t_{ik}} \qquad \forall i \subset 1..N_{FE} \tag{3.8}$$

Note that the polynomial of the control variable is of order N_{CP}, whereas the state variable is of order N_{CP+1}. Such polynomials are used to compute the value of the state variables at any time point, based on the value obtained at the collocation points in the optimization stage. The two polynomials have different orders because

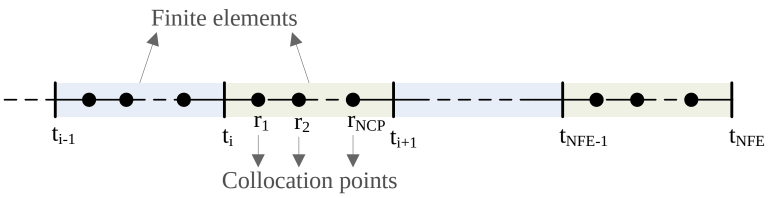


Figure 3.1: Time discretization for the orthogonal collocation on finite elements.

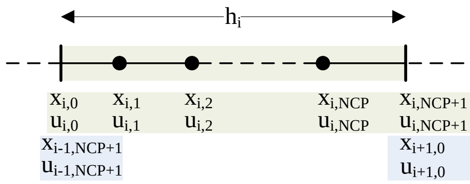


Figure 3.2: Detail of control and state variables for one finite element of the orthogonal collocation on finite elements.

the state variables have initial state conditions; so the method has an additional point for adjustment.

The collocation points location in the finite element is decided based on the roots of some polynomials that have stability properties (Biegler, 2010), namely the shifted Legendre or Radau polynomials.

The differential equations are approximated at the collocation points as algebraic equations. Specifically, residual equations are posed at those points, by using the basis functions normalized over each element ($\Delta\zeta_i, \tau \in \{0,1\}$). Such residual equations (Equation 3.9) determine the dynamic behavior of the system. The element lengths h_i can also be included as decision variables.

$$\Delta\zeta_i r\left(t_{ik}\right) = \sum_{j=0}^{N_{CP}} x_{ij}\dot{\varphi}_j\left(\tau_k\right) - \Delta\zeta_i F\left(t_{ik}, x_{ik}, u_{ik}\right) \qquad \forall i \subset 1..N_{FE}, \forall k \subset 1..N_{CP} \tag{3.9}$$

Additionally, it is necessary to define the continuity of the state variables at the element endpoints (Equation 3.10, Figure 3.2).

$$x_{N:CP+1}^{i-1} = x_0^i \quad \text{where} \quad x_{N_{CP}+1}^i = \sum_{j=0}^{N_{CP}} x_{ij}\varphi_j\left(\tau = 1\right) \quad \forall i \subset 1..N_{FE} \tag{3.10}$$

In contrast, control variables have discontinuities at these endpoints. Control profiles are constrained by its bounds at collocation points.

Moreover, in order to assess the value of the integration functions, the Gauss Quadrature rule of integration which approximates the integrals as the weighted sum of the value of the integrated functions at specific points is employed. Therefore, the general expression of the Gauss Quadrature rule is given by Equation 3.11. More information regarding this integration method can be found in Stoer and Bulirsch (1993).

$$\int_{-1}^{1} g(x)\, dx \approx \sum_{i=1}^{n} c_i g(x_i) \tag{3.11}$$

The OCFE method is thoroughly described in Cizniar *et al.* (2005) and Biegler (2010). The application of this method to the integration of the scheduling and control problems is detailed in section 8.4 of Chapter 8.

3.8 Multi-criteria Decision Making

In industry, decisions must be continuously taken under multiple and usually conflicting criteria. Precisely, multicriteria decision making (MCDM) is a discipline that deals with the methodology and theory to treat complex problems entailing conflicting objectives, such as cost, performance, reliability, safety, sustainability and productivity among others (Wiecek *et al.*, 2008). In presence of multiple criteria, a large number of solutions may be suitable. Multiple objective programming is an area of MCDM which aims at finding suitable solutions of mathematical programs with multiple objectives, whereas decision maker-driven multiple criteria decision analysis (MCDA) is an area which encompasses decision makers' judgments and preferences to derive a preferred decision becoming the policy to be implemented for the problem. This thesis applies multiobjective programming techniques to obtain solutions of multiobjective problems, and uses some criteria of multiple criteria decision analysis to reach objectively good solutions.

According to the way the decision maker intervenes in the optimization process, several approaches can be defined, namely a priori, interactive and a posteriori. The former approach focuses on establishing a priority in the objectives, and solving them iteratively. So first objective is optimized, then that value is established as a constraint when solving the second objective, and this procedure is repeated for all the objectives. However, the final solution depends on the selected order for optimizing each objective. Hence, this approach is only applicable for those cases where priority is clearly determined. The interactive approach is based on directing the search using the information obtained in the optimization process. Finally, the a posteriori search consists of producing a set of solutions covering the trade-off region comprising the best compromise solutions. As a result of the multiobjective optimization problem, a set of solutions which are said to be Pareto optimal is obtained. This thesis applies the latter approach to deal with multiobjective decision problems, which can be mathematically formalized as follows:

$$\left.\begin{array}{lc} & \min\limits_{x} \{\mu_1(x)\, \mu_2(x) \ldots \mu_n(x)\} \qquad n \geq 2 \\ \text{subject to} & \\ & g(x) \leq 0 \\ & h(x) = 0 \\ & x^L \leq x \leq x^U \\ & \text{where} \quad x \in \chi \subset \Re^n \end{array}\right\} \tag{3.12}$$

A Pareto solution is one for which any improvement in one objective can only take place if at least one other objective worsens. One solution dominates another if the values of all objective criteria for the former are better than for the latter. In addition,

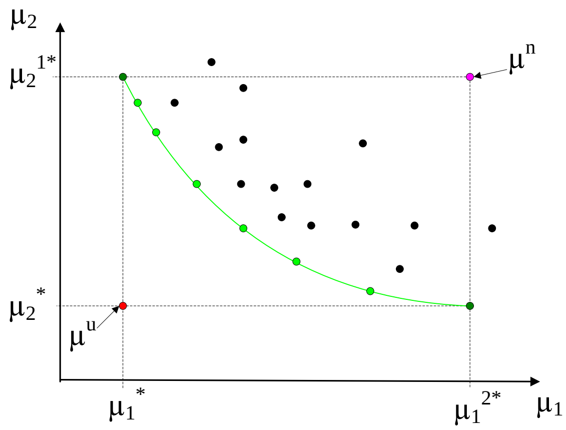


Figure 3.3: Pareto frontier and dominated solutions for a biobjective minimization problem.

one solution does not dominate another if at least one of the objective criteria of the former is equal or worse than the values for the second solution. Given two solutions, if none of them dominates the other, then both of them are non-dominated solutions with respect to one another. In fact, the Pareto optimal solutions are a set of non-dominated solutions. Figure 3.3 presents a set of solutions and the Pareto frontier for a biobjective minimization problem. Black points represent the set of solutions that are dominated by the solutions in the Pareto frontier (green points). Additionally, the darker green points are called anchor points($[\mu_1^*, \mu_2^{1*}], [\mu_1^{2*}, \mu_2^*]$), which result from the optimization problems considering one single criteria at a time(μ_1). Moreover, the utopia point ($\mu^u : [\mu_1^*, \mu_2^*]$) is defined by the optimal values for each objective function, whereas the nadir point ($\mu^n : [\mu_1^{2*}, \mu_2^{1*}]$) is given by the worse values of the objective functions.

A Pareto optimal solution involves first generating a set of solutions, from which one will be selected. This thesis applies two generation techniques for the set of solutions, namely the Normalized Constraint method and MultiObjective Genetic Algorithm, which are introduced in the following subsection.

However, many other techniques exist. For example, one of the typical approaches to generate solutions is to systematically vary the numerical scalar weights in an aggregate objective function, whereby each set of weights results in a possible Pareto solution.

In order to evaluate the quality of the solutions, different methods can be applied (Gandibleux, 2004). For example, the multiobjective problem can be transformed into a single-objective by combining the various criteria into a single scalar value by setting weights to each criterion and add them together; however, this process is quite complicated because meaningless numerical weights are usually involved. Such methods as goal programming and physical programming offer important advantages in this regard (Messac *et al.*, 2003). Another approach refers to optimizing one criterion at a time, while imposing constraints on the others. Unfortunately, the order in which the optimization is done may lead to totally different final solutions. Therefore, the most accepted evaluation consists of the Pareto based evaluation (Gandibleux, 2004). Some of the techniques reported in the literature that can also be used to generate Pareto solutions are the weighted sum, the physical programming or the normal boundary intersection.

Even though multiobjective criteria has traditionally been used for process design, process operation is affected by multiple conflicting criteria as well. Therefore, Chapter 9 tackles the trade-offs arising in process scheduling when considering environmental and economic criteria.

3.8.1 Multi-Objective Optimisation

Normalized Constrained Method (NC) The Normalized Constrained method is proposed by Messac *et al.* (2003). It consists of a solution generation algorithm, after which a Pareto solution filtering step is necessary in order to delete dominated solutions. The NC method consists of normalizing the objective functions in their own domain, and next consecutively optimizing one normalized objective function, while imposing constraints on the other normalized objective functions. Algorithm 3.2 encompasses the method for a two objective problem. The general algorithm for n objective functions can be found in Messac *et al.* (2003).

Algorithm 3.2: Normalized Constraint method for a bi-objective problem presented in Messac *et al.* (2003).

Data: A MO problem (Problem 3.12), a prescribed number of solutions in a space direction (m_1) and an optimality tolerance (tol).
Result: A set of points that are potential Pareto solutions of the MO problem.
begin
- solve the single objective problems to obtain the anchor points μ_n^*;
- set $Utopiahyperplane \longleftarrow \mu_1^*, ..., \mu_N^*$;
- define Utopia and Nadir points μ^u, μ^n;
- set $L = [l_1, l_2] \longleftarrow [\mu^N - \mu_u]$;
- set $\bar{\mu}_i \longleftarrow \frac{\mu_i - \mu_i^*}{l_i}$;
- define Utopia line vector: $\bar{N}_i = \bar{\mu}^{2*} - \bar{\mu}^{1*}$;
- generate normalized increments: $\delta_1 \longleftarrow \frac{1}{m_1 - 1}$;
- $j \longleftarrow 1$;
- $\alpha_{1j} \longleftarrow 0$;
- **while** $j \leq m_1$ **do**
    - $\alpha_{2j} \longleftarrow 1 - \alpha_{1j}$;
    - $\bar{X}_j = \alpha_{1j}\bar{\mu}^{1*} + \alpha_{2j}\bar{\mu}^{2*}$;
    - jth point generation: solve Problem 3.13 ;

$$\left.\begin{array}{c} \min \bar{\mu}_2 \\ \text{subject to} \\ g(x) \leq 0 \\ h(x) = 0 \\ \bar{N}_1 \left(\bar{\mu} - \bar{X}_j\right)^T \leq 0 \\ x^L \leq x \leq x^U \\ \bar{\mu} = [\bar{\mu}_1 \quad \bar{\mu}_2] \end{array}\right\} \tag{3.13}$$

    - $j \longleftarrow j + 1$;
    - $\alpha_{1j} \longleftarrow \alpha_{1(j-1)} + \delta_1$;

Multi-Objective Genetic Algorithm (moGA) This stochastic method is based on the evolutionary algorithm. As in the NC method, it is necessary to apply a Pareto filter in order to obtain the Pareto frontier after solution generation.

Genetic algorithms are inspired in biological evolution. Unlike other stochastic optimization methods, genetic algorithms move from one set of points (termed population) to another set of points. Thus, populations of strings, usually named chromosomes represent the underlying set of parameters. A simple genetic algorithm exploits three basic operators, namely reproduction, crossover and mutation. The first operator consists of generating new population sets stemming from those already existing based on the objective function value. The crossover involves the combination of two strings to generate the offspring. Such operator works as a local search operator and spreads good properties among the population. Finally, mutation creates new strings by randomly changing parts of strings with a low probability of occurring. The genetic algorithm works by first generating an initial population randomly and evaluating its fitness (value of the objective function). The three operators are applied and the new population is obtained. Such evolution is done iteratively. In the case of multiple objective functions, after the fitness evaluation the Pareto filtering must be applied in order to obtain the ranking criteria of each generation.

Further references to multiobjective evolutionary algorithms are given by Coello *et al.* (2007).

In this thesis, the multiobjective genetic algorithm is combined with a rigorous mathematical local search, resulting in a hybrid optimization method. The features of the moGA applied in this thesis are discussed next (see Algorithm 3.3), and it is applied to solve the functional integration as described in section 9.6.2 of Chapter 9.

Solutions representation. According to the genetic algorithm, each solution of the problem is referred to as an individual of the population, which represents the whole set of solutions to be evaluated, i.e. its phenotype is to be calculated. The information regarding each individual genotype is characterized in its chromosome formed by different genes. In this thesis, the production schedule is encoded as a chromosome which consists of a string divided in three genes represented by vectors (Figure 3.4). The length of each gene (i.e vector) corresponds to the total number of batches (NB) associated to the final products that may be performed including the initial and final still-state. The first and last vectors represent the ordered set of batches, whereas the second vector contains the permutation of the number of batches, defining the order in which batches are processed.

Specifically, the first vector represents the decision of the batches to be performed, and it directly corresponds to the binary variable W_i of the mathematical formulation. Therefore, if a given batch i is processed, its position in the vector (i) contains a value of 1 ($W_i = 1$); otherwise, its value is 0, consequently its alleles are 0 or 1. In the chromosome second vector, the first and last entries correspond to the initial and final batches, representing the still-state. The other positions in the vector represent the sequence in which batches are performed, consequently its values i.e. alleles are integers from 1 to $NB-1$. The last vector of the chromosome contains the inter-batch cleaning method (CL_i) that precedes each batch, and its alleles range from 1 to the number of possible interbatch cleaning methods available. It must be noted that the batch representing the initial still state does not have any precedence cleaning method.

As a result, if a number of $N0$ batches are to be produced then the chromosome

Algorithm 3.3: moGA hybrid algorithm.

Data: Input parameters for the MOGA (N_{pop}, N_{gen}, N_{rep}, N_{ls}, N_{PFs}, P_{ran}, P_{mut1}, P_{mut3}, P_{swi}, P_{pos}, P_{crx} and T_{lim})
Result: A reliable Pareto frontier estimate PF^*
begin
  solve the batching problem;
  generate the initial population POP_0 with N_{pop} individuals;
  evaluate the objective function of POP_0, calling the feasibility test math program;
  $j \longleftarrow 1$;
  while $time \leq T_{lim}$ *or* $j \leq N_{gen}$ **do**
    obtain the Pareto frontier estimate, $PF_j^1 \xleftarrow[Paretofiltering]{} [POP_{j-1}]$;
    check the end criteria for N_{rep} consecutive PF;
    gather the mating pool $MatPool_j \leftarrow [PF_j^1 \mid$ and selected $N_{PFs}]$, from POP_{j-1};
    if $mod(\frac{j}{N_{ls}}) = 0$ **then**
      obtain PF_j^{1ls} and the N_{PFs}^{ls} from the $MatPool_j$ using bit-change local search;
      gather $MatPool_j \xleftarrow[Paretofiltering]{} [PF_j^1 | PF_j^{1ls} \mid$ selected N_{PFs} |and $N_{PFs}^{ls}]$;
    generate the off-spring population POP_j using as parents $MatPool_j$;
    evaluate the objective function of POP_j, calling the feasibility test math program;
    $j \longleftarrow j + 1$;
  $PF^* \longleftarrow PF_j^1$

representing a given set of decisions will have $N0 \cdot N0 \cdot N0$ values, Figure 3.4. The second and third vectors of the chromosome correspond to the binary variable $X_{ii'c}$ of the mathematical formulation, which stands for the assignment of cleaning method c to changeover when batch i is produced immediately before batch i'.

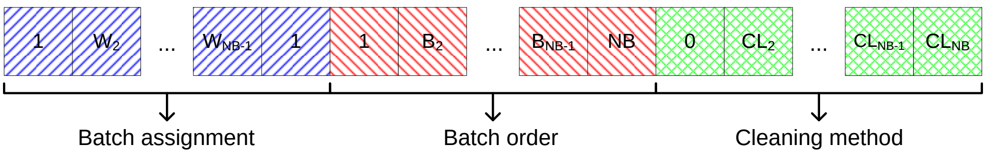


Figure 3.4: String representation of a solution.

Feasible integer solution generation. A simple algorithm for generating feasible sequence solutions satisfying the minimum demand is used. In the first vector of the chromosome, the first and last batches corresponding to the still state as well as a minimum number of batches to fulfill the demand are compulsory, and are fixed to have value $W_i = 1$. To decide whether the remaining batches are performed, random assignment criterion is used. The second vector of the chromosomes, which can contain any integer value from 1 to NB, is defined by assigning random values to each batch position and then sorting them according to these values. The value assigned is the order that each batch receives when sorted, this procedure allows for generating sequences where integer values would not repeat themselves. Anyhow, the first and last positions of this vector are fixed to the batches corresponding to still state. The final vector of the individuals is initialized by randomly assigning a cleaning method to precede each batch.

The former guidelines for generating the chromosomes allow for disregarding a large amount of possible solutions which would otherwise be generated: solutions where minimum demand is not achieved, or where some batches have the same sequence order. Therefore, the adopted representation and its generation creates sequence feasible solutions of the scheduling problem, since the binary variables of the mathematical formulation are completely defined. Thus, the batches allocation, sequence and cleaning method of each solution can be directly introduced in the mathematical formulation, so that timing constraints may be checked and the value of the continuous variables, such as starting and finish times of the operations, makespan or costs, can be obtained solving a LP problem.

Genetic algorithm operators. The optimization strategy seeks to improve the individuals of the population at each iteration, i.e. generation. Therefore, for a given population, a number of individuals is first selected and the operators are then applied to these individuals, called after parents, to generate their off-spring. According to Coello *et al.* (2007), three main operators are applied in genetic algorithms: mutation, recombination and selection.

As for the solution representation, several operators can be applied to different chromosome sections. For the first vector, the mutation operator is adopted to choose the assignment of those batches that are optional to fulfill the minimum demand, otherwise integer solutions will be generated, that are a priori known to be unfeasible. The second vector of the chromosome is replicated by means of several variations of the recombination operator: switching between any two genes, inversion of the genes order between any two points, and crossover between two parents. The third part of the chromosome evolves using only the mutation operator. A proportion of P_{mut} individuals of the total population is obtained by the application of the mutation operator, another proportion (P_{rnd}) is generated using the feasible integer solution generator and is considered to be random, while the remaining individuals are generated using recombination (P_{rec}). Thus the amount of individuals generated by each operator is defined a priori by setting the former proportions ($P_{rnd}+P_{mut}+P_{rec} = 100\%$). The operators applied to the parents solutions pool are explained next:

- *Mutation.* A random gen of the chromosome is selected to be mutated, and its value is changed in each generation (Figure 3.5). The chromosome genes that are susceptible for mutation are those related to the first and third gene vectors. Mutation only affects a single gen.

- *Switch of two batches.* This operator is applied to the gene coding the batches sequence. The positions of two batches are switched thus providing a new sequence as shown in Figure 3.6. Two randomly selected genes of the vector are exchanged and consequently only two batches are modified in sequence. The amount of individuals generated using this operator is P_{swi}.

- *Inversion of batches between two points.* This operator inverts the order of the genes between two randomly chosen positions (Figure 3.7). It is applied to obtain a percentage P_{pos} of the whole population. The number of batches modified in sequence using this operator depends on the sequence positions choice.

- *Chromosomes crossover.* This operator is applied using two parents. The children are generated by choosing a random position in the chains (i_1 and i_2) of the respective parents. Consequently, four possible situations arise in which each child preserves half of the parent sequence without changes (B_{i_1} and B_{i_1+1} for parent 1, and B_{i_2} and B_{i_2+1} for parent 2). The rest of the sequence chain for each child is fulfilled with the lacking batches in the order associated with the other parent (Figure 3.8). Hence, R_{i_1+1} contains the sequence of batches from parent 2 that are not considered in B_{i_1}. The former way of performing the sequence crossover guarantees the generation of feasible sequences. Only two randomly chosen children are selected and the amount of individuals total proportion of P_{crx} individuals is generated by choosing the same amount of pairs of individuals of the total population.

Finally, the selection operator is used to generate the mating pool for the next generation. In this case, instead of allowing all the population to be selected as parents based on roulette selection rules, only the best possible solutions which are part of the different Pareto fronts are considered. The individuals are classified according to the Pareto frontier they belong to. The individuals pertaining to the first Pareto front are known as PF^1, to obtain the 2nd best individuals a filter is applied disregarding those individuals belonging to PF^1, thus obtaining PF^2. Similarly, the other i-th best individuals are classified by disregarding during the filtering procedure the $i-1$-th previously selected individuals, which in general terms can be considered of higher rank. The mating pool is generated from these sets of solutions (PF^i) by selecting the number of PFs (N_{PFs}) that are included or not. Please note that in all cases PF^1 is the problem solution, and therefore the one that is the most interesting to generate.

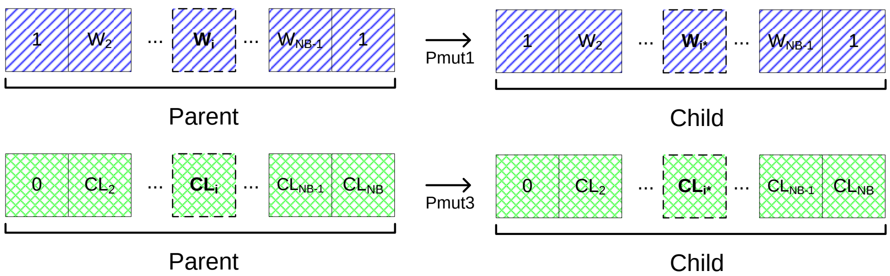


Figure 3.5: Mutation operation for a given chain.

The steps of the genetic algorithm procedure applied in this work are detailed next.

Initialization. The very first step consists of initializing the population. The construction of a given set of individuals is performed using the criteria previously described for the generation of feasible integer solutions.

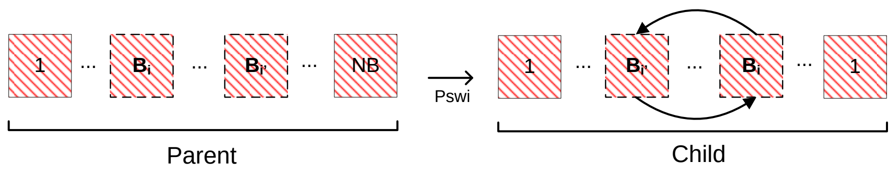


Figure 3.6: Switch of two positions operation.

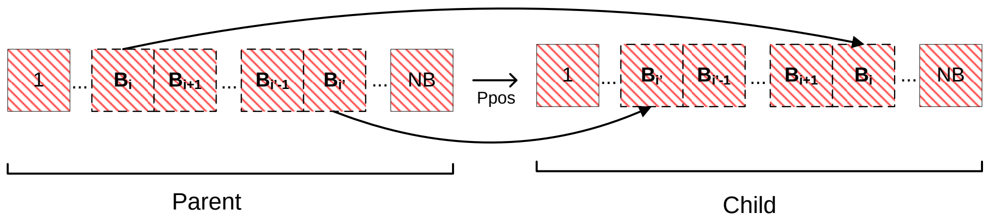


Figure 3.7: Reverse operator between two positions operation.

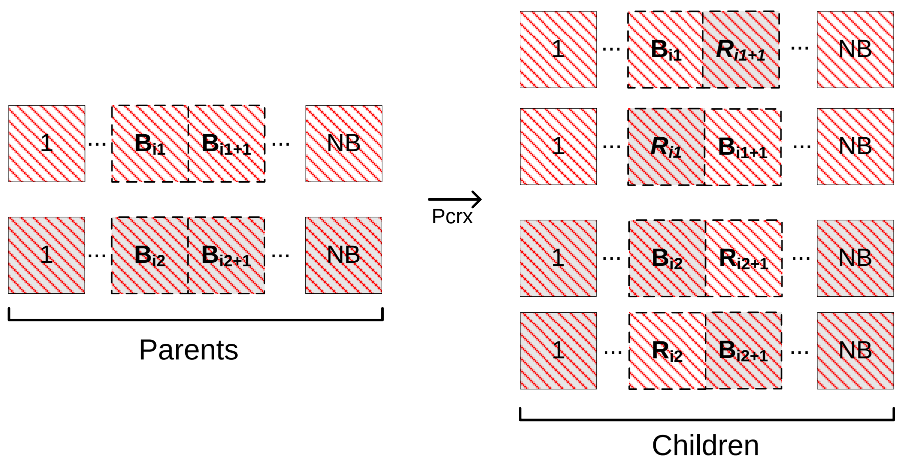


Figure 3.8: Crossover of two individuals operation.

Objective functions evaluation and ranking criterion. After defining the individuals of the population, the objective functions are assessed, and the potential unfeasibilities are checked. The goodness of the individuals is next measured by feeding each individual from the entire solution pool to the mathematical program solved by fixing the integer variables. This step serves two purposes: checking the feasibility of each solution and assessing each individual according to the objective functions. In the case of a single objective optimization, the value of the objective function could be used as performance metrics. In multi-objective optimization, the Pareto frontier criterion is selected to rank the individuals.

Rigorous local search. The solutions of the mating pool can be further improved using a local branching strategy in a rigorous branch and bound mathematical programming framework. Therefore, the solutions, namely the values of the binary variables W_i and $X_{ii'c}$, are introduced as parameters pW_i and $pX_{ii'c}$ in the mathematical scheduling problem. A maximum number of changes in these integer variables is allowed in such a model by means of Equations 3.14 and 3.15. Due to the mathematical formulation, any change in the number of batches (W_i) entails at least 3 changes in the sequence and cleaning variables ($X_{ii'c}$). Two possible integers parameters are added consequently for constraining the amount and type of changes allowed, $N_X^{changes}$ is the number of changes allowed in binary variables $X_{ii'c}$, while $N_W^{changes}$, applies to the batching binary variables.

$$\sum_{i|pW_i=0} (W_i - pW_i) + \sum_{i|pW_i=1} (pW_i - W_i) \leq N_W^{changes} \tag{3.14}$$

$$\sum_{i,i',c|pX_{ii'c}=0} (X_{ii'c} - pX_{ii'c}) + \sum_{i,i',c|pX_{ii'c}=1} (pX_{ii'c} - X_{ii'c}) \leq 3 * N_W^{changes} + N_X^{changes} \tag{3.15}$$

The problems are optimized considering all objective functions, one at a time. As a result, as many new solutions as objective functions times are obtained from each individual in the mating pool. These new solutions are considered together with their parents (which in this case are the initial individuals that started the local search) and filtered to obtain the improved mating pool. This optimization may be time consuming. Hence, to expedite the overall procedure, a local search is applied to the mating pool solutions every N_{ls} number of generations. A maximum allowable time is defined for the optimizer (T_{lim}^{ls}).

Replication. The next generation is obtained defining the mating pool of individuals as parents. A certain percentage of the next generation, P_{ran}, is created following the procedure explained in the initialization procedure, and disregarding any information from the mating pool. This allows keeping a population diverse enough while reducing the risk of providing local solutions.

The rest of the population is created applying the previously discussed operators to randomly selected individuals of the mating pool. Specifically, P_{mut} is the total percentage of individuals of the next population created by mutation, while the remaining is created by using recombination operators: switch (P_{swi}), crossover (P_{crx}) and inversion (P_{pos}), note that $P_{rec} = P_{swi} + P_{crx} + P_{pos}$. In the case of mutation, $P_{mut} = P_{mut1} + P_{mut3}$, where P_{mut1} is the percentage of individuals of the next population created by mutation of the first chromosome part, while P_{mut3} represents the percentage created by mutation of the third chromosome part.

End criteria. Different termination criteria can be used. Three possible end criteria are proposed, namely:

- *Maximum time.* A maximum time limit T_{lim} is defined. The algorithm is stopped whenever such time limit is reached.
- *Maximum number of generations.* The algorithm is stopped when a maximum number of generations, N_{gen}, is exceeded.
- *Number of consecutive identical PF estimations.* The algorithm finishes when the estimation of the PF is the same for a N_{rep} number of consecutive generations.

Table 3.2: Parameters of the hybrid MOGA.

Parameter	Definition
N_{pop}	Number of individuals of the population.
N_{gen}	Maximum number of generations.
N_{rep}	Number of equal PF to meet end criterion.
T_{lim}	Maximum time for the algorithm implementation
N_{ls}	Number of generations between two consecutive local search procedures.
T_{lim}^{ls}	Maximum time available for local search optimisation
$N_X^{changes}$	Number of changes of the binary variables $X_{ii'c}$ the local search procedure.
$N_W^{changes}$	Number of changes of the binary variables W_i the local search procedure.
N_{PFs}	Number of PFs that are included in the mating pool.
P_{ran}	Percentage of random individuals of the population .
P_{mut1}	Percentage of the population obtained by mutation in the first vector in the chromosome.
P_{mut3}	Percentage of the population obtained by mutation in the third vector in the chromosome.
P_{swi}	Percentage of the population obtained by switching positions of the second vector in the chromosome.
P_{pos}	Percentage of the population obtained by inverting the chain of genes between two positions of the second vector in the chromosome.
P_{crx}	Percentage of the population obtained by crossover of two parents in the second vector of the chromosome.

Part II

Scheduling Modeling Issues Toward Process and Scheduling Integration

Chapter 4

Modeling Changeover Operations in Multiproduct Batch Plants

4.1 Motivation

One of the main advantages of batch process operation consists of a more thorough control over process operation and conditions, compared to continuous operation. Therefore, operations which require regular cleaning for fouling or product purity maintenance reasons are usually performed in batch mode. Such cleaning tasks are executed during the batch changeover operations, which increase the complexity of the scheduling problem. As a result, some problem features may be simplified in order to deal with the associated complexity, for example changeover times may be considered as a part of the overall processing time. However, such approach is not always possible because changeover times may depend on the batch sequence. Thus, changeovers may have a significant effect on cycle time and makespan (Smith, 2005) since they may introduce a decrease in overall equipment utilization. Therefore, the efficiency of single-product vs mixed-product campaigns should be carefully studied. In addition, equipment cleaning may generate significant waste quantities which can result in an environmental problem. Furthermore, changeovers increase the complexity of the process control, and so the system results to be more susceptible to errors. On the whole, batch changeovers may lead to production time loss, equipment underutilization and higher environmental impact. Hence, highly effective formulations for scheduling which consider changeovers are needed to optimize plant performance. This chapter presents an effective formulation for multiproduct processes based on the immediate precedence concept, which is able to consider alternative cleaning methods among products. The mathematical formulation also counts for product batching, multiple units per stage and different timing and storage constraints between stages. In addition, this chapter defines different objective functions to be considered in the scheduling problem depending on the scheduler criteria, regarding economic, timing and environmental metrics. Therefore, a scheduling model which can be a main building block for further improvements and integration with other decision levels is pursued. The proposed formulation is illustrated in two examples, namely the scheduling of a multiproduct acrylic fiber production plant and a multistage sequential batch process with multiple units per stage and finite intermediate storage resources.

4.2 Introduction

According to material routing, process scheduling problems can be broadly classified in network and sequential processes Méndez *et al.* (2006). In the former, batches can be split or mixed at the end of the stages to produce intermediate products for further processing. In contrast, sequential processes maintain batch integrity along all process stages. Additionally, sequential processes may be considered as multiproduct, if the sequence of stages is the same for all products in the plant, and multipurpose, in the other instances.

This chapter aims at adequately modeling batch process scheduling of multiproduct multistage plants when alternative methods for batch changeover are available. In general the process of converting a line or equipment from running one batch to another, i.e. product changeover, is time consuming and it may involve a variety of operations such as cleaning or unit configuration. One significant issue to be considered when product changeover occurs is concerned with cleaning operations, that may be regularly performed between two consecutive batches for the sake of product quality or plant safety. Thus, the consideration of multiple changeover possibilities increases the number of production schedules to be considered.

Several mathematical formulations have been recently proposed to solve the scheduling problem of multistage batch plants under sequence dependent changeovers, but none of them considers alternative cleaning methods within their formulation. Erdirik-Dogan and Grossmann (2008) present a time slot based formulation which incorporates mass balances and propose a bilevel decomposition algorithm for dealing with medium sized problems. Maravelias and Grossmann (2003b) propose a continuous time MILP model, based on the state task network (STN) representation and apply it to the case of multiproduct batch plants. A resource task network (RTN) based representation is adopted by Castro and Novais (2009), which contains 4-index binary variables. Their formulation considers multiple product batches, sequence-dependent changeovers and poses mass balances. Furthermore, Castro *et al.* (2011) present a greedy algorithm for multistage batch plants with a large number of orders, which are usually intractable to solve with full-space mathematical approaches. As presented in Chapter 2, alternative formulations, which can deal specifically with sequential processes, are based on the general and immediate precedence concepts. The former was firstly introduced by Mendez *et al.* (2001), whereas Gupta and Karimi (2003) presented an immediate precedence model for multiproduct batch plants including sequence dependent changeover time. Recently, Sundaramoorthy and Maravelias (2008) present a formulation based on the general precedence model for considering simultaneous batching and scheduling, multiple storage policies, and extend their formulation to include sequence dependent changeovers by introducing immediate precedence binary variables.

Compared to the general precedence formulation, the immediate precedence model eases the mathematical formulation required for the consideration of sequence dependent schedules because sequencing binary variables are directly related to consecutive batches. For this reason, this Chapter models the scheduling problem using the immediate precedence model initially proposed by Gupta and Karimi (2003). Such model has been extended to consider the possible use of different product changeover cleaning methods, multiple alternative units at each stage, policies of limited storage, product batching and allocation and timing constraints.

Different criteria can be applied to evaluate the solutions of the scheduling problem. The choice of the objective function depends on the decision maker criteria, which are based on his/her experience, the company's goals and the nature of the problem. Hence, a unique objective function is not suitable for all scheduling problems. Therefore, several possible objective functions and their scope are discussed along this chapter. Plant productivity and profit are considered; and their reduction to time metrics is discussed. Moreover, environmental metrics are also proposed and discussed. Precisely, Chapter 9 considers the multiobjective problem when considering simultaneously conflicting objectives, such as profit maximization and environmental impact minimization.

4.3 Problem statement

The multistage scheduling problem which is posed in this chapter can be stated as follows. Considering:

Process operations planning data

- a given time horizon;
- a set of materials: final products, intermediates and raw materials;
- the associated minimum and maximum demands;
- a fixed batch topology consisting of a set of equipment technologies for processing stages;
- a set of fixed product recipes for processing, concerning mass balance coefficients, resources utilization and processing times, with fixed batch size and defined storage policies;
- a set of different product changeover methods;

Economic data

- direct cost parameters such as production or utilities and raw material consumption costs;
- indirect cost parameters such as storage costs;
- changeover cost parameters associated with every possible product sequence combination;
- selling price for every final product;

Environmental data

- raw material production environmental interventions;

- product manufacturing environmental interventions;
- equipment change over environmental interventions;

The goal is to determine:

- the number of batches required to meet the demand (batching);
- the assignment and sequencing of batches (scheduling);
- the allocation of batches to units and storages;
- the appropriate changeover methods required between batches;
- the amount of final products to be sold;

such that a given economic or environmental performance metric as discussed in section 4.5 is optimized.

4.4 Mathematical scheduling model

The model presented by Gupta and Karimi (2003) has been extended to consider different interbatch cleaning methods, product batching, additional objective functions (such as makespan, productivity and environmental impact), possible multiple units at each stage, alternative storage policies, and timing constraints regarding operation simultaneity. The model is decomposed into two parts. First, the product batching problem is considered based on demand and fixed product batch sizes. This allows for the subsequent scheduling problem to determine the number of batches to be produced instead of fixing them beforehand. In this sense, the maximum number of batches has to be set according to the upper bound set on the demand and the fixed batch size.

Next, the allocation, sequencing and timing of the batches resulting from the first problem and associated tasks (i.e. cleaning) are modeled and optimized along a production time horizon according to different objective functions. Scheduling decisions, such as product sequencing, affect environmental and economic considerations. The two stage batching-scheduling formulation is presented in the following subsections. Some relevant modeling features concerning the proposed formulation are considered first.

Multiple units per stage. First, a set of process stages for which alternative units are available is considered (M_k). Such set is ruled by additional variables and must accomplish specific constraints. If all stages had a single possible unit to be performed, the sequential multistage nature of the process would require a single set of sequencing variables, valid for all stages. However, the consideration of multiple units per stage entails additional sequencing variables because only a subset of batches visits each alternative unit.

Storage policies. The original formulation is able to cope with unlimited intermediate storage policies, as well as zero-wait time and no intermediate storage, as a result of the timing constraints. However, intermediate storage policies were not considered. In this work, intermediate storage is modeled as an additional stage of the production process belonging to a special kind of task ($kstor$) since it has no operation time, but only waiting time. In addition, storage tasks are optional, so the sequencing variable for that stage is only defined if the storage is used. Thus, for formulation purposes, these storage stages are included in the set M_k, even though they may be assigned to only one single storage vessel.

Task timing. Both consecutive timing constraints between two stages modeled as well as simultaneity conditions are modeled. Specifically, consecutive, loading and unloading, as well as semicontinuous operations are represented by operation related times at each stage. The relationship among process stages is also distinguished beforehand.

4.4.1 First stage: product batching

The first stage consists of the assignment of production to batches, so that the demand of each product can be fulfilled in the second stage. The number of batches considered must be enough to allow the potential assignment of the complete demand. Each batch i can be assigned to at most one product p (equation 4.1), and the total demand of each product must be fulfilled (equations 4.2 and 4.3). Given that the problems being addressed consider a fixed batch topology, product batch sizes BS_p are fixed. The amount of product produced must lie within the lower (D_p^{min}) and upper bounds (D_p^{max}).

$$\sum_p Y_{ip} \leq 1 \qquad \forall i \tag{4.1}$$

$$\sum_i BS_p Y_{ip} \leq D_p^{max} \qquad \forall p \tag{4.2}$$

$$\sum_i BS_p Y_{ip} \geq D_p^{min} \qquad \forall p \tag{4.3}$$

An additional aim at this stage consists of the definition of process features for each batch, that is, the assignment to each batch of its corresponding product features, such as processing time through the different processing stages, selling price, cost and environmental impact. Therefore, constraints 4.4 and 4.5 establish the time required to fulfill stage k of batch i, and the related o operations: loading ($load$), preparation (pre), processing (pro) and unloading (unl) which all depend on the product p assigned to that batch. Constraints 4.6 to 4.8 are posed for determining the batch benefit, batch size and product environmental impact.

$$T_{ik} = \sum_p time_{pk} Y_{ip} \qquad \forall i,k \tag{4.4}$$

$$T_{ik}^o = \sum_p time_{pk}^o Y_{ip} \qquad \forall i,k \tag{4.5}$$

$$BP_i = \sum_p BP_p Y_{ip} \qquad \forall i \tag{4.6}$$

$$BS_i = \sum_p BS_p Y_{ip} \qquad \forall i \tag{4.7}$$

$$EnvIm_i = \sum_p EnvIm_p Y_{ip} \qquad \forall i \tag{4.8}$$

Changeover cleaning times have been assumed to only depend on products sequence, and different cleaning methods can not be used between the same two batches. Equations 4.9 and 4.10 define the changeover time between any pair of batches for a given cleaning method c, depending on the products assigned to the batches. Identical expressions are considered for the changeover cost and environmental impact associated with every stage k and each pair i, i' of batches.

$$ChT_{ii'kc} \geq chanT_{pp'kc} - M \cdot (2 - Y_{ip} - Y_{i'p'}) \qquad \forall i, i', p, p', k, c \mid i \neq i' \tag{4.9}$$

$$ChT_{ii'kc} \leq chanT_{pp'kc} + M \cdot (2 - Y_{ip} - Y_{i'p'}) \qquad \forall i, i', p, p', k, c \mid i \neq i' \tag{4.10}$$

Finally, constraint 4.11 enforces that each batch can only be assigned if all previous ones have already been, in order to avoid degenerated solutions.

$$\sum_p Y_{ip} \leq \sum_p Y_{i+1p} \qquad \forall i \mid i < max\,(i) \tag{4.11}$$

The objective function of the first stage of the formulation is the total profit as presented in Section 4.5 regardless time horizon constraints. As a result, the maximum number of batches can be pre-assigned, and the second stage regarding batch scheduling is not artificially restricted.

4.4.2 Second stage: batch scheduling

After solving the batching problem, the production and sequencing of the previously assigned batches, which are gathered in a set ($dynI$), are determined. A special feature of the proposed formulation is the production of a starting and finishing batch, required to address the cleaning for the first and last batches, which produce no product, but represent the initial and final still state (S) of the plant. To facilitate the modeling task, an unreal product, whose processing time, cost and environmental impact are zero, is assigned to the previous two batches.

Demand satisfaction. Equation 4.12 imposes that a minimum demand for each product p must be fulfilled. Thus, this equation allows variable production quantities between the minimum demand and the maximum number of batches established in the first stage. Precisely, this constraint assumes that all product batch sizes are fixed, that is, they do not vary between batches of the same product and such batch sizes are problem parameters which were assigned to batches in the first stage of the formulation.

$$\sum_{i \in dynI} W_i \overline{BS_i} \geq D_p^{min} \qquad \forall p \tag{4.12}$$

Allocation constraints. For those stages where alternative units may perform a given stage (M_k), it is necessary to define an assignment variable which relates the processing unit to the batch (Z_{iu}). Specifically, a set U_{uk} which contains the units u which are able to perform stage k is defined. Therefore, constraint 4.13 imposes that at most one unit performs a stage with multiple units, if batch i is produced. However, the previous assignment constraint is only limited to process stages, since storage stages are only assigned if they are actually used in the batch processing. Hence, equation 4.14 is applied instead of 4.13, for batch assignment at storage stages. Constraints 4.15 and 4.16 are necessary to force the value of the binary value to 0, if the storage stage is disregarded. Storage stages have no processing time, but waiting time, so if there is no waiting time for a given k storage task ($k \in kstor$), then the assignment, Z_{iu}, is set to zero. Otherwise, the assignment Z_{iu} is equal to one by means of a bigM constraint (equation 4.16).

$$\begin{gathered}\sum_{u \in U_{uk}} Z_{iu} = W_i \\ \forall\,(i,k)|\, i \in dynI, i \neq 1, i \neq \max(dynI), k \notin \text{kstor}, k \in M_k\end{gathered} \tag{4.13}$$

$$\begin{gathered}\sum_{u \in U_{uk}} Z_{iu} \leq W_i \\ \forall\,(i,k)|\, i \in dynI, i \neq 1, i \neq \max(dynI), k \in \text{kstor}, k \in M_k\end{gathered} \tag{4.14}$$

$$\begin{gathered}\sum_{u \in U_{uk}} Z_{iu} \leq Tw_{ik} \\ \forall\,(i,k)|\, i \in dynI, i \neq 1, i \neq \max(dynI), k \in kstor, k \in M_k\end{gathered} \tag{4.15}$$

$$\begin{gathered}M_2 \cdot \sum_{u \in U_{uk}} Z_{iu} \geq Tw_{ik} \\ \forall\,(i,k)|\, i \in dynI, i \neq 1, i \neq \max(dynI), k \in kstor, k \in M_k\end{gathered} \tag{4.16}$$

The previous assignment equations are valid for all batches assigned to real products. For the batches representing the initial and final still state of the plant, the variables of assignment to units are fixed to one (constraints 4.17 and 4.18). As a result, the initial and final still state is ensured in all units, even if they do not produce any batch.

$$Z_{iu} = 1 \qquad \forall i, k, u \mid i \in dynI, (u,k) \in U_{uk}, k \in M_k, i = 1 \tag{4.17}$$

$$Z_{iu} = 1 \qquad \forall i, k, u \mid i \in dynI, (u,k) \in U_{uk}, k \in M_k, i = \max\left(dynI\right) \tag{4.18}$$

Sequencing constraints. It is necessary to define the sequence in which batches are processed. Therefore, any batch i, with the exception of the first and the last, must have an immediate predecessor and an immediate successor. This condition is enforced by means of constraints 4.19 and 4.20, respectively.

$$\sum_{i',c | i' \in dynI, i \neq i'} X_{ii'c} = W_i \qquad \forall i \mid i \in dynI,\; i < max\left(dynI\right),\; i > 1 \tag{4.19}$$

$$\sum_{i',c | i' \in dynI, i \neq i'} X_{i'ic} = W_i \qquad \forall i \mid i \in dynI,\; i < max\left(dynI\right),\; i > 1 \tag{4.20}$$

The sequencing conditions for the first and last batches, which are fixed and assigned to the still state, are imposed by constraints 4.21 to 4.24.

$$\sum_{i',c|i'\in dynI,i\neq i'} X_{ii'c} = 1 \qquad \forall i,p \mid i=1,\, p=S,\, \overline{Y_{ip}}=1 \tag{4.21}$$

$$\sum_{i',c|i'\in dynI,i\neq i'} X_{i'ic} = 0 \qquad \forall i,p \mid i=1,\, p=S,\, \overline{Y_{ip}}=1 \tag{4.22}$$

$$\sum_{i',c|i'\in dynI,i\neq i'} X_{ii'c} = 0 \qquad \forall i,p \mid i=max\,(dynI)\,,\, p=S,\, \overline{Y_{ip}}=1 \tag{4.23}$$

$$\sum_{i',c|i'\in dynI,i\neq i'} X_{i'ic} = 1 \qquad \forall i,p \mid i=max\,(dynI)\,,\, p=S,\, \overline{Y_{ip}}=1 \tag{4.24}$$

Equations 4.19 to 4.24 are the sequencing constraints for those stages where a single unit is available. In contrast, for those stages with alternative units (M_k), other sequencing conditions must be imposed since not all products visit the same units. Hence, assignment variable Z_{ik} is of crucial importance, and a sequencing variable $X^M_{ii'uc}$ for the units at these stages U_{uk} must be defined. Any batch i assigned to unit u, with the exception of the first and the last, must have an immediate predecessor and an immediate successor at a given unit u ($Z_{iu}=1$). This condition is enforced by constraints 4.25 and 4.26, respectively.

$$\begin{gathered}\sum_{i',c|i'\in dynI,i\neq i'} X^M_{ii'uc} = Z_{iu}\\ \forall i,k,u \mid i\in dynI,\, i<max\,(dynI)\,,\, i>1,\, (u,k)\in U_{uk},\, k\in M_k\end{gathered} \tag{4.25}$$

$$\begin{gathered}\sum_{i',c|i'\in dynI,i\neq i'} X^M_{i'iuc} = Z_{iu}\\ \forall i,k,u \mid i\in dynI,\, i<max\,(dynI)\,,\, i>1,\, (u,k)\in U_{uk},\, k\in M_k\end{gathered} \tag{4.26}$$

Accordingly, the sequencing conditions for the first and last batches, which are fixed and assigned to the still state, are imposed by constraints 4.27 to 4.30.

$$\begin{gathered}\sum_{i',c|i'\in dynI,i\neq i'} X^M_{ii'uc} = 1\\ \forall i,k,p,u \mid i=1,\, (u,k)\in U_{uk},\, k\in M_k,\, p=S,\, \overline{Y_{ip}}=1\end{gathered} \tag{4.27}$$

$$\begin{gathered}\sum_{i',c|i'\in dynI,i\neq i'} X^M_{i'iuc} = 0\\ \forall i,k,p,u \mid i=1,\, (u,k)\in U_{uk},\, k\in M_k,\, p=S,\, \overline{Y_{ip}}=1\end{gathered} \tag{4.28}$$

$$\begin{gathered}\sum_{i',c|i'\in dynI,i\neq i'} X^M_{ii'uc} = 0\\ \forall i,k,p,u \mid i=max\,(dynI)\,,\, p=S,\, \overline{Y_{ip}}=1,\, (u,k)\in U_{uk},\, k\in M_k\end{gathered} \tag{4.29}$$

$$\begin{gathered}\sum_{i',c|i'\in dynI,i\neq i'} X^M_{i'iuc} = 1\\ \forall i,k,p,u \mid i=max\,(dynI)\,,\, p=S,\, \overline{Y_{ip}}=1,\, (u,k)\in U_{uk},\, k\in M_k\end{gathered} \tag{4.30}$$

Timing constraints. As for timing constraints, equation 4.31 determines the end time of stage k of batch i, from the starting time (Ts_{ik}), the operation o time ($\overline{T_{ik}^{o}}$) and the waiting time (Tw_{ik}), provided such batch is eventually produced, that is, the binary variable (W_i) is 1. By definition, the operation time of storage stages is 0, and only waiting time is possible.

$$Tf_{ik} = Ts_{ik} + \overline{T_{ik}} W_i + Tw_{ik} \qquad \forall i,k \mid i \in dynI \tag{4.31}$$

In addition, timing constraints among the different stages are necessary. Constraint 4.32 enforces the condition that for two consecutive stages, the unloading start time of the first one must be equal to the loading start time of the following one. This equation assumes that there is no intermediate storage of products between consecutive stages; however, by modeling storages as additional stages, intermediate storage policies can be adopted.

$$Ts_{ik+1} + \overline{T_{ik+1}^{prep}} = Tf_{ik} - \overline{T_{ik}^{unlo}} \qquad \forall i,k \mid i \in dynI,\, k \in kcon \tag{4.32}$$

When two stages are simultaneous, that is, their loading, operation and unloading occur at the same time, constraint 4.33 enforces the load starting time of both stages to be equal. This constraint also models fed-batch stages, e.g. a filter that requires a feed and outlet pump to work simultaneously for its operation.

$$Ts_{ik+1} + \overline{T_{ik+1}^{prep}} = Ts_{ik} + \overline{T_{ik}^{prep}} \qquad \forall i,k \mid i \in dynI,\, k \in kpar \tag{4.33}$$

Equation 4.34 forces that the loading start time of a given $k+1$ stage to be equal to the time at which the operation of the previous stage k starts. This condition is useful for semicontinuous operations.

$$Ts_{ik+1} + \overline{T_{ik+1}^{prep}} = Tf_{ik} - \overline{T_{ik}^{unlo}} - \overline{T_{ik}^{proc}} \qquad \forall i,k \mid i \in dynI,\, k \in kpum \tag{4.34}$$

An additional timing constraint is defined to handle batch changeover times. The production sequence affects both the changeover time and changeover method c. Equation 4.35 defines the changeover time for two consecutive batches in a given stage k, according to the cleaning method used. Therefore, the binary variable $X_{ii'c}$ is 1 in case batch i is immediately processed before batch i' using cleaning method c.

$$\begin{gathered} Ts_{i'k} \geq Tf_{ik} + \overline{ChT_{ii'kc}} X_{ii'c} - M_2 \left(1 - X_{ii'c}\right) \\ \forall i,i',k \mid (i,i') \in dynI,\, i \neq i',\, k \notin M_k \end{gathered} \tag{4.35}$$

Equation 4.36 defines the changeover time for two consecutive batches in a given stage $k \in M_k$, depending on the cleaning method used. Therefore, the binary variable $X_{ii'uc}^{M}$ is 1 in case batch i is immediately processed before batch i' using cleaning method c, if batches i and i' are assigned to unit u.

$$\begin{gathered} Ts_{i'k} \geq Tf_{ik} + \overline{ChT_{ii'kc}} X_{ii'uc}^{M} - M_2 \left(1 - XM_{ii'uc}\right) \\ -M_2 \left(2 - Z_{iu} - Z_{i'u}\right) \qquad \forall i,i',u,k \mid (i,i') \in dynI,\, i \neq i', \\ k \in M_k,\, (u,k) \in U_{uk} \end{gathered} \tag{4.36}$$

The production horizon H defines the maximum time at which the last stage of any batch is allowed to finish (equation 4.37).

$$W_i H \geq Tf_{ik} \qquad \forall i,k \mid i \in dynI \tag{4.37}$$

In addition, timing constraints related to order due dates dd_i may be imposed. Therefore, equation 4.38 defines the tardiness of a batch i (Tar_i), whereas equation 4.39 the earliness (Ear_i).

$$Tar_i = \max\{0, Tf_{ik} - dd_i\} \qquad \forall i,k \mid i \in dynI \tag{4.38}$$

$$Ear_i = \max\{0, dd_i - Tf_{ik}\} \qquad \forall i,k \mid i \in dynI \tag{4.39}$$

Rescheduling extension. The set $dynI$ facilitates the modeling of the rescheduling problem in a straightforward manner. Specifically, when a certain production plan is interrupted, the pending orders are included again in the former set, and the starting time of the stages k of such batches is enforced to be lower than the time availability of their corresponding units.

4.5 Objective functions selection

The main objective of batch production planning and scheduling is to optimize capacity utilization of batch manufacturing facilities and fulfill customer orders within a specific time horizon (Barker & Rawtani, 2005). As a main building block of enterprise-wide optimization, the scheduling level pursues the overall company objectives which arise from economic, environmental and social aspects.

Economic criteria are of utmost importance in process industry. Hence, multiple economic objectives can be adopted in process scheduling, depending on the decision maker preferences. Thus, either an absolute economic measure, such as total profit, or a time relative measure, such as productivity or profitability could be adopted to assess the decisions. The former criteria could be more suitable for those industrial environments where prices and demand have low uncertainty, and working hours are fixed; whereas process productivity and profitability are more interesting in those environments where late orders may arrive and variable costs are more important than fixed costs, and consequently the main objective is to produce the most profitable products using the least time. In academic studies related to scheduling, the economic objective function is usually quantified via time metrics, such as makespan, lateness or earliness (Korovessi & Linninger, 2006; Méndez *et al.*, 2006). However, profitability maximization is only equivalent to makespan minimization under certain conditions. Specifically, they are equivalent under the same trends in cost and time for changeover, if (i) the produced quantity is fixed, or (ii) under time constraints and variable production quantities if all products are equivalent from a profitability point of view, that is, they have the same profit and production time along the different stages. Only in such cases, profitability maximization may be reduced to makespan minimization.

Companies must face nowadays tighter environmental regulations. Hence, environmental objectives have to be considered as part of the optimization process (Cano-Ruiz & McRae, 1998). The objectives could be again expressed in absolute measures, for example, the minimization of the total environmental impact, which requires the definition of a minimum demand satisfaction to avoid zero production rates; or a relative measure, such as the minimization of the total environmental impact per mass of product produced. In this case, the lack of production leads to higher penalties.

4.5.1 Economic goals

The total profit objective function, which considers product benefits ($\overline{BP_i}$) and changeover costs ($\overline{ChCost_{ii'kc}}$), is defined by equation 4.40. Parameter $\overline{BP_i}$ includes the production revenues ($\overline{PR_i}$) minus its raw materials and utilities (e.g. electricity, heat and water) costs ($\overline{PC_i}$) associated directly with its production, while $\overline{ChCost_{ii'kc}}$ considers the cost associated with the inter-batch cleaning operations. Thus, the economic function could include costs associated to storage usage. The estimation of profit using Equation 4.40 has been similarly performed by other authors (Erdirik-Dogan & Grossmann, 2008) and is widely used in the scheduling decision level. Profitability (equation 4.41) results from dividing the total profit by the production schedule makespan (equation 4.42). Thus, productivity (equation 4.43) is defined as the total amount produced divided by the makespan.

$$z^{profit} = \sum_i (\overline{PR_i} - \overline{PC_i})W_i - \sum_{i,i',c|i\neq i'} X_{ii'c} \sum_k \overline{ChCost_{ii'kc}} \tag{4.40}$$

$$z^{profy} = \frac{z^{profit}}{z^{Mk}} \tag{4.41}$$

$$z^{Mk} = Tf_{ik} \qquad \forall i,k \mid k = max\,(k)\,,\ i = max\,(i) \tag{4.42}$$

$$z^{prod} = \frac{\sum_i (\overline{BS_i})W_i}{z^{Mk}} \tag{4.43}$$

As previously mentioned, it is common practice to reduce economic criteria to timing goals at the scheduling level. Hence, objective functions related to production order due dates are usually posed. Examples of this are the minimization of total tardiness (Equation 4.44) or total earliness (Equation 4.45).

$$z_T^{tard} = \sum_i Tar_i \tag{4.44}$$

$$z_T^{earl} = \sum_i Ear_i \tag{4.45}$$

However, time related goodness measures are not adequate to capture the whole complexity of plant process operations, whereas economic metrics provide a means for the integration of scheduling with other decision levels such as process or planning areas. For example, the consideration of variable processing recipes, whose operating conditions may have an influence over cost, can lead to improved decisions from an integrated perspective as will be shown in Part Part III.

4.5.2 Environmental goals

Environmental criteria must be assessed considering the specific features of the production process. The total environmental impact accounts for both the impact of the production process ($\overline{EnvIm_i}$) and that associated to changeover tasks ($\overline{EnvIm_{ii'kc}}$) (equation 4.46). A relative environmental impact can also be obtained dividing the total environmental impact by the produced quantity (equation 4.47). Chapter 9 further discusses the methodology for evaluating the environmental impact.

$$z^{ei} = \sum_{i,i',c|i \neq i',i' \in dynI} X_{ii'c} \sum_{k} \overline{EnvIm_{ii'kc}} + \sum_{i|i \in dynI} W_i \overline{EnvIm_i} \tag{4.46}$$

$$z^{rei} = \frac{z^{ei}}{\sum_{i|i \in dynI} W_i \overline{BS_i}} \tag{4.47}$$

4.5.3 Formulation issues

The former objective functions are either lineal (equations 4.40, 4.42, 4.44, 4.45 and 4.46) or fractional (equations 4.41 and 4.47). Therefore, insomuch as the mathematical scheduling model is formulated using lineal constraints, the objective function will determine the problem type. Specifically, if a lineal objective function is considered, the resulting scheduling problem is a MILP. In contrast, for a fractional objective, a MINLP problem must be tackled.

In this case, it must be pinpointed that the non-linearity is associated with the objective function and not with the scheduling equations that are all linear (constraints 4.1 to 4.39). Furthermore, the non-linear objective functions are the ratio of two linear functions. These MINLP problems belong to a special type of MINLP, namely mixed-integer linear fractional programs (MILFP). Because of the non-linear nature of the MILFP problems and the combinatorial complexity of the scheduling problem, the resulting model may result computationally intractable specially for large instances. You *et al.* (2009) extend the Dinkelbach's algorithm, which originally exploits the relationship between nonlinear fractional programming and nonlinear parametric programing to solve convex nonlinear fractional problems in order to obtain the global optimal solution of an MILFP problem by solving a sequence of MILP subproblems. As a result, large-scale MILFP problems can be tackled better than with standard MINLP methods.

4.6 Examples

Two examples illustrate the capabilities of the presented mathematical formulation, which has been implemented in GAMS, and solved using CPLEX 11.2 for the MILP problem and BARON 8.1 for the MINLP cases in a 2.26 GHz Intel Core Duo computer.

4.6.1 Example 1: Multi-stage batch plant with alternative units

This example was originally presented by Sundaramoorthy and Maravelias (2008). It considers a multi-stage batch plant which processes 10 batches of known fixed sizes.

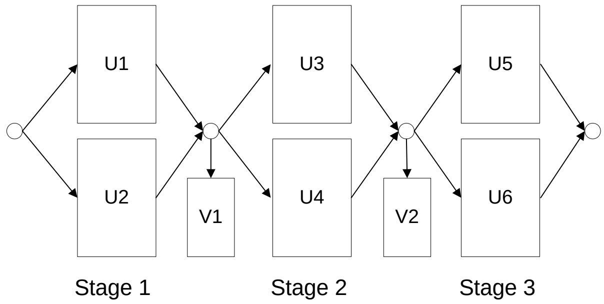


Figure 4.1: Multi-stage batch plant structure of Example 1.

Each batch has to sequentially undergo three processing stages (Figure 4.1) which have two alternative units. Thus, after stages 1 and 2, there is a shared-exclusive storage tank. Additionally, sequence dependent batch times are considered. Table 4.1 contains the data of processing time at the different stages and Table 4.2 presents the data for sequence-dependent changeover times.

Table 4.1: Processing times [h] for Example 1.

Unit-Batch	A	B	C	D	E	F	G	H	I	J
U1	4.66	7.02	3.89	2.67	5.46	6.23	6.24	2.67	12.01	9.99
U2	5.01	9.32	3.00	5.00	5.01	6.99	9.35	3.33	13.32	10.68
U3	4.16	6.00	3.34	3.01	4.16	4.16	4.57	9.00	10.01	6.00
U4	5.01	8.00	3.99	4.18	6.00	5.01	5.85	5.83	6.68	8.00
U5	5.99	9.79	7.01	5.33	7.01	8.01	6.66	7.12	11.00	9.79
U6	5.84	4.99	4.16	5.99	5.00	7.50	5.50	6.75	7.01	6.00

Table 4.2: Sequence-dependent changeover times [h] for Example 1.

	A	B	C	D	E	F	G	H	I	J
A	0	1	1	2	3	4	3	2	4	3
B	2	0	2	1	2	2	1	2	1	2
C	1	4	0	2	2	3	2	1	2	1
D	2	2	3	0	1	3	1	2	1	2
E	3	1	1	1	0	1	2	1	1	2
F	2	2	2	1	4	0	3	4	2	2
G	1	1	4	2	1	3	0	2	2	1
H	2	2	3	2	3	2	1	0	1	1
I	1	3	1	1	2	1	3	2	0	2
J	2	4	1	2	2	3	4	3	1	0

In this case, no batching stage is needed to assign products to batches. In addition, the production of all batches is compulsory. Intermediate storage is available and multiple units are possible at each stage. Hence, the scheduling model includes equations 4.13 to 4.18, 4.25 to 4.32 and 4.36 to 4.39. Two objective functions are considered, namely total tardiness minimization and total earliness minimization for two cases with different release and due times (Table 4.3).

Table 4.3: Release and due times [h] for Example 1.

	[Total tardiness]		[Total earliness]	
	Release time [h]	Due date [h]	Release time [h]	Due date [h]
A	0	10	0	40
B	0	25	10	30
C	5	15	5	40
D	20	40	10	50
E	5	30	20	40
F	20	40	0	50
G	10	30	10	50
H	15	40	5	40
I	30	50	20	70
J	5	50	15	70

The proposed scheduling formulation successfully represents the specific features of this example, and the problem is also solved to optimality in both cases (Figures 4.2 and 4.3).

It is noteworthy to mention that the results obtained in this work improve those values reported in the original paper, in which the tardiness was 24.2h and earliness 42.7h. As for tardiness minimization, the order in which batch processing starts corresponds to the batch due times, respecting the batch order release times. As a result, a total tardiness of 20.29h is obtained. Regarding earliness minimization, all orders are sequenced in such a way that their completion time is equal to or higher than their due time, so the optimal total earliness is 0h. Thus, it can be observed that storage utilization for total earliness minimization is much higher than in the other case.

4.6.2 Example 2: Multi-product fiber plant

The proposed formulation is applied to the scheduling of a multi-product batch process originally posed by Grau *et al.* (1996). The plant produces three acrylic fiber formulations (A, B and C) by a suspension polymerization process (Figure 4.4) comprising 14 processing stages. Due to minimization of inventory costs, the possible storage of polymer (considered as intermediate product) after stages deaeration (stages 11, 12) has been disregarded and polymer extrusion (stage 13) is performed right after polymer deareation is done. Production recipes contain a detailed description of the product batch sizes and energy demands based on the real production plant (Grau *et al.*, 1996). Data regarding operation times, production costs, product prices, environmental impact as well as alternative changeover methods are provided in Appendix D based on the data of the original authors. This example stands for an illustrative case study which is revisited along this thesis in order to illustrate the proposed approaches.

In this example, the first stage of the formulation related to product batching is needed in order to assign products to batches. Thus, since a non-intermediate storage policy is considered and a single unit is available at each stage, equations 4.12, 4.19 to 4.24, 4.31 to 4.35 and 4.37 are used for modeling the scheduling problem. Two different sized problems are posed: (i) a case with large demand considering a single cleaning method, and (ii) a case with medium sized demand and two alternative cleaning methods. Two objective functions are considered, namely profit and profitability maximization. The former criterion results in a MILP, whereas the latter gives rise

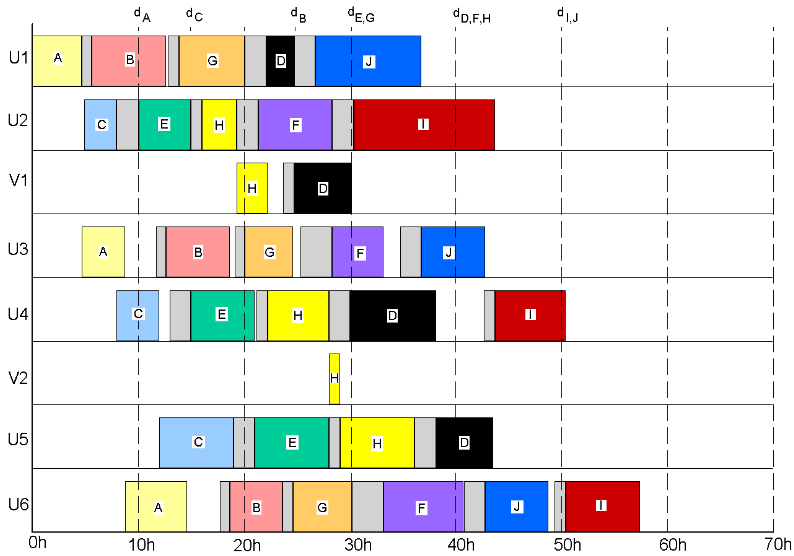


Figure 4.2: Gantt chart of an optimal schedule considering total tardiness minimization of Example 1, total tardiness is equal to 20.29h.

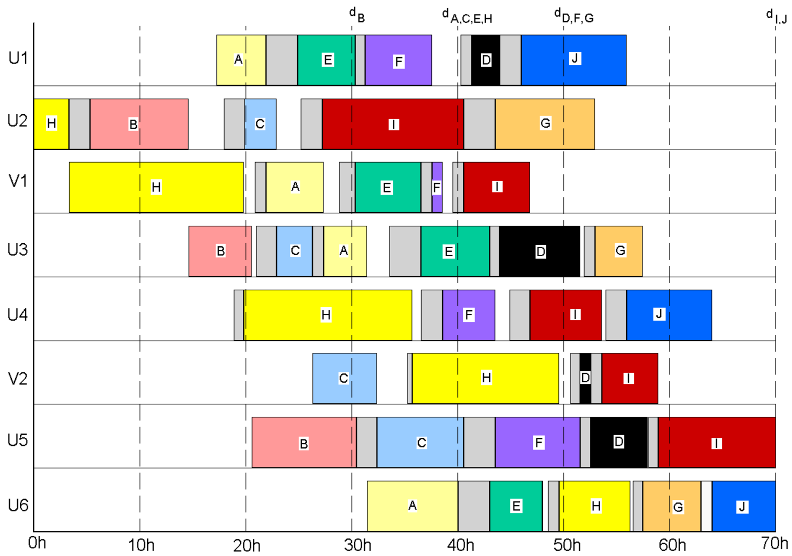


Figure 4.3: Gantt chart of an optimal schedule considering total earliness minimization of Example 1, total earliness is equal to 0.0h.

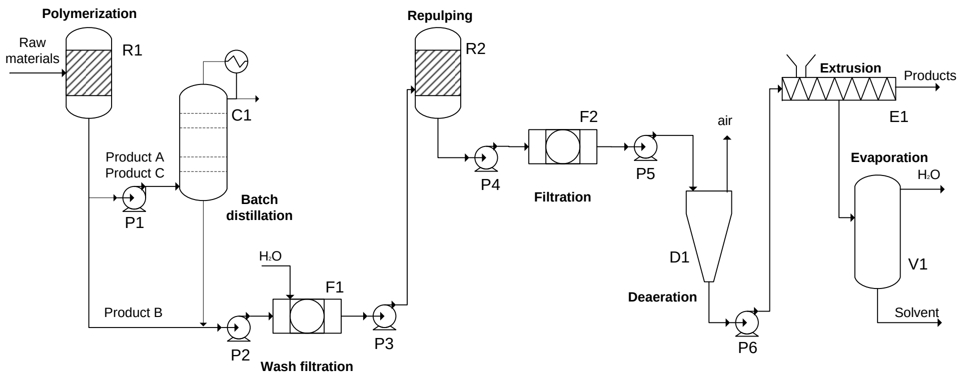


Figure 4.4: Plant structure of the production process of acrylic fibers (Example 2).

to a MILFP problem which is solved using Dinkelbach's algorithm. The effectiveness of Dinkelbach's algorithm is checked by comparing its performance with the MINLP problem for case (ii). Economic data include batch production costs and changeover costs, as shown in Appendix D.

Case i considers the fulfillment of the whole demand presented in Table D.2 in a time horizon of 120h. Changeover method 1 (Figures D.3 and D.5) is selected in this case for illustrating purposes.

Figures 4.5 and 4.6 show the Gantt charts resulting from profit and profitability maximization, respectively. Differences between the two results stem from production sequences. In the case of profit maximization, three single product campaigns are sequentially processed, namely firstly five batches of fiber B, next seven batches of C, and finally the whole demand of fiber A is fulfilled. In contrast, for profitability maximization, fiber A is firstly produced, next five batches of fiber C, followed by the whole demand of fiber B and finally the remaining batches of product C. Although the profit of the latter solution is lower, the corresponding makespan makes up for such decrease since the combination of campaigns of products B and C reduces considerably the production time, and the overall profitability is better than for the former solution (Table 4.4).

It must be also noted that only for profit maximization, optimality is guaranteed; whereas regarding profitability maximization, the optimality of the reported solution cannot be guaranteed, since the maximum time for each iteration, 7200 sCPU, was reached in the some of the 4 iterations of the Dinkelbach's algorithm that were necessary to fulfill the end condition. Indeed, the nonlinear problem involving production makespan is a hard problem from a computational point of view, and alternative strategies should be applied. Hence, in Chapter 9, an hybrid method consisting of metaheuristics and mathematical local search is proposed for large sized problems.

Thus, the Gantt charts present the capabilities of the formulation for modeling timing constraints between stages. Specifically, the synchronization of the loading and unloading operations related to equipment and pumping units has been successfully modeled, and the simultaneity conditions for filtration, deaeration and extrusion stages are illustrated. Therefore, not only can consecutive batch stages be modeled with the proposed formulation, but also semicontinuous and transfer operations.

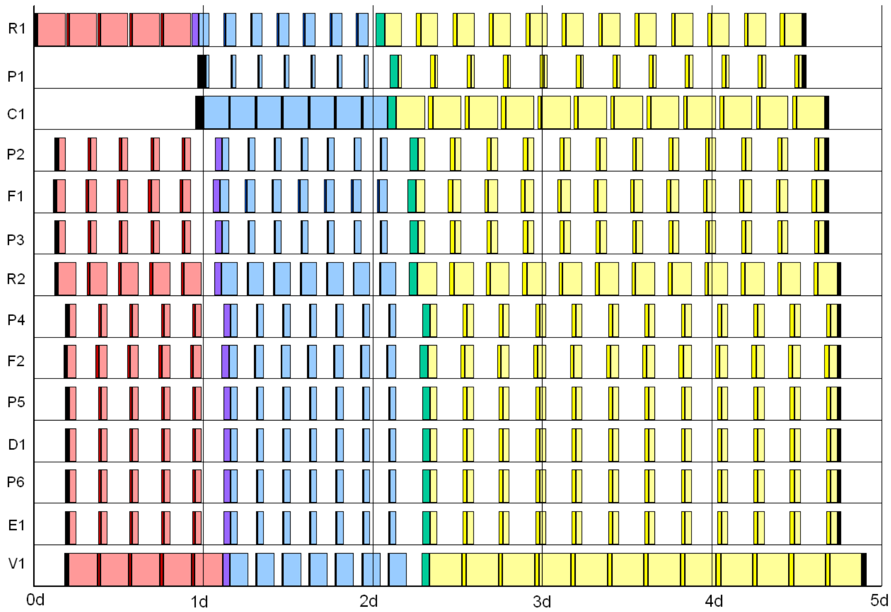


Figure 4.5: Gantt chart of an optimal schedule considering profit maximization in Example 2(i), profit is equal to $170.41{\cdot}10^3$ m.u.(black: starting and finishing cleaning tasks; yellow, red and blue: fibers A, B and C, respectively; darker colored areas represent changeover).

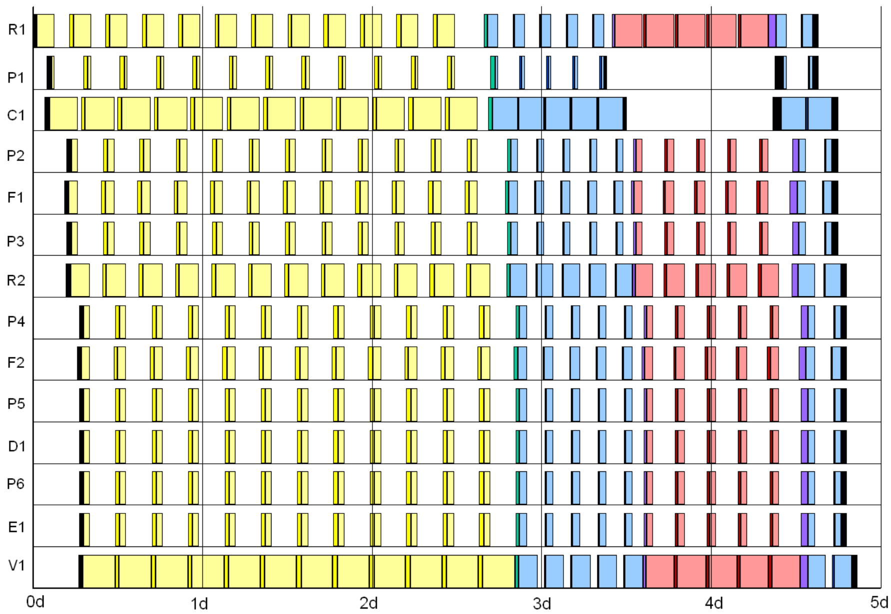


Figure 4.6: Gantt chart considering profitability maximization in Example 2(i), profitability is equal to $1.457{\cdot}10^3$ m.u./h (black: starting and finishing cleaning tasks; yellow, red and blue: fibers A, B and C, respectively; darker colored areas represent changeover).

Table 4.4: Profit and profitability for the different subcases of Example 2.

		z^{profit} [$\cdot 10^3$m.u.]	z^{profy} [$\cdot 10^3$ m.u./h]	Mk [h]
Case i	max z^{profit}	**170.41**	1.445	117.90
	max z^{profy}	170.09	**1.457**	116.75
Case ii	max z^{profit}	**40.98**	1.041	39.35
	max z^{profy}	38.65	**1.166**	33.15

Table 4.5: Computational comparison for productivity maximization using MINLP and Dinkelbach's algorithm for Example 2.

Method	z^{profy} [m.u./h]	Optimality gap [%]	Time [sCPU]
MINLP	1.166	42.28	7200
Dinkelbach's algorithm	1.166	0	202

Case ii considers the fulfillment of the demand of two batches of each product in a time horizon of 40h. Changeover methods 1, 2 and 3 are possible alternative cleanings in this case (Figures D.3 and D.5).

Table 4.4 presents the results for this case and figures 4.7 and 4.8 show the Gantt charts of the optimal schedules resulting from profit and profitability maximization, respectively. Differences between the two subcases stem from production sequences and changeover methods. As for profit maximization, changeover method 2, which is the most economic but most time consuming, is selected for most changeover operations but not all of them, in order to meet the time horizon constraint (initial and final changeovers, as well as the changeovers to batches B are performed with method 1). The resulting production sequence consists of producing two batches of A, next two batches of C, and finally two batches of polymer B. In contrast, for profitability maximization changeover method 1 is selected in all cases, and the most profitable changeover production sequence consists of producing two batches of A, next one batch of product C, two batches of B, and finally one batch of product C. Such production sequence is more profitable than the single campaign production mode, because the reduction of makespan is more significant than the increase in cost (note that fiber B does not require the separation stage after polymerization).

As for the computational efficiency, it is noteworthy to mention that the profit minimization problem is solved in 8 sCPU in this case. However, the computational cost of the productivity maximization is much higher. Table 4.5 shows that using Dinkelbach's algorithm the problem can be solved to optimality in about 200 sCPU; however, using the MINLP strategy based on branch-and-bound, the global optimality could not be guaranteed and the difference between the upper and lower bounds was over 42.3 % in two hours CPU. Therefore, the efficiency of Dinkelbach's algorithm to solve MILFP is proved in this example.

4.7 Final remarks

Highly effective scheduling formulations which model different problem features necessary for the decision maker are required. In this sense, this chapter presents a MILP formulation based on the immediate precedence concept which is able to

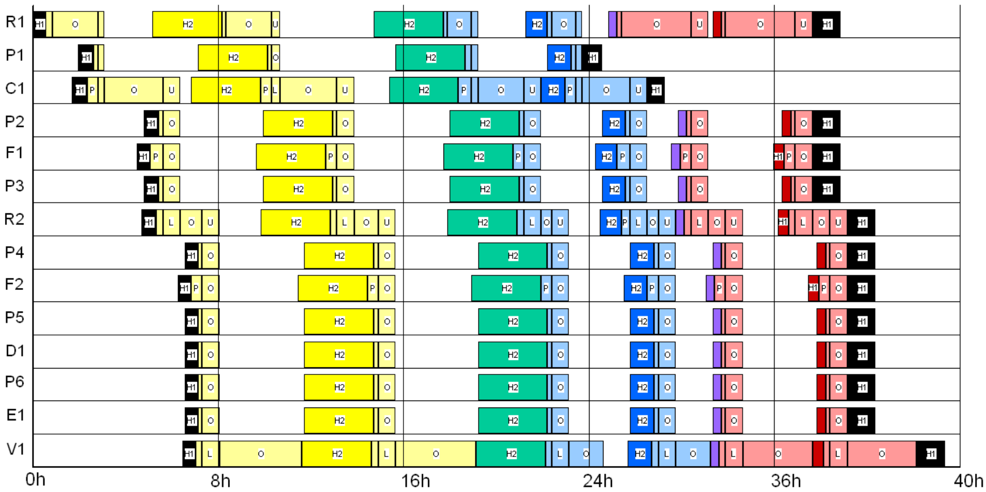


Figure 4.7: Gantt chart of an optimal schedule considering profit maximization in Example 2(ii), profit is equal to $40.98{\cdot}10^3$ m.u.(black: starting and finishing cleaning tasks; yellow, red and blue: fibers A, B and C, respectively; darker colored areas represent changeover methods).

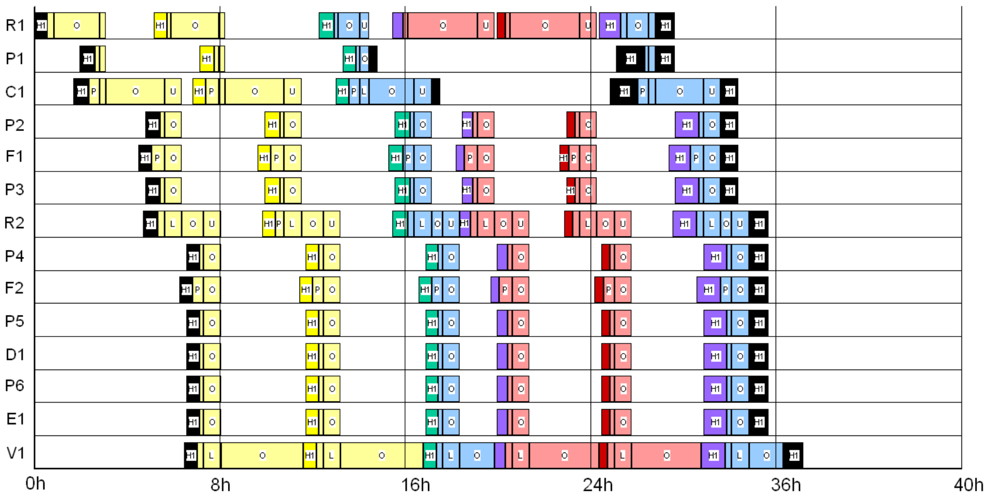


Figure 4.8: Gantt chart of an optimal considering profitability maximization in Example 2(ii), profitability is equal to $1.166{\cdot}10^3$ m.u./h (black: starting and finishing cleaning tasks; yellow, red and blue: fibers A, B and C, respectively; darker colored areas represent changeover methods).

consider alternative changeover methods, product batching, alternative processing units, different storage policies and timing constraints. Additionally, several objective functions related to economic, environmental and timing criteria can be considered in the scheduling problem. Economic objectives stand for the more general metrics to be applied for integrating purposes. Such criteria can only be reduced to timing metrics under certain conditions. Chapter 9 presents the multiobjective optimization of the scheduling problem under conflicting objectives.

The different capabilities of the proposed formulation to deal with specific scheduling features in a wide range of applications are highlighted through two examples. Therefore, a multistage multiproduct batch plant with multiple units per stage and limited intermediate storage has been optimized considering time related objective functions. In addition, a multiproduct batch plant producing acrylic fibers with alternative changeover methods has been studied for different problem sizes and different economic metrics. In this example, timing constraints related to the synchronization between operations of different stages have been considered. Furthermore, the computational complexity of large size problems and non-linear models motivates the exploration of alternatives to pure mathematical optimization techniques, which will be presented in Chapter 9.

On the whole, the applicability of the proposed model has been tested in different scenarios. The scheduling model presented herein can be regarded as a building block for further improvements in the modeling and convergence for integration with other decision levels, as it is discussed in the following chapters.

4.8 Nomenclature

Sets and subsets

c	Cleaning modes between products.
i	Batches.
k	Stages.
p	Products (product S simulates plant 'still' state).
u	Units.
$dynI$	Batches i that have been assigned to a product.
$kcon$	Stages k whose following stage operation is parallel to their unload.
$kpar$	Stages k which are parallel in operation to the following one.
$kpum$	Stages k whose following stage is being loaded while they are operating.
$kstor$	Stages k which are used to model storage vessels.
M_k	Stages k which may be performed in multiple units.
U_{uk}	Units u which may be used to perform stage k.

Parameters

BP_p	Batch price of product p.
$\overline{BP_i}$	Benefit resulting from the production of batch i.
BS_p	Batch size of product p (which is fixed).
$\overline{BS_i}$	Batch size of batch i.
$chanT_{pp'kc}$	Changeover time between products p and p' in stage k with cleaning mode c.
$\overline{ChCost_{ii'kc}}$	Changeover cost between batches i and i' for stage k using changeover type c.
$\overline{ChT_{ii'kc}}$	Changeover time between batches i and i' for stage k using changeover type c.
D_p^{MIN}	Minimum demand of product p that has to be accomplished.
D_p^{MAX}	Maximum demand of product p that can be accomplished.
dd_i	Due date of batch i.
$EnvIm_p$	Production impact resulting of producing a batch of product p. It includes: raw materials, electricity, residues, steam, water and emissions.
$\overline{EnvIm_i}$	Production impact resulting of producing a batch i.
$EnvIm_{ii'kc}$	Environmental impact associated with changeover type c between batches i and i' for stage k.
H	time horizon.
M	Parameter with a big value, in this case its minimum value is 3 times the maximum cost, environmental impact or time between any pair of products.
M_2	Parameter with a big value, in this case its minimum value is the time horizon.
$\overline{PC_i}$	Production costs associated to batch i.
$\overline{PR_i}$	Product revenues associated to batch i.
$ptime_{pk}$	Total processing time before stage k of product p.
rd_i	Release date of batch i.
$\overline{T_{ik}}$	Total processing time of stage k of product i.
$\overline{T_{ik}^{operation}}$	Time parameter of stage k in batch i for the different operations, i.e. preparation, loading, cleaning, operation and unloading.

$\overline{Y_{ip}}$	States if product p is being carried out in batch i (it is defined after the first stage, which assigns products to batches).

Continuous variables

$ChT_{ii'kc}$	Changeover time of doing i and then i' in stage k through cleaning method c.
Ear_i	Earliness of batch i, defined as positive variable.
pT_{ik}	Time of stage k in order i.
Tar_i	Tardiness of batch i, defined as positive variable.
Ts_{ik}	Starting time of stage k of batch i.
Tf_{ik}	Completion time of stage k of batch i.
Tw_{ik}	Waiting time of stage k of batch i.
z_T^{ear}	Objective function that aims at minimizing the total earliness.
z^{ei}	Objective function that aims at minimizing the environmental impact.
z^{Mk}	Objective function that aims at minimizing the makespan.
z^{prod}	Objective function that aims at maximizing productivity.
z^{profit}	Objective function that aims at maximizing profit.
z^{profy}	Objective function that aims at maximizing profitability.
z^{rei}	Objective function that aims at minimizing the relative environmental impact.
z_T^{tar}	Objective function that aims at minimizing the total tardiness.

Binary variables

W_i	Production of batch i.
$X_{ii'c}$	Assignment of cleaning method c to changeover, if batch i is produced immediately before batch i'.
$X_{ii'uc}^M$	Assignment of cleaning method c to changeover, if batch i is produced immediately before batch i' in unit u.
Y_{ip}	Assignment of product p to batch i.
Z_{iu}	Assignment of batch i to unit/vessel u.

Chapter 5

Modeling Transfer Operations in Multipurpose Plants

5.1 Motivation

A useful model must represent all those problem characteristics affecting the decision maker's decisions. Precisely, one of the key complex features to be considered in batch processes operations is the representation of the material transfer between process stages. A non-zero time as well as certain conditions and resources are always required to move 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. As a result, equipment availability is restricted by transfer tasks, and overall resource utilization time is affected.

Therefore, an effective short-term scheduling formulation must simultaneously tackle several problem difficulties commonly arising in batch processes preserving the problem complexity to a manageable level. In this sense, 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, it is demonstrated in this chapter 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 the serious limitations of current MILP-based scheduling approaches, a general precedence based framework which models batch transfer features is introduced. In addition, an algorithm which is able to identify those transfers that would lead to unfeasible schedules is proposed, and could be introduced in existing formulations to disregard such sequences beforehand. The defined problem and the proposed solution procedure are illustrated in two different examples from the literature.

5.2 The nature of transfer operations

Scheduling, not only for manufacturing operations, is a common requirement for process industry, management and services. Broadly speaking, scheduling may be defined as a decision-making process of allocation of scarce resources to a set of activities over time. 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, 1997; French, 1982; Baudin, 1990). Nevertheless, operations research particularly focus on mechanical plants rather than chemical factories, applying fairly simple process models that are unable to capture the complexity of standard chemical operations. For this reason, the scheduling of chemical industry operations are 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 next stage 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 which is a common feature in mechanical industries. 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 processing stages, complex process layouts and topological implications which usually have a significant influence on the short-term scheduling problem complexity. 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. These simplifications are not applicable to standard batch chemical facilities. 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 scheduling of chemical plants, the assumption of negligible transfer times is generally accepted. The complexity of managing these transfer operations is usually avoided 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 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. Hence, transfer times are usually neglected or assumed to be lumped into the batch processing time. 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 (Mendez *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 highlights that the assumptions introduced during the modeling process must be carefully analyzed in order to avoid generating schedules with unfeasible operational sequences. Although simplifications are necessary to reduce problem complexity, they must be carefully considered in order to avoid unreal solutions. Firstly, an example illustrates the generation of unfeasible scheduling solutions when transfer tasks are disregarded in the modeling step. In addition, some of the most well-known mathematical programming scheduling formulations are reviewed and analyzed in order to identify the sources of unfeasible schedules. A detailed mathematical formulation for modeling transfer operations based on the general precedence formulation is presented, along with two stage approach for its efficient solution. Finally, two examples from the literature are employed to illustrate the drawbacks of disregarding transfer tasks and the optimal feasible schedules obtained with the defined strategy.

5.3 An illustrative example

Let us consider a multipurpose plant producing 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 1 for 3 hours, and then transferred to unit 2 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 2 for 2 hours and then in unit 1 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.

For this straightforward example, the solution depicted in the Gantt chart of Figure 5.1 (a) 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 1 to unit 2 and

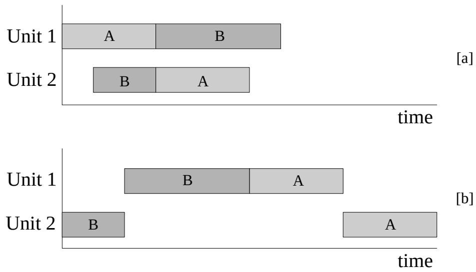


Figure 5.1: Illustrative example regarding the importance of transfer operations in multipurpose plants: [a] unfeasible situation, [b] feasible schedule.

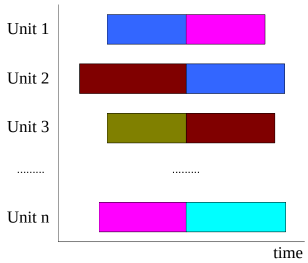


Figure 5.2: Generalization to n products.

also material from unit 2 to unit 1 at the same time. When the first stage of product A is finished, unit 2 needs to be completely empty to receive material from unit 1. But in fact, unit 2 is trying to discharge product B to unit 1. 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 5.1(b) shows the feasible optimal solution with a makespan of 12 hours, almost 86% greater than the unfeasible one shown in Figure 5.1(a). Here it is worth remarking that the unfeasible schedule cannot be slightly modified in order to be feasible by only performing left or right shifting. New sequencing decisions are required to guarantee its feasibility.

An intermediate storage tank does not guarantee feasibility in all instances. For example, in 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 5.2) if the materials are simultaneously transferred between them and there are less than n-1 additional intermediate storage units.

5.4 Problem identification

Analyzing the situation described in the illustrative example, the following claims are derived:

Claim 1. The unfeasible situations may only appear in multipurpose plant configurations, where bidirectional flow is permitted. In multipurpose plants, routes of different products may undergo equipment units in reverse direction. In contrast, multiproduct batch plants always operate following unidirectional flow that cannot generate unfeasible sequences. In this particular case, non-zero 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 the condition of non-UIS. Storage policy assumption is an important issue. Under the simple consideration of unlimited intermediate storage (UIS), unfeasible solutions do not occur since additional storage 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 root 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, tasks synchronization among units is ignored, leading to unfeasible solutions in practice as shown in the previous example. Hence, the modeling of multipurpose batch plants with limited storage policies must consider transfer times. Otherwise, synchronization among tasks regarding transfer times is completely disregarded, 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.

5.5 Mathematical programming formulations for multipurpose plants

The scheduling problem has received a great attention over the last decades as a manner to improve the efficiency of batch chemical processes usually aimed at the production of high-added value products. Optimization models for batch scheduling are usually classified according to their time representation. Continuous time formulations allow scheduling events to occur at any time point along time horizon, whereas discrete time formulations have only a fixed number of time points. Discrete time models entail large size problems and even unfeasible schedules may be generated, however they have proved to be very efficient, adaptable and convenient for many industrial applications. Continuous time representation reduces the number of variables, and results in more flexible solutions, but at the same time it entails more complicated constraints, thereby increasing the model complexity.

Moreover, scheduling models can be distinguished according to the way they arrange the event points over the time horizon. Event points are used to guarantee that resource limitations are not exceeded. Discrete time formulations use global time intervals, whereas continuous formulations have a wider range of possible representations. On the one hand, the global time point representation corresponds to a generalization of global time intervals where timing of intervals is treated as a new model variable. This formulation employs a predefined time grid that is valid for all shared resources involved in the scheduling problem. On the other hand, unit specific time events use a different time grid for each resource. Other kinds of formulations are time slots and batch precedence based. Both of them are oriented toward sequential processes. The former defines a set of time intervals of unknown duration, whereas the latter enforces the sequential use of resources through model variables and constraints. In addition, scheduling problems are also classified depending on the representation of the material balances. These methods are able to deal with arbitrary network processes involving complex product recipes. These models employ the state-task network (STN) or the resource task network (RTN) concept to represent the problem. The STN-based models represent the problem assuming that processing tasks produce and consume states. The RTN-based formulations employ a uniform treatment and representation framework for all available resources through the idea that processing and storage tasks consume and release resources at their beginning and ending times, respectively. In this section, three of the most widely used scheduling continuous-time formulations available in the literature have been selected and implemented for examples of multipurpose batch plants with different storage policies in a minimization makespan problem. The aim is to analyze these formulations in order to check whether the lack of consideration of transfer times which leads to incorrect task synchronization. The main features of MILP models based on the STN and RTN global time points and unit specific time events are briefly described next. The general precedence model is completely developed in the next section because it is taken as a basis for developing an extended formulation capable of avoiding infeasible schedules due to the omission of transfer times. Further details related to the other alternative existing MILP optimization models can be found in Méndez *et al.* (2006).

5.5.1 State-Task-Network based continuous formulation

There have been many efforts to develop a continuous-time formulation (Giannelos & Georgiadis, 2002; Maravelias & Grossmann, 2003b) based on the original STN representation proposed by Kondili *et al.* (1993). For testing purposes, we have selected the work by Maravelias and Grossmann (2003b) 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 includes 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 is to increase the number of time intervals from a relative small number until no improvement in the objective function is achieved. 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.

5.5.2 Resource-Task-Network based continuous formulation

The RTN-representation was firstly introduced by Pantelides (1994). Further improvements were presented by Castro *et al.* (2001) and Castro *et al.* (2004). The improved model version developed by Castro *et al.* (2004) was selected for this case. 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 render 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 ending of a given task. The RTN representation considers two types of items: resources and tasks. 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. RTN continuous models for multipurpose plants reported in the literature do not consider transfer times in their formulation. Therefore, in cases where neglecting transfer times influences the synchronization of tasks, unfeasible optimal results may appear, as will be shown in the examples.

5.5.3 Unit-Specific Time Event

The original idea of unit-specific events was firstly presented by Ierapetritou and Floudas (1998) 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. It should be noticed that this iterative method provides good solutions, but it is not guaranteed to converge to the global optimum of the problem. 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 employed for testing purposes. As it can be seen, this formulation also neglects transfer times and so, the synchronization of tasks. Hence, the optimal solutions obtained with this model may be unfeasible, as it will be discussed in the examples addressed.

5.6 Solution approach

The aforementioned kind of unfeasible schedules was firstly identified under NIS conditions by Sanmartí *et al.* (2002) using a S-graph representation for the scheduling problem. The S-graph uses a graph representation and in contrast to MILP-based methods embeds modeling aspects into the solution algorithm. A graph is used to represent recipes, where the nodes represent the production tasks and the arcs the precedence relationships among them. Then, a branch-and-bound procedure eventually generates the optimal schedule. In the bounding procedure the unfeasible solutions can be pre-detected beforehand. They appear as directed cycles in the graph which are identified and excluded using the algorithm described by Cormen *et al.* (1997). Later on, Romero *et al.* (2004) extended the use of the S-graph framework to include the common intermediate storage policy and then applied a similar algorithm to cycle detection and thus discarding unfeasible solutions. And finally, Ferrer-Nadal *et al.* (2006) carried out a comparative study between the S-graph and a MILP formulation highlighting advantages and inconveniences of both representations. Although S-graph has proved to be a very efficient framework for solving the scheduling problem, it must be still extended in order to include some relevant modeling aspects in complex plant configurations including the handling of specific resource constraints.

The solution approach consists of considering the transfer tasks in the modeling step of the scheduling problem. In this section, a formulation based on the general precedence concept is defined, and a two stage algorithm for accounting for negligible transfer times is proposed. As a result, the synchronization is embedded in the formulation and the resulting solution will be inherently feasible.

5.6.1 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 (Mendez *et al.*, 2001; Mendez & Cerda, 2004) 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 5.3, this model defines a couple of sets of binary variables in order to sequence ($X_{pisp'i's'}$) and allocate (Y_{pisu}) processing tasks.

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

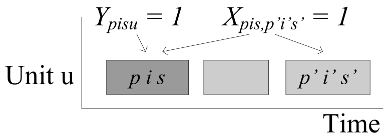


Figure 5.3: General precedence representation.

the sequencing decisions, $X_{pis,p'i's'}$ is a binary variable which establishes the general precedence relationship between a pair of tasks pis and $p'i's'$ executed at the same processing unit (otherwise $X_{pis,p'i's'}$ is meaningless). If $X_{pis,p'i's'}$ is equal to 1, task pis 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 pis in the same unit,$X_{pis,p'i's'}$ takes the value zero. It is worth noting that the six sub index defined for sequencing variables are needed to deal with the general scheduling problem arising in multipurpose batch plants, where 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 pis 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 sub index related to the stages in the sequencing variables are not longer required.

Allocation constraint. Unit allocation constraint 5.1 states that a single processing unit must be assigned to every required processing task.

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

Sequencing constraints. Constraints 5.2 and 5.3 sequence two batches of two different products processed in the same unit. Constraint 5.2 is active if task pis precedes task $p'i's'$ while constraint 5.3 is active in the opposite case. In order to reduce the number of binary variables $X_{pis,p'i's'}$ constraints 5.2 and 5.3 are only used when product p appears before p' $(p < p')$ or if $p = p'$ for $s < s'$.

$$\begin{gathered} Ts_{p'i's'} \geq Tf_{pis} - M\left(1 - X_{pisp'i's'}\right) - M\left(2 - Y_{pisu} - Y_{p'i's'u}\right) \\ \forall (p,i,s),(p',i',s'), u \in (U_{ps} \cap U_{p's'}) : (p < p') or (p = p',\, and,\, s < s') \end{gathered} \tag{5.2}$$

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

Timing constraints. Constraint 5.4 expresses starting (Ts_{pis}) and completion time of a task (Tf_{pis}), from the overall time required to perform the loading of material or transfer time from the previous stage $(tt_{pu'})$, the batch processing operation itself (pt_{ps}), a possible waiting time in the processing unit (Tw_{pis}) and 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 5.4.

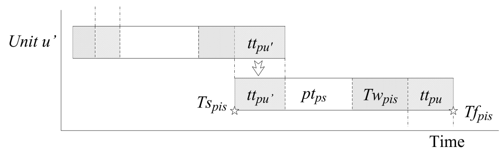


Figure 5.4: Task duration.

$$Tf_{pis} = Ts_{pis} + \sum_{u' \in U_{p(s-1)}} tt_{pu'} + \sum_{u \in U_{ps}} (pt_{psu} + tt_{pu})Y_{pisu} + Tw_{pis} \qquad \forall p, i, s \quad (5.4)$$

Constraint 5.5 sequences two batches i and i' of the same product p at the same stage s. The number of these constraints is significantly reduced by considering that batch i is processed before batch i' ($i < i'$).

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

The task precedence constraint 5.6 is defined for every pair of consecutive tasks that must be sequentially performed for a particular product. One 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.

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

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 Mendez and Cerda (2003) must be further generalized here by allowing selective interconnection between processing units and storage tanks facilities. For this purpose, a new binary variable AT_{pist} is defined denoting whether task pis is sent to storage tank t. Then, constraint 5.7 expresses that material from task pis may be stored in a tank t only if processing unit u is connected to storage tank t.

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

Constraint 5.8 works together with 5.6 in order to sequence two stages of a batch. In case intermediate storage is not used, both constraints give rise to a single equality constraint. Otherwise, Constraint 5.8 is relaxed.

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

Storage task sequencing constraints 5.9 and 5.10 (Figure 5.5) define the order of storage tasks pis and $p'i's'$ assigned to the same tank. Constraint 5.9 is only active when task $p'i's'$ precedes task pis while Constraint 5.10 is active in the opposite case.

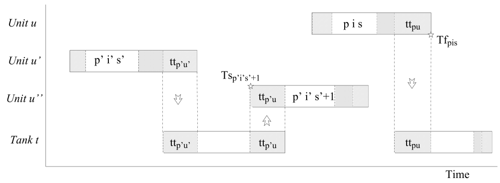


Figure 5.5: Illustrative representation of storage constraint 5.10.

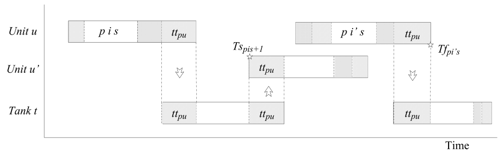


Figure 5.6: Illustrative representation of storage constraint 5.11.

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

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

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 5.6).

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

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

$$z^{Mk} \geq Tf_{pis} \quad \forall p,i,s \tag{5.12}$$

The general precedence model allows including transfer times in tasks. If transfer times are assumed negligible, this model may fail to observe the synchronization of tasks, thus leading to unfeasible schedules. This will be proved in the testing examples.

5.6.2 Two-stage algorithm for negligible transfer times

The second alternative consists of a two-stage algorithm (Figure 5.7) 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 the problem is again solved specifying zero transfer times in step two. In this second step, a LP problem, in which units can be synchronized by

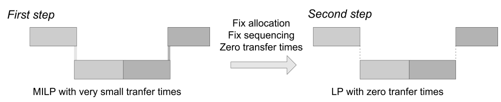


Figure 5.7: Two-stage algorithm for considering negligible transfer times.

performing left or right shifting, is solved. Furthermore, the computational effort of this stage is almost negligible because all the binary variables are already fixed. This makes this strategy very suitable for large problems.

5.7 Examples

Two examples are 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 in its formulation, 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. The two stage solution approach is applied and the new results are compared to the unfeasible solutions generated with the previous models. The mathematical models have been implemented in GAMS and solved using the MILP solver CPLEX 9.0.

5.7.1 Example 1

The scheduling problem presented in this example was originally proposed by Kim *et al.* (2000), and later solved by Mendez and Cerda (2003). A multipurpose batch plant comprises four products which have to sequentially undergo several processing stages (Figure 5.8). One single batch of each product is assumed to be manufactured. In the original definition of the problem transfer times were neglected as they were much smaller than the processing times. The production sequences and processing times for the recipe of each product are stated in Table C.1. 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 according to 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). This problem stands for an illustrative example of multipurpose batch plant and will be further examined in Chapter 7.

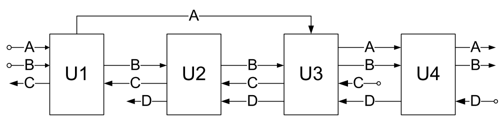


Figure 5.8: Multi-purpose batch plant structure of Example 1.

Figure 5.9 presents the comparison between the results obtained after applying the four MILP formulations and the results using the two stage approach. MILP formulations yield solutions with lower makespan values because, as expected, unfeasible transfer operations within these solutions are encountered.

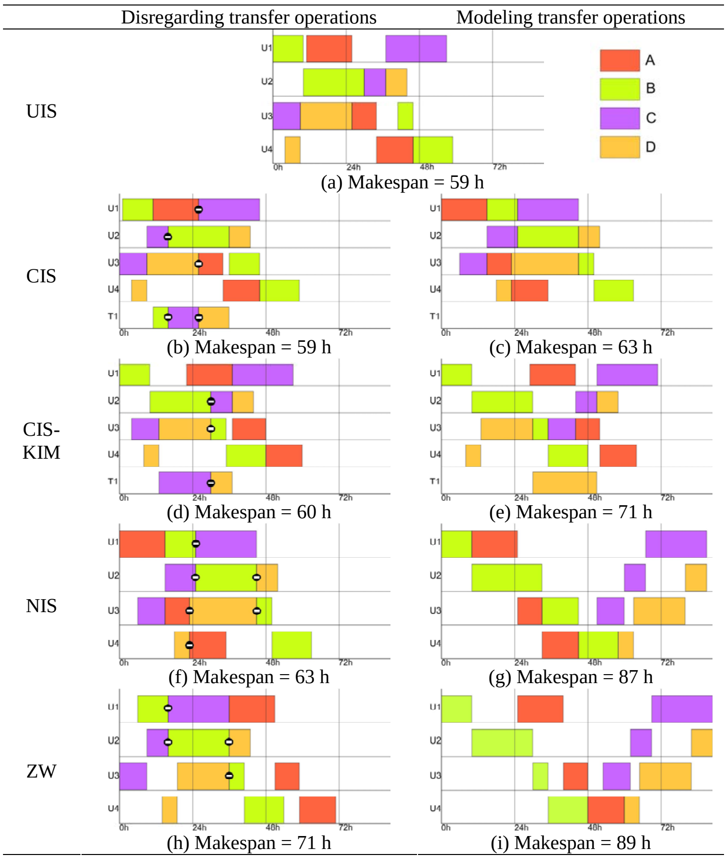


Figure 5.9: Gantt charts for the optimal schedules disregarding and considering batch transfer operations for different storage policies (Example 1).

Only in the case of UIS policy, MILP formulations lead to an optimum feasible schedule (Figure 5.9(a)).

For the case in which one tank can be shared by all units (CIS), two unfeasible sequences appear, as shown in the Gantt chart in Figure 5.9(b). 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 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 5.9(c) shows the feasible solution obtained.

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 5.9(d)). The difference between the values of the makespan of this unfeasible solution (60 hours) and the feasible one shown in Figure 5.9(e) (71 hours) is almost of 20%.

A similar situation corresponds to the NIS policy, where unfeasible sequences take place at times 23 h, 25 h and 45 h (Figure 5.9(f)). For this example, 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 5.9(h)).

5.7.2 Example 2

A multipurpose production plant proposed by Papageorgaki and Reklaitis (1990) is revisited in this example. The scheduling problem comprises the production of five products which have to sequentially undergo different processing stages with alternative units (Figure 5.10). Two batches of each product are assumed to be manufactured. The production sequences and processing times for the five products are stated in Table C.2. The alternative storage policies contemplated are unlimited intermediate storage (UIS), non intermediate storage (NIS), common intermediate storage tank (CIS) and zero-wait time (ZW).

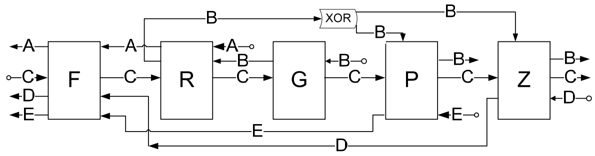


Figure 5.10: Multi-purpose batch plant structure of Example 2.

Figure 5.11 presents the results obtained applying the four MILP formulations and those using the strategy described in this chapter. As in the previous example, MILP formulations lead to solutions with lower makespan values because unfeasible transfer operations within these solutions are encountered. In the case of UIS policy, MILP formulations lead to an optimum feasible schedule (Figure 5.11(a)). If one tank can be shared by all units (CIS), two unfeasible sequences appear in the Gantt chart of the solution shown in Figure 5.11(b). The first unfeasible transfer takes place at time 11.6 h with two products involved, the first batch of product C and the second batch of B, which are simultaneously transferred from unit R2 to unit G and vice versa. At time 12.1 h, a second unfeasible transfer occurs with the first batch of A and the second batch of C, which are simultaneously transferred from unit R1 to unit F. Figure 5.11(c) shows the feasible solution obtained by applying the proposed strategy. If a NIS policy is considered, one unfeasible sequence takes place at time 11.6 h (Figure 5.11(d)). Finally, three unfeasible sequences arise when adopting a ZW policy. The first batches of products C and B are transferred simultaneously between units R1 and G at time 7.8 hours, the second batches of the same products are transferred between units R2 and G at time 16.4 hours, and the first batch of product A and the second batch of product C are simultaneously transferred from unit R2 to F at time 8.8 h (Figure 5.11(f)).

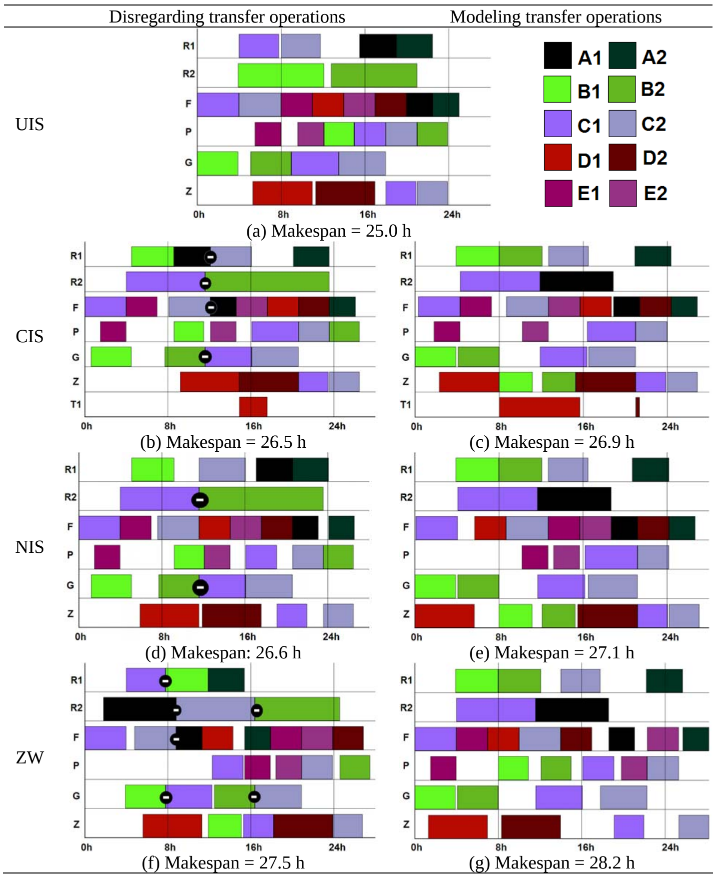


Figure 5.11: Gantt charts for the optimal schedules disregarding and considering batch transfer operations for different storage policies (Example 2).

5.8 Final remarks

A wide variety of MILP-based optimization methods for the short-term scheduling of batch plants have been developed in the last years. Although they have showed a good computational performance in a wide variety of scheduling problems, most of them have only focused the attention on the modeling aspects of processing and changeover tasks, ignoring the important role of material transfer operations. Although transfer times may represent a very small percentage of time regarding the whole duration of processing tasks in the batch units, loading and unloading operations may play a crucial role in the synchronization of material transfer.

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 actual transfer times and their corresponding effect on the task synchronization, optimal but actually unfeasible solutions in practice may be reached, particularly in those cases involving shared units and storage tanks, material recycles or bidirectional flows of products.

The flaw stems from disregarding at the modeling stage the transfer among tasks. Hence, in order to avoid this situation, a continuous time MILP framework based on the general precedence notion that can explicitly consider non-zero transfer times has been introduced. Despite the direct representation of transfer activities, neither new variables nor additional constraints are required and consequently the computational effort remains almost the same. Therefore, in order to deal with negligible transfer times, a two-stage algorithm is proposed. First, a small value is assigned to the transfer times in order to force a proper synchronization, and in step two the problem is solved fixing the previous schedule, but assigning zero transfer-times, and assessing the new starting and finish times of tasks.

On the whole, it is shown that standard scheduling models can produce unfeasible solutions in the two previous examples where different storage policies are considered. These unfeasible solution schedules have been compared to the feasible ones obtained by using the proposed approach, highlighting the effect of transfer operation restrictions on scheduling decisions. To avoid this situation, specific restrictions and resources related to transfer tasks should be systematically incorporated in any optimization scheduling approach.

5.9 Nomenclature

Sets and subsets

i, i'	Batches.
p, p'	Products.
s, s'	Stages.
t, t'	Storage tanks.
u, u', u''	Equipment units.
pis	Batch operation.
U_{ps}	Available units for processing product p at stage s.
T_u	Set of storage tanks available after unit u.

Parameters

pt_{psu}	Processing time of stage s of product p in unit u.
tt_{pu}	Transfer time of product p from unit u.
M	A very large number.

Continuous variables

Tf_{pis}	Completion time of task pis.
Ts_{pis}	Starting time of task pis.
Tw_{pis}	Waiting time of task pis.
z^{Mk}	Makespan.

Binary variables

$X_{pis,p'i's'}$	Sequencing of tasks pis and $p'i's'$, if the former is processed before the latter, it is equal to 1, and 0 otherwise.
Y_{pisu}	Assignment of task pis to equipment unit u.
AT_{pist}	Assignment of material from task pis to transfer to storage tank t.

Part III

Process Models Integration in Scheduling

Chapter 6

Variable Production Rates in Semi-continuous Plants

6.1 Motivation

Chemical plants are moving toward more flexible environments in order to adapt faster to market changes. To achieve this flexibility, batch and continuous units must be integrated and operated along the same processing route in a semi-continuous mode, featured by production campaigns of finite duration. Up to now, at the planning level, constant production rates for the whole operation have been usually considered, so production rate adjustments were achieved through storage resources, production line stops and multiple product campaigns.

The aim of this chapter is to improve the production schedules by developing a new concept for flexible and integrated manufacturing which allows to program production rate profiles within each semi-continuous operation campaign according to process units and storage availability. Therefore, an initial attempt to including process variables into the scheduling formulation is presented for semi-continuous batch plants.

6.2 Introduction

A first approach to the introduction of process dynamics in the scheduling model consists of considering linear process variables within the scheduling model. Such parameters increase the process flexibility, which is an important parameter in ensuring that market demands are met effectively. Flexible batch plants provide an adaptable solution for highly dynamic and uncertain environments and have grown in popularity at the expense of mass production, a more rigid continuous production mode. Precisely, in continuous processes, a very limited number of products are produced at constant rates over long production periods. In contrast, batch plants produce much smaller quantities of a wider range of products during shorter production periods. Therefore, the flow of material in batch production is discontinuous: input products are loaded at the beginning of the production period and output products are only available after the entire operation has been completed.

Semi-continuous operation mode can be regarded as an intermediate mode between the batch processes and continuous processes. This type of process improves the efficiency with which equipment is used to process medium quantities of several products simultaneously in a continuous facility. Semi-continuous operation is characterized by an overall processing rate, in which equipment runs continuously with periodic start-ups and shutdowns for product transitions. Processing times in semi-continuous operation are usually relatively long periods called campaigns, in which a single product is produced. Individual campaigns are often used to produce feed stocks for downstream processes that produce more specialized final products (Papageorgiou & Pantelides, 1996). In fact, most process plants in the chemical industry work in semi-continuous mode by combining continuous operations and batch processes. Typical campaign lengths range from a few hours to several days. However, too many short campaigns can produce unrealistic and non-cost-effective operating conditions due to the expense of switching production from one product to another (Mendez & Cerda, 2002) .

Intermediate product storage is again important factor in the operational management of semi-continuous chemical plants since it allows to decouple different upstream and downstream production rates. Hence, a satisfactory storage policy has a strong influence on the efficiency and flexibility of plants working in semi-continuous mode. This operation mode is complex but has a large number of potential applications.

Sahinidis and Grossmann (1991) addressed the problem of cyclic multiproduct scheduling for continuous parallel production lines. They identified a combinatorial part (the assignment of products to lines and their sequencing in each line) and a continuous part (the duration of production runs and the frequency of production), and formulated a slot-based MINLP model which was linearized in the space of the integer variables. Pinto and Grossmann (1994) extended this work and modeled the cyclic scheduling problem in multi-stage continuous processing plants. They developed a solution based on a generalized Benders decomposition and an outer approximation which used explicit inventory breakpoints to handle the inventory profiles of intermediate storage tanks. Zhang and Sargent (1996) developed a MINLP formulation based on the RTN representation. The resulting model was linearized to create a very large-scale MILP model. Ierapetritou and Floudas (1998) used a STN representation to formulate the problem taking into account multiple intermediate due dates, while storage requirements were handled using approximated storage task timings. Mockus and Reklaitis (1999a) proposed a global event-based MINLP able to handle resource constraints such as limited availability of utilities and manpower. Giannelos and Georgiadis (2002) developed a similar model to that of Ierapetritou and Floudas (1998) but relaxed time durations and eliminated big-M constraints. However, they also assumed equal start and end times of the tasks producing/consuming the same state, which could lead to suboptimal solutions if the material is allowed to bypass the storage. Mendez and Cerda (2002) used a continuous-time formulation based on a general precedence notion, which generated a very small and compact model. An important assumption they made is that every intermediate or final product should be produced by a single production campaign. Castro *et al.* (2004) developed a MILP formulation based on the RTN representation and highlighted the benefits of using a uniform time grid continuous-time representation. Shaik and Floudas (2007) extended the work of Ierapetritou and Floudas (1998) to handle different storage requirements.

All of the studies cited in the literature examined how to use semi-continuous processes to increase production flexibility. Consequently, in most of them, multiple campaigns can clearly be considered suitable for increasing flexibility despite the fact that they may often be infeasible in industrial practice. However, other rigid features can be found in most of the formulations presented in these studies. Precisely, one of the most significant variable from a flexibility point of view are constant and invariable production rates, which are usually implicitly assumed for the whole operating period of a given campaign. Furthermore, no transitions in the production rate are made within each campaign so, production adjustments are made by using production line stops, storage resources and multiple product campaigns.

This chapter aims at improving production schedules by developing a new formulation allowing flexible fabrication opportunities by means of the consideration of adjustable production rates within each production campaign. Next, a mathematical formulation based on this concept is introduced and the flexible manufacturing concept for semi-continuous processes is presented. Moreover, two examples illustrate the formulation and results are compared with those reported in the literature. Finally, the potential applications of the extended formulation, which uses adjustable processing rates to increase the scheduling flexibility, are discussed in the final remarks.

6.3 Mathematical formulation

The proposed MILP formulation is designed in order to determine the optimal sequence of campaigns that maximizes production, by satisfying a minimum demand for each product. It is based on the model introduced by Mendez and Cerda (2002), which makes one major assumption by considering single campaigns for the production of each product. Although this assumption may produce sub-optimal solutions, the previous authors argued that a large number of campaigns creates a much higher demand for manpower and generates financial losses through excessive equipment idle time. For this reason, such assumption is also considered in this work.

This section describes the MILP formulation in detail. The indexes, parameters, and variables included in the model are defined in the nomenclature section.

Constraint 6.1 ensures that only one production line j can be assigned to each processing campaign i.

$$\sum_{j \in J_s} Y_{ij} \leq 1 \qquad \forall i \in I_s^+, s \in S \tag{6.1}$$

Constraint 6.2 ensures that each campaign finishes before a pre-specified time horizon (H), while equation 6.3 restricts the duration of each campaign (L_{ij}) to a minimum value l_{sj}^{min}.

$$C_i \leq H \qquad \forall i \in I \tag{6.2}$$

$$l_{sj}^{\min} Y_{ij} \leq L_{ij} \leq H Y_{ij} \qquad \forall i \in I_s^+, j \in J_s, s \in S \tag{6.3}$$

Constraint 6.4 ensures that the product release and unit set-up processes cannot begin before the time values (ro_i) and (ru_j), respectively.

$$C_i - \sum_{j \in J_s} L_{ij} \geq \sum_{j \in J_s} \max\left[ru_j, ro_s\right] Y_{ij} \qquad \forall i \in I_s^+, s \in S \tag{6.4}$$

Constraint 6.5 imposes minimum and maximum limits on the total production of a campaign (Q_i) according to its duration and the maximum production rate of the corresponding equipment unit. In addition, at the end of the time horizon, the amount of every final product from each campaign must satisfy the minimum demand (d_s) specified by constraint 6.6.

$$\sum_{j \in J_s} r_{sj}^{\min} L_{ij} \leq Q_i \leq \sum_{j \in J_s} r_{sj}^{\max} L_{ij} \qquad \forall i \in I_s^+, s \in S \tag{6.5}$$

$$d_s \leq \sum_{i \in I_s^+} Q_i \qquad \forall s \in S^P \tag{6.6}$$

Constraints 6.7 and 6.8 sequence a pair of campaigns i and i' assigned to the same semi-continuous line. Constraint 6.7 is only active if campaign i precedes i', and is inactive otherwise. Since this model also takes into account a possible changeover time between different products ($uch_{ii'j}$), the value of M_1 should be equal to $H + max\{uch_{ii'j}\}$, to obtain the tightest relaxation.

$$\begin{gathered} C_{i'} - L_{i'j} \geq C_i + uch_{ii'j} - M_1\left(1 - X_{ii'}\right) - M_1\left(2 - Y_{ij} - Y_{i'j}\right) \\ \forall i, i' \in I, i < i', j \in J_i \cap J_{i'} \end{gathered} \tag{6.7}$$

$$\begin{aligned} & C_i - L_{ij} \geq C_{i'} + uch_{i'ij} - M_1 \cdot X_{ii'} - M_1\left(2 - Y_{ij} - Y_{i'j}\right) \\ & \forall i, i' \in I, i < i', j \in J_i \cap J_{i'} \end{aligned} \tag{6.8}$$

Equation 6.9 establishes the mass balances between a campaign i producing and a campaign i' consuming the intermediate state s. In this equation, $F_{ii'}$ is a continuous variable which represents the amount of material transferred between the two campaigns.

$$Q_i = \sum_{i' \in I_s^-} F_{ii'} \qquad \forall i \in I_s^+, s \in S^I \tag{6.9}$$

Similarly, equation 6.10 adjusts the amount of material consumed by a campaign i which receives material from i'. The amount of material consumed by this campaign is used to determine the campaign production by using the coefficient ρ_{is}.

$$\rho_{is} Q_i = \sum_{i' \in I_s^+} F_{i'i} \qquad \forall i \in I_s^-, s \in S \tag{6.10}$$

Equation 6.11 defines a binary variable ($U_{ii'}$) that equals one if $F_{ii'}$ is greater than zero, i.e. campaign i supplies material to campaign i'. In this case, M_2 must be greater than any value of $F_{ii'}$. The variable $U_{ii'}$ is used to ensure that campaign i which supplies the material can never start later than the campaign i' during which the material is received (constraint 6.12).

$$F_{ii'} \leq M_2 \cdot U_{ii'} \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{6.11}$$

$$C_i - \sum_{j \in J_i} L_{ij} \leq C_{i'} - \sum_{j \in J_{i'}} L_{i'j} + H\left(1 - U_{ii'}\right) \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{6.12}$$

Constraints 6.1 to 6.12 represent an unlimited intermediate storage (UIS) policy, which is the least restrictive storage policy, in which sufficient storage resources are assumed to be available at any time. In contrast, a non-intermediate storage (NIS) policy is a scenario in which no storage is available and materials have to be transferred directly between production lines. Constraint 6.13 accounts for the NIS case by ensuring that a campaign i cannot finish until any consuming campaign i' has finished.

$$C_i \geq C_{i'} - H\left(1 - U_{ii'}\right) \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{6.13}$$

In addition to UIS and NIS, this model can account for different finite intermediate storage (FIS) policies. In these cases, material can either be stored in a limited number of tanks with restricted capacity, or bypassed directly between production lines. Therefore, in addition to constraints 6.1 to 6.12 for the UIS case, the following constraints are applied to the cases in which there are limited storage resources.

Constraint 6.14 ensures that the beginning of the storage period for intermediate material supplied by campaign i (IT_i) coincides with the beginning of that campaign.

$$IT_i \geq C_i - \sum_{j \in J_s} L_{ij} \qquad \forall i \in I_s^+, s \in S^I \tag{6.14}$$

Constraint 6.15 states that the storage of an intermediate material must not end (CT_i) before all of the production campaigns consuming this material have been completed.

$$CT_i \geq C_{i'} - H\left(1 - U_{ii'}\right) \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{6.15}$$

The model includes sequencing constraints for the storage tasks, equations 6.16 and 6.17, that are similar to those used for production campaigns. In this case, an additional binary variable, W_{it}, indicates whether the intermediate material produced by a campaign i is transferred to tank t. The parameter $tch_{ii't}$ stands for the changeover time between different products in the same tank; thus, M_3 should be equal to $H + max\{tch_{ii't}\}$ to achieve the tightest relaxation of the model.

$$\begin{gathered} IT_{i'} \geq CT_i + tch_{ii't} - M_3\left(1 - X_{ii'}\right) - M_3\left(2 - W_{it} - W_{i't}\right) \\ \forall i, i' \in I, i < i', t \in T_i \cap T_{i'} \end{gathered} \tag{6.16}$$

$$\begin{gathered} IT_i \geq CT_{i'} + tch_{i'it} - M_3 \cdot X_{ii'} - M_3\left(2 - W_{it} - W_{i't}\right) \\ \forall i, i' \in I, i < i', t \in T_i \cap T_{i'} \end{gathered} \tag{6.17}$$

Constraint 6.18 assigns a value of 1 to the binary variable $Z_{ii'}$ if a campaign i producing an intermediate consumed by campaign i' finishes before i' starts.

$$C_{i'} - \sum_{j \in J_{i'}} L_{i'j} - C_i \leq H Z_{ii'} \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{6.18}$$

Constraint 6.19 introduces a continuous variable ($V_{ii'}$) which represents the amount of intermediate material consumed by i' at the end of the campaign i in which the material is produced. Therefore, if $Z_{ii'} = 1$, that is, i and i' do not coincide in time, then $V_{ii'}$ must be equal to zero.

$$V_{ii'} \leq M_2 \left(1 - Z_{ii'}\right) \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{6.19}$$

Equations 6.20 and 6.21 constrain the value of $V_{ii'}$: equation 6.20 forces $V_{ii'}$ to be at most as large as $F_{ii'}$, and constraint 6.21 establishes an upper bound for the value of $V_{ii'}$ assuming that campaign i' consumes material from i at its maximum rate capacity during the period in which i and i' run simultaneously.

$$V_{ii'} \leq F_{ii'} \qquad \forall i \in I_s^+, i' \in I_s^-, s \in S \tag{6.20}$$

$$\begin{aligned} V_{ii'} \leq \rho_{i's} \cdot \min\left(r_{ij}^{\max}, r_{i'j'}^{\max}\right) \cdot \left(C_i - C_{i'} + L_{i'j}\right) + M_2 Z_{ii'} + M_2\left(1 - U_{ii'}\right) \\ +M_2\left(1 - Y_{ij}\right) + M_2\left(1 - Y_{i'j'}\right) \\ \forall i \in I_s^+, \forall i' \in I_s^-, s \in S, j \in J_i, j' \in J_{i'} \end{aligned} \tag{6.21}$$

Finally, constraint 6.22 restricts the use of a storage tank to its maximum volumetric capacity (v_t).

$$Q_i - \sum_{i' \in I_s^-} V_{ii'} \leq \sum_{t \in T_s} v_t W_{it} \qquad \forall i \in I_s^+, s \in S \tag{6.22}$$

The objective function of this problem is to maximize the production of final products:

$$\max \sum_{i \in I_s^+, s \in S^P} Q_i \tag{6.23}$$

subject to:

- Constraints 6.1 - 6.12, for the UIS case.
- Constraints 6.1 - 6.13, for the NIS case.
- Constraints 6.1 - 6.12 and 6.14 - 6.22, for the FIS case.

Additionally, mixed storage policies, which combine the different storage policies for the different products could be considered.

6.4 Examples

6.4.1 Example 1

A semi-continuous plant that produces six final products (P1-P6) from three intermediate products (I1-I3), as shown in Figure 6.1, is considered. It is based on a simplification of the academic case study used as Example 2, in order to better show the main advantages of the proposed formulation. There are two parallel available units to produce the intermediate products in step one and two additional units to produce the final products in step two (Figure 6.2). The maximum processing rates of the units are given in Table 6.1. The problem objective seeks to maximize the total production while satisfying a minimum demand for each product (Table 6.2).

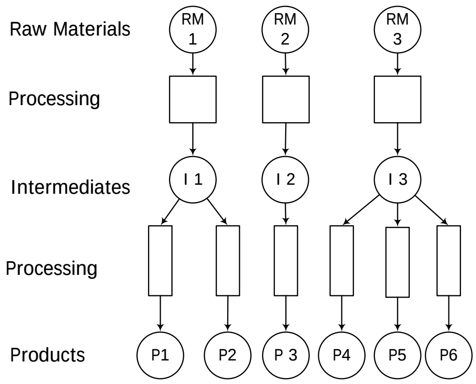


Figure 6.1: STN representation of the process in Example 1.

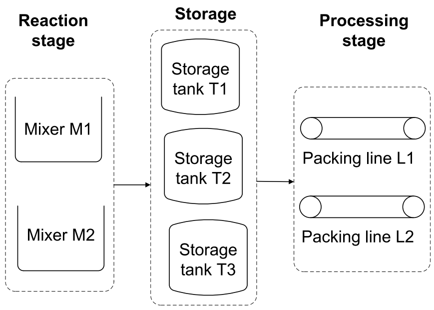


Figure 6.2: Schematic representation of the plant in Example 1.

Table 6.1: Maximum rate capacities and unit suitability for the Example 1.

Product	Available units	r_{max}, [ton/h]
I_1, I_2, I_3	M1	14.0
I_1, I_2, I_3	M2	16.0
P_1	L1	5.0
P_2	L2	7.0
P_3	L1	8.0
P_4	L1	8.0
P_5	L2	4.0
P_6	L1	4.0

Table 6.2: Minimum demand requirements for Example 1.

Product	Demand [ton]
P_1	60.0
P_2	300.0
P_3	100.0
P_4	300.0
P_5	80.0
P_6	10.0

Different storage policies for intermediate products may be considered. If unlimited intermediate storage is available in the plant, then a total optimal production of 1694 u. is achieved (Figure 6.3). However, unlimited storage capacity is unreal since there are usually storage limitations in process plants. Hence, let us suppose that there are three intermediate storage tanks of 60 ton capacity available, and the processing rates are set constant and equal to the maximum for every batch. If a single batch per product is allowed, then the maximum attainable production is 1482.5 u., and the three storage units would be required, as shown in Figure 6.4. Alternatively, if multiple batches were carried out for producing intermediate products, then the maximum production would be reached. However, it is clear that multiple batches and storage resources entail additional costs that are not considered in the model.

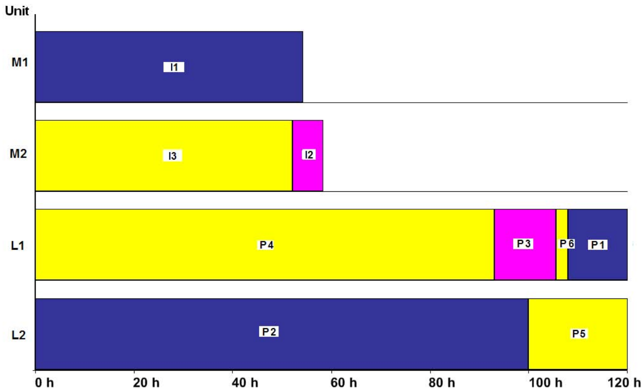


Figure 6.3: Optimal schedule for the UIS policy for Example 1.

However, in this case, if the variation of the unit processing rate along a batch was allowed, then one batch per intermediate product would also reach the value of the maximum production obtained with unlimited storage policy, 1694 u. (Figure 6.5).

To summarize, the fact of considering variable processing rates within batch allows obtaining schedules requiring lower number of batches as well as lower consumption of storage resources.

6.4.2 Example 2

This example illustrates the capabilities of the flexible formulation introduced. It refers to a fast-moving consumer goods manufacturing plant. This is a classic case that has been addressed many times in the literature on scheduling strategies for semi-continuous facilities (Zhang & Sargent, 1996; Ierapetritou & Floudas, 1998; Mendez & Cerda, 2002; Giannelos & Georgiadis, 2002; Castro *et al.*, 2004; Shaik & Floudas, 2007) . The plant has the structure shown in Figure 6.6. It consists of three parallel mixers that send material to five packing lines operating in semi-continuous mode, and a set of three storage tanks that buffer the production stocks. Three raw materials with non-restricted availability are blended in the corresponding mixers to produce seven intermediates (I1-I7). These intermediates are then combined in a series of packing lines to produce fifteen final products (P1-P15). Figure 6.7 shows the STN representation for the plant. Table 6.3 shows the maximum production rate for each product and the

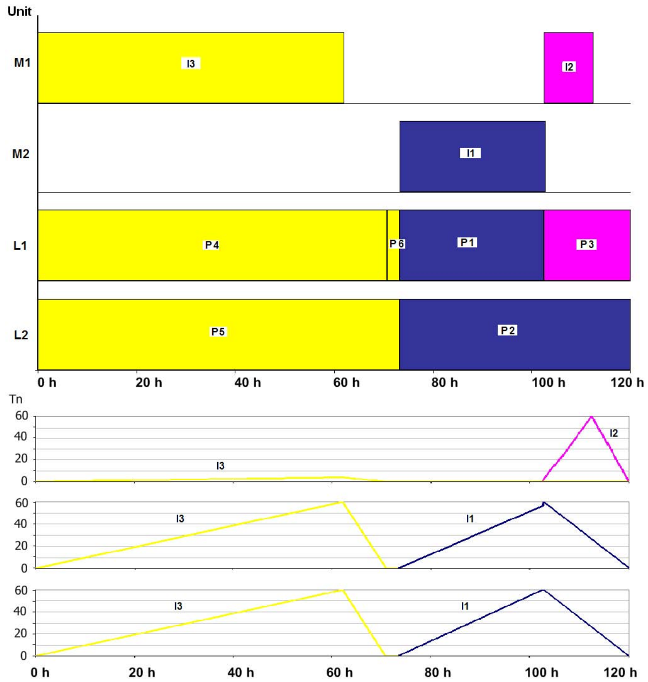


Figure 6.4: Optimal schedule for the FIS policy with three tanks and fixed processing rates for Example 1.

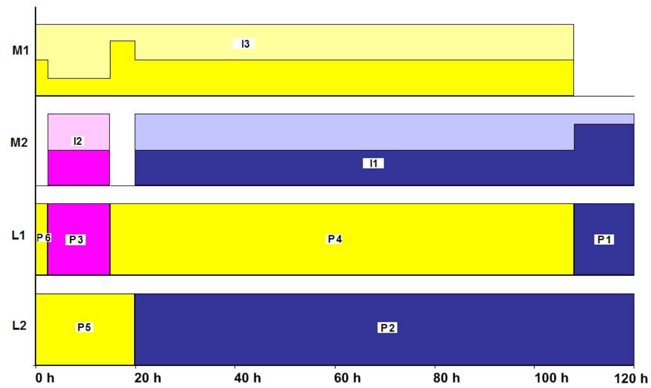


Figure 6.5: Optimal schedule for the NIS policy with variable processing rates for Example 1 (the height of the boxes represent the percentage of utilization of r_{sj}^{max}).

availability of the semi-continuous units in which they are processed. Table 6.4 gives the sequence-dependent changeover times between different products.

The objective function maximizes production to ensure that a minimum final product demand (Table 6.5) is met over a time horizon of five working days (120 hours). Several storage policies are considered, namely unlimited intermediate storage (UIS), no intermediate storage (NIS) and finite intermediate storage (FIS), under

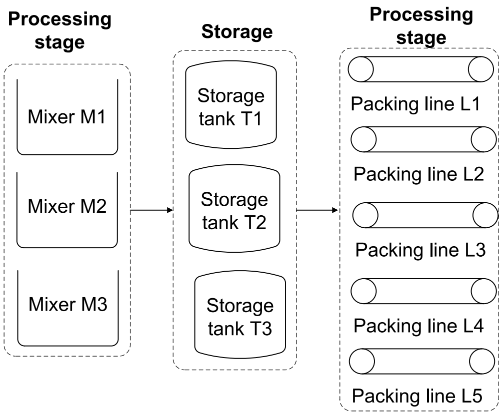


Figure 6.6: Schematic representation of the plant in Example 2.

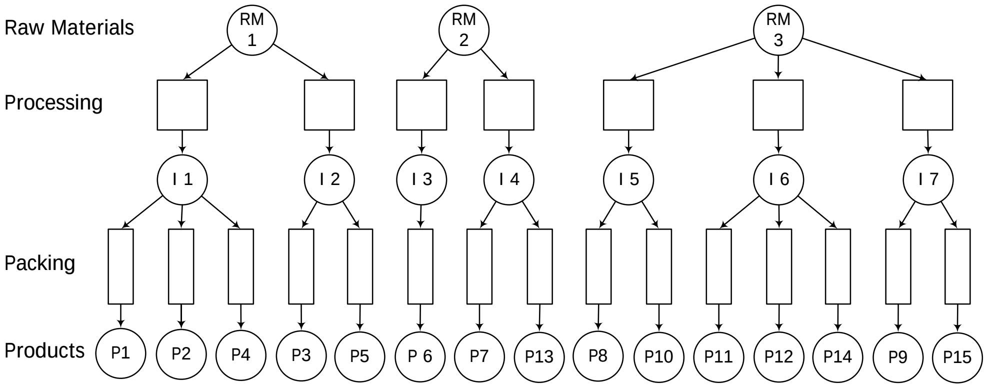


Figure 6.7: STN representation of the plant for Example 2.

Table 6.3: Maximum rate capacities and unit suitability for Example 2.

Product	Available units	r_{max}, [ton/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

different allocation and demand constraints. These cases are implemented in GAMS and, solved using CPLEX 10.0 in a 3 GHz computer and compared with the results reported in the literature.

Table 6.4: Changeover requirements for Example 2 [h].

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 6.5: Minimum demand requirements for Example 2.

Product	Demand [ton]
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

Unlimited intermediate storage. The optimal schedule obtained under an UIS policy is shown in Figure 6.8. This solution represents the highest attainable profit, given the demand requirements and unlimited storage capacity for the intermediate products. Since there are no restrictions on the storage capacity, all mixers in the processing stage can work at their maximum rate, and the idle times are reduced. However, in the packing stage, units are working during the whole time horizon because the processing rates of the packing lines are much lower than those of the mixers. Therefore, whereas mixers are partially idle in this example, production is limited by the processing rates of the packing units.

Table 6.6 shows the results for the formulation proposed in this case study and those given in literature. All formulations yield the same optimal profit value. Intermediates are produced in more than one campaign in the solutions reported by Castro *et al.* (2004) and Shaik and Floudas (2007) . In contrast, the solution presented here and that of Mendez and Cerda (2002) use a single campaign for each intermediate and final product, reflects a more realistic industrial practice. Multiple campaigns lead to higher operational costs because of the higher demand for resources such as manpower that may be required to carry out changes in production lines. Consequently, the cost-efficiency of the plant increases as the number of campaigns decreases.

There is no constraint on the storage in the proposed solution (Figure 6.8), so most intermediate products are stored. Figure 6.9 shows the stock profiles for this solution

Table 6.6: Results for the UIS policy for Example 2.

Model	Profit [m.u.]	Campaigns/ product	Bin., cont., rows	Time [s CPU]
Shaik and Floudas (2007)	2695.32	> 1	108 356 1040	1.03
Castro *et al.* (2004)	2695.32	> 1	236 762 894	58.5
Mendez and Cerda (2002)	2695.32	1	38 44 140	4.77
This approach	2695.32	1	38 85 140	2.84

that requires as many storage units as intermediates. As it can be seen, the maximum tank capacity needed is more than 350 ton, which is a highly inefficient solution that would generate high operational and fixed costs. Therefore, this scenario is unrealistic, because industries usually have storage space limitations that need to be taken into account by the production scheduler. However, this case is particularly relevant to the example because it represents an upper bound for more restrictive cases. In addition, the UIS policy represents the best solution in terms of productivity if the cost of storage is disregarded. More realistic approaches are presented in the following paragraphs.

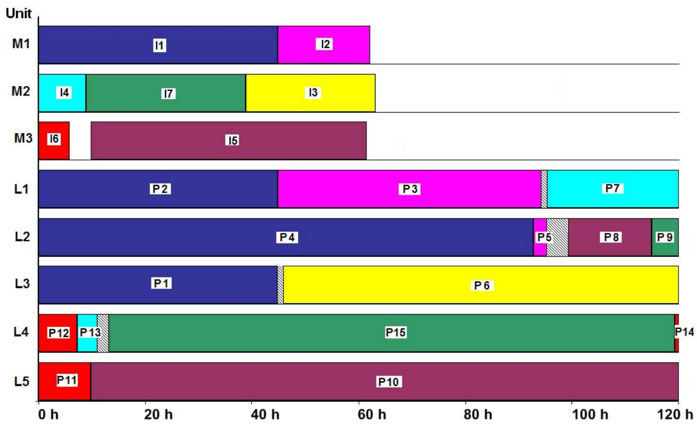


Figure 6.8: Optimal schedule for the UIS policy for Example 2.

No intermediate storage. In this case, there is no storage available to buffer the mismatching production rates between upstream and downstream processes. Table 6.7 shows the results reported by different authors with those obtained in this work. The model used by Shaik and Floudas (2007) yielded a profit of 2689.75 m.u., whereas that of Castro *et al.* (2004) only produced 2672.50 m.u. Mendez and Cerda (2002) did not solve the problem for this storage policy; hence results with their formulation are not included. The formulation presented in this example yields a solution of 2688.31 m.u., which is better than the one obtained by Castro *et al.* (2004) , and slightly worse than the optimal value reported in the literature (0.05%). However, the solution proposed uses a single campaign per intermediate product, reflecting a more realistic industrial practice.

Figure 6.10 shows the Gantt chart corresponding to the solution obtained in this case. This solution is achieved by regulating the processing rate of the mixers according to the requirements of the packing lines. For example, when mixer M1 processes

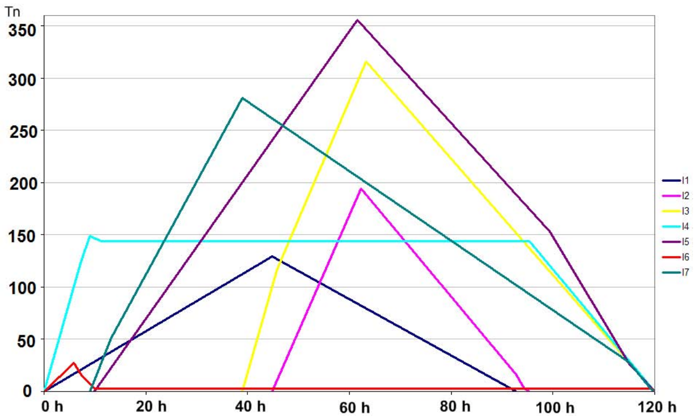


Figure 6.9: Profile of surplus intermediate material for the UIS in Example 2.

Table 6.7: Results for the NIS policy for Example 2.

Model	Profit [m.u.]	Campaigns/ product	Bin., cont., rows	Time [sCPU]
Shaik and Floudas (2007)	2689.75	>1	108 328 1240	157.9
Castro *et al.* (2004)	2672.50	>1	228 762 894	2701
This approach	2688.31	1	38 119 202	10.7

intermediate I1, its processing rate changes from 8.280 ton/h to 14.113 ton/h and then to 11.407 ton/h. These transitions enable downstream facilities to process the material generated by the mixers without the need for intermediate storage.

In this example, the production bottleneck occurs during the packing stage because the packing lines have much lower processing rates than the mixers. The results for the NIS policy are likely to be unrealistic in most cases because intermediate storage tanks are usually available in semi-continuous plants. However, the case study does prove the generality of the scheduling approach presented. Likewise, the solution of this case represents a lower bound for the solution for less restrictive cases. Thus, the solution for the scheduling problem will be bounded by the solutions of the UIS case (upper bound) and NIS case (lower bound).

Finite intermediate storage. Previous studies considered a scenario with three 60 ton storage tanks. The results are shown in Table 6.8. Shaik and Floudas (2007) and Castro *et al.* (2004) reported a profit of 2695.32 m.u., which is the same as the value obtained with unlimited intermediate storage. Therefore, the optimal profit obtained with three 60 ton tanks can not be increased by using additional tanks. Mendez and Cerda (2002) obtained a lower profit (2670.28 m.u.), because their formulation considers a single campaign per product and fixed processing rates. The model proposed in this chapter yields the optimal profit (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 60 ton tanks is large enough to be considered equivalent to the unlimited storage case.

A scenario is considered in which the storage capacity is reduced to two 60 ton tanks. In this scenario, the optimal profit (2695.32 m.u.) is also achieved with a

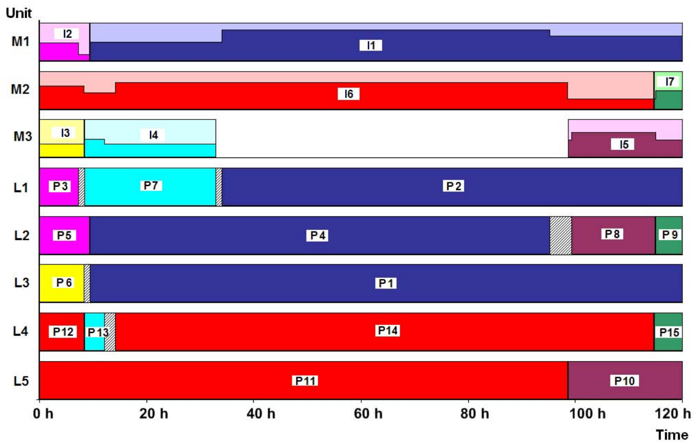


Figure 6.10: Optimal schedule for the NIS policy for Example 2(the height of the boxes represent the percentage of utilization of r_{sj}^{max}).

Table 6.8: Results for the FIS policy with three tanks for Example 2.

Model	Profit [m.u.]	Campaigns/ product	Bin., cont., rows	Time [sCPU]
Shaik and Floudas (2007)	2695.32	>1	276 580 4267	465.6
Castro *et al.* (2004)	2695.32	>1	330 927 1127	162
Mendez and Cerda (2002)	2670.28	1	60 87 361	398.9
This approach	2695.32	1	84 148 402	5.72

single campaign per product. Table 6.9 shows the results obtained with the proposed formulation, and figure 6.11 shows the Gantt chart of the solution. The optimal schedule was obtained by adjusting the semi-continuous equipment processing rates. For example, when mixer M3 processes intermediate I6, the processing rate is adjusted first from 7.598 ton/h to 5.357 ton/h at 7.36 h, and then to 8.690 ton/h at 13.16 h. Therefore, the same optimal objective function value is obtained with lower storage resources consumption. This is a direct consequence of a flexible formulation allowing variable processing rates. Since the production capacity of intermediates may be adjusted below its maximum capacity for those non-limiting stages, the storage requirements for these intermediates are accordingly reduced.

Table 6.9: Results for the FIS policy with two tanks for Example 2.

Model	Profit [m.u.]	Campaigns/ product	Bin., cont., rows	Time [sCPU]
This approach	2695.32	1	77 148 360	12.9

Finite intermediate storage under restricted allocation. A more restricted intermediate storage policy is considered in the work by Shaik and Floudas (2007) . In this case, a specific storage tank is available for each product. The results for both models are reported in Table 6.10. The optimal value objective function corresponds

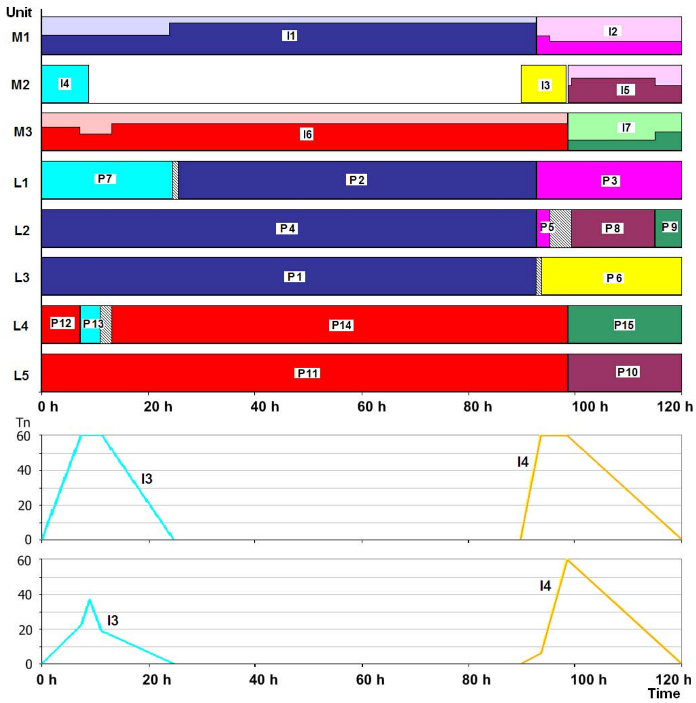


Figure 6.11: Optimal schedule for the FIS policy in CS1 with two storage tanks for Example 2 (the height of the boxes represent the percentage of utilization of r_{sj}^{max}).

to 2695.32 m.u. The proposed approach also reaches the optimal value of the objective function. As in the previous cases, the difference between both approaches lies in the number of campaigns for each product. The work by Shaik and Floudas (2007) considers up to two campaigns for each intermediate product, with changeovers in the mixers. In contrast, this approach considers only one campaign for each product, with variable processing rate inside a campaign. Hence, the same optimal production value is achieved, but the costs of changeovers are avoided, although they are not explicitly quantified in the objective function.

Table 6.10: Results for the FIS policy with restricted storage allocation for Example 2.

Model	Profit [m.u.]	Campaigns/ product	Bin., cont., rows	Time [sCPU]
Shaik and Floudas (2007)	2695.32	>1	164 412 2065	662.58
This approach	2695.32	1	63 148 309	21.23

Finite intermediate storage with maximum demand limits. An additional case is presented in the work by Shaik and Floudas (2007) that a maximum demand for each product. Two cases are presented with three intermediate storage tanks of 60 ton, namely common intermediate storage (CIS) and restricted storage allocation (RSA). The results using different approaches are compared in Table 6.11. The optimal objective function value corresponds to 1388 m.u. in both cases. The model proposed in this chapter also reaches the optimal value. Two important differences may observed in the solution obtained from the proposed formulation when compared to the results obtained by Shaik and Floudas (2007) : a single campaign for each product and no storage resources consumption. Figure 6.12 shows the corresponding Gantt chart for this case with the formulation proposed in this thesis. Again, production is adjusted by changing processing rates inside a given campaign, for example, at time 50.22 h processing rate for product I1 raises from 8.28 ton/h to 14.113 ton/h; next at time 90.61 h, it is adjusted to 8.542 ton/h, and finally at 93.85 h to 5.833 ton/h.

Table 6.11: Results for the FIS policy with maximum demand limits for Example 2.

Model	Profit [m.u.]	Storage	Campaigns/ product	Bin., cont., rows	Time [sCPU]
Shaik and Floudas (2007)	1388	CIS	>1	276 580 4282	8.81
Shaik and Floudas (2007)	1388	RSA	>1	164 412 2080	6.56
This approach	1388	CIS	1	84 148 440	0.20
This approach	1388	RSA	1	63 184 324	0.09

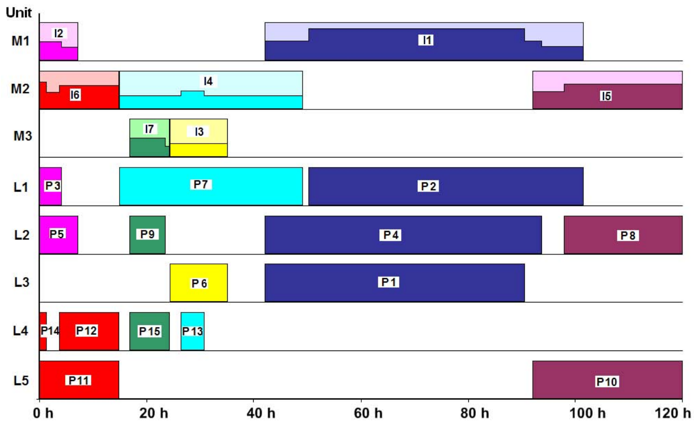


Figure 6.12: Optimal schedule for the FIS policy with maximum demand constraints for Example 2(the height of the boxes represent the percentage of utilization of r_{sj}^{max}).

Mixed intermediate storage. Next, a scenario is considered in which the storage policies are mixed. To be more specific, intermediate products I1, I4 and I5 have a no intermediate storage policy, I6 and I7 have a limited storage policy with one tank of 60 ton available, and I2 and I3 have unlimited intermediate storage capacity.

In this scenario, the optimal profit of 2695.32 m.u. is also achieved with a single campaign per product. Table 6.12 shows the results obtained with the proposed formulation, and Figure 6.13 shows the Gantt chart of the solution. The optimal schedule is obtained by adjusting the processing rates of the semi-continuous equipment, as well as by adjusting the use of the available tank to store intermediate product I6.

Table 6.12: Results for a mixed storage policy with one tank for Example 2.

Model	Profit [m.u.]	Campaigns/ product	Bin., cont., rows	Time [sCPU]
This approach	2695.32	1	46 130 233	3.19

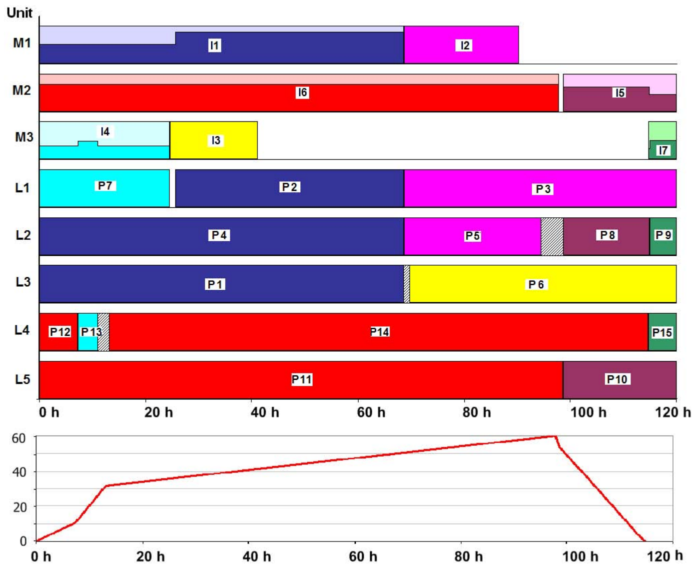


Figure 6.13: Optimal schedule for the mixed policy with a storage tank for Example 2 (the height of the boxes represent the percentage of utilization of r_{sj}^{max}).

6.5 Final remarks

This chapter considers how semi-continuous campaigns with production rate transitions can be used to improve production flexibility and storage management in chemical plants by means of an adequate representation of the plant features. Traditionally, constant rates are maintained throughout production operations, and adjustments are made by using storage resources, production line stops and multiple product campaigns. In contrast, this thesis proposes to adjust the processing rates of a campaign to the specific production requirements. The formulation presented can also be used for fixed operation by tightening the parameters and bounds of the model, and it is also more realistic than other formulations because it allows adjusting production rates and campaign lengths; thus this formulation may be also expected to produce more robust schedules. This proposed operation policy, if feasible according to production and control requirements, allows comparable results to those reported in previous studies, but uses individual product campaigns and requires less storage resources, which makes the system more flexible and cost efficient. These advantages increase storage management flexibility, thereby reducing the capital cost of the plant.

6.6 Nomenclature

Sets and subscripts

i	semi-continuous processing campaigns.
j	semi-continuous production lines.
s	States (intermediate or final products).
t	Storage tanks.
I	Campaigns.
I_s^-	Campaigns that consume state s.
I_s^+	Campaigns that supply state s.
J_i	Available production line for campaign i.
J_s	Available production line for manufacturing state s.
S	States.
S^I	Intermediate states.
S^P	Final states.
T_I	Available tanks to store state from campaign i.
T_S	Available tanks to store state s.

Parameters

d_s	Minimum demand for state s.
H	Time horizon.
l_{sj}^{min}	Minimum allowed length campaign at production line j producing state s.
M_1	A very large number equal to $H + max\{uch_{ii'j}\}$.
M_2	A very large number.
M_3	A very large number to $H + max\{tch_{ii't}\}$.
ρ_{is}	Amount of state s required per unit size of supplying campaign i.
ro_i	Release time of processing campaign i.
r_{sj}^{max}	Maximum production rate at line j generating state s.
r_{sj}^{min}	Minimum production rate at line j generating state s.
ru_j	Ready time of production line j.
$uch_{ii'j}$	Changeover time between campaigns i and i' at production line j.
$tch_{ii't}$	Changeover time between campaigns i and i' at storage tank t.
v_t	Volume capacity of storage tank t.

Continuous variables

C_i	Completion time of campaign i.
CT_i	Completion time of storage task receiving material from i.
$F_{ii'}$	Amount of material supplied by i and consumed by i'.
IT_i	Starting time of stage task receiving material from i.
L_{ij}	Length of campaign i in production line j.
Q_i	Overall production of campaign i.
$V_{ii'}$	Amount of accumulated material consumed by i' at the completion time of its supplying campaign i.
z^{Mk}	Makespan.

Binary variables

$U_{ii'}$	Supply of material from campaign i to i'.
W_{it}	Assignment of material from campaign i to storage tank t.

$X_{ii'}$	Decision on the fact that campaign i is run or stored before i' (equals to 1).
Y_{ij}	Assignment of campaign i to production line j.
$Z_{ii'}$	Decision on the fact that campaign i supplying material to i' starts after i has finished (equals to 1).

Chapter 7

Process Dynamics in Scheduling using Variable Recipes

7.1 Motivation

Batch scheduling and control decision levels are usually optimized separately, even though integrated decisions would potentially increase the overall performance of the plant. This chapter aims at including in the scheduling problem the operating conditions related to the control function, which are traditionally optimized and fixed beforehand, to shed light to the benefits that may be obtained when using variable recipes rather than fixed recipes.

A rigorous approach to including process dynamics entails the consideration of a model based on differential equations, which largely increases the problem complexity as shown in Chapter 8. For this reason, a simpler strategy capable of capturing the effect of operating conditions in the scheduling problem can be extremely useful. Since operating conditions directly affect operational cost and time, the relationship between time and cost can be straightforward derived. Therefore, by introducing cost functions of time in the scheduling problem, operating conditions can be indirectly included in the decision making process. This chapter presents several approximation strategies for the cost function representing the process behavior in the objective function of the scheduling problem.

The actual benefits and drawbacks of the proposed approach are illustrated in three examples, namely two multiproduct batch plants of different complexity and a multipurpose plant under several intermediate storage policies.

7.2 Introduction

The current challenges of process industries have been widely identified and reported in the literature (Grossmann, 2005; Edwards, 2006; Charpentier, 2007; Klatt & Marquardt, 2009). Factors such as globalization of trade, market uncertainty and fierce competition entail dwindling margins in enterprises. In addition, companies must face tighter demands, whose nature is not only economic. Particularly, safety and environmental regulations are increasingly stringent, and enterprises must cope in turn with other issues, such as corporative image and social acceptance. Hence, technological and operational advantages become of utmost importance, and the requirements for flexibility and quick time-to-market are crucial business drivers in many industries. In addition, several authors highlight the importance of integrating information and decisions along the different time and space levels that are found in enterprise structures, and the need of improved models as well as integration and information tools is clear (Grossmann, 2004; Kallrath, 2005; Varma *et al.*, 2007; Grossmann *et al.*, 2008).

In this general scenario, plant management deals with the scheduling function which formalizes decisions about resource allocation and timing of the activities performed in a production plant within a time ranging from days to weeks in order to directly satisfy client requirements or demands issued from a production plan. The decision making process related to the scheduling function is crucial to meet the global goals of the company (Barker & Rawtani, 2005), and its integration with other company functions has received increasing attention in the literature. On the one hand, many works have been proposed to meet the integration with the planning level, whose objective is to fulfill customers' demands minimizing total cost, including inventory and production costs. Maravelias and Sung (2009) present a thorough review on the integrated production planning and scheduling, and point out the current challenges and opportunities in this field. Different modeling approaches are described, and the solution strategies are broadly classified in three categories, namely hierarchical, iterative and full-space methods. On the other hand, the integration between the scheduling and control level, which copes with the real time execution over time and the optimization of dynamic trajectories of process variables, has been only scantly tackled in the literature (Harjunkoski *et al.*, 2009). Although such integration would lead to improved overall plant operability, several hurdles are found to meet it, for example the information transfer, the different time scale domains, the different problem perspective and boundaries and the wide variety of approaches to formulate and solve such problems (Shobrys & White, 2002).

The scheduling task is usually mathematically posed as a MILP or MINLP depending on the problem structure, whereas the control one focuses on the optimization of differential algebraic equations, which usually give rise to nonlinear models of the process. Therefore, the integration of these two problems requires the solution of mixed-integer dynamic optimization problems (MIDO). Several approaches have been identified in the literature to solve these problems. Particularly, the contributions toward the integration of scheduling and control can be broadly classified in three groups (Harjunkoski *et al.*, 2009). The first is the transformation of the MIDO problem into large scale MINLP/MILP problems, as proposed by Chatzidoukas *et al.* (2003). The second relies on the decomposition of the integrated problem into scheduling and control subproblems, as suggested by Nystrom *et al.* (2005). The third entails the use of agent-based systems that employ metaheuristic rules to achieve the integration (Pawlewski *et al.*, 2009). Most of these works focus on decision-making integration for continuous processes. For example, Terrazas-Moreno *et al.* (2007) presented a scheduling formulation that accounts for process dynamics for a cyclically operating continuous plant of polymer manufacturing, in order to calculate the optimal transition duration and profiles that affect decisively the optimal profit achieved.

Busch *et al.* (2007) integrate control and scheduling for a wastewater treatment plant, and solve it using heuristic rules and General Disjunctive Programming. This strategy was later applied to a continuous polymerization process (Prata *et al.*, 2008).

Batch production has gained much importance in the last decades, due to its flexibility to deal with plant demand fluctuations in the fine chemicals industry. Batch processing decisions pertain to the domain of optimal state and control variables profile since the associated operations are dynamic in nature. The very first piece of information for batch scheduling to manufacture a product is the process recipe (Korovessi & Linninger, 2006), specified by a set of processing tasks. These tasks have an associated processing time which may depend on the specific type of equipment employed, the amount of material, the assigned resources and the selected state variables (Reklaitis *et al.*, 1996). In process industries, it is common practice to define and fix the control strategy and processing times for any batch of a given product (Biegler *et al.*, 1999). In some cases such as in the pharmaceutical industry, fixed accurate recipes are compulsory owing to the nature of the products and the process. In other cases, fixed production recipes are obtained from a design optimization stage or determined by previous knowledge and experience over the process behavior. As a result, degrees of freedom of the dynamic problem are disregarded in the scheduling problem and actual production scenarios cannot be globally optimized. Thus, works involving alternative options related to the integration of optimal control profiles in scheduling problems are still very scarce. The main difficulties arise from the computational requirements associated with the solution of the underlying MIDO problem, the lack of appropriate process models, and the large time needed to develop and reduce process models which require both sufficient detail to represent the process and enough simplicity to be solved by the available optimization tools (Bhatia & Biegler, 1997).

According to Harjunkoski *et al.* (2009), *"the integration of scheduling and control into large-scale MIDO problems is probably only useful in certain selected cases, where the complexity of the dynamic plant model is well in proportion to the size of the scheduling problem."* In their work, it is also highlighted that possible solutions to the integration problem may probably remain in the domain of including in the scheduling problem some indicators related to the dynamic part of the problem. In this context, some work has already been presented in the literature, which may be regarded as intermediate strategies, since they define processing times as a function of batch sizes or state variables (Bhatia & Biegler, 1996). Romero *et al.* (2003) present a framework that includes the possibility of recipe adaptation in the optimization of batch processes. Specifically, a linear-based recipe model is integrated into a multipurpose scheduling algorithm called S-graph. In that work, the productivity maximization was established as objective function and the cost of modifying process variables was considered negligible. In addition, Ferrer-Nadal *et al.* (2008) incorporate the concept of recipe flexibility as an additional rescheduling action in the reactive batch operation of multipurpose batch plants. They assume a linear model in a predefined flexibility region around nominal operating conditions, penalize any deviation from the optimal operating conditions and solve a MILP based on the general precedence model (Mendez *et al.*, 2001). The aforementioned authors assume that there is a single optimal nominal recipe, and changes may be produced linearly around such operating point. Moreover, there is no reference in the literature regarding the consequences that may be derived from including in the scheduling problem batch-to-batch variable processing times from a general scheduling perspective.

Thus, the main objectives that may lead to an optimal integrated decision of the scheduling and control problems are important to be considered (Harjunkoski *et al.*, 2009). From the scheduling perspective, the objective function depends on the decision maker criteria, which are based both on his/her experience and the nature of the plant. Hence, a unique objective function may not be suitable for all scheduling problems. Typical objective functions consider the minimization of makespan, earliness, tardiness, or overall cost (Hoogeveen, 2005). Regarding control specifications, they are usually supposed to be determined as part of the design activity, and a suitable criterion consists of either measuring the economic success such as batch time or total production cost per unit product including annualized investment and operating costs (Korovessi & Linninger, 2006) or using processing criteria (Edgar, 2004). The determination of the optimal operating profiles is usually referred to as the optimal control problem (Seider *et al.*, 2004).

The traditional hierarchical approach established between the scheduling and control levels consists of an initial optimization stage which provides as output a nominal recipe with a set of operating conditions and processing times to be implemented at the scheduling level. As a result, the operating conditions and times are fixed, and any deviation from those values is even considered negative for the process performance. However, such simplification of the problem at the scheduling level ignores possible trade-offs between operating conditions and times.

This chapter aims at gaining insight into the benefits that may be obtained when integrating process level knowledge in scheduling problems of different complexity. The proposed approach is based on the use of flexible recipes and economic functions for both the scheduling and control problems. Economic functions are employed to determine the so-called optimal recipe in the control level; and the cost associated with the variation of the state variables is then represented in the objective function of the scheduling problem. As a result, the dynamics of the processes are implicitly considered in the objective function by allowing variable batch processing times, thus including the trade-offs between batch time and state variables variations. Therefore, the scheduling level indirectly decides on the batch conditions that optimize the overall profit, in such a manner that trade-offs between process conditions and scheduling decisions of complex scheduling problems can be tackled. Moreover, the proposed recipes are feasible in an operational range wider than for those defined around nominal operation reported in previous works. Likewise, this chapter assesses the benefits of allowing variable batch-to-batch processing times. As for the economic cost function, several approximations of different complexity are proposed. Specifically, linear, quadratic and piece-wise linear functions represent the relationship between time and cost; and heuristic strategies based on linear cost approximations are also presented specially aimed to deal with large sized problems. Overall, by including variable processing times at the scheduling level, the search space of the scheduling optimization problem broadens, and so does the potential for solution improvement.

7.3 Problem statement

This work considers variable batch-to-batch processing recipes in contrast to fixed nominal recipes in order to improve the overall plant performance. The corresponding problem is defined as follows. Considering:

Process dynamics

- a dynamic process model that describes the process;
- a set of control variables, such as temperature, feed flows or pressure;
- a set of state variables that are ruled by the process model;
- a set of constraints imposed over the process conditions;

Process operations planning data

- a specific time horizon;
- a set of materials: final products, intermediates and raw materials;
- a set of expected final products with minimum and maximum demands;
- a fixed batch plant topology consisting of a set of equipment technologies for processing stages;
- a set of production recipes containing the production sequences or stages;
- a set of fixed production stages whose processing times, mass balance coefficients and resources utilization are optimized and fixed beforehand;
- a set of variable production stages defined by their corresponding process dynamics;
- a set of intermediate materials storage policies;

Economic data

- a selling price for every final product;
- direct cost parameters such as labor, energy, raw material costs and unfulfilled demand costs;
- a set of environmental, quality or safety constraints that have an associated economic cost;

The goal is to determine:

- the number of batches required to meet the demand;
- the assignment and sequencing of the batches;
- the amount of final products to be sold;
- the processing times of the variable stages of each batch;
- the operating conditions of the variable production stages of each batch;

such that the adopted performance metrics are optimized. In this case, economic functions characterize the overall plant performance at both scheduling and control levels. Accordingly, economic indicators, namely profit (equation 7.1) and profitability (equation 7.2), may be used as objective functions. In both functions, a thorough definition of profit, including revenues, operating costs (such as electricity, steam or water consumption), raw material costs, and unfulfilled demand penalty, is adopted.

$$\begin{aligned} z^{profit} = & \, Revenues - OperatingCost - RawMaterialCost \\ & - DemandPenalty \end{aligned} \tag{7.1}$$

$$z^{profitability} = \frac{z^{profit}}{z^{Mk}} \tag{7.2}$$

7.4 Solution procedure

As introduced in section 7.2, the integration between scheduling and control is managed in this chapter by the use of economic functions, which can be used to relate the variables of scheduling and control. Specifically, economic performance indicators, such as profitability or profit, may be affected by the main controlled variables in each case, namely assignment and sequencing at the scheduling level, and operating conditions at the control level.

The proposed solution procedure of the scheduling problem with control related information comprises three main stages (Figure 7.1), namely the definition of flexible recipes, the formulation of the scheduling problem considering variable processing times, and finally the implementation of the recipes. Thus, this procedure can be extended to consider heuristic approaches for the scheduling problem. This indirect approach to consider process variables in the scheduling problem is applied in Chapter 8 for comparison purposes.

Flexible recipe definition. Traditionally, the problem solved at this step has a rigid structure, since the recipe obtained at the design step is usually adopted for the scheduling without any further consideration regarding cost or operating conditions. However, in this work the nominal recipe may be later modified during the scheduling solution by exploiting recipe flexibility to improve the overall performance of the plant.

First, those process variables with larger influence on the process dynamics should be identified along with their operational cost. Such variables are then used to define the flexible recipe, since they offer additional degrees of freedom at the scheduling level. Hence, these variables will be regarded as free decision variables in the flexible recipe. The selection of such variables may be either given by process experience or determined by a sensitivity analysis of the process considering economic implications.

Next, the relationship between the free decision variables and both processing time and operating cost should be formulated. As a result, a correspondence between time and operating cost is explicitly established and introduced in the scheduling problem through the objective function. Therefore, for each time, the set of free decision variables that correspond to the minimum cost must be obtained, either by determining the Pareto optimal curve resulting from historical data or by an optimization based strategy. Thus, the free decision variables must be time independent along the batch time since variable profiles cannot be accounted by means of this strategy. When a single free decision variable is considered, the relationship between time and the

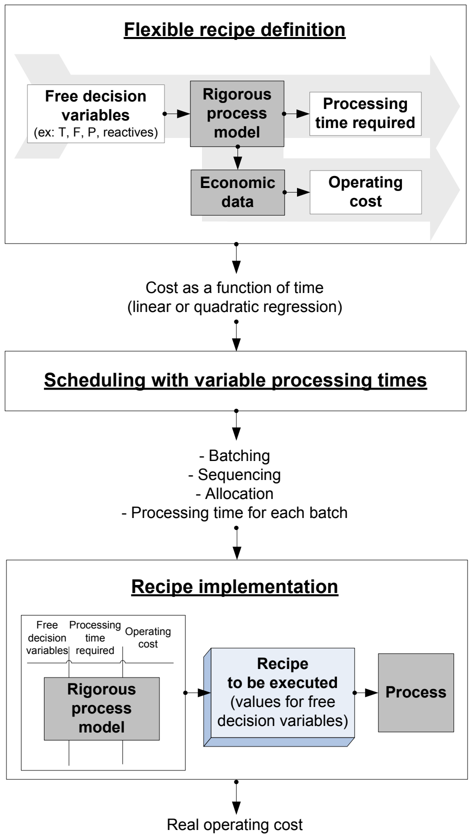


Figure 7.1: Steps of the solution procedure for integrated decision of scheduling and control by means of the objective function.

independent variable is straightforward, which allows an easier implementation of the recipes obtained at the scheduling step. Hence, after establishing the processing time for each batch, it is possible to obtain the value of the free decision variables. Anyhow, working with more free variables is possible by determining for every batch processing time, the optimal combination that would result in a minimum economic indicator.

Additionally, it must be considered that in order to efficiently tackle the combinatorial complexity of the scheduling problem optimization, simple cost functions are required. For example, linear, quadratic or piecewise linear functions could be used to define the operating cost as function of time.

Scheduling with variable processing times. Several formulations have been proposed in the literature to tackle the scheduling problem, as stated in Méndez *et al.* (2006). Different classifications can be adopted, which can be broadly classified as network based and sequential based from a process representation perspective. Specifically, sequential based plants may be further divided in multiproduct and multipurpose batch plants.

Any of the existing mathematical formulations may be used to solve the scheduling problem with variable processing times as long as they allow variable times and the use of economic indicators as objective function, so that they can successfully model the relationship between cost and time. This work considers sequential based plants, and builds upon the immediate and general precedence models presented in Chapters 4 and 5, since they adequately deal with the modeling of multiproduct and multipurpose batch plants. Thus, such formulations have been adequately extended to consider the aforementioned issues. Next, the additional constraints regarding the general precedence model are thoroughly described. As for the immediate precedence model, such equations are directly applicable since sequencing constraints are not involved in the proposed modifications.

Objective function. The integration strategy is based on the use of an economic objective function at the scheduling level, which determines the final production rates given a minimum demand and a known time horizon. Therefore, using flexible recipes allows to study the trade-offs between the economic factors and the influence that overall available time has on the processing time. Equation 7.3 describes how the profit is determined in this chapter. The main idea consists of including the revenues and the total costs. In this case, batch price includes revenues and most of the operating costs, whereas those variable costs depending on the processing time of the variable stages (E_{ips}), such as energy cost and raw materials costs and unaccomplished demand (DS_{ps}), which is additionally penalized, are independently considered.

$$z^{profit} = \sum_{ip} BP_p W_{ip} - \sum_{ips \in DS_{ps}} E_{ips} - \sum_{p} CD_p \cdot AD_p \tag{7.3}$$

Batch assignment. An important consideration refers to batch assignment and demand fulfillment. Therefore, equation 7.4 defines that if a batch i of a product p is produced ($W_{ip} = 1$), then all stages of its recipe have to be assigned to an available unit. In addition, a variable is defined to account for the percentage of the demand that cannot be fulfilled (AD_p) (constraint 7.5). Additionally, it is necessary to avoid problem degeneration. Hence, Equation 7.6 forces that a given batch of a product can only be assigned if the previous batch in the set has been assigned.

$$\sum_{u} Y_{ipsu} = W_{ip} \qquad \forall p, i, s \tag{7.4}$$

$$\sum_{i} W_{ip} BS_p + AD_p \geq D_p^{min} \qquad \forall p \tag{7.5}$$

$$W_{i+1p} \leq W_{ip} \tag{7.6}$$

Timing constraints and variable cost function definition. The relationship between time and cost must be defined for those stages whose dynamics is considered at the scheduling level. Therefore, depending on the adopted cost function representation, alternative equations may be considered. In all cases, processing times of the corresponding dynamic stages (td_{ips}) must lie between a minimum and a maximum bounds established in the flexible recipe, if the batch is performed (constraints 7.7 and 7.8).

$$td_{ips} \leq T_{ps}^{max} + M\left(1 - W_{ip}\right) \qquad \forall i, p, s \mid (p, s) \in DS_{ps} \tag{7.7}$$

$$td_{ips} \geq T_{ps}^{min} - M\left(1 - W_{ip}\right) \qquad \forall i, p, s \mid (p, s) \in DS_{ps} \tag{7.8}$$

On the one hand, the variable cost of the dynamic stages takes a value according to the approximated cost function used, which may be typically either linear or quadratic (equations 7.9 and 7.10). The value of the variable cost of a given batch at the scheduling level is only determined if the batch is actually performed.

$$E_{ips} = a_0 W_{ip} + a_1 td_{ips} \qquad \forall i, p, s \mid (p, s) \in DS_{ps} \tag{7.9}$$

$$E_{ips} = a_0 W_{ip} + a_1 td_{ips} + a_2 td_{ips}^2 \qquad \forall i, p, s \mid (p, s) \in DS_{ps} \tag{7.10}$$

On the other hand, piecewise linear functions may be considered. For such case, the cost function can be approximated by using an additional SOS2 type variable (RS_{ir}). Specifically, the operating cost as a function of time for each product is divided in $r-1$ intervals, for which the time and cost coordinates of the piece boundaries are defined ($tval_{pr}$ and $cval_{pr}$). Therefore, the assignment variable of a piece of function to the batch must be only considered if the batch is actually performed (constraint 7.11). Additionally, the value of time and cost must be computed according to the active interval of the cost function (equations 7.12 and 7.13).

$$\sum_{r \in RP_{pr}} RS_{ir} = W_{ip} \qquad \forall i, p \tag{7.11}$$

$$td_{ips} = \sum_{r \in RP_{pr}} tval_{ir} \cdot RS_{ir} \qquad \forall i, p \tag{7.12}$$

$$E_{ips} = \sum_{r \in RP_{pr}} cval_{ir} \cdot RS_{ir} \qquad \forall i, p \tag{7.13}$$

Moreover, the finishing time has to be adequately defined according to the fixed and variable stages. Therefore, if a stage has a fixed processing time, the finishing time will depend on the realization of the batch (equation 7.14), whereas for dynamic stages, the processing time is a decision variable (equation 7.15). Finally, the time horizon cannot be exceeded under any circumstance (constraint 7.16).

$$\begin{gathered} Tf_{ips} = Ts_{ips} + \sum_{u' \in U_{p(s-1)}} tt_{pu'} + \left(\sum_{u \in U_{ps}} pt_{psu} + tt_{pu} \right) Y_{ipsu} + Tw_{ips} \\ \forall i, p, s \mid (p, s) \notin DS_{ps} \end{gathered} \tag{7.14}$$

$$\begin{gathered} Tf_{ips} = Ts_{ips} + \sum_{u' \in U_{p(s-1)}} tt_{pu'} + td_{pis} + \sum_{u \in U_{ps}} tt_{pu} Y_{ipsu} + Tw_{ips} \\ \forall i, p, s \mid (p, s) \in DS_{ps} \end{gathered} \tag{7.15}$$

$$Tf_{ips} \leq H \qquad \forall i, p, s \tag{7.16}$$

Recipe implementation. The scheduling optimization problem provides as output the processing times of the variable stages. For the recipe implementation, the values of the process variables corresponding to such processing times must be determined, so that they can be implemented in the process. When a rigorous model of the process stage exists, the values of the free decision variables can be directly obtained from the optimization of the model subject to the timing decided at the scheduling level. In contrast, if only historical data are available, the values of the free decision variables may be either rounded to the best existing historical data or interpolated between the existing points, leading to solutions that may be unrealistic considering the actual plant performance. Additionally, this step can lead to the reoptimization of the scheduling problem, if necessary. For example, a tighter range of values of times or the free decision variables could be used to reformulate a better estimation of the variable operating costs introduced at the scheduling level. This idea is presented next as a recursive heuristic algorithm.

Recursive heuristic algorithm. The use of a recursive heuristic algorithm (Figure 7.3) allows improving the quality of the adjusted functions by consecutively reducing and adjusting the time interval, thus solving a more detailed scheduling problem at each iteration. First, the approximation to the cost function must be decided. It can be either a linear regression within a given interval, or tangent to the function at a given time point (convex envelope) (Figure 7.2). An additional consideration is the way the time interval must be reduced at every iteration around the time value of the previous approximation for every batch. In this case, a fixed percentage (TI_{red}) of the whole time interval of the previous iteration is reduced at every iteration. Therefore, the strategy consists of iteratively repeating the cost function approximation for all batches and solving the scheduling problem until a stop criterion is met. Specifically, different stop criteria may be set, such as a maximum number of iterations (N_{iter}) of the algorithm, a tolerance (tol) between the profit estimated by the adjusted function and the actual profit or obtaining the same actual profit in two consecutive iterations of the recursive algorithm.

7.5 Examples

Three examples have been posed to study the effects of introducing process dynamics at the scheduling level. They consist of two multiproduct plants of different complexity and a multipurpose batch plant with different products under different storage policies. The immediate precedence formulation presented in Chapter 4 has been extended to consider variable processing times and economic objective functions as defined in the previous section, and applied for the former plant structure, whereas the adapted general precedence model (Chapter 5) is applied to the multipurpose facility. In all cases, the considered objective function consists of the maximization of profit, which includes the benefit of each batch, the operating cost associated with the stage of variable processing time, related to the process variables to be modified, and an unfulfilled demand penalty. The mathematical model has been implemented in GAMS and solved using the MILP solver CPLEX 9.0 for linear based cost functions, and the MINLP solver BARON 8.1 in the case of quadratic approximations, in a 2.26 GHz Intel Core Duo computer.

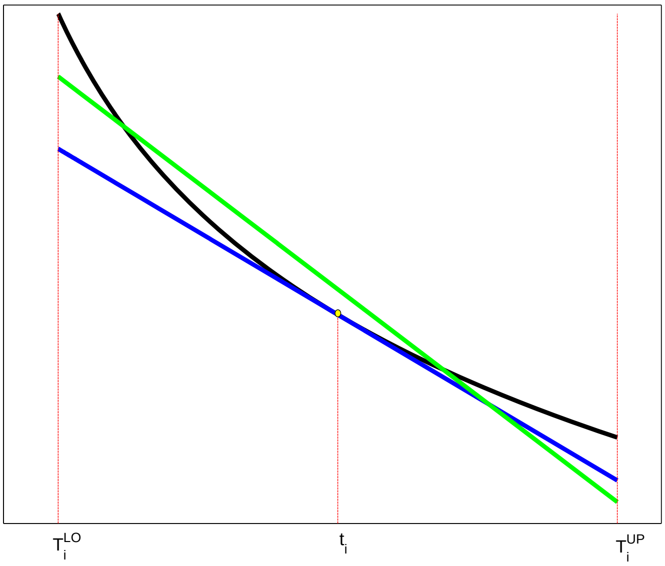


Figure 7.2: Cost adjustment strategies for the heuristic approaches for a nonlinear cost function (black) around a batch time t_i within a time interval defined by T_i^{LO} and T_i^{UP}: linear regression within the whole interval (green), convex envelope around the previous solution point (blue).

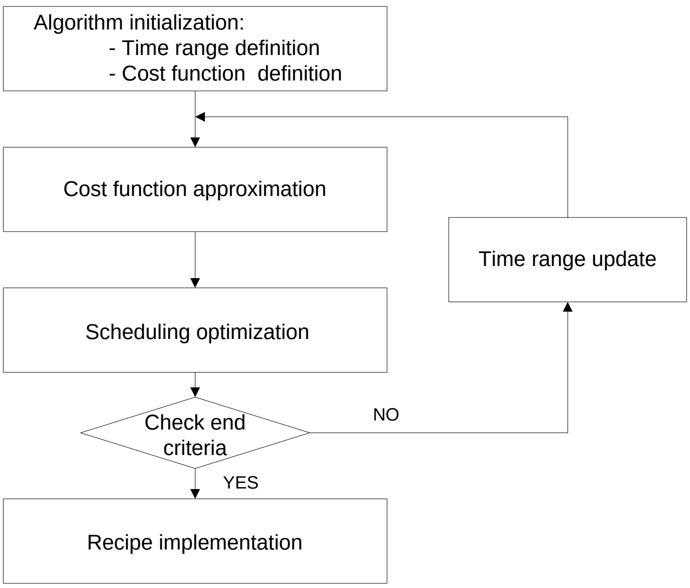


Figure 7.3: Recursive algorithm scheme for heuristic approaches.

The former two examples consider variable energy costs. First, in a multiproduct plant, approximated cost functions have been adjusted as linear and quadratic regression functions of time, as well as piecewise linear functions, and the recursive algorithms have been applied as well. Next a multipurpose plant with approximated costs based on linear regression and several storage policies are studied. The third example considers both energy and raw material variable costs for a multiproduct fiber plant considering two free decision variables for recipe definition. In this case, the same strategies for approximating cost functions as in the first example, but for the quadratic regression, have been studied.

7.5.1 Example 1

A multiproduct batch plant processes two products, i.e. A and B, through three stages (Figure 7.4). The first stage is an isothermal reaction process where a conversion of 95% must be achieved, and whose processing time is a function of the temperature (Figure 7.5). Product batch sizes, production times and product demands are given in Tables 7.1 and 7.2. A single unit is available for each stage and the non intermediate storage policy is adopted. A finite production time horizon of 6 hours is considered. Product changeover times and costs are disregarded in order to specifically characterize the effect of variable processing conditions over the scheduling results.

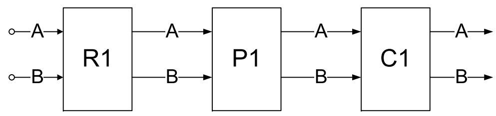


Figure 7.4: Plant flowsheet for the three stage multiproduct plant in Example 1.

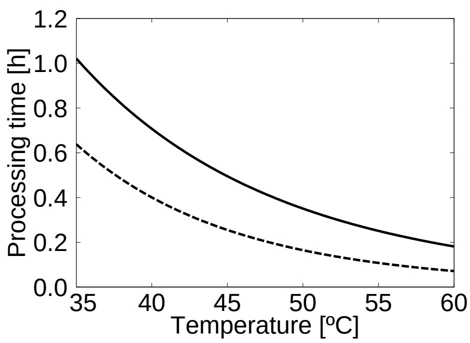


Figure 7.5: Processing time dependence upon temperature in stage 1 for the production of: A (solid line) and B (dashed line) in Example 1.

Dynamics of the reaction stage follow a second order kinetics for both products with the reaction parameters from Table 7.3, and they are fully characterized in Matlab (Mathworks, 2009). The reaction system consists of a continuous stirred tank reactor. A feedback temperature control system is also modeled, leading the processing temperature to the desired value through a heating jacket. A classical proportional integral (PI) controller is considered (K_P = 0.964 and K_I = 0.030

Table 7.1: Product prices and lot sizes for Example 1.

Product	Batch Benefit [m.u./batch]	Batch size [ton/batch]	Demand [ton]	Unitary energy cost [m.u./MWh]
A	30	5	20	90
B	40	6	24	90

Table 7.2: Recipe stage times for Example 1 [h].

Product	Stage 1		Stage 2		Stage 3	
	Unit	Time [h]	Unit	Time [h]	Unit	Time [h]
A	R1	Variable	P1	0.5	C1	0.5
B	R1	Variable	P1	0.8	C1	0.4

s^{-1}). All this information allows to characterize the processing time dependence upon temperature, being the higher temperature the lower production time. However, high temperature entails high energy cost. Therefore, a trade-off exists between minimum time and production cost. The objective of the scheduling optimization problem consists of maximizing the total profit in a time horizon of 6 hours (equation 7.3). Six approximation strategies for cost optimization are examined, namely: i) the nominal recipe implementation; ii) linear regression cost approximation; iii) quadratic regression cost approximation; iv) piecewise linear cost approximation; and two algorithmic iterative strategies (Figure 7.3), v) based on linear regression, and vi) based on the convex envelope for approximating costs.

Table 7.3: Kinetic parameters in reaction stage for products A and B in Example 1.

Product	Reaction rate constant $[m^3/(s \cdot kg)]$	Activation Energy $[\cdot 10^3 kJ/(molK)]$
A	$2.1 \cdot 10^5$	59.029
B	$1.8 \cdot 10^8$	74.826

Case i: Nominal recipe. The traditional approach consists of optimizing the operation of a given product, and next applying the nominal recipe to any batch of that product. In this case, the maximization of total profitability for the reaction stage of a single batch is considered, obtaining the energy cost from the energy balance (equations 7.17 and 7.18) resulting in the optimum processing conditions described in Figure 7.6. The minimum and maximum reaction times (Table 7.4) are determined by the operating restrictions regarding processing temperatures (Figure 7.6).

$$z^{profitability} = \frac{BP_p - EC_p}{t_p^{reaction}} \tag{7.17}$$

$$EC_p = [\Delta H_{Rp} \cdot BS_p \cdot X_{Rp} + c_p \cdot BS_p \cdot (T - T_0)] \cdot PE \tag{7.18}$$

Table 7.4: Optimal recipe and limiting conditions for Example 1.

Product	Optimal reaction time [h]	Energy cost [m.u./batch]	T_{sp}^{min}[h]	T_{sp}^{max}[h]
A	0.5846	22.7162	0.1813	1.0216
B	0.3105	32.8366	0.0713	0.6385

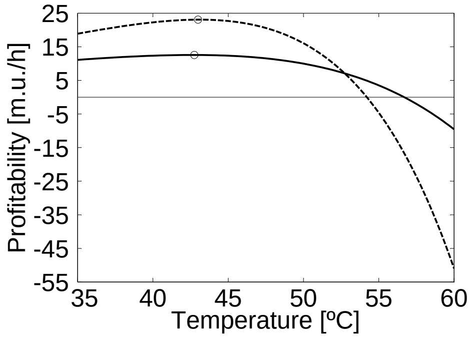
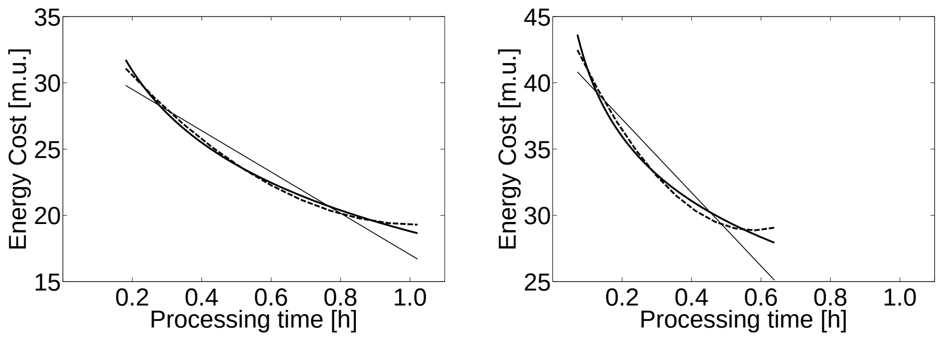


Figure 7.6: Profitability as a function of temperature for the production of: A (solid line) and B (dashed line) in Example 1. Circles represent the maximum value for each case.

Cases ii, iii and iv: linear, quadratic regressions and piecewise linear approximations. In order to exploit the flexibility of the reaction stage in the scheduling problem, the relationships between processing time and temperature, and temperature and energy cost are introduced in the scheduling by adjusting the energy cost as a function of time (Figure 7.7). As a result, linear and quadratic regressions of the energy cost are obtained (Table 7.5 and Figure 7.7) between the maximum and minimum reaction times. Additionally, a piecewise linear approximation to the original cost function has been defined by dividing the reaction time into 5 equally spaced time intervals.

Table 7.5: Linear and quadratic regressions for variable cost approximations for Example 1.

Product	Linear Cost [m.u./batch]	Quadratic Cost [m.u./batch]
A	$32.6260 - 15.5879t$	$36.7037 - 34.0498t + 16.6491t^2$
B	$42.7954 - 27.6950t$	$46.5673 - 61.3099t + 53.0707t^2$



(a) Production of A.

(b) Production of B.

Figure 7.7: Energy cost as a function of time in Example 1: real data (thick solid line), linear regression (thin solid line) and quadratic regression (dashed line).

Cases v and vi: iterative regression and convex envelope. The recursive heuristic algorithms are applied, considering 1000 points between the maximum and minimum processing times to approximate the regression. A reduction of 90% of the initial time interval (TI_{red}) is considered at every iteration. The end criteria consist of: i) a tolerance (tol) between the actual profit and the estimated one of 10^{-3}; ii) no improvement in the actual profit, and iii) a maximum number of 20 iterations (N_{iter}).

Results. Table 7.6 summarizes the main results for the aforementioned cases. The use of the nominal recipe leads to the worst optimal profit from all the values obtained with the proposed strategies. When adjusting a linear regression, the optimal solution improves significantly. Moreover, the results with all the remaining strategies, namely the piecewise linear function, the quadratic function and the iterative approaches, are identical in terms of batch times, and all of them lead to the maximum profit.. The difference between the profit obtained with the linear regression costs and the other strategies is relatively small in this case, even though the operating times are widely different (Table 7.7). Anyhow, those times and their corresponding optimal conditions are completely different from those obtained using the nominal recipe. Although the objective function used for designing the nominal recipe, i.e. profitability, is not the same as the one used at the scheduling level, namely total profit, the obtained nominal processing times allow to produce the whole demand. In contrast, if process conditions corresponding to the minimum batch cost for each product had been used, the nominal processing time would have been the maximum possible time and it would have not been possible to fulfill the total demand in the fixed horizon resulting in a reduction of the overall economic performance.

Table 7.6: Computational results for the different approaches in Example 1.

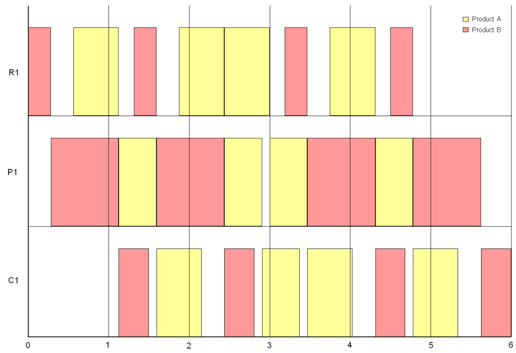
Approximated model/ Strategy	Actual profit [m.u.]	Iterations	Bin., cont., eqs.	Time [sCPU]
Nominal recipe	58.07		100,164,416	0.09
Linear regression	71.87	-	100,184,444	0.42
Quadratic regression	76.67	-	100,184,444	7200[1]
Piecewise linear function	76.67	-	100,232,444	6.14
Iterative linear regression	76.67	7	100,184,444	197.8
Iterative convex envelope	76.67	5	100,184,444	95.72

[1]Maximum computational time exceeded without global optimality.

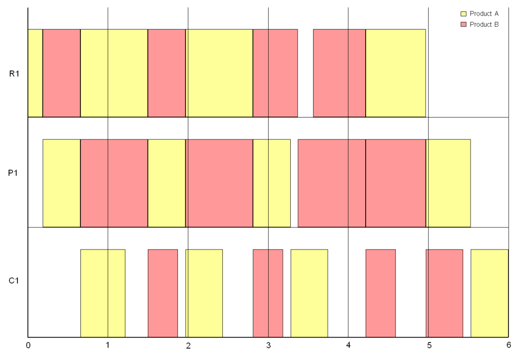
Table 7.7: Optimal reaction times for Example 1 [h].

Approximated Model/Strategy	t_{A1}	t_{A2}	t_{A3}	t_{A4}	t_{B1}	t_{B2}	t_{B3}	t_{B4}
Nominal recipe	0.58	0.58	0.58	0.58	0.31	0.31	0.31	0.31
Linear approximation	0.18	0.80	0.80	0.80	0.50	0.50	0.63	0.61
Piecewise linear, quadratic approximations and iterative approaches	0.80	0.80	0.80	0.80	0.30	0.50	0.50	0.50

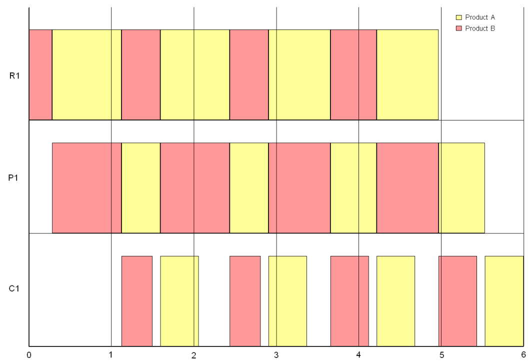
Additionally, it is observed in Figure 7.8 that the operating times of the better schedules derive from the adaptation of the recipes to the start and finishing of the plant activity, as well as to the processing times of the different products in the plant. Such features can only be considered at the scheduling level, so the potential improvement of including variable process recipes is evident. Moreover, it should be noted that the different processing times result in different production sequences (Figure 7.8).



(a) Nominal recipe.



(b) Linear function of energy costs.



(c) Quadratic function, piecewise linear function and iterative algorithms for energy costs.

Figure 7.8: Gantt charts for the different cost approaches in Example 1.

As shown in Table 7.6, the final solutions in both iterative algorithmic strategies correspond to the best possible solutions obtained with the quadratic and piecewise linear functions. Figures 7.9 and 7.10 show the evolution along the different iterations of the adjusted profit and the actual profit. Both iterative strategies start at the same estimated scheduling derived from the linear regression and the convex envelope around the midpoint considering the whole time range for each batch product. Such time range decreases at every iteration for each batch, and both algorithms converge after a given number of iterations, 7 iterations considering the regression approximation and 5 for the convex envelope approximation. In the former approximation, the trajectories along the iterations of the profit estimation and the actual profit cross each other at several points and their values do not coincide at the end criteria, whereas in the latter approximation, both functions converge to the same value. In fact, the convex envelope strategy underestimates the actual cost, thus provides with upper bounds of the actual profit.

Model dimensions and solution times for the different strategies are compared in Table 7.6. The quadratic regression function approximation consumes the maximum time. Hence, it is highly difficult to consider such kind of functions for large problems. As for the other approximations, the piecewise linear strategy is the one which consumes the least time, and reaches the optimal solution. However, for large sized problems, it introduces a large number of additional variables, which may increase the computational complexity of the problem. Hence, the heuristic recursive approximations may be an interesting approach for large size problems, since linear functions are considered at every iteration.

7.5.2 Example 2

The illustrative multipurpose batch plant presented in Example 1 of Chapter 5 is considered. The plant processes four products which sequentially undergo different stages. The production sequences and the original processing times for each product are given in Appendix C. Additionally, it has been assumed that an isothermal reaction process, similar to the one presented in Example 1, takes place in unit U2, affecting products B, C and D. Hence, times affecting this stage are now variable.

A minimum demand of three batches of each product is to be manufactured, and a maximum of one additional batch of each product are considered to be produced in a time horizon of 170 hours. The maximization of total profit (equation 7.3) has been established as the overall objective function. Table 7.8 contains the batch product benefits and the penalties for unaccomplished demand. Thus, the effect of adopting different intermediate storage policies over the total profit and the influence of variable processing times are studied. The storage policies considered are unlimited intermediate storage (UIS), non intermediate storage (NIS), one common intermediate storage only available after unit U3 (CIS) and zero-wait time (ZW). Product transfer times are assumed negligible, but only feasible solutions from the transfer point of view are considered as discussed in Chapter 5.

In this example, the optimal processing time is regarded as the one that minimizes energy cost, so the nominal recipe (Table 7.9) corresponds to the one presented in previous works (Kim *et al.*, 2000). Energy cost is calculated as in Example 1 (equation 7.18) with unitary energy cost of 80 m.u./MWh.

However, such processing time can be reduced at cost expenses, and a linearly approximated function that relates time to energy cost (Figure 7.11 and Table 7.9) is provided and introduced in the scheduling problem in order to integrate processing

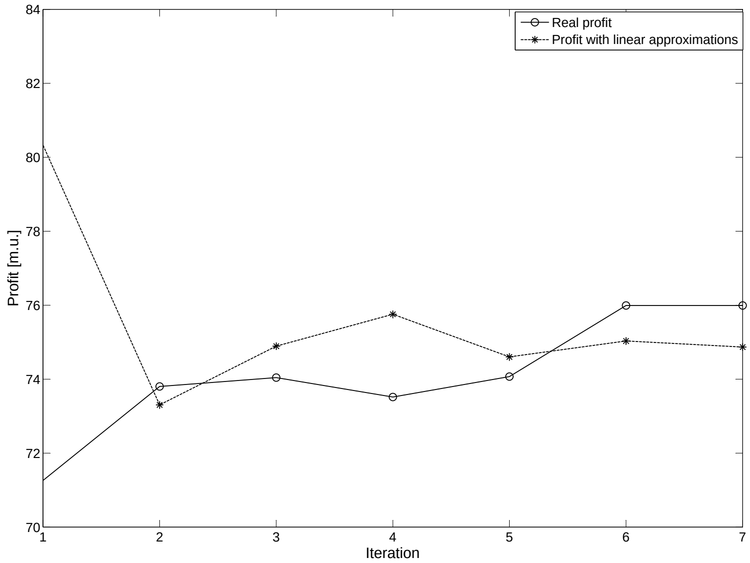


Figure 7.9: Evolution of adjusted profit and actual profit in the algorithmic procedure considering linear regressions for each batch in Example 1.

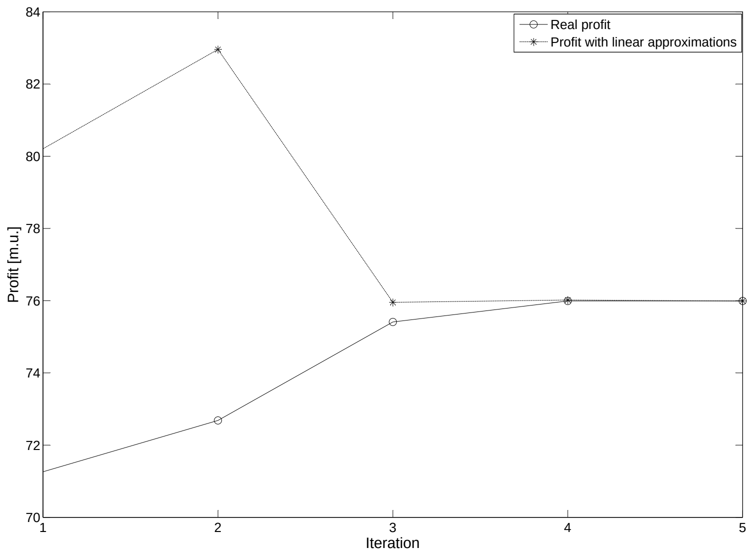


Figure 7.10: Evolution of adjusted profit and actual profit in the algorithmic procedure considering convex envelopes for each batch in Example 1.

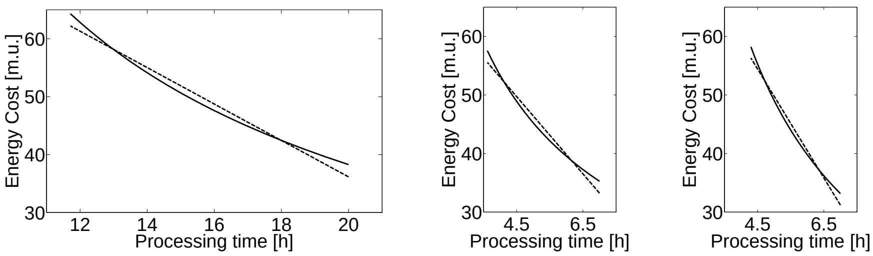
Table 7.8: Processing times, production benefit and unfulfilled demand penalty for Example 2.

Product	Batch benefit [m.u./batch]	Unfulfilled demand penalty [m.u./batch]	Stage 1		Stage 2		Stage 3		Stage 4	
			Unit	Time [h]	Unit	Time [h]	Unit	Time [h]	Unit	Time [h]
A	85.0	125.0	U1	15	U3	8	U4	12	-	-
B	81.2	140.0	U1	10	U2	Variable	U3	5	U4	13
C	88.8	120.0	U3	9	U2	Variable	U1	20	-	-
D	86.4	135.0	U4	5	U3	17	U2	Variable	-	-

Table 7.9: Nominal recipe, limiting conditions and energy costs as a linear function of time for processing stage at unit U2 for Example 2.

Product	Nominal time [h]	Nominal cost [m.u./batch]	T_{sp}^{min}[h]	T_{sp}^{max}[h]	Energy cost [m.u./batch]
A	-	-	-	-	-
B	20	38.2	11.71	20	$99.04 - 3.144t$
C	7	35.2	3.61	7	$79.41 - 6.597t$
D	7	33.1	4.30	7	$96.28 - 9.299t$

conditions decisions.



(a) Production of B. (b) Production of C. (c) Production of D.

Figure 7.11: Energy cost as a function of time in Example 2: real data (solid line) and linear regression (dashed line, see Table 7.9).

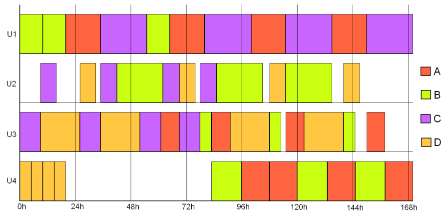
Figure 7.12 presents the Gantt charts comparing the nominal recipe and variable time recipe approaches for the different intermediate storage policies, and the profit associated with the results obtained by the proposed approach.

Regarding the UIS policy (Figure 7.12(a)), the optimization results in the same number of batches and processing times as the nominal recipe, since they can be done in the whole time horizon. This storage policy satisfies the minimum demand of all products in the given time horizon, and an additional batch of all products with the exception of B can be processed as well. For the other storage policies, when using the recipes with the nominal time, there is no possibility to fulfill the minimum demand, whereas the variable processing time approach allows to achieve the minimum demand in all cases. Hence, total profit is higher when recipe flexibility is considered, and the higher energy cost is compensated by the time decrease for the reaction stage.

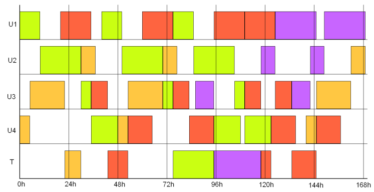
As for the CIS policy, the result using nominal recipes (Figure 7.12(b)) cannot satisfy the minimum production of C by one batch, but an additional fourth batch of A can be processed; in contrast, the use of variable recipes (Figure 7.12(c)) allows to fulfill the minimum demand of all products and an additional batch of products A and D. In the latter case, in order to fully satisfy the demand, the processing times of the three batches of product B in unit U2 are reduced from 20h to 11.71h, 12.29h and 15.00h.

Under NIS policy, again the minimum demand cannot be fulfilled using the nominal recipes, and one batch of product C cannot be produced (Figure 7.12(d)). However, the introduction of variable time allows to satisfy the demand (Figure 7.12(e)), by reducing the processing time of two batches of product B, and one of product C.

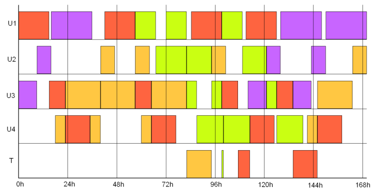
The ZW policy, which is the most restrictive one, does not allow to fulfill two batches of B under the nominal recipe assumption (Figure 7.12(f)), but an additional batch of product C can be produced instead. When assuming variable processing times, the minimum demand is satisfied (Figure 7.12(g)), at the cost of decreasing the reaction times of all the batches of product B, and one batch of products C and D.



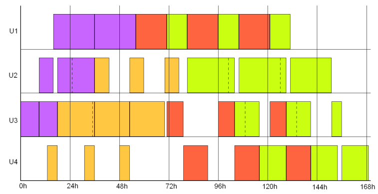
(a) UIS, 896.43 m.u.



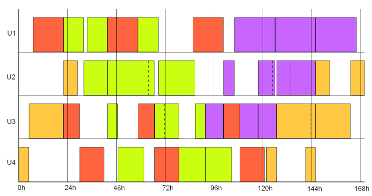
(b) CIS nominal recipe, 616.0 m.u.



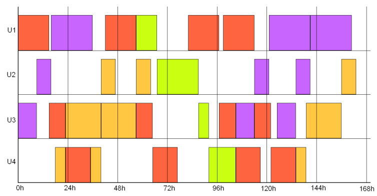
(c) CIS variable recipe, 781.3 m.u.



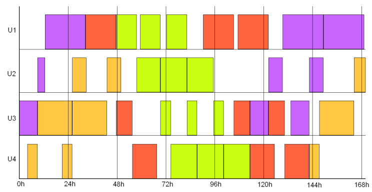
(d) NIS nominal recipe, 531.0 m.u.



(e) NIS variable recipe, 666.1 m.u.



(f) ZW nominal recipe, 391.87 m.u.



(g) ZW variable recipe, 607.0 m.u.

Figure 7.12: Gantt charts and actual profit for Example 2.

As a whole, even though production costs increase when allowing shorter processing times, the total profit increases as well when compared to the nominal recipe, because the total demand can be fully satisfied. Therefore, both the production of the total demand and the inclusion of more batches in a given time horizon justifies the reduction of the processing times, even though operational costs increase. These results apply to processing stages where operating cost has a certain weight in the overall cost associated with the product. In scenarios where the considered operating costs are either in a high or low ratio regarding the overall cost, it should be predicted that extreme solutions with minimum or maximum costs respectively are the optimal ones, and the traditional approach using fixed recipes may be used.

Summarizing, for all storage policies the solution, both the production sequence and the operating conditions, are different from those obtained with independent optimizations at scheduling and control level, and lead to better solutions. The proposed approach leads to better solutions and the economic benefits are noteworthy.

7.5.3 Example 3

The illustrative multiproduct fiber plant presented in Example 1 of Chapter 4 is further studied in this example. Specifically, the first polymerization stage is fully characterized. The process dynamics is thoroughly described in Section D.2. In this case, two free decision variables are considered, namely the initial amount of each of the two monomers of the copolymerization. The total conversion is fixed. Total cost and processing time of the polymerization stage depend on the initial amount of these two monomers in the reactor. The amount of residual monomer should be separated in the distillation column, which implies an additional cost. However, such cost is disregarded because it is orders of magnitude lower than the cost associated with the reaction stage. Thus, total production costs of the dynamic stage have been assessed separated from the rest of the production process. Therefore, Table 7.10 contains the value of the batch benefit disregarding the reaction stage cost, which is presented for each fiber in Figure 7.13.

Table 7.10: Batch benefit disregarding operating costs of the dynamic stage in Example 3.

Fiber	Profit [m.u./batch]
A	14.75
B	10.53
C	8.89

A production of two batches of each product must be fulfilled in a time horizon of 30 hours maximizing total profit. Unfulfilled demand is penalized with 10^4 m.u./ton. Therefore, since demand is highly penalized and in order to appreciate more clearly the influence of process dynamics over the production schedule, this example disregards sequence dependent changeovers.

Five approximation strategies for cost optimization are examined, namely: i) the nominal recipe implementation; ii) linear regression cost approximation; iii) piecewise linear cost approximation; and two algorithmic strategies, iv) based on linear regression, and v) based on the convex envelope for approximating costs. The results obtained using flexible recipes are compared with the nominal ones.

Case i: Nominal recipe. The nominal recipe implementation is derived from optimizing the operation of one single batch of each product, and next applying the

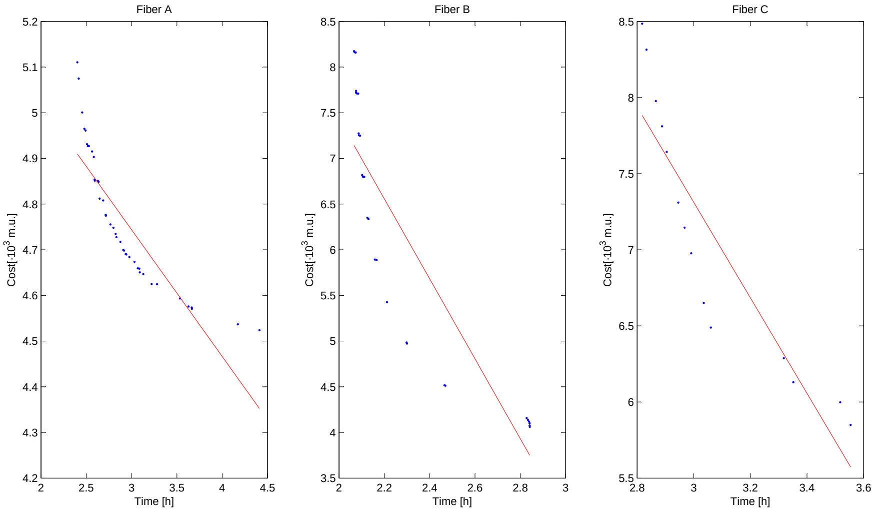


Figure 7.13: Total cost in the reaction stage as a function of time in the multiproduct fiber plant for fibers A to C: real data (blue points) and linear regression (red line, see Table 7.12).

obtained values to any batch of that product. In this case, both the minimization of total cost (Case ia) and the maximization of profitability (Case ib) for the reaction stage of a single batch are considered. Table 7.11 contains the values of the nominal processing times which are almost the same for fibers B and C using both approaches. The reduction of production time of fiber A in the nominal recipe when maximizing profitability (Case ib) could be highly beneficial in specific situations for including additional batches and fulfill products demand, which would be not possible for the nominal recipe considering minimum total cost (Case ia) with larger processing time in the reaction stage.

Cases ii and iii: linear and piecewise linear approximations to total cost. In order to exploit the flexibility of the dynamic stage in the scheduling problem, the relationships between processing time and processing cost are introduced in the scheduling by obtaining the total cost as a function of time (Figure 7.13) for different combinations of the initial amounts of monomers, and applying a Pareto filter to the obtained points. Next, a linear regression of the processing cost is obtained (Table 7.12) between the maximum and the minimum reaction time. Thus, a piecewise linear approximation to the original cost function has been adopted by taking all the points that belong to the Pareto frontier of the time-cost function.

Cases iv and v: iterative regression and convex envelope. In this example, the number of points considered to approximate the linear functions are those resulting from time-cost Pareto frontier. In both cases, a reduction of 60% of the initial time interval is considered at every iteration according to the algorithm presented in Figure 7.3. The end criteria consist of: i) a tolerance between the actual profit and the estimated one of 10^{-3}; ii) no improvement in the actual profit, and iii) a maximum number of 10 iterations.

Table 7.11: Processing times for the three polymer fibers according to the nominal recipes [h] in Example 3.

Fiber	Case ia	Case ib
A	4.412	3.666
B	2.863	2.842
C	3.554	3.554

Table 7.12: Linear regression for total cost approximation in the reaction stage for the multiproduct fiber plant.

Fiber	Linear Cost [m.u./batch]
A	$5.5758 - 0.2773t$
B	$16.1845 - 4.3761t$
C	$16.7247 - 3.1376t$

Results. Figure 7.14 presents the Gantt charts and actual profit for the previous five cases. It shows that there is significant discrepancy between the values of actual profit obtained with the nominal recipe (Figures 7.14(a) and 7.14(b)) compared to the profit of the schedules using adjusted processing time values based on variable recipes (Figures 7.14(c) and 7.14(d)).

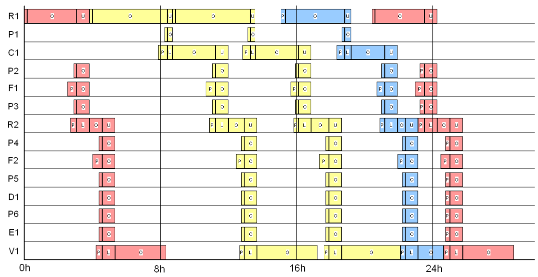
None of the nominal recipe approaches succeeds in providing with the optimal estimation of the actual optimal processing times for the scheduling problem, because it is not possible to fulfill the total demand in the fixed time horizon. Specifically, when the nominal recipes are applied, one batch of fiber C cannot be fulfilled. As a result, significantly lower profits are obtained. Additionally, it is observed that using the nominal recipe based on profitability maximization results in slightly worse actual profit ($21.34 \cdot 10^3$ m.u.) compared to the nominal recipe based on total cost minimization ($21.39 \cdot 10^3$ m.u.). The reason of such difference stems from the fact that in no case the full demand can be accomplished and the idle time within the production time is better used applying total cost minimization.

When variable processing times are allowed, the duration of the dynamic stage is adjusted so that the total production can be fulfilled at the minimum cost. The strategies with the piecewise linear function and the iterative strategy using a linear regression around the previous points lead to identical results in terms of batch times and actual profit. On the other hand, the results with linear regression and the iterative strategy with convex envelope around the previous solution times, result in a very similar objective function value, slightly worse than the previous, basically due to different processing times of the batches. Anyhow, those times and their corresponding optimal conditions are rather different from those obtained using the nominal recipe.

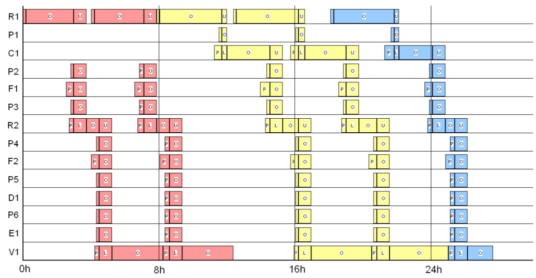
Figures 7.15 and 7.16 show the evolution of the adjusted and the actual profit for the iterative strategies. The algorithmic procedure considering linear regression meets the end criteria after 11 iterations, whereas the one considering the convex envelope reaches in 3 iterations. Furthermore, Figures 7.17 and 7.18 present the evolution along the iterations of the processing times of the different batches for both cases.

7.6 Final remarks

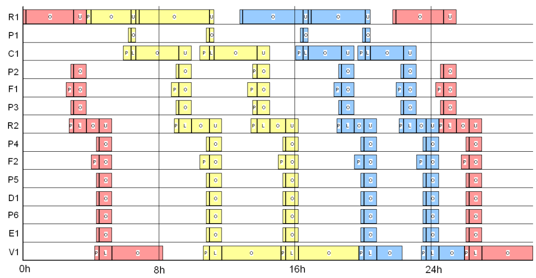
The integration of scheduling and control functions leads to overall operational improvements in the enterprise structure, but several difficulties must be overcome



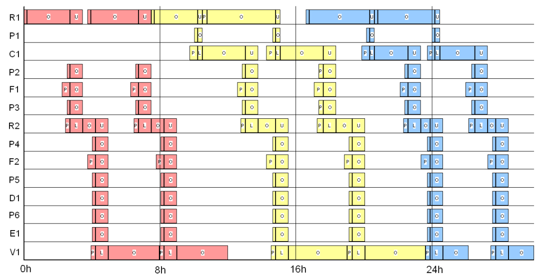
(a) Case ia: Nominal recipe minimizing total cost in the reaction stage, $21.39 \cdot 10^3$ m.u.



(b) Case ib: Nominal recipe maximizing profitability for the reaction stage, $21.34 \cdot 10^3$ m.u.



(c) Cases ii and v: Linear regression and iterative convex envelope of total costs, $38.33 \cdot 10^3$ m.u.



(d) Cases iii and iv: Piecewise linear function and iterative linear regression function of total costs, $38.40 \cdot 10^3$ m.u.

Figure 7.14: Gantt charts and actual profit for cases i-v in Example 3. (yellow, red and blue: fibers A, B and C, respectively)

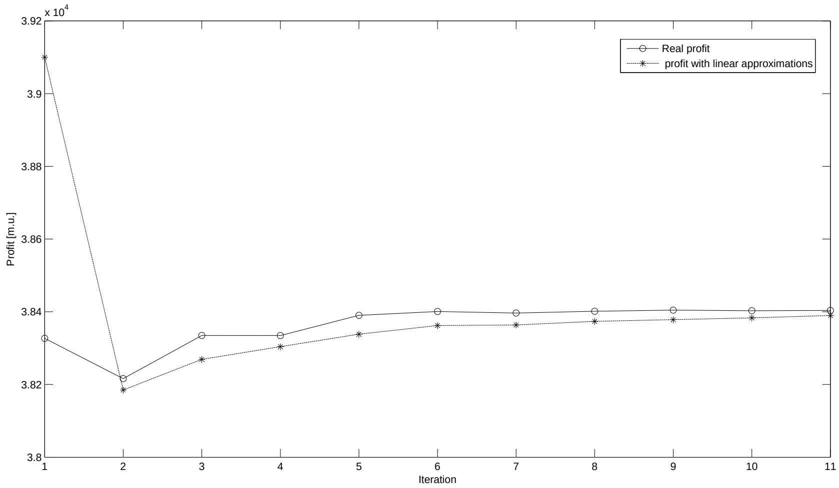


Figure 7.15: Evolution of adjusted profit and actual profit in the algorithmic procedure considering linear regressions for each batch in Example 3, case iv.

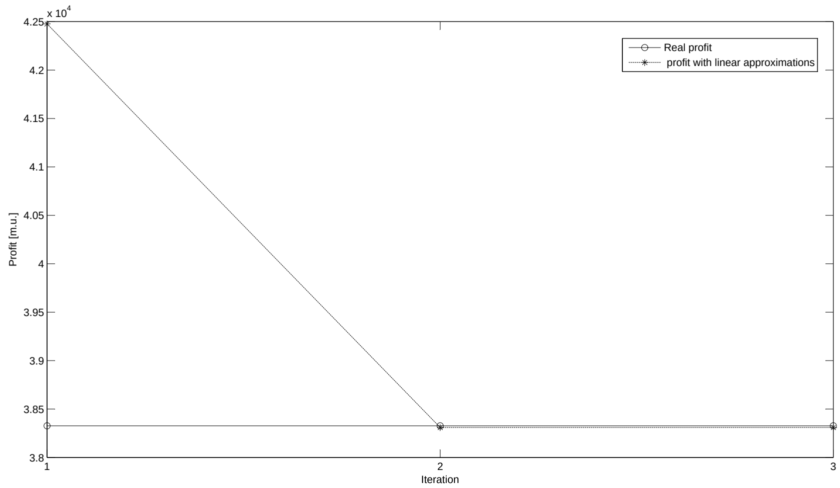


Figure 7.16: Evolution of adjusted profit and actual profit in the algorithmic procedure considering convex envelopes for each batch in Example 3, case v.

first. This chapter presents an indirect approach to manage typical control decisions at the scheduling level through the management and characterization of process recipe variability, as well as the evaluation of the potential improvements that can be achieved. Specifically, an economic objective function reflects the influence that operational variability, expressed in terms of time, has over the process performance. Although this should be considered as an indirect approach, it is demonstrated that by introducing recipe variability at the scheduling level, plant economic performance improves significantly regarding its value obtained considering fixed recipe conditions.

Two basic cost function approximation strategies have been considered. On the one hand, the approximation of costs as a direct function of time in the whole time interval,

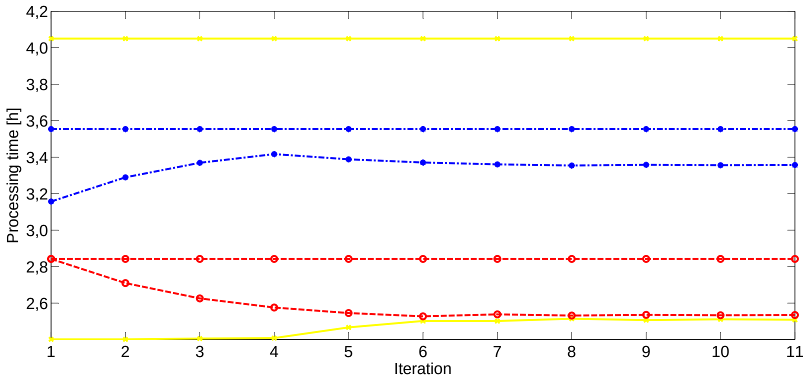


Figure 7.17: Evolution of batch processing times in the algorithmic procedure considering linear regressions for each batch in Example 3, case iv. (yellow, red and blue: fibers A, B and C, respectively)

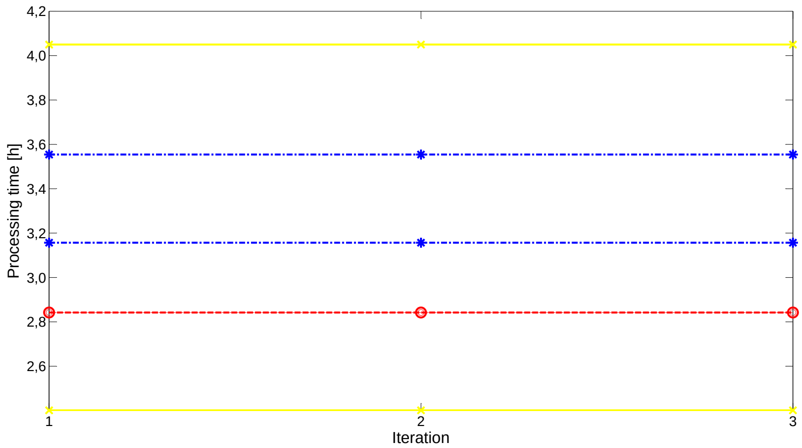


Figure 7.18: Evolution of batch processing times in the algorithmic procedure considering convex envelopes for each batch in Example 3, case v. (yellow, red and blue: fibers A, B and C, respectively)

namely by linear regression, quadratic regression and piecewise linear functions. On the other hand, the use of an heuristic approach, which iteratively reduces the time range for linear based cost functions, in order to successfully improve the function adjustment.

The heuristic approaches are better suited to large scale problems, since they may achieve good results using linear functions at every iteration instead of computationally costly non linear functions (i.e. quadratic regression functions) or highly combinatorial problems (i.e. piecewise linear approximations). Anyhow, it must be pinpointed that the resulting solutions may be not result in the global optimal of the problem.

The results for two multiproduct batch plants and a multipurpose batch plant under different intermediate storage policies have been used to illustrate the potential benefits of the proposed strategies. All cases include a dynamic reactor stage. For the isothermal reactors, the value of temperature influences both energy cost and processing time. For the polymerization plant, the initial amount of monomers in the reactor and the added quantities of monomers affect raw materials and energy cost. Total profit has been maximized in a given time horizon. Therefore, processing times in the reactor stage are not fixed, but obtained as a result of the proposed optimization procedure, and schedules and operating conditions different from the nominal have been found. As a result of introducing processing variability, total profit improved significantly in both cases. Such improvements stem from the additional number of batches that could be processed, the adjustment of operation start-ups and shut-downs, and also from different coordination with the rest of product stages. Basically, the improvement routes from the fact that the whole demand cannot be fulfilled in the time horizon using the most cost efficient production, instead other process conditions can be adopted to meet the production goals. The integration of the resulting simplified model in an overall scheduling approach leads to new decision making trade-offs and consequently, to further optimization opportunities.

On the whole, variable batch-to-batch recipes increase process flexibility and allow better economic results compared to nominal fixed recipes, since now trade-offs between processing times and scheduling actions can be contemplated. These results with constant values of the free decision variables are significant to foresee that greater improvements could be obtained working with variable batch sizes and time variable profiles for the additional degrees of freedom, as it will be presented in the next chapter.

7.7 Nomenclature

Sets and subsets

i	Batches.
p	Products.
r	Element of the piece-wise linear function.
s	Stages.
u	Processing unit.
DS_{ps}	Stages s of product p which are time variable.
U_{ps}	Available units for processing product p at stage s.

Parameters

BP_p	Batch price resulting from the production of a batch of product p.
BS_p	Batch size of product p.
c_p	Specific heat in reaction media of the reaction stage.
CD_p	Unitary cost of unfulfilled demand.
D_p^{min}	Minimum demand of product p.
H	Time horizon.
M	A parameter with a big value.
N_{iter}	Maximum number of iterations of the recursive algorithm.
PE	Unitary energy cost.
PD_p	Minimum demand of product p.
pt_{psu}	Processing time of fixed stage s of product p in unit u.
$SellingPrice_p$	Batch price resulting from the production of a batch of product p.
T_0	Initial temperature of the reaction stage [ºC].
T_{ps}^{max}	Maximum processing time of the variable stage s of product p.
T_{ps}^{min}	Minimum processing time of the variable stage s of product p.
TI_{red}	Percentage of the time interval that is reduced at every iteration of the recursive algorithm.
tol	Tolerance between the profit estimated by the adjusted function and the actual profit.
tt_{pu}	Transfer time from unit u for product p.
X_{Rp}	Conversion in reaction stage of product p.
ΔH_{Rp}	Reaction energy for the generation of product p [kJ/kg].

Continuous variables

AD_p	Positive variable that counts for the unfulfilled demand of product p.
E_{ips}	Energy cost of batch i of product p at dynamic stage s.
EC_p	Energy cost of the production of one batch of product p.
T	Temperature of the reaction stage [ºC].
$t_p^{reaction}$	Processing time in reaction stage of product p .
td_{ips}	Time of variable processing stage s of batch i of product p.
Ts_{ips}	Starting time of stage s of batch i of product p.
Tf_{ips}	Finishing time of stage s of batch i of product p.
Tw_{ips}	Waiting time of stage s of batch i of product p.
z^{Mk}	Makespan.
z^{profit}	Objective function defining profit.
$z^{profitability}$	Objective function defining profitability.

Binary variables

RS_{ir}	Assignment of the piecewise linear function element r to batch i.
W_{ip}	Production of batch i of p.
Y_{ipsu}	Assignment of batch i of product p at stage s to unit u.

Chapter 8

Full Process Models in the Scheduling Problem

8.1 Motivation

The previous chapter demonstrates that considering process dynamics issues in the scheduling problem allows for improvement in overall process performance. However, the control decision variables were fixed along time, and their effect was indirectly regarded in the scheduling problem by means of objective cost functions depending on time. In this chapter the potential of directly including control variables with time varying values and variable batch sizes is explored. Thus, the effects of complete integration of control and scheduling decision levels is assessed herein.

At the process design stage, product batch size, processing conditions as well as processing times are usually established and fixed for plant operation. Nevertheless, production conditions vary from design forecasts, in such a manner that the predicted optimal design conditions are not the best in practice. Consequently, the plant usually operates under non-optimal conditions, but if the process is flexible, its processing conditions may be adapted to actual plant needs. Hence, it is crucial the adoption of models and optimization tools which help to assess the consequences of process integration and the resulting improvement in the plant.

In general, the optimization of process conditions, which are time varying, results in a dynamic optimization problem. Therefore, the inherent complexity of combinatorial scheduling problems is further increased by the adoption of process models for integrating control decisions. Hence, it is important to assess the benefits of additionally increasing the problem complexity for the scheduling problem.

8.2 Introduction

The need for improved models and tools that allow for decision level integration of the scheduling function with the process conditions has been demonstrated in the previous chapter. Specifically, a thorough review in the literature revealed the claims made by many researchers concerning the benefits of such integration. Despite these potential advantages, little work has been actually done toward the full integration of process models with scheduling formulations for batch processes, because of the complexity of the resulting dynamic problem (Harjunkoski *et al.*, 2009). In this chapter, further emphasis is given to fully integrate the process model in the scheduling model, which is referred to as direct approach in this thesis. Initial attempts to such integration are described in Chapter 7, but specific contributions to including process models in the scheduling problem are scarce.

One of the very first works to consider the full process dynamics in the batch scheduling problem was presented by Bhatia and Biegler (1996). They include dynamic models of processing tasks within the design and scheduling formulation for a special kind of batch problems, namely flowshop plants with unlimited intermediate storage and zero wait transfer policies with one unit per stage. In their work, processing decisions are resolved by discretizing the dynamic process models through collocation on finite elements; the authors prove that dynamic process considerations can contribute significantly to increase profitability. Their work was further extended to deal with product and plant uncertainty (Bhatia & Biegler, 1997). On the other hand, Mishra *et al.* (2005) broadly classify the scheduling problem formulation in two categories, namely standard recipe approach and the overall optimization approach. The former defines a recipe beforehand either empirically or by single batch optimization; and next, the scheduling problem is posed on the basis of these standardized recipes. The latter approach directly includes process dynamics in the scheduling problem and restores degrees of freedom, so it can yield a better solution. They compare both approaches for a single product plant and a multiproduct plant. The standard recipes are modeled as polynomials that relate duration and reaction heat to the processed quantities. They prove the superiority of the overall optimization approach in terms of solution quality but they also highlight that one of the major drawbacks is the large size of the resulting problems and the computational difficulty in solving large-scale problems. In addition, the authors point out the influence of the costs structure on the results.

Previous works as well as the ones cited in Chapter 7 either assume a cyclic operation strategy or are limited to specific intermediate storage policies. However, none of them dare to present a global approach for dealing with short-term dynamic scheduling problems. In this chapter, a general approach to short-term scheduling problems is proposed. Specifically, the dynamic optimization problem is discretized by means of a simultaneous discretization method, namely the orthogonal collocation method over finite elements as proposed by previous authors in similar applications (Bhatia & Biegler, 1996; Mishra *et al.*, 2005), which results in a non linear formulation. The combination of the dynamic problem with any general scheduling formulation results in a mixed integer non linear problem. Therefore, this chapter aims at unveiling the benefits of such proposed approach when compared to the traditional fixed recipes and approximations to optimal process performance based on polynomials.

8.3 Problem statement

This chapter considers full process dynamics at the scheduling level, in contrast to fixed nominal processing recipes or time cost varying functions as presented in Chapter 7, so that overall plant performance may be improved by means of control variables of the dynamic stages adaptation. The problem statement defined in the previous Chapter is almost identical, since the process operations planning data, the process dynamics and the economic data are the same. However, an additional goal is considered, namely the batch sizes of each product must be also determined in this chapter. Therefore, the scope of the previous models must be enlarged, as shown in the following section. By including the batch size as a decision variable, the problem flexibility also is enlarged. In contrast, variable batch-to-batch processing times are not included because such additional degree of freedom increases the problem complexity and could prevent the solution of the problem.

Additionally, since the scheduling problem formulation includes the process model, it is expected that the set of control variables, which may be time dependent in this problem, are controlled within the scheduling level.

In this chapter, similar performance metrics to those of the previous chapter are optimized. In this case, an economic function characterizes the overall plant performance at the scheduling level. Such function includes the actual cost of the dynamic stages operation; and so it may result in a better adjustment of such conditions to actual plant needs. In this case, the economic indicator, namely profit (equation 7.1), includes product revenues, operating costs and unfulfilled demand penalty. For designing the fixed recipes, the profitability measure (equation 7.2), is used as objective function.

8.4 Solution procedure

In order to solve the scheduling problem accounting for process dynamics, it is necessary to formalize mathematically the process model which defines its behavior. Next, such dynamic model, which is generally a system of ordinary differential equations, must be discretized in order to be combined with the scheduling formulation, which may yield improved overall solutions. The following subsections describe the details regarding the process dynamic model discretization and the modifications that must be considered for the scheduling problem.

8.4.1 Dynamic model considerations

The very first step consists of establishing the mathematical model that represents the dynamic system, namely a set of differential equations that determine the evolution of the state variables over time as a function of the control variables. As a result, the control variablesbecome decision variables that can be manipulated at the scheduling. Since these variables have an influence over the desired state variables (e.g., final concentration or batch time), it is expected that the integrated solution will improve the economic performance of the plant.

In this thesis, the total discretization method based on Orthogonal Collocation on Finite Elements (OCFE) (Cuthrell & Biegler, 1987, 1989; Biegler, 2010) is adopted to tackle the dynamic optimization of the batch process. This dynamic model is combined with the scheduling formulation, giving rise to a large-scale MINLP, which simultaneously accounts for scheduling and process decisions. The OCFE method discretizes the differential equations at specific points, adjusting the actual values of the function at those points. Such approach leads to a fully open formulation which allows a great deal of sparsity and structure, as well as flexible decomposition strategies to solve the problem efficiently and allows to avoid convergence difficulties of other solvers and sensitivity calculations from the solver. However, efficient NLP solvers are required and a careful formulation of the nonlinear program is required. The fundamentals of this method are described in Chapter 3.

The analysis and discretization of a case study is next presented, consisting of a dynamic model of kinetic system in which two competitive reactions (Reactions 8.1) that take place in a reactor of a batch plant to further clarify the application of the OCFE method in scheduling problems.

$$\begin{cases} A \rightarrow B \\ A \rightarrow C \end{cases} \qquad \forall p \tag{8.1}$$

The process kinetics is described by means of two differential equations 8.2 and 8.3, which stand for the reaction rates of species A and B. Therefore, the concentrations of both species (state variables) are totally determined by the previous two equations, considering the initial concentrations and the control variable u_t, which is temperature related, time variable and determines heating requirements of the reaction operation. Heating requirements for all products are determined as the integral along time of the control variable related, as expressed by Equation 8.4. Such heating requirement can take part in the objective function. Anyhow, the integrals of the control variables can be used in the process model which is included in the scheduling problem. In general, any number of control variables, which may be time variable, can be considered in the scheduling model.

$$\dot{x}_A^p = -\left(u_p + \alpha_p u_p^{\beta_p}\right) x_A^p \qquad \forall p \tag{8.2}$$

$$\dot{x}_B^p = \alpha_p u_p^{\beta_p} x_A^p \qquad \forall p \tag{8.3}$$

$$heat_p = \int^{[0,time_p]} u_p dt \qquad \forall p \tag{8.4}$$

The previous differential equations are fully discretized in terms of time by means of the orthogonal collocation on finite elements. As a result, the dynamic optimization problem that embeds a set of differential equations is expressed as a NLP. A key point of the application of such a method is time discretization. Since the final time is a free decision variable for batch processing problems, time increments cannot be exactly defined beforehand. In this work, time increments have been chosen to be equally distributed along the integration time. Therefore, their values for each product are a function of the processing time, which is a decision variable, and the total number of finite elements considered, as posed by Equation 8.5.

$$inc_p NF = time_p \qquad \forall p \tag{8.5}$$

Moreover, it is necessary to define the differential equations at the collocation points of the finite elements. In this case, equations 8.2 and 8.3 are transformed into equations 8.6 and 8.7, which stand for the reaction rate for each product.

$$\sum_{c' \in [0,NC]} Ac_{fc'p} \dot{\varphi}_{c'c} = inc_p \left(- \left(uc_{fcp} + \alpha_p uc_{fcp}^{\beta_p} \right) Ac_{fcp} \right) \qquad \forall f, c, p \mid c \in [1, NC] \tag{8.6}$$

$$\sum_{c' \in [0,NC]} Bc_{fc'p} \dot{\varphi}_{c'c} = inc_p \left(\left(\alpha_p uc_{fcp}^{\beta_p} \right) Ac_{fcp} \right) \qquad \forall f, c, p \mid c \in [1, NC] \tag{8.7}$$

According to the OCFE method, the value of a decision variable at the final point of each finite element is a function of its value at the collocation points, which are decision variables as well, and at the beginning of each finite, which is an initial condition for the first finite element and a result of the continuity equations for the other elements. Equations 8.8 and 8.9 represent the species concentrations at the final point of each finite element.

$$\sum_{c' \in [0,NC]} Ac_{fc'p} \varphi_{c'} = A_{fcp} \qquad \forall f, c, p \mid c = NC + 1 \tag{8.8}$$

$$\sum_{c' \in [0,NC]} Bc_{fc'p} \varphi_{c'} = B_{fcp} \qquad \forall f, c, p \mid c = NC + 1 \tag{8.9}$$

Additionally, the continuity condition of the state variables between finite elements must be ensured (Equations 8.10 and 8.11).

$$Ac_{f-1cp} = A_{fc'p} \qquad \forall p, f > 1, c, c' \mid c' = 0, c = NC + 1 \tag{8.10}$$

$$Bc_{f-1cp} = B_{fc'p} \qquad \forall p, f > 1, c, c' \mid c' = 0, c = NC + 1 \tag{8.11}$$

On the other hand, continuity of the control variable u_t is not compulsory between finite elements. For this variable, the value at the initial and final point of each finite element is a function of its values at the collocation points, as defined by Equations 8.12 and 8.13.

$$\sum_{c' \in [1,NC]} uc_{fc'p} \theta_{0c'} = u_{fcp} \qquad \forall p, f, c \mid c = 0 \tag{8.12}$$

$$\sum_{c' \in [1,NC]} uc_{fc'p} \theta_{NC+1c'} = u_{fcp} \qquad \forall p, f, c \mid c = NC + 1 \tag{8.13}$$

Finally, the integration of the control variable u_t over time provides the heating requirement, which can be expressed in terms of Equation 8.14, which makes use of the Gauss quadrature rule.

$$heatc_{fp} = \sum_{f' < f} heatc_{f'p} + 0.5 inc_p \sum_{q \in [1,NQ]} weightsG_q \sum_{c' \in [1,NC]} \theta_{qc'} u_{fc'p} \qquad \forall p, f \tag{8.14}$$

Therefore, Equations 8.5 to 8.14 represent the discretization of the dynamic equations 8.2 to 8.4.

8.4.2 Scheduling considerations

Once the process dynamic model (Equations 8.2 to 8.4) is translated into a NLP (Equations 8.5 to 8.14). This NLP must be then combined with the scheduling formulation, which introduces binary variables related to assignment, sequencing and allocation decisions. The overall problem takes the form of a MINLP problem. The adopted scheduling formulation is based on the general precedence model, which has been also described in Section 7.4. However, in this chapter, variable batch size is introduced and several storage policies are considered.

The number of batches to be produced of each product, which is denoted by the binary variable W_{ip}, taking values of 1 if batch i of product p is performed, and 0 otherwise. Thus, equations 8.15 to 8.20 assign the batch size, processing time and reaction heat to each batch provided the batch is produced ($W_{ip} = 1$) by means of bigM constraints. In the presented case, the final product batch size corresponds to the concentration of species B at the end of the reaction time, that is its value at the last collocation point of the last finite element, Bc_{fcp}.

$$BSB_{ip} \leq Bc_{fcp} + M\,(1 - W_{ip}) \qquad \forall i, p, s, f, c \mid (p, s) \in DS_{ps} \tag{8.15}$$

$$BSB_{ip} \geq Bc_{fcp} - M\,(1 - W_{ip}) \qquad \forall i, p, s, f, c \mid (p, s) \in DS_{ps} \tag{8.16}$$

$$BSB_{ip} \leq M\,(W_{ip}) \qquad \forall i, p, s \mid (p, s) \in DS_{ps} \tag{8.17}$$

$$TB_{ip} \leq time_p + M\,(1 - W_{ip}) \qquad \forall i, p, s \mid (p, s) \in DS_{ps} \tag{8.18}$$

$$TB_{ip} \geq time_p - M\,(1 - W_{ip}) \qquad \forall i, p, s \mid (p, s) \in DS_{ps} \tag{8.19}$$

$$TB_{ip} \leq M\,(W_{ip}) \qquad \forall i, p, s \mid (p, s) \in DS_{ps} \tag{8.20}$$

$$HB_{ip} \leq heatc_{fcp} + M\,(1 - W_{ip}) \qquad \forall i, p, s, f, c \mid (p, s) \in DS_{ps} \tag{8.21}$$

$$HB_{ip} \geq heatc_{fcp} - M\,(1 - W_{ip}) \qquad \forall i, p, s, f, c \mid (p, s) \in DS_{ps} \tag{8.22}$$

$$HB_{ip} \leq M\,(W_{ip}) \qquad \forall i, p, s \mid (p, s) \in DS_{ps} \tag{8.23}$$

Note that these equations avoid nonlinear products of binary variables and continuous variables that would arise in the objective function, time horizon or demand satisfaction constraints.

As mentioned previously in Chapter 7, the proposed model assumes that part of the demand can be left unsatisfied due to limited production capacity of low profitability. Hence, part of the total demand is produced in the plant, while the rest may be not fulfilled. for the latter amount a penalty that is appended to the objective function. Therefore, considering the new continuous variable that stands for the total quantity produced in a given batch (BSB_{ip}), Equation 8.24 enforces the total product demand to be satisfied, either as a result of the production plant (BSB_{ip}) or as additional demand (AD_p).

$$\sum_{i} BSB_{ip} + AD_p = PD_p \tag{8.24}$$

Additionally, for those stages whose time is defined by the dynamic process equations, timing equations similar to those posed in the previous chapter must be considered, namely Equations 7.14 and 7.15. Specifically, td_{pis} corresponds to the variable defined in this section as TB_{ip}.

The objective function accounts for production revenues, operation costs and unfulfilled demand penalization, as stated in Equation 8.25.

$$z^{profit} = \sum_{ip} BSB_{ip} \cdot PRICEp - \sum_{ips \in DS_{ps}} HB_{ip} \cdot costQ - \sum_{p} CostD \cdot AD_p \tag{8.25}$$

As a whole, Equations 8.2 to 8.25 combined with those from the general precedence model described in Chapter 5 (Equations 5.1 to 5.3 and 5.5 to 5.11) and revised in Section 7.4 (Equations 7.4, 7.6 and 7.14 to 7.16), define a general integrated scheduling and process model consisting of a MINLP that is applied to two scheduling problems, highlighting the advantages of such decision level integration.

8.5 Examples

In order to shed light to the benefits of introducing process dynamics at the scheduling level, the optimization of profit along a finite time horizon in two multiproduct batch plants is analyzed using three different approaches, namely: case (i) the traditional approach which optimizes beforehand the batch duration and fixes its value at the scheduling level, that is, the nominal case; case (ii) a time variable approach which relates batch conversion to time and introduces polynomial functions in the scheduling problem relating concentration and heat to batch duration, that is, an indirect approach similar to that presented in Chapter 7; and case (iii) the consideration of the full process dynamics in the scheduling formulation, using orthogonal collocation on finite elements. Even though multipurpose plants could be also tackled with the proposed strategy, this chapter focuses on multiproduct batch plants for maintaining the problem complexity within reasonable limits.

The mathematical formulation for the two examples has been implemented in GAMS interfacing with CPLEX 11.2 and DICOPT to solve the MILP and MINLP models, respectively.

8.5.1 Example 1: Multiproduct single stage batch plant

A batch plant consisting of a single stage reaction for three products, based on the case study presented by Bhatia and Biegler (1996), is considered. A competitive reaction takes place in the reactor, and species B is the desired final product in all cases (Reaction 8.1). The reaction temperature may vary over time. The kinetic parameters, demand of final species B for each product and unitary price are included in Table 8.1. The total time horizon is 12 hours, in which a maximum of 5 batches of each product can be processed in the single available unit. Additionally, the following assumptions are made:

- The batch load consists of 1 ton raw material.

Table 8.1: Kinetic parameters and prices of the products for the small case study.

	Product 1	**Product 2**	**Product 3**
α	2	2	3
β	0.5	0.4	0.5
$Price[m.u./ton]$	3	12	3
$Demand[tonB]$	1.8	1.8	1.8

- There are no sequence dependent set up times, but a set-up time of 0.5 h for each batch is necessary.
- The minimum and maximum reaction times for each product are 0.5 and 3 h. The actual reaction time values are decision variables of the scheduling problem.
- The temperature related measure u_t which is used in the kinetic model of the process may vary between 0.05 and 2 units. Its integration results in the energy consumption required to perform the batch.
- Production costs are related to energy consumption (6 m.u./e.u.).
- Unfulfilled demand is penalized with 1000 m.u./ton.
- All batches of a given product are considered to be performed under the same conditions.

Case i: Nominal recipe. The traditional approach consists of optimizing the reaction stage at the design phase, and using the obtained variable profiles and batch processing times for the scheduling problem. Specifically, the objective at the design stage consists of maximizing the profitability, considering a batch preparation time of 0.5h for each product. The optimal processing times, the final concentration and reaction heat for all products are reported in Table 8.2.

Table 8.2: Fixed recipes for the reaction stage of the different products.

	Product 1	Product 2	Product 3
Final concentration A [-]	0.589	0.403	0.497
Final concentration B [-]	0.360	0.474	0.450
Batch time [h]	0.818	0.548	0.586
Reaction heat $\cdot 10^{-2}$ [e.u.]	1.743	6.068	1.993

Case ii: Indirect approach. The second approach consists of solving the problem according to a simplified version of the dynamic model. Specifically, the temperature profile for a single batch of each product is optimized maximizing total profit at discrete processing times. Figure 8.1 presents the optimal values of the final concentration of species B and the corresponding reaction heat consumption over time.

The relationships between final concentration and reaction heat with time are approximated as third and first order polynomials respectively (Equations 8.26 and 8.27).

$$x_{Bp}(t) = C_3^p t^3 + C_2^p t^2 + C_1^p t + C_0^p \tag{8.26}$$

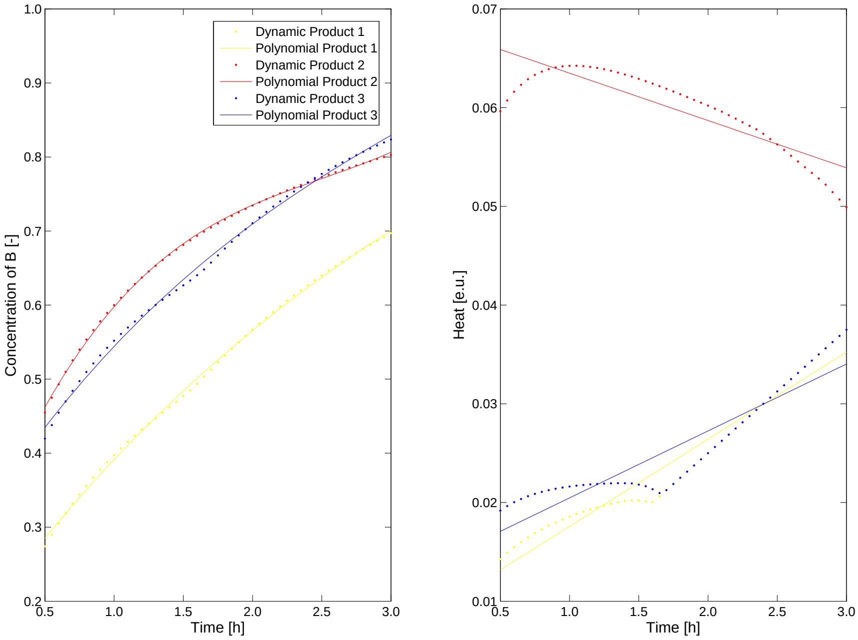


Figure 8.1: Optimal concentration and heat profiles for the three products along time.

$$heat_p\,(t) = H_1^p t + H_0^p \tag{8.27}$$

Tables 8.3 and 8.4 contain the adjusted parameters for the previous polynomials, which are introduced at the scheduling level as an approximation to the dynamic model in order to obtain the batch processing times. Therefore, the heat consumption and concentration values used in the scheduling level are approximations to the actual optimal values, owing to the error in the polynomials adjustment. The batch processing times obtained in the scheduling optimization are used in a following step in order to assess the actual values of the final product concentration, heat consumption along with the objective function.

Table 8.3: Adjusted parameters for the polynomials that describe the concentration of B along time [h] for each product.

	Product 1	Product 2	Product 3
$C_3^p(\cdot 10^{-3})$	2.013	22.33	5.179
$C_2^p(\cdot 10^{-2})$	-3.173	-16.71	-5.389
$C_1^p(\cdot 10^{-1})$	2.556	4.825	2.908
$C_0^p(\cdot 10^{-1})$	1.654	2.598	3.021

Table 8.4: Adjusted parameters for the linealisation of reaction heat along time [h].

	Product 1	Product 2	Product 3
$H_1^p(\cdot 10^{-3})$	8.821	-4.798	6.782
$H_0^p(\cdot 10^{-2})$	0.877	6.829	1.369

Case iii: Direct approach. The third approach consists of introducing the whole process dynamics into the scheduling problem. First, the adequacy of the method for introducing process dynamics must be checked, and the parameters must be tuned.

In this example, it has been observed that for the orthogonal collocation (OC) on finite elements (FE) method, two collocation points are adequate to adjust the state variables profile. Both the Legendre and Radau roots are considered as collocation points. However, Legendre roots adjust better than Radau roots for this particular case. In order to decide the number of finite elements, several alternatives have been tried using different energy costs and different number of finite elements, namely 4, 12 and 20. The temperature profiles obtained with the orthogonal collocation method have been introduced in Matlab, to obtain the corresponding state variable profiles using the Runge-Kutta (RK) integration. Next, the state variables and reaction heat after 2 hours of reaction time for different energy costs, as well as the average distance between both integration methods at the collocation points have been compared.

Table 8.5 contains values of the final concentration of species B and the reaction heating requirements for the profit optimization considering a two hour period and an energy cost of 6 m.u./e.u. using orthogonal collocation, and the corresponding values of the integration by the Runge Kutta method and the temperature profile obtained at the optimization stage for the three possible products. Additionally, Table 8.6 presents the average distance between the results of tFrom these numerical results, it can be concluded that the orthogonal collocation method accurately describes the behavior of this dynamic process. Increasing the number of finite elements improves the accuracy of the approximation at the expense of increasing the computational complexity. In this case, it has been found a good compromise between accuracy and computational tractability using 4 FE (Table 8.6).

Figure 8.2 shows the proximity between the orthogonal collocation points and the Runge Kutta integration results regarding the concentration profiles using two collocation points, four finite elements and different energy costs for Product 1. Results for Products 2 and 3 present the same behavior.

Table 8.5: Values of the final concentration of species B, reaction heat for the optimization along a two hour period considering an energy cost of 6 m.u./e.u. using orthogonal collocation, and the corresponding values of the integration using Runge Kutta using the obtained temperature profile.

		Final Concentration B			Heating requirement [e.u.]		
		Product 1	Product 2	Product 3	Product 1	Product 2	Product 3
4 FE	OC	0.66075	0.77687	0.77807	0.1920	0.3230	0.1674
	RK	0.66079	0.77697	0.77816	0.1921	0.3236	0.1675
12 FE	OC	0.66076	0.77687	0.77810	0.1920	0.3229	0.1674
	RK	0.66076	0.77688	0.77811	0.1921	0.3229	0.1674
20 FE	OC	0.66076	0.77687	0.77810	0.1920	0.3229	0.1674
	RK	0.66076	0.77685	0.77811	0.1920	0.3229	0.1674

Results. The results of the scheduling optimization for the three previously described cases are reported in Table 8.7. In the first case, global optimality of the final solution is guaranteed, whereas in the two latter cases is not guaranteed, since standard gradient-based methods are likely to fall in local optima due to the presence of non-convexities.The results highlight the benefits of including process variability in

Table 8.6: Average distance in concentration of species B at collocation points comparing the orthogonal collocation on finite elements and the Runge Kutta integration considering an energy cost of 6 m.u./e.u. along a time period of 2h for different number of finite elements (FE).

Number of FE	Product 1	Product 2	Product 3
4	$5.132{\cdot}10^{-5}$	$5.004{\cdot}10^{-5}$	$1.397{\cdot}10^{-4}$
12	$1.103{\cdot}10^{-6}$	$2.144{\cdot}10^{-6}$	$3.018{\cdot}10^{-6}$
20	$1.844{\cdot}10^{-7}$	$2.171{\cdot}10^{-6}$	$5.050{\cdot}10^{-7}$

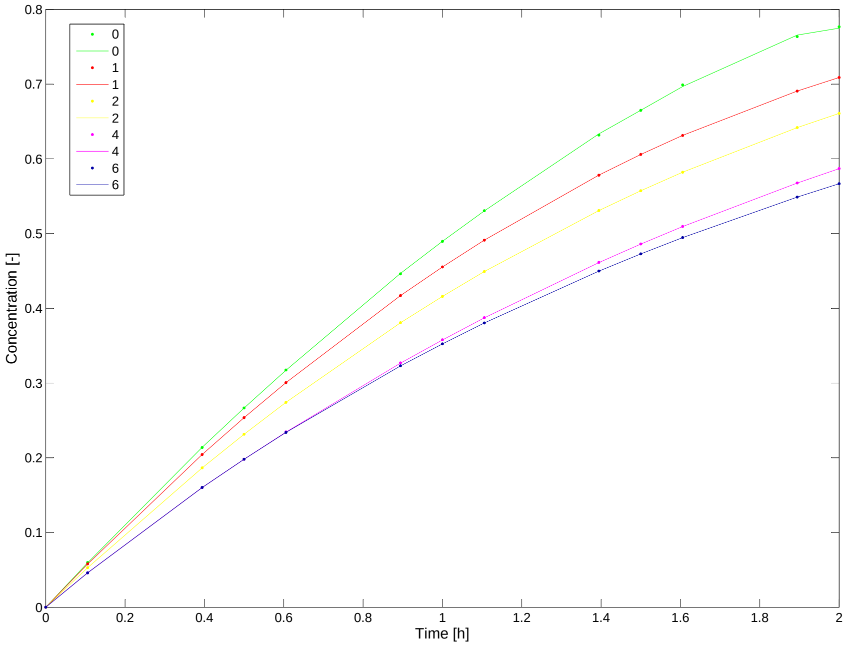


Figure 8.2: Results for the optimization of total profit using different energy costs using orthogonal collocation on finite elements (dot points) and comparison to the values obtained using Runge-Kutta integration (line).

the scheduling problem.

In the first case, since batch processing times are fixed from a previous optimization stage, the scheduling problem only handles the number of batches of each product that are produced. As a result, the demand is not fully covered and the total profit is the worst of the three cases.

In the second case, the batch duration is not fixed, but the amount of product obtained at every time is given by an initial optimization. Therefore, both the number of batches and their processing times are decided in the scheduling optimization, but the demand cannot be fully fulfilled either.

Finally, the inclusion of process dynamics in the scheduling problem produces a solution that fully covers the total demand. This is accomplished at the expense of increasing the heat consumption in each time point. These results are explained by the large penalties assigned to the uncovered demand, which force the model to increase heat consumption in order to expedite the reaction and attain the production targets. Note that the number of batches and the batch processing times differ from one case

Table 8.7: Scheduling results for the three products in Example 1, where case (i) stands for fixed recipe, case (ii) for polynomial adjustment and case (iii) for orthogonal collocation on finite elements.

	z^{profit}	N. batches			Batch time [h]			Concentr. of B			Add. Demand [ton]		
Product	[m.u.]	1	2	3	1	2	3	1	2	3	1	2	3
Case (i)	-1051.7	2	4	4	0.818	0.548	0.586	0.360	0.474	0.450	1.080	0.000	0.000
Case (ii)	-691.1	4	4	4	0.500	0.500	0.500	0.285	0.462	0.435	0.658	0.000	0.061
Case (iii)	18.1	3	3	3	0.999	0.945	0.557	0.600	0.600	0.600	0.000	0.000	0.000

to another. It is also clearly illustrated how the additional process flexibility leads to better overall economic performance.

8.5.2 Example 2: Multiproduct multistage batch plant

This example tackles a multiproduct three stage batch plant, whose first stage corresponds to the reaction process described in the previous example, and the other two processing stages have fixed processing times as shown in Table 8.8. Two different time horizons and demand scenarios are considered for analyzing the effect of including process dynamics in the scheduling problem (Table 8.9) under unlimited and zero wait intermediate storage policies. The same assumptions as in Example 1 are considered.

Table 8.8: Processing times [h] for stages 2 and 3 of the multiproduct plant in Example 2.

	Stage	
Product	2	3
1	1.9	1.3
2	1.5	2.4
3	1.4	1.8

Table 8.9: Demand and time horizon for the two scenarios in Example 2.

	Demand [ton]	
Product	Scenario 1 (sc1) [20h]	Scenario 2 (sc2) [40h]
1	2.0	4.0
2	2.0	4.0
3	1.5	3.0

Results. The three alternative solution schemes previously described, namely the nominal recipe, the indirect and the direct approaches, are studied in two scenarios for estimating the processing time of the dynamic stage, and the effect of multiple stages is additionally analyzed. Tables 8.10 and 8.11 contain the optimization results for the two scenarios, and Figure 8.3 presents the Gantt charts for scenario 2.

Case (i) is the worst approximation to the dynamic stage processing time in both scenarios. In such case, the processing time, the final concentration and the heat consumption are fixed beforehand. As a result, the scheduling problem only deals with the sequencing and timing of the product batches in order to fulfill the demand as much as possible, since production costs related to energy consumption cannot be modified in this case. Thus, the total demand cannot be fulfilled in any scenario, and the production makespan does not match the total time horizon for this case, as can be observed in Figure 8.3.

If processing times of the dynamic stage were not fixed (as in cases (ii) and (iii)), production costs regarding heat consumption could be reduced by extending the reaction time(note that the same conversion can be reached with less heat consumption at the expense of increasing the processing time). In the first scenario (sc1), the same number of batches of each product is assigned for both storage policies. In contrast, in the second scenario (sc2), the zero wait policy, which is more restrictive than the unlimited intermediate storage policy, , produces 1 batch less of product 1, which worsens the total profit (Table 8.11). This clearly illustrates the main disadvantage of fixing processing times at the design step and using them in the scheduling problem. Specifically, the batch processing time of the dynamic stage is decoupled from the rest of stages. Consequently, the idle production time of the dynamic stage is neglected, loosing opportunities for possible total cost reductions.

Both cases (ii) and (iii) perform better than case (i), basically because they include process dynamics in the scheduling problem. Specifically, case (ii) results from an initial optimization design stage, which provides the optimal concentration for each processing time, in terms of energy production costs and revenues. Such optimization allows for a time dependent concentration and cost that are input data to the scheduling problem. In contrast, case (iii) includes the whole dynamic process description in the scheduling formulation, in such a manner that final concentration and energy consumption are decided at such decision level. Therefore, the flexibility of case (iii) is much higher than in case (ii). In scenario 1, both cases result in the same number of batches for each product, namely 3 batches of product 1 and 2, and 2 batches of product 3 (Table 8.10). However, the production times for the latter two products is different, with case (iii) being the only one that fully fulfills the total demand. The two cases mainly differ in the energy consumption profile. Case (iii) allows for a higher energy consumption, which results in a higher conversion for the same production time, at the expense of increasing the cost. Comparing case (iii) with case (ii), it is observed a slight increase in the production time of the dynamic stage of product 2 and a decrease in the production of product 3. The final concentration of the desired species increases in both products and the demand is totally fulfilled.

For scenario 2 (sc2), the limited intermediate storage policy restricts the overall batch assignment for cases (ii) and (iii); modifying the processing times of the dynamic stage (Table 8.11) should be modified. In both cases, the total number of batches of products 2 and 3 is reduced. In case (ii), one batch less of products 2 and 3 is produced, which penalizes the fulfilled demand in spite of the increase in production time.

In contrast, for case (iii) two batches of product 3 are eliminated in the zero wait intermediate storage policy. In the UIS policy, the production time of the dynamic stage of product 2 is reduced (from 2.120h to 1.910h), and the corresponding time of product 3 is notably increased (from 0.614h to 1.484h), but the total demand is fulfilled using both storage policies, which avoids the high penalization stemming from unaccomplished demand. The main reason of the successful results of case (iii) lie in the integration of control decisions in the scheduling level.Specifically, by including variable control profiles at the scheduling level, the relationship between state and control variable values are subject to optimization. As a result, the model is provided with more flexibility at a higher decision level (i.e., scheduling), which increases the overall process performance.

Table 8.10: Scheduling results for Example 2 considering different storage policies for scenario 1, where case (i) stands for fixed recipe, case (ii) for polynomial adjustment and case (iii) for orthogonal collocation on finite elements.

Storage policy		z^{profit} [m.u.]	N. batches			Batch time [h]			Concentration of B			Additional Demand [ton]		
			1	2	3	1	2	3	1	2	3	1	2	3
UIS	Case (i)	-1867.9	1	4	3	0.818	0.548	0.586	0.360	0.474	0.450	1.640	0.104	0.150
	Case (ii)	-556.3	3	3	2	1.900	1.319	1.572	0.550	0.657	0.646	0.349	0.030	0.208
	Case (iii)	24.6	3	3	2	1.900	1.460	1.360	0.667	0.670	0.750	0.000	0.000	0.000
ZW	Case (i)	-1867.9	1	4	3	0.818	0.548	0.586	0.360	0.474	0.450	1.640	0.104	0.150
	Case (ii)	-816.6	3	3	2	1.500	1.400	1.300	0.484	0.669	0.600	0.548	0.000	0.299
	Case (iii)	22.2	3	3	2	1.500	1.466	1.234	0.667	0.671	0.750	0.000	0.000	0.000

Table 8.11: Scheduling results for Example 2 considering different storage policies for scenario 2, where case (i) stands for fixed recipe, case (ii) for polynomial adjustment and case (iii) for orthogonal collocation on finite elements.

Storage policy		z^{profit} [m.u.]	N. batches			Batch time [h]			Concentration of B			Additional Demand [ton]		
			1	2	3	1	2	3	1	2	3	1	2	3
UIS	Case (i)	-3013.7	4	8	6	0.818	0.548	0.586	0.360	0.474	0.450	2.560	0.208	0.300
	Case (ii)	-140.9	6	6	6	2.463	1.383	0.787	0.632	0.667	0.500	0.205	0.000	0.000
	Case (iii)	56.6	6	6	6	1.900	2.120	0.614	0.667	0.738	0.500	0.000	0.000	0.000
ZW	Case (i)	-3374.8	3	8	6	0.818	0.548	0.586	0.360	0.474	0.450	2.920	0.208	0.300
	Case (ii)	-852.6	6	5	5	2.067	1.900	1.298	0.576	0.727	0.600	0.545	0.367	0.000
	Case (iii)	53.8	6	6	4	1.900	1.910	1.484	0.667	0.720	0.750	0.000	0.000	0.000

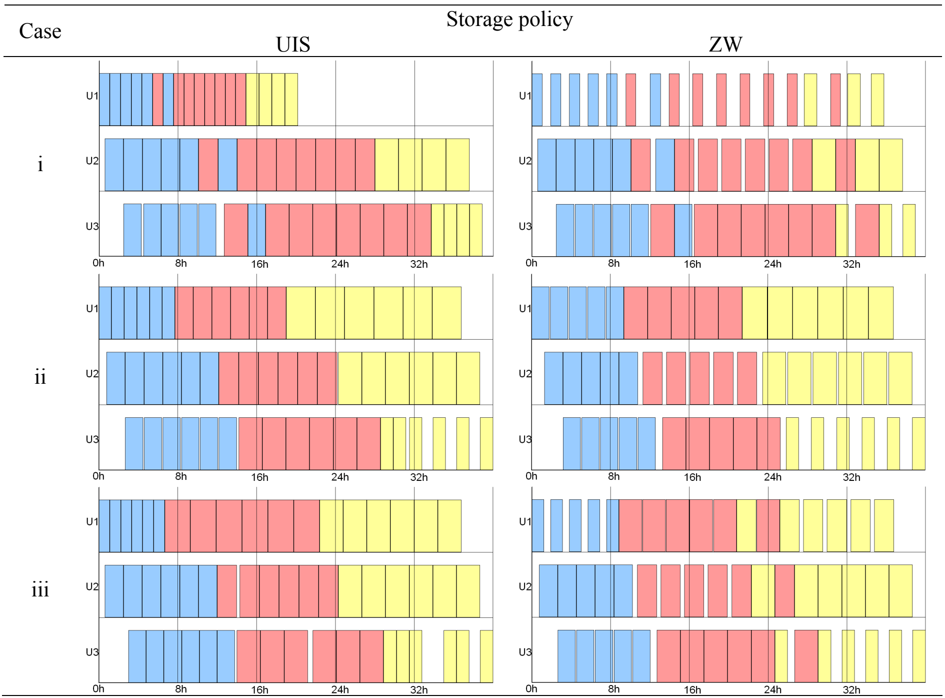


Figure 8.3: Gantt Charts obtained for scenario 2 in Example 2. (Product 1 in yellow, Product 2 in red, and Product 3 in blue)

8.6 Final remarks

The relevance of integrated models for short-term scheduling has been illustrated in this chapter.

The traditional fixed recipes are adequate for those cases in which batch conditions must be carefully preserved and batches must not be altered. However, such approach does not allow the process to adapt to changing environment conditions and tight demand constraints. In contrast, the introduction of process conditions in the scheduling problem yields better results. Basically, processing times can be adequately adapted to actual units availability, time limitations and demand specifications.

The way in which process conditions are introduced in the scheduling problem affects the final results. On the one hand, a design based optimization can be done beforehand, so that the relationship between state variables and time can be derived and included in the scheduling optimization method. An alternative approach is to combine the whole process model with the scheduling formulation so that the of all decisions is performed simultaneously. The former approach is somehow limited as compared to the latter, since control variables are fixed at the scheduling level, and there is no possibility to adapt process conditions to actual plant needs. It has been clearly shown that total integration of between process and scheduling decision variables provides more flexibility and leads to larger benefits.

As a whole, the integration of process dynamics in scheduling represents a generalization of this problem, and eventually leads to better results at the expense of increasing the problem complexity. For this reason, this approach should be preferred for those cases where a considerable potential for improvement is identified.

Further efforts should focus on the benefits of including batches of the same product with different durations, and particularly on studying the effect of various dynamic stages and different cost structures.

8.7 Nomenclature

Sets and subsets

c	Collocation points.
f	Finite elements.
i	Batches.
p	Products.
q	Gauss quadrature collocation points.
s	Stages.
DS_{ps}	Stages s of product p which are variable.

Parameters

$Acini_p$	Initial concentration of raw material for product p.
α_p	Kinetic parameter.
$Bcini_p$	Initial concentration of final product for product p.
β_p	Kinetic parameter.
BS_p	Batch size of product p.
C_k^p	Adjusted coefficients for concentration as a function of time.
$COSTQ$	Energy unitary cost [m.u./e.u.].
$COSTD$	Penalty for additional demand [m.u./ton.].
H_k^p	Adjusted coefficients for reaction heat as a function of time.
M	A parameter with a big value.
NC	Number of collocation points.
NF	Number of finite elements.
NQ	Number of quadrature points.
PD_p	Demand of product p.
$PRICE_p$	Unitary selling price of product p.
θ_{0c}	Multiplier values for the control variables at the collocation points at the initial point of the finite element.
θ_{NC+1c}	Multiplier values for the control variables at the collocation points at the final point of the finite element.
ϕ_c	Multiplier values for the state variables at the collocation points .
$\dot{\phi}_{cc'}$	Multipliers values for the state variables at the collocation points for the derivatives collocation.
$rootsL_c$	Lagrange roots values.
$\theta_{qc'}$	Multiplier values for the control variables at the collocation points at the Gauss quadrature points of the finite element.
$weightsG_q$	Gauss quadrature weights.

Continuous variables

Ac_{fcp}	Concentration of product p in element A at collocation point c of finite element f.
AD_p	Positive variable that counts for the unfulfilled demand of product p.
B_p	Batch size for product p.
Bc_{fcp}	Concentration of product p in element B at collocation point c.
BSB_{ip}	Batch size of batch h for product p. It has value if W_{ip} is 1.

HB_{ip}	Heat consumption of batch bp for product p. It has value if W_{ip} is 1.
$heat_p$	Heat for producing one batch of p.
$heatc_{fp}$	Heat consumed to produce product p at the end of finite element f.
inc_p	Time increment for product p.
TB_{ip}	Time of batch h for product p. It has value if W_{ip} is 1.
$time_p$	Time to produce product p.
$x_{Bp}(t)$	Concentration of B as a function of time, from the polynomial adjustment.
$\dot{x}_A^p$	Reaction rate of A for product p.
$\dot{x}_B^p$	Reaction rate of B for product p.
x_A^p	Concentration of B profile for product p.
x_B^p	Concentration of B profile for product p.
u_p	Temperature profile for product p.
uc_{fcp}	Temperature of reactor for reaction of product p at collocation point c of the finite element f.
z^{profit}	Objective function defining profit.
$z^{profitability}$	Objective function defining profitability.

Binary variables

W_{ip}	Production of batch i of p.

Part IV

Functional Integration of the Scheduling Problem

Chapter 9

Environmental and Economic Issues in the Scheduling problem

9.1 Motivation

The integration of decision making along the hierarchical structure of the enterprise entails the simultaneous consideration of multiple and often conflicting criteria, which can be regarded as a functional integration of the decision levels. In process scheduling, the consideration of different objectives brings forth important production trade-offs which should be assessed in order to reach overall improved solutions. The common economic goals are expressed in terms of plant profitability and productivity, whereas environmental objectives are evaluated by means of specific metrics.

An important operational issue which has a high impact over both economic and environmental results is product changeover. In general, the process of converting a line or equipment from running one product batch to another, i.e. product changeover, is time consuming and it may involve a variety of operations such as cleaning or unit configuration. One significant issue to be considered when product changeover occurs is concerned with cleaning operations, that may be regularly performed between two consecutive batches for the sake of product quality or plant safety. In addition, their environmental impact and economic cost may vary largely depending on the cleaning technique. Thus the consideration of multiple changeover possibilities increases the number of production schedules to be considered, giving rise to eventual trade-offs. Precisely, this chapter aims at gaining insight into those trade-offs in batch process scheduling when alternative methods for product changeover are available.

This chapter introduces a novel approach for scheduling that accounts for the simultaneous consideration of economic and environmental concerns by using multiobjective optimization and life cycle assessment (LCA) principles. The modeling approach leads to complex formulations that require high computational effort even for small instances, which seriously compromises its practical applicability to day-to-day operation. Hence, the modeling framework is complemented by an efficient multiobjective hybrid optimization solution method based on the combined use of evolutionary algorithms and local search. Furthermore, the use of different metrics is investigated to select a possible compromise between the criteria considered in the analysis based on the distance to the utopian solution (i.e., the one whose objective function values are all optimal). Thus, this chapter provides a deeper insight into the selection of metrics for the environmental and economic assessment of schedule and the inherent trade-offs arising between them.

9.2 Introduction

Process industry faces increasing environmental, social and economic requirements which entail complex decision making. Specifically, process scheduling, which is important for the maximization of the production facility utilization specially in batch processes (Korovessi & Linninger, 2006), should cope with a wide variety of criteria to obtain good schedules according to the decision maker's preferences. In this respect, the consideration of multiple criteria decision making (MCDM) provides the path to deal with complex problems involving multiple and conflicting objectives. As a result, a set of compromise solutions, known as Pareto solutions (Wiecek *et al.*, 2008), is usually obtained; from them, the decision maker should choose the most suitable.

In the scheduling problem, the objective function depends on the decision maker criteria, which are based both on his/her experience and the nature of the problem.. The definition of a universal objective function for all scheduling problems is not possible. Several possible objective functions and their scope presented in Chapter 4 are compared and discussed along this chapter. Along with the obvious and traditional economic objective functions, such as plant productivity and profit, environmental metrics related to the production process are gaining importance in production decisions. Thus, makespan is also considered as a process wide resource usage efficiency metric.

Regarding the increasing environmental concerns in chemical industry, more accurate approaches to assess process sustainability are required. Several authors highlight the importance of considering life-cycle assessment of production processes at process synthesis, product design and its integration with processing (Grossmann, 2004; Barbosa-Povoa, 2007).

Therefore, Stefanis *et al.* (1997) propose a methodology that embeds principles from life cycle assessment (LCA) in order to incorporate environmental considerations in the optimal design and scheduling of batch and semi-continuous processes. Process economics and pollution metrics are adopted as design objectives in a multiobjetive formulation.

A combinatorial process synthesis is proposed by Chakraborty and Linninger (2002); Chakraborty *et al.* (2003) using multiobjective goal programming under economic and environmental criteria. The decision variables are operational variables, which depend on the design superstructure being optimized, and the presented case study addresses the design of plant-wide waste treatment facilities related to the batch industry. The economic function beholds operating cost and the environmental function uses the waste reduction algorithm (WAR (Young & Cabezas, 1999; Cabezas *et al.*, 1999)).

Dietz *et al.* (2006) define a multicriteria design framework for multi-product batch plants, which aims at minimizing both investment costs and environmental impact. The problem is solved through a multi objective genetic algorithm (moGA), and a discrete event simulation environment is used to solve the scheduling and planning problem level in the design process. Waste minimization, material recovery and utilities rationalization have been mainly dealt as integral parts at the design stage of batch plants (Yao & Yuan, 2000; Barbosa-Povoa, 2007; Melnyk *et al.*, 2001).

Once the plant design is fixed, process operation decisions, i.e. scheduling related, are the only ones subject to modifications. By appropriately modifying these decisions, it is possible to obtain significant economic and environmental savings. It is important to note that the combinatorial nature of the scheduling problem poses serious computational difficulties. The consideration of more than one objective in a multiobjective optimization framework further increases the problem complexity. Therefore, global optimal solutions for multiobjective scheduling problems can only be obtained for models of limited complexity using the computational tools available nowadays.

Several authors have explored the use of mathematical programming for the simultaneous optimization of environmental and economic criteria. Song *et al.* (2002) presented a MILP formulation for the scheduling of a refinery process taking into account the environmental impact. The ϵ-constraint method is used to obtain a set of Pareto solutions in which the global environmental impact is quantified by means of the critical surface-time 95 (CST95) assessment methodology. Berlin *et al.* (2007) consider a case study of the dairy industry, where the production sequencing affects the environmental impact from a life-cycle perspective. They developed a heuristic method to minimize production waste based on production rules. Their methodology is further applied by Berlin and Sonesson (2008) to a case study with two dairy products. The authors conclude that the environmental impact of processing cultured milk products can be greatly reduced by adopting sequences with fewer changes of products. Park *et al.* (2007) present a goal constrained programming (GCP) algorithm for the multiobjective optimization with priority for the scheduling of cutting papers, which produced various optimal schedule sets.

As previously mentioned, multiobjective formulations in the context of scheduling models may lead to large computational burdens. The application of evolutionary optimization represents an alternative to efficiently handle these formulations. The computational savings may be obtained, however, at the expense of sacrificing global optimality. Many works have used different evolutionary algorithms for multiobjective optimization. A review of this method can be found in the work by Coello-Coello and Landa-Becerra (2009). Specifically, different works tackle the batch scheduling problem using metaheuristic multiobjective optimization techniques. For example, ant colony optimization (ACO) was applied by Jayaraman *et al.* (2000) for the optimal design and scheduling of batch chemical processes. Arnaout *et al.* (2010) have also used ACO for minimizing makespan in unrelated parallel machines with sequence-dependent setup times. The authors divided the scheduling problem into two-subproblems: assignment and sequencing, each of which was solved using different ant trails collaboratively. The use of other heuristics such as simulated annealing (SA) (Li & Ierapetritou, 2007) or particle swarm optimization (PSO) (Guo *et al.*, 2009) for solving the scheduling problem has also been exemplified.

Genetic algorithms (GA) have also been used for solving the problem of unrelated parallel machine scheduling, in some cases including planning (Dayou *et al.*, 2009) or job sequence and machine dependent setup times (Chyu & Chang, 2010). Most of these works rely on multiobjective formulations (MOGA), that attempt to optimize several objectives simultaneously such as total weighted flow time, total weighted tardiness, makespan and tardiness. He and Hui (2007) present a heuristic approach based on GA for solving large-size multi-stage multi-product scheduling problem in batch plants suitable for different scheduling objectives, such as total process time, total flow time, etc. They present techniques that greatly reduce the search space and expedite the overall solution procedure. In addition, the present a penalty method for handling the constraints in the problem, which avoids infeasibilities during the GA search and greatly increases the search speed. Other authors combine different optimization heuristics. Simulated Annealing (SA) has been combined with GA in the work of Ponnambalam and Mohan-Reddy (2003). In this context, it is also noteworthy to mention the use of discrete-event simulation (DES) models to represent dynamically the production system behavior as in (Azzaro-Pantel *et al.*, 1998), or the replacement of such DES by neural networks (ANN) (Senties *et al.*, 2009, 2010). Other developments include fuzzy logic (Aguilar-Lasserre *et al.*, 2009) or probabilistic (Bonfill *et al.*, 2008) representations for uncertainty treatment.

Although the previous contributions consider multiobjective problems, they are either focused on economic criteria or they only tackle partially the environmental concerns of the production process, disregarding trade-offs between economic and environmental aspects typically arise in scheduling problems. This chapter aims at gaining insight into those trade-offs, with emphasis on the effect of product changeovers, in the environmental performance of batch plants from an integrative perspective of the whole production process. The proposed functional integration approach is illustrated for the case of the multiproduct acrylic fiber production plant, in which special attention is focused on the influence of product changeovers.

The analysis of the decision maker's alternatives under conflicting objectives is performed by means of multi-objective optimization. Specifically, the normalized normal constraint method presented by Messac *et al.* (2003), as justified in Chapter 3, is applied to obtain a set of Pareto solutions, representing the compromise between the criteria considered in the analysis. Such approach is computationally expensive, since it entails solving a large number of MILP problems, or MINLP according to the problem structure. Integrating the aforementioned issues at the operational level is not a trivial task, since day-to-day decisions require almost immediate solutions, which can only be provided for models of limited complexity using the computational tools currently available. Hence, a hybrid optimization method that combines an evolutionary algorithm based on genetic algorithms with mathematical local search that implements a bit change method is presented for solving large scale instances. Furthermore, different metrics are proposed to select a compromise among the Pareto solutions.

9.3 Problem statement

This chapter represents a comprehensive step over the approaches presented in the former section by systematically assisting in the product scheduling under economic and environmental impacts considerations. The problem statement is identical to the one presented in Chapter 4, giving special emphasis to simultaneous optimization of the environmental impact and economic performance of the resulting schedules.

The scheduling model proposed in section 9.5 is solved under multiple objectives by using: i) a rigorous mathematical moMILP/MINLP algorithm, for tackling small instances, and ii) a hybrid moGA coupled with a local search algorithm for large scale problems, as described in section 9.6.

9.4 Environmental assessment

The main driving forces for incorporating environmental aspects process optimization are the pressures from regulation policies and the recent global trend toward sustainability in businesses (Clift & Azapagic, 1999). Different studies have been carried out in order to identify the most significant environmental effects of a process and to suggest modifications with the aim to achieve environmental improvements. As a result, a wide range of process design frameworks have been proposed. The methodology for obtaining minimum environmental impact processes (MEI, or MEI methodology) (Stefanis *et al.*, 1997), the waste reduction algorithm (WAR) (Cabezas *et al.*, 1999) proposed by the United States Environmental Protection Agency (US-EPA), which uses the pollution balance concept, the introduction of "eco-vectors" (Castells *et al.*, 1994) for the calculation of life cycle inventories for process industries and the environmental fate and risk assessment tool (EFRAT) (Chen & Shonnard, 2004), are only some representative examples. Most of them embed the concepts of Life Cycle Assessment (LCA), developed to set an environmental management system (EMS) through the ISO 1404X series (ISO14001, 2004). Within LCA, the overall life cycle of a process or product is analyzed, taking into account upstream and downstream flows from the cradle to the grave of the process. This approach avoids shifting burdens from part of the product supply chain to another, which would eventually lead to larger environmental damages. Consequently, the LCA technique is also selected in this work to assess the environmental performance of the scheduling tasks.

The implementation of a LCA for a given process or product requires data associated with process environmental interventions (e.g. raw material consumption, uncontrolled emissions and waste generation). This set of data is organized in a life cycle inventory (LCI) which is the basis for the environmental impact calculation, as specified in the ISO 1404X series. Within this model, and in order to avoid double counting in the emissions calculation, raw material emissions are not aggregated, whereas cleaning environmental interventions are considered separately.

Waste generation, fugitive emissions and raw material or utility consumption are the key components of the LCIs. Specifically, in the case of batch industries, the LCI is directly determined from product recipes and product changeover procedures.

9.5 Mathematical scheduling model and objective functions

In order to model the scheduling problem, the mathematical formulation presented in Chapter 4 based on the immediate precedence concept (Gupta & Karimi, 2003) has been adopted. The original model has been extended to consider different interbatch cleaning methods, additional objective functions (e.g. makespan, productivity and environmental impact) and product batching. The model is decomposed into two parts. First, the product batching problem (Equations 4.1 to 4.11) is considered based on the demand to be fulfilled and product batch sizes. The scheduling problem (Equations 4.12 to 4.39) is next solved to select the number of batches to be produced.

In this chapter, the environmental impact associated with products and different cleaning methods for changeovers is assessed. As a result, the mathematical programming model considers product flows, raw materials and utilities consumptions, and changeover operations to simultaneously deal with environmental and productivity features. The considered objective functions are presented and discussed in section 4.5. It is worth mentioning that the proposed multiobjective approach is still valid regardless of the selected mathematical model and could be further combined with the structural integration of Part III as proposed in the future work.

9.6 Multiobjective approaches and metrics selection

Different objective functions may be used in scheduling according to the decision maker's criteria. Multiple objective programming methods aim at finding suitable solutions of mathematical problems with multiple conflicting objective functions, and different alternative strategies can be applied to solve a multiobjective problem (Gandibleux, 2004; Wiecek *et al.*, 2008).

One typical approach consists of aggregating the different objectives in a single objective function with varying numerical weights. Unfortunately, these coefficients usually lack physical meaning, and entail an arbitrary assignment of values. Thus, there is not a unique optimal solution for multiobjective problems, but rather a set of feasible solutions which may be suitable. The preferred approach consists of providing a set of Pareto optimal solutions. This method provides further insight into the problem, allowing for the identification of solutions leading to large environmental improvements at a marginal increase in cost. A Pareto solution is one for which any improvement in one objective can only take place if at least another objective worsens. Pareto optimal solutions are also termed dominating solutions, while the remaining feasible solutions are dominated. This latter approach implies that the decision maker is interested in all possible trade-off solutions resulting from no previous articulation of the decision-maker's preferences. Particularly in the case of objective functions related to the environment, economic metrics are always prioritized in companies and constraints on the environmental interventions (emissions, concentrations and others) are given by stringent environmental policies. However, a view of process operation that considers the environment as an objective and not just as a constraint on operations can lead to the discovery of operating policies that improve both environmental and economic performance (Cano-Ruiz & McRae, 1998).

The techniques for generating a set of Pareto optimal solutions should have some desirable properties. Namely, they should be able to find all available Pareto points, generate them evenly along the possible solutions in the feasible region (understood as the collection of points that satisfy all problem constraints), and they should not generate and explore dominated solutions (Messac *et al.*, 2003). However, all the available techniques present deficiencies in some of the former aspects. For example, the weighted sum must be carefully applied since it does not generate all available Pareto points, and the Pareto frontier does not represent an evenly set of solutions of the feasible region (Steuer, 1986). Normal boundary intersection (NBI) (Das & Dennis,

1998) and normal constraint method (NC) (Messac *et al.*, 2003) generate points that are not in the Pareto frontier, being NBI more prone to generate dominated solutions. In general, all previous procedures require a filtering step to distinguish and classify dominated from non-dominated solutions.

In practice, the combinatorial nature of the scheduling problem poses serious computational difficulties when using rigorous mathematical approaches. Thus, the integration of multiobjective issues at the operational level increases the problem complexity. Therefore, rigorously multiobjective optimal solutions for scheduling problems, which entail almost immediate results for day-to-day decisions, can only be provided for models of limited complexity given the capabilities of the current software-hardware systems. Hence, the application of evolutionary optimization represents an alternative to efficiently solve large scale problems at the cost of disregarding the optimality proof.

One important aspect of using metaheuristics is that the scheduling problem can be considered as a black box model and no information regarding first or higher order derivatives is required, so only the objective function value and some constraints satisfaction is checked.

More importantly, most metaheuristic formulations are inherently multiobjective including the Pareto efficiency concept in the way new solutions are tested and gathered.

Thus, in metaheuristic optimization, the problem is not formulated as a mathematical program since the solution method is based on procedural search techniques, and the violation of constraints is handled through penalty functions. Hence, they may be problematic for problems involving complex constraints and continuous variables. As a result, it may be difficult to find feasible solutions.

9.6.1 Rigorous mathematical multiobjective approach

The Pareto frontier (PF) associated with the problem at hand is discrete and results from a set of integer variables being defined (e.g. sequence, cleaning method), consequently evenly separated solutions cannot be expected. This work proposes the use of the Normalized Constraint (NC) method described in Messac *et al.* (2003) modified to obtain a reliable set of possible Pareto solutions, and applies a Pareto filter algorithm developed by Cao (2009).

A key point in the NC method is the number of solutions that should be generated to obtain evenly separated Pareto solutions over the PF. Thus, the application of the NC method requires special attention. The selection of the number of solutions to be explored is performed by dividing the utopian line (hyperplane, in case of more than two objectives being considered), and exploring each constrained segment. This utopian hyperplane is obtained by the solution of the single objective optimizations as described in Messac *et al.* (2003). Exploring a high number of points would lead to an excessive computational effort, whereas an inadequate number of solutions would result in an incomplete PF containing dominated solutions due to unexplored Pareto optimal solutions.

In addition, in a strategy based on constraints, if the solution space is discrete, an increase in the number of divisions of the utopian hyperplane in question does not guarantee the generation of new Pareto solutions. Although the total number of problems discussed can be increased, their solution can lead to already explored discrete solutions.

Hence, to overcome some of the limitations mentioned above, an iterative approach is proposed to generate a reliable estimation of the PF. The number of divisions of the utopian hyperplane is incremented on each iteration and the points explored are added as new solutions. Different termination criteria are possible, (i) PF similarity and (ii) PF similarity percentage. The first termination criterion consists of checking the PF at the end of each iteration, if no changes are found in two consecutive iterations the PF is accepted as solution to the multiobjective problem. The latter termination criterion imposes the end of the iteration procedure, when the number of new Pareto solutions divided by the total number of explored solutions is lower than a specific tolerance (tol) percentage. Specifically in our case, a minimum of fifty points (nd_0) are initially generated and in the next iteration at least fifty new different points are further studied (nd_1). These parameters values (nd_j and tol) can be changed according to the problem characteristics.

The convergence of the proposed algorithm depends strongly on the global convergence of the optimization method used to solve each of the constrained problems, which in some cases might require the estimation of an initial starting point, particularly in the case of dealing with nonlinear models. The algorithm is shown next, Algorithm 9.1.

Algorithm 9.1: Pareto frontier generation.

Data: Number of utopian line divisions (nd_0), tolerance (tol).
Result: A reliable Pareto frontier estimate PF^*
begin
  explore S_0 solutions using nd_0 and count $np_0^{explored}$;
  generate first Pareto frontier estimate PF_0 from S_0;
  count Pareto points np_0^{PF};
  $j \longleftarrow 1$;
  $np_j^{PF}, np_j^{explored} \longleftarrow np_0^{explored} + 1$;
  while $np_j^{PF} \neq np_{j-1}^{PF}$ or $\frac{np_j^{PF} - np_{j-1}^{PF}}{np_j^{explored}} \geq tol$ **do**
    select j-th number of utopian line divisions nd_j;
    explore j-th solutions S_j using nd_j;
    $S_j \longleftarrow [S_j, S_{j-1}]$;
    perform a Pareto filter of explored solutions PF_j from S_j;
    count Pareto points np_j^{PF};
    count total explored solutions $np_j^{explored}$ in S_j;
    $j \longleftarrow j + 1$;
  $PF^* \longleftarrow PF_j$

9.6.2 Hybrid metaheuristic approach

The implementation of the multiobjective genetic algorithm (moGA) consists of creating and improving a set possible solutions defined by the integer decisions related to batch assignment, sequencing and cleaning method, as defined in Figure 9.1. The aforementioned decision variables have been properly coded to implement the multiobjective genetic algorithm and encompass the solution genotype. Therefore, such decision variables are properly coded to implement the moGA as presented in Chapter 3.

Next, the resulting solutions are transformed into binary variables and passed to the mathematical formulation, so that the continuous variables such as processing times or makespan are evaluated for those decisions, and the problem constraints, such as demand satisfaction, due dates or time horizon, are consequently checked.This step also allows for the problem constraints to be checked by the mathematical problem. The GA uses the objective function values for its selection operator. The inner level mathematical program optimization can be run in two different modes: one where integer variables are fixed rendering an LP using the values that the GA has set, and using the LP solver (CPLEX) as feasibility solution tester, where makespan is minimized; or other where a local search strategy based on a bit-code heuristic is used to improve the solutions. In this last case if linear functions are used then a MILP is solved for each objective function, while if a nonlinear objective function is selected then a MINLP or a sequence of MILPs are solved using the algorithm proposed for MILFPs. For the solution of MILPS, CPLEX solver is used. The scheduling problem is modeled by the mathematical formulation defined in section 9.5.

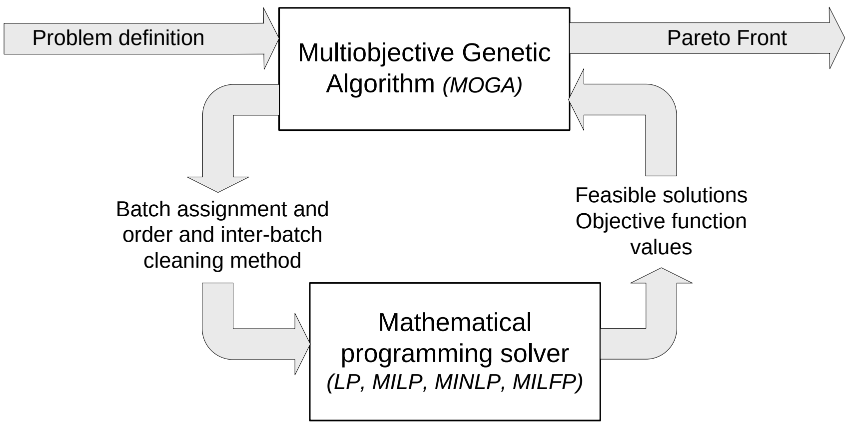


Figure 9.1: Overall algorithm outline.

9.6.3 Multiobjective performance metrics

Once the PF is generated, the decision maker should choose the solution to be adopted (Wiecek *et al.*, 2008). Different metrics have been defined to assist decision-makers in this task. These metrics are typically derived from the values of the objectives expressed in terms of the normalized distance from a given solution. The point which considers the best possible single objective outcomes is known as utopian point, while the one associated with the worst solutions is the nadir point. Routing from these points, several authors have proposed different compromise solutions.

On the one hand, a possible way to identify the best compromise solution is to select the point that minimizes the overall distance to the utopian point (Equation 9.1), as proposed by Hwang and Yoon (1981) in the Technique for Order by Similarity to Ideal Solution (TOPSIS).

$$\mu^{best} \rightarrow \min \left\{ \sum_{g} \left(\frac{\mu_g^* - \mu_g}{\mu_g^* - \mu_g^0} \right)^2 \right\} \tag{9.1}$$

An alternative strategy consists of measuring the distance from the PF solutions to the nadir point. Hence, an additional compromise solution can be chosen as the one whose geometric distance to the nadir is maximum (Equation 9.2).

$$\mu^{best} \rightarrow \max \left\{ \sum_{g} \left(\frac{\mu_g - \mu_g^0}{\mu_g^* - \mu_g^0} \right)^2 \right\} \tag{9.2}$$

9.7 Case study process description

The proposed methodology is illustrated through its application to the multi-product batch process plant presented in Chapter 4, which produces three acrylic fiber formulations by a suspension polymerization process requiring 14 processing stages. The detailed description of the recipes and costs are provided in Appendix D.

A changeover operation is performed between any two batches. Three different changeover cleaning methods, which differ in time, cost and environmental impact, are defined as summarized in Table 9.1.

Table 9.1: Cleaning methods description.

Cleaning method	Time	Cost	Env. Impact	Method based on the use of
1	Very low	Medium	Medium	Steam
2	Very high	Very low	Low	Water
3	Medium	High	Medium	Organic solvent

To ease the computation of the environmental impacts, instead of adding up all the LCI results associated with the consumption/use of raw materials, utilities and cleaning agents, the Life Cycle Impact Assessment (LCIA) results from each of the activities (e.g. water use, steam generation or raw material production) have directly been used. These LCIA results hold the combined environmental impact of each activity from a cradle to gate point of view. The LCIA methodology applied is IMPACT 2002 (Humbert *et al.*, 2005). Simapro (de Schryver *et al.*, 2006) has been selected to calculate these LCIAs from the corresponding LCIs (EcoinventV2.0, 2008) and the LCIA information is used in the model. It is found that the environmental impact of raw materials is quite large compared to the remaining quantities. This fact was expected given that this impact is significantly larger than either the environmental impact associated with the use of utilities or changeover operations. Hence, this analysis distinguishes between them accordingly. As for environmental impact of the production itself, the LCI entailing residues, non-controlled emissions, raw materials, steam, water, and electricity consumption is calculated using good engineering practices, and it is based on the available literature data. Section D.3 contains the assumptions and results of these calculations.

9.8 Results

9.8.1 Example 1: Rigorous multiobjective approach

The scheduling of the multiproduct fiber plant is solved considering a demand of 2 batches of each product, and assuming a minimum demand satisfaction level of the 50%.

Three different combinations of objective functions are studied which result in different multiobjective problems, namely case (i) a three-objective optimization considering makespan, profit and environmental impact, and two biobjective optimization problems which consider: case (ii) profitability and environmental impact, and case (iii) profitability and relative environmental impact.

These problems were selected bearing in mind the "extensive" and "intensified" system characteristics. The extensive characteristics depend on the amount of product produced, while the latter focus on efficiency, by relating the metric directly linked to production to others such as time or amount produced.

Therefore, the three case studies have been chosen in order to consider only extensive metrics (case i), such as profit, makespan or total environmental impact, only intensified metrics (case iii), such as profitability and relative environmental impact, and a mixture of them (case ii).

The mathematical formulation and the NC method have been implemented in GAMS, and solved using CPLEX 11.2 for the MILP (case i), and BARON 8.1 for the MINLP (cases ii and iii). The computation effort in solving each constrained optimization is highly dependent on the starting point. A four thread processor 3 GHz each has been used for the solution of this example.

The Pareto filtering procedure has been implemented in Matlab (Mathworks, 2009; Cao, 2009), along with the algorithmic strategy (Algorithm 9.1) using Matgams (Ferris, 2005) to inferface both software packages (i.e., Matlab and GAMS).

Case i considers the multiobjective optimization of profit, environmental impact and makespan. Figure 9.2 contains the Pareto solutions in the three dimensional space. Given the fact that fixed batch sizes are considered, the Pareto frontier is a collection of points that represent different production sequences. The evolution of the proposed algorithm in terms of the resulting Pareto solutions are presented in Table 9.2. A total of 5143 MILP have been solved to optimality, which result in 89 non-dominated solutions. The average solving time for each optimization problem was about 44 seconds. The iterative procedure has been stopped when the percentage of new Pareto solutions divided by the total number of explored points is below 0.1%, (tol=$1{\cdot}10^{-5}$).

PFs of the two dimension projections do not contain all the Pareto points of the three dimensional problem, but show existing trade-offs between any two objectives. Therefore, the projections of the solutions on two dimensional planes and their respective Pareto points are further discussed.

Figure 9.3 presents the PF for the two-objective optimization of total profit and total environmental impact, which was considered separately (as Case ia) from the 3 objective Case (i). A total number of 3000 points along the utopian line have been solved to optimality (green crosses), from which 24 non-dominated Pareto solutions (blue circles) are obtained after applying the Pareto filter.

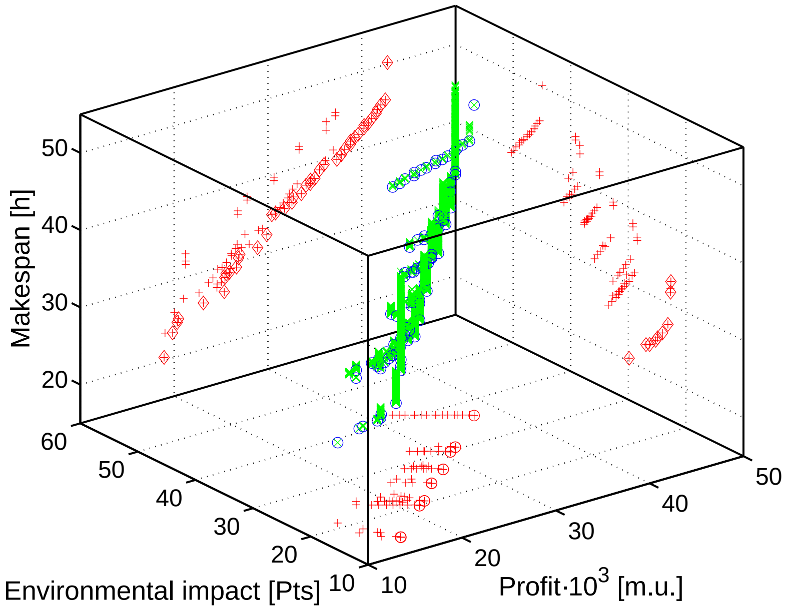


Figure 9.2: Case (i), solutions for three objective optimization considering profit, environmental impact and makespan. Green crosses are all explored solutions; non-dominated solutions are encircled in blue (Pareto frontier); red plus symbols are projections of all explored solutions in their corresponding two dimensional planes and red diamonds solutions are non-dominated solutions in such planes.

Table 9.2: Case (i), iterations in the number of Pareto points generation, for the multiobjective optimization considering profit, environmental impact and makespan.

Iteration	0	1	2	3	4	5	6
Number of utopian line divisions (nd_j)	11	21	31	41	46	51	56
Number of explored points	58	256	701	1479	2468	3679	5143
Total Pareto solutions (np_j^{PF})	26	42	59	71	76	85	89
Changing Pareto frontier solutions	26	16	20	12	6	10	4
Pareto solutions z^{profit} - z^{ei}	10	11	13	15	15	16	16
Pareto solutions z^{profit} - Mk	10	18	31	34	36	40	42
Pareto solutions z^{ei} - Mk	4	4	5	7	7	9	9
Computation time $\cdot 10^3$[s CPU]	3.60	11.97	30.25	69.40	107.89	152.43	210.45

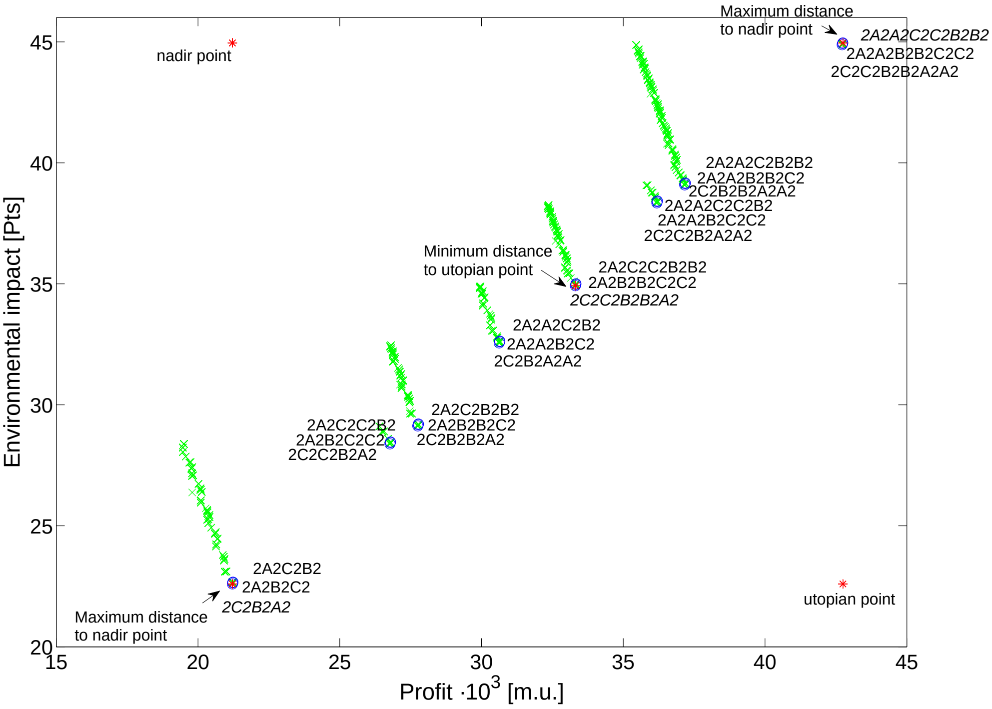


Figure 9.3: Case (ia), solutions for two-objective optimization considering profit and environmental impact. Green crosses are all explored solutions; non-dominated solutions are encircled in blue (Pareto frontier); red stars are nadir, utopian points; and sequences in italics represent compromise solutions shown in Table 9.3.

The solution with highest profit satisfies the total demand (i.e. 2 batches of each product), whereas the most environmentally friendly option only processes the minimum amount of each product (1 batch for each product). In any case, the same changeover cleaning method 2 is selected in all solutions, because it is the most economic and environmental advantageous (see Figure D.3 and Figure D.4), in spite of the time required, which is not considered in this case. Pareto points are found to be grouped between the two extreme optimal solutions in six clusters, whose difference consists of the number of batches of each product. Regarding the most environmentally friendly solution cluster, product C offers more increment in profit and less environmental impact. The following less environmentally advantageous sequence with higher gain in profit includes an additional batch of product B instead of C; and then, a batch of A instead B or C. Next, an additional batch is considered in the production sequence, and finally, the complete fulfillment of demand entails the highest economic profit. In every cluster, solutions differ in the production sequences. To start producing with fiber C is slightly more environmentally friendly and less economically profitable than with fiber A.

Table 9.3 shows that the compromise solution according to the minimum distance to the utopian point consists of sequence 2A2A2C2B2 (such string represents the ordered sequence of batches, where the capital letters A, B and C, stand for the product, and the numbers 1, 2 and 3 for the cleaning method being used), which is

Table 9.3: Case (ia), utopian, nadir and solutions of compromise according to the different metrics considering profit and environmental impact (* defines utopia and $^-$ defines nadir). Distances are reported normalized.

$z^{profit} \cdot 10^3$ [m.u.]	z^{ei} [Pts]	Sequence	Distance utopian	Distance nadir
21.213$^-$	22.595*	2C2B2A2	1.000	**1.000**
33.310	34.921	2A2A2C2B2	**0.704**	0.719
42.7455*	44.956$^-$	2A2A2C2C2B2B2	1.000	**1.000**

Table 9.4: Case (i) utopian, nadir and solutions of compromise according to the different metrics considering total profit, environmental impact and makespan (* defines utopia and $^-$ defines nadir) for Example 1. Distances are reported normalized.

$z^{profit} \cdot 10^3$ [m.u.]	z^{ei} [Pts]	Mk [h]	Sequence	Distance utopian	Distance nadir
21.213	22.595*	33.000	2C2B2A2	0.998	1.159
42.745*	44.956$^-$	50.200$^-$	2A2A2C2C2B2B2	1.285	1.018
18.931$^-$	29.861	20.400*	1A1C1B1	1.034	1.243
30.417	33.069	34.820	2A2A2C2B1	**0.803**	0.941
20.327	25.251	24.427	2A2B1C1	0.956	**1.253**

located approximately in the middle of the whole range of both objective functions. If the maximum distance to the nadir point was selected as decision criterion, there would be two possibilities: either the solution of maximum profit or the solution of minimum environmental impact, since both of them have the same maximum normalized distance to the nadir solution.

On the other hand, the biobjetive projections for environmental impact vs makespan (case ib), and profit vs makespan (case ic), are given in Figure 9.4 and Figure 9.5. The solution with lowest makespan contains one batch of each product, and includes changeover 1, whose time is the shortest, as it could be expected (see Figure D.5). Sequences starting with fiber A have higher environmental impact but lower makespan than those with C. In addition, those sequences starting with product A dominate other sequences in the profit and makespan biobjective problem, even though starting with product A has the highest cost regarding the other two products.

For the overall three objective optimization, the utopia, nadir and solutions of compromise selected according to the proposed criteria are shown in Table 9.4. Sequence 2A2A2C2B1 is the one whose distance to the utopian is minimum; whereas solution 2A2B1C1 has the highest distance to the nadir point.

It is important to note that in this case, single objective optimal solutions are bounded by the minimum and maximum demand requirements. Regarding minimum requirements, in the case of environmental impact and makespan, their ultimate minimum will be zero which is associated with not producing any product, while in the case of profit, its optimization fulfills all required demand. If these bounds are changed the behavior would be the same, consequently special attention has to be put in the modeling of demand requirements given that for these metrics, its selection will be of paramount importance.

Case ii considers the analysis of the scheduling results when profitability and environmental impact are compared. Figure 9.6 presents the PF with 38 non-

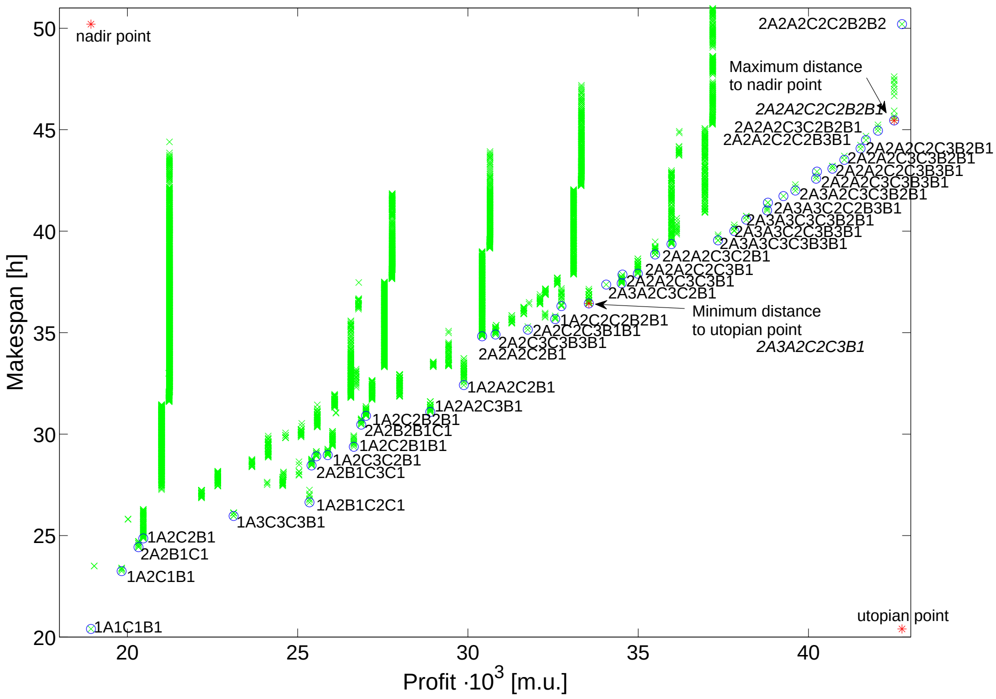


Figure 9.4: Case (ib), solutions for two-objective optimization considering profit and makespan. Green crosses are all explored solutions; blue circles the non-nominated solutions (Pareto frontier); red stars are nadir, utopian points; and sequences in italics represent compromise solutions.

Table 9.5: Case (ii), iterations in the number of Pareto points generation, for the multiobjective optimization considering profitability and environmental impact for Example 1.

Iteration	0	1	2
Number of utopian line divisions	51	101	151
Number of explored points	51	101	201
Total Pareto solutions	31	38	37
Changing Pareto frontier solutions	31	7	7
Computation time $\cdot 10^5$[s CPU]	1.31	2.67	5.28

dominated Pareto solutions (blue circles) for the biobjective optimization of profitability and environmental impact. In this case, the utopian line is divided iteratively in multiples of 50, from 50 up to 150 (see Table 9.5). As a result, a total number of 200 points along the utopian line have been solved. In about 13% of all problems, the MINLP solver (BARON) was not able to guarantee global optimality, after a reasonable computational effort (7200 CPU seconds). The average solving time for each optimization problem was found to be 2635 seconds. The iterative procedure has been stopped when the percentage of new solutions is below 5% ($tol = 5{\cdot}10^{-2}$).

The most productive sequence consists of producing full demand of the three products with changeover method 1, which is the one that takes the least time. It is worth noting that the former sequence consists of AACBBC, which entails three

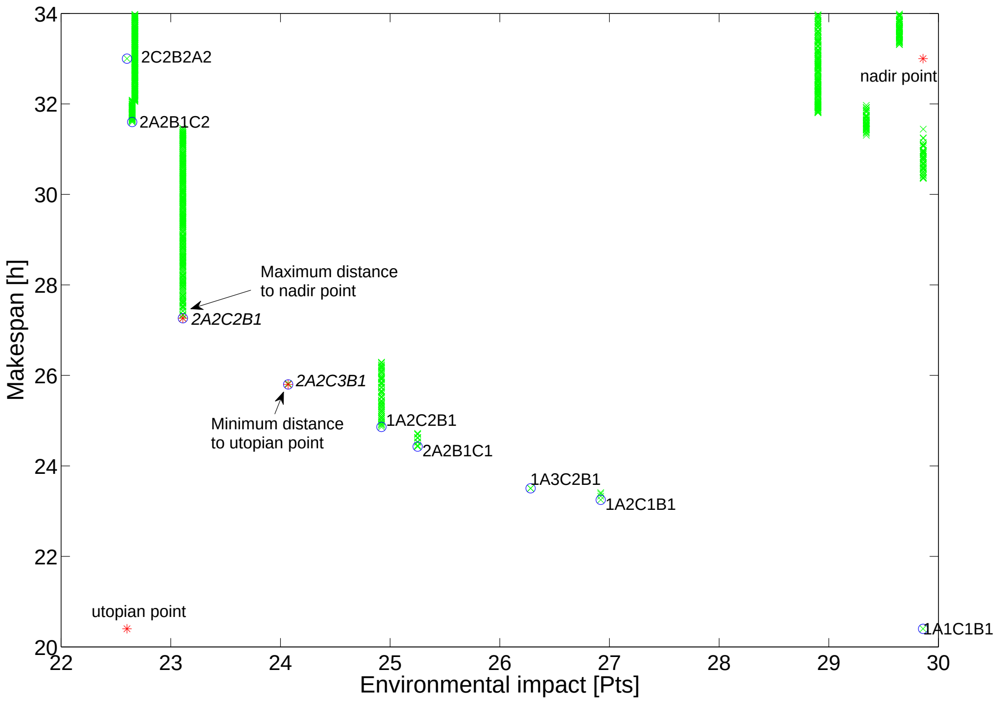


Figure 9.5: Case (ic), solutions for two-objective optimization considering total environmental impact and makespan. Green crosses are all explored solutions; blue circles the non-nominated solutions (Pareto frontier); red stars are nadir, utopian points; and sequences in italics represent compromise solutions.

inter-product changes and with higher overall changeover time than sequences such as AACCBB (with two inter-product changes). The reason for this issue is not evident and it can be understood from the Gantt charts in Figure 9.7. In sequence AACCBB, there are two pieces of equipment that are bottlenecks (C1 and V1); which results in a total makespan of 33.75h (Figure 9.7(b)). However, sequence AACBBC avoids the bottleneck in equipment C1 and has a total makespan of 33.15h (Figure 9.7(a)); consequently, its profitability increases in spite of the higher costs incurred by sequence changes.

Table 9.6 contains the solutions of compromise according to the different metrics. Note that in this case, the solution whose distance to the utopian point is minimum includes one batch of each product using cleaning method 1. In addition, Figure 9.6 highlights the relative position of the compromise solutions regarding the other Pareto solutions.

Case iii encompasses the analysis of scheduling results considering profitability and relative environmental impact metrics. In Figure 9.8, Pareto solutions differ in the number of batches of each product, the sequence in which they are produced, and cleaning method used. Some of these solutions have already appeared when optimization of total profit is considered, although they are still valid, most of them are not part of the PF for this case. In the Pareto frontier solutions are not grouped as in the two-objective case of total profit and environmental impact.

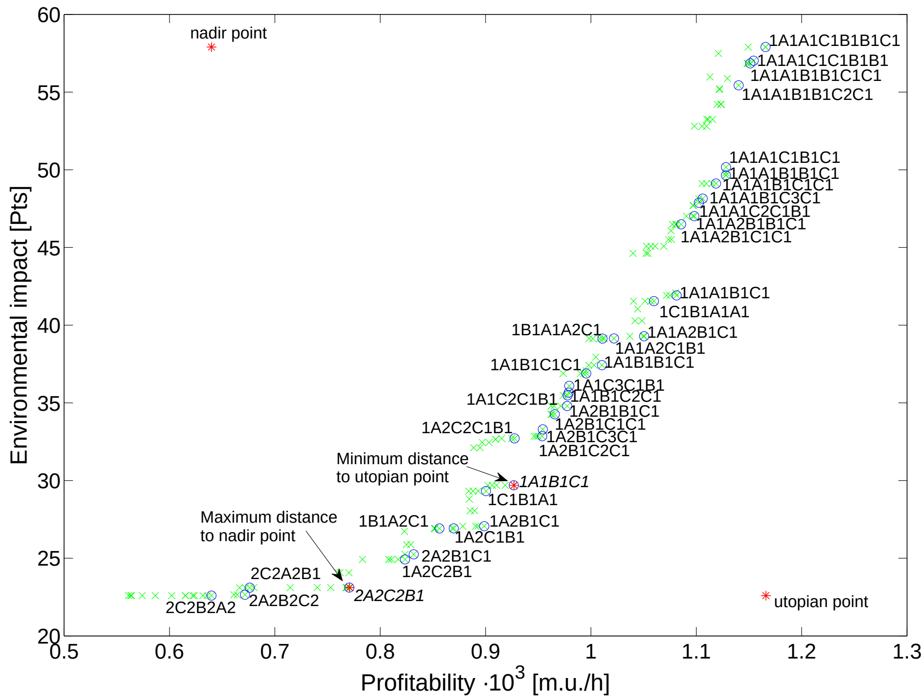
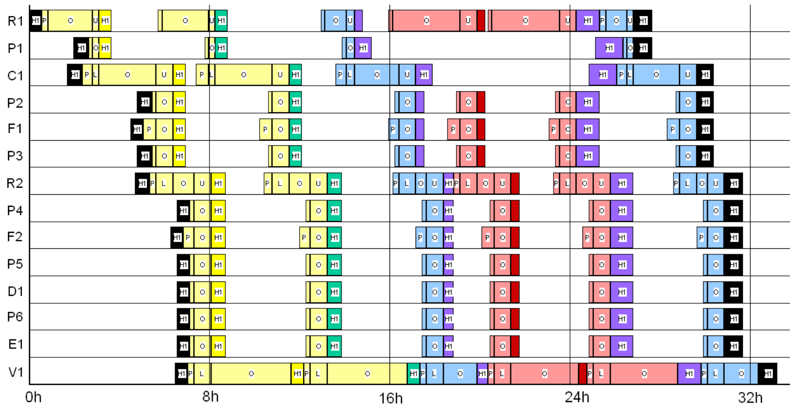


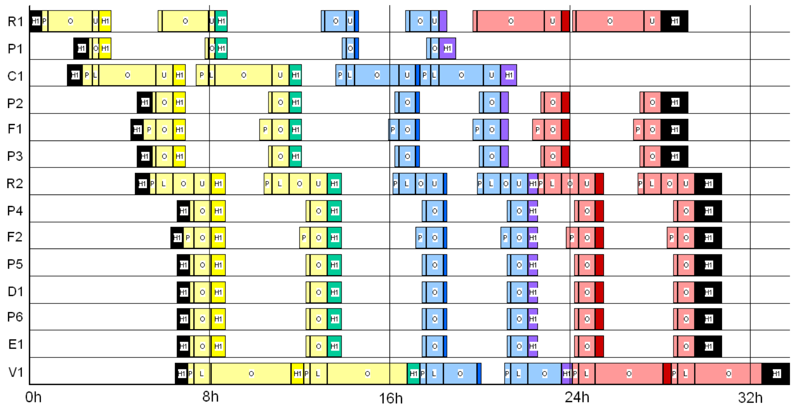
Figure 9.6: Case (ii), solutions for two-objective optimization considering profitability and environmental impact for Example 1. Green crosses are all explored solutions; blue circles the non-nominated solutions (Pareto frontier); red stars are nadir, utopian points; and sequences in italics represent compromise solutions shown in Table 9.6.

Table 9.6: Case (ii), solutions of compromise according to the different metrics considering profitability and environmental impact (* defines utopia and $^-$ defines nadir) for Example 1. Distances are reported normalized.

$z^{prod} \cdot 10^3$ [m.u./h]	z^{ei} [Pts]	Sequence	Distance utopian	Distance nadir
0.640^-	22.595^*	2C2B2A2	1.000	1.000
0.927	29.691	1A1B1C1	**0.497**	0.968
0.771	23.110	2A2C2B1	0.752	**1.016**
1.166^*	57.898^-	1A1A1C1B1B1C1	1.000	1.000



(a) Sequence 1A1A1C1B1B1C1.



(b) Sequence 1A1A1C1C1B1B1.

Figure 9.7: Gantt charts for sequences AACBBC and AACCBB in Example 1, case ii. (black: starting and finishing cleaning tasks; yellow, red and blue: fibers A, B and C, respectively; darker colored areas represent changeover methods)

The Pareto frontier for the two-objective optimization of profitability and relative environmental impact contains 34 non-dominated solutions (Figure 9.8). In this case, the utopian line is divided iteratively in multiples of 50, from 50 up to 100 (see Table 9.7), when the percentage of new Pareto solutions is below 10%. The average solving time for each optimization problem was found to be 5121 seconds. When minimizing the environmental impact per unit of product, both the sequence and cleaning method is the same as when minimizing the total environmental impact, but an additional batch of fiber B is produced. The main reason stems from the fact that by dividing the produced quantity, producing the smallest quantity of the products is not advantageous from the environmental point of view. Therefore, this relative objective function measures the most environmentally efficient way of producing.

Table 9.8 contains the solutions of compromise according to the different metrics. In this case, both solutions are different to the extreme points. Figure 9.8 highlights the relative position of the solutions of compromise according to Equations 9.1 and

Table 9.7: Case (iii), iterations in the number of Pareto points generation, for the multiobjective optimization considering profitability and relative environmental impact for Example 1.

Iteration	0	1
Number of utopian line divisions	51	101
Number of explored points	51	101
Total Pareto solutions	31	34
Changing Pareto frontier solutions	31	10
Computation time $\cdot 10^5$[s CPU]	2.35	5.17

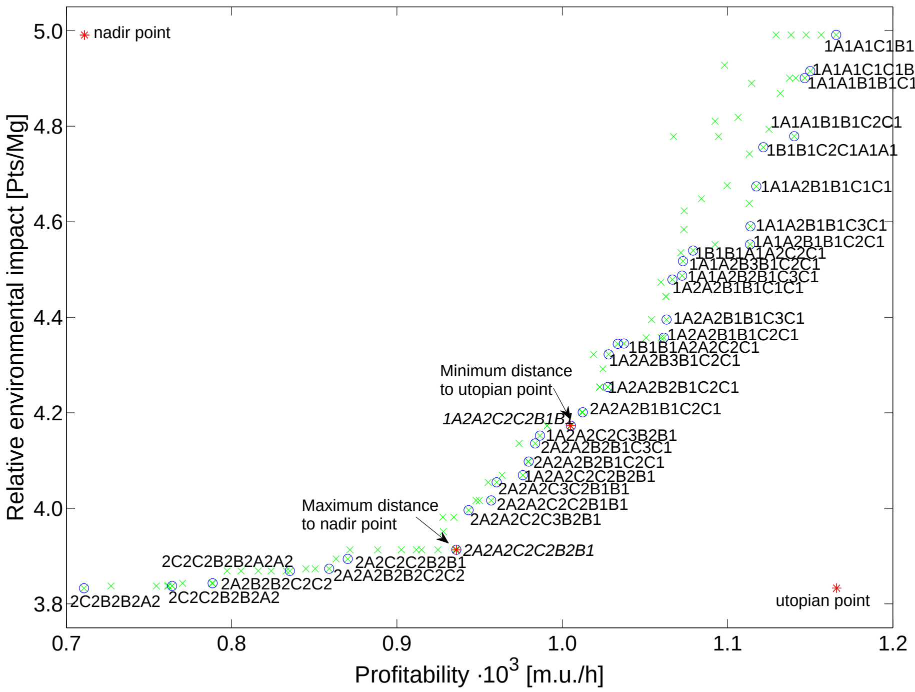


Figure 9.8: Case (iii), Solutions for two-objective optimization considering profitability and relative environmental impact. Green crosses are all explored solutions; blue circles the non-nominated solutions (Pareto frontier); red stars are nadir, utopian points; and sequences in italics represent compromise solutions shown in Table 9.8.

9.2, which are both different to the single objective optimal solutions. Both selected sequences produce the same amount of products and in the same order, but they differ in the cleaning methods used for the changeover between pairs of batches.

To sum up, we have considered the relative environmental impact and profitability metrics for comparison. In Figure 9.9 it can be seen that the solutions obtained for the other metrics optimization (case i and ii), are not contained in the PF found for the relative environmental impact and profitability (case iii). It can be seen that the solution with optimal profit is dominated by other solutions whose cleaning methods are the same, but its production sequence is different. With regard to the makespan (Mk) optimization solution it is found be far way from the PF, while the environmental

Table 9.8: Case (iii), utopian, nadir and solutions of compromise considering profitability and relative environmental impact (* defines utopia and $^-$ defines nadir). Distances are reported normalized.

$z^{prod} \cdot 10^3$ [m.u./h]	z^{rei} [Pts/Mg]	Sequence	Distance utopian	Distance nadir
0.711^-	3.833^*	2C2B2B2A2	1.000	1.000
0.936	3.913	2A2A2C2C2B2B1	0.510	**1.054**
1.005	4.173	1A2A2C2C2B1B1	**0.459**	0.958
1.166^*	4.991^-	1A1A1C1B1B1C1	1.000	1.000

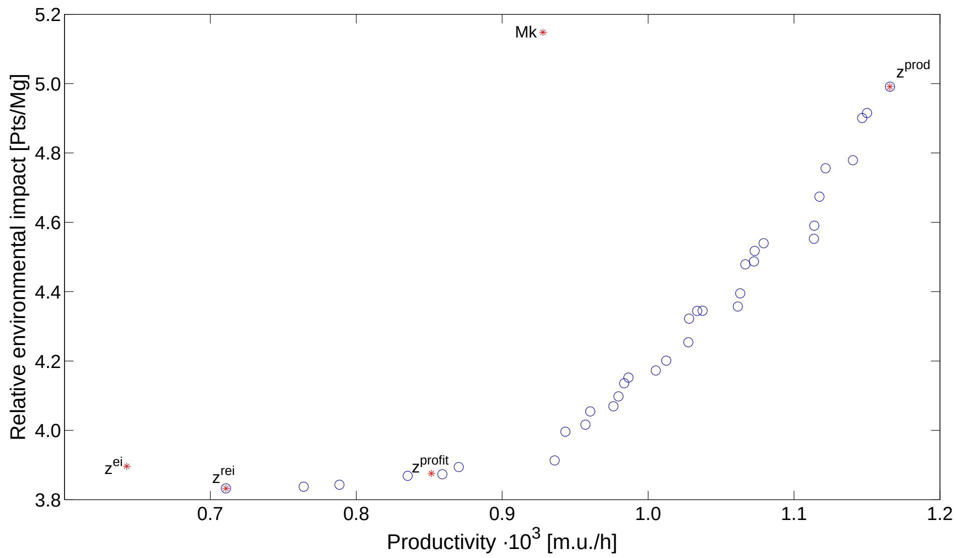


Figure 9.9: Pareto frontier for two-objective optimization considering profitability and relative environmental impact, and optimal single objective solutions (non-dominated solutions are encircled in blue; red stars are single objective optimal solutions).

impact optimization is closer.

As we can see from the computational times reported in Tables 9.2, 9.5 and 9.7, the application of this algorithm is highly dependent on the solving time required for the optimization of each constrained problem. We found that MILPs are easier to solve, while MINLPs require longer times. Clearly the applicability of the presented model and algorithm, to practical day-to-day operation decisions is far from being optimal due to the excessive computational required time, however we have shown the algorithm conceptual validity, which is independent of the model used. Given that the bottleneck of the presented algorithm resides in the optimization step, any method or technique for decreasing this time will improve the overall algorithm solution time. These techniques might involve: an initial point estimator or the application of decomposition techniques (e.g. Benders or Lagrange) to the model. On the other hand, an algorithm improvement might lie in the selection of the new constrained problems which in our case was done blindly and systematically by sub-dividing the utopian hyperplane in smaller divisions. The application of any of the former techniques will render an algorithm which might be suitable for day to day operation.

9.8.2 Example 2: Multiobjective hybrid GA approach

The scheduling corresponding to a large size demand is considered in this case (Table D.2). A minimum of 80% of the demand of each product must be satisfied. Two bi-objective optimization problems considering profit and environmental impact are considered, namely (iv) under no time horizon restrictions (200h); and (v) under strict time horizon constraints (140h), which results in a high initial number of non-feasible solutions. In order to tune the parameters of the GA, case (ia), whose exact PF is known from Example 1, is used. The moGa algorithmic strategy and Pareto filtering of the solutions have been implemented and solved in Matlab (Mathworks, 2009; Cao, 2009), and the whole solving process automated using Matgams (Ferris, 2005). The mathematical formulation and local search have been implemented in GAMS, and solved using CPLEX 11.2. A four thread processor MMMM has been used for the solution of this example.

In order to test the former algorithm, many of its parameters must be selected and decided upon. The discussion of its selection is done next, while its application to large size problems is done in cases (iv) and (v).

Algorithm tuning

As discussed by Conn *et al.* (2009), tuning the parameters of a derivative free optimization algorithm can itself be thought as an optimization problem, where different criteria are to be met and the algorithm parameters are optimization variables. To analyze the results of a given set of parameters different criteria were analyzed: (i) number of model runs (N_{mruns}) (ii) the fraction of Pareto solutions that the last iteration contains compared to a "true" PF ($F_{PF} = \frac{N_{PF}}{N_{PF}^{true}}$), and (iii) the time elapsed for its execution. In the first case the number of generations, and the number of model runs in both modes (feasibility/OF evaluation and local search) are considered. For calculating the fraction of PF solutions, an estimate of the "true" PF is required (PF^{true}), this estimation is done by considering all the numerical experiments that were run, or optimizations that were done using directly the mathematical program solving the MILP.

Instead of using an optimization approach to tune the algorithm parameters, we focus on different selections of those parameters based on a design of experiments (DOE) and check the optimization results of such selections, this approach is similar to the one adopted in Arnaout *et al.* (2010), for tuning ACO algorithm parameters.

The parameters that must be decided and fixed beforehand are specified in Table 3.2. However many of them were predefined using widely accepted heuristics, thus minimizing the amount of parameters to consider. According to the GA toolbox from Matlab Mathworks (2009), the number of individuals of a population, N_{pop}, must be equal to or larger than 15 times the number of variables, consequently we have set this value to $N_{pop} = 15 * N_{vars}$. Given that the variables considered are the number of parameters in the string that can be changed, namely sequence, cleaning type and batch allocation then the N_{vars} value is changed depending on the problem size. The maximum number of generations, N_{gen}, is the limit of iterations that the algorithm performs consequently the number of iterations will result according to the end criteria presented in the previous paragraphs, related to: T_{lim} and N_{rep}. These

Table 9.9: Values for the combinations of parameters that result in alternative parameters tuning (in bold, the selected values).

N_{PFs}	F_{PF}	N_{mruns}	F_{PF}/N_{mruns}
'1'	0.6479	24.4	0.02716
'2'	0.7438	26.6	**0.03191**
'3'	0.8063	30.0	0.03120
Mean	0.7326	27.0	0.03009

Table 9.10: Values for the combinations of parameters that result in alternative parameters tuning (in bold, the selected values).

	Combination 1			Combination 2			Combination 3		
r	0.00	0.25	0.50	0.15	0.25	0.40	**0.15**	0.25	0.40
β	0.00	0.25	0.50	0.00	0.20	0.40	**0.70**	0.80	0.90
m	0.00	0.25	0.50	0.00	0.20	0.40	**0.25**	0.50	0.75
c_1	0.00	0.25	0.50	0.00	0.20	0.40	**0.25**	0.50	0.75
c_2	0.00	0.25	0.50	0.00	0.20	0.40	0.25	**0.50**	0.75

three parameter values were fixed to: N_{gen}=1000, T_{lim}=7200s and N_{rep}=5, which provide with adequate solution times.

Regarding the mating pool, three possibilities were analyzed $N_{PFs} = 1$, where the mating pool only considers the first Pareto Front, PF^1, while the other possibilities considered $N_{PFs} = 2$ or 3, where the mating pool consisted of the PF^1 and the first and second best Pareto fronts (PF^2 and PF^3). In this case, it has been found that better results where found when $N_{PFs} = 2$, i.e. when the mating pool considers the PF^1 together with its second best PF estimate (PF^2). Table 9.9 shows the N_{mruns} and F_{PF} values for the N_{PFs} previously discussed and all the remaining parameters fixed. The values reported are for 20 different random seeds.

In order to choose the percentage of the new individuals that are derived from the mating pool using each operator (P_{ran}, P_{mut1}, P_{mut3}, P_{swi}, P_{pos}, P_{crx}), different numerical experiments were performed. It has been found that when a single operator is used, i.e. P_{any}=1, the results are worst than when combinations of them are used, consequently a given combination has to be set. Given that the total fraction must add to one, only 5 of the previous parameters are independent. In order to select the best combination of such percentages, a statistical analysis has been performed based on 5 parameters which allow for calculating the former 6, namely they have been modeled as: $P_{ran} = r$ $P_{mut1} = (1-r)*\beta*m$ $P_{mut3} = (1-r)*\beta*(1-m)$ $P_{swi} = (1-r)*(1-\beta)*c_1$ $P_{pos} = (1-r)*(1-\beta)*(1-c_1)*c_2$ $P_{crx} = (1-r)*(1-\beta)*(1-c_1)*(1-c_2)$ where the experiment parameters are: r, β, m, c_1 and c_2. Three different experiments have been performed considering three different levels for each parameter. Each experiment was run using three different random generator seed to avoid eventual "lucky draws".

The experiments results were compared considering F_{PF} and F_{PF}/N_{mruns}. The average value across different random seeds was considered for each model parameters combination. The experiment with the highest mean value for F_{PF}/N_{mruns} and F_{PF} was r=0.15, β=0.7, m=0.25, c_1=0.25 and c_2=0.5, which corresponds to the following probabilities/percentages: P_{ran}=0.150, P_{mut1}=0.446, P_{mut3}=0.149, P_{swi}=0.064, P_{pos}=0.143 and P_{crx}=0.048.

Table 9.11: Mean values for F_{PF} and Elapsed Time (ET) [sCPU], grouped along the different tested parameter values ($N_X^{changes}$, $N_W^{changes}$).

Variable values		F_{PF}	ET [sCPU]	F_{PF}/ET$\cdot 10^{-4}$ [1/sCPU]
Mean value		0.8056	3618	2.481
$N_X^{changes}$	'4'	0.7937	**3070**	**2.730**
$N_X^{changes}$	'6'	**0.8135**	**3420**	**2.603**
$N_X^{changes}$	'10'	**0.8095**	4364	2.110
$N_W^{changes}$	'0'	0.8016	**2279**	**3.527**
$N_W^{changes}$	'1'	**0.8254**	4126	2.051
$N_W^{changes}$	'2'	0.7897	4448	1.866

Table 9.12: Tuned values of the parameters of the GA.

Parameter	Value
N_{pop}	$15\cdot Nvars$
N_{rep}	5
N_{ls}	3
$N_W^{changes}$	1
$N_X^{changes}$	6
P_{ran}	0.150
P_{mut1}	0.446
P_{mut3}	0.149
P_{swi}	0.064
P_{pos}	0.143
P_{crx}	0.048

Similarly to the selection of N_{PFs}, the parameters $N_X^{changes}$ and $N_W^{changes}$ have been studied. In this case, N_{ls} was set to 3 (1 local search runs and 2 consecutive feasible runs), while T_{lim}^{ls}=5 sCPU. These last two values were selected based on a T_{lim}=7200s. Note that longer algorithm runs might enable different values for both parameters. The time elapsed when the solver is running an optimization run is sensitively higher than when the solver has all binary variables fixed. Table 9.11 shows the Elapsed Time [sCPU] and the F_{PF} values while all the remaining parameters are fixed.

It can be seen from Table 9.11, that there are different possible combinations of $N_X^{changes}$ and $N_W^{changes}$ whose value produces higher mean values than the overall mean (see bold figures). In terms of F_{PF} the best value regarding $N_X^{changes}$ is '6', while in terms of $N_W^{changes}$ is '4'. Concerning the ETs the behavior is similar, the parameters values do not coincide for the best values, being '1' for the case of $N_X^{changes}$ and '0' for $N_W^{changes}$. Given that we are prioritizing the F_{PF}, we will select: $N_X^{changes}$='6' and $N_W^{changes}$='1'. As a whole, the parameters of the moGA for this work are summarized in Table 9.12.

Case (ii) is revisited to check performance with non linear objective functions, using the local search. The resulting PF contains 38 solutions (see Figure 9.10), and was solved in 34 generations using 3960.6sCPU. Next, this case has been solved recursively to obtain the actual PF, which consists of 51 solutions. Therefore, a total of 29 of the former 38 solutions of the Pareto frontier actually belong to the 51 solution PF. The compromise solution whose distance to the utopia is minimum is different from that obtained in case (ii). In addition, this PF as good the one first PF reported in

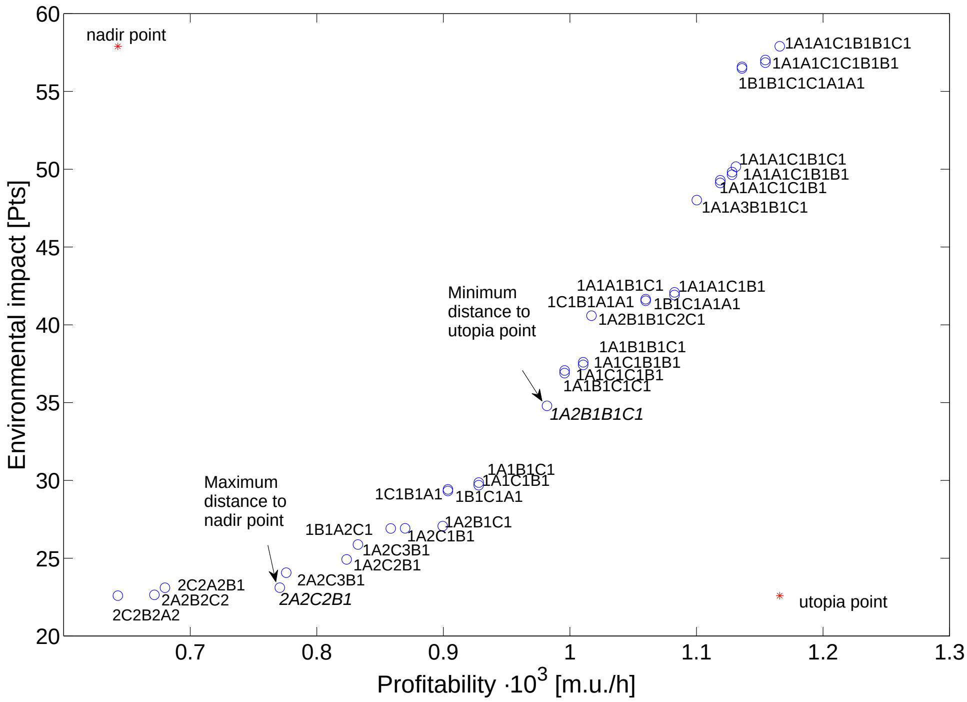


Figure 9.10: Case (ii) revisited. Solutions for two-objective optimization considering profitability and environmental impact. Non-dominated solutions are encircled in blue (Pareto frontier); red stars are nadir, utopian points; and sequences in italics represent compromise solutions. Note that for the sake of clarity some sequences are not explicitly shown in the Figure.

Example 1 case (ii) at iteration 0, which contained only 31 solutions, three of which were in fact dominated solutions of the real PF. From a computational point of view, the moGA is able to find as good solutions as the rigorous approach for this case, in approximately 33 times less CPU resources.

Case iv explores the trade-offs arising profit between and total environmental impact considering a large demand. The parameters of the moGA for this case are summarized in Table 9.12.

Figure 9.11 presents the PF for the bi-criteria optimization (profit vs total environmental impact). The estimated PF in this case contains 36 non dominated solutions (blue circles), and is generated after 20 generations with a total time of $7.76 \cdot 10^3$CPUs. Such Pareto frontier must correspond to the actual PF, since the problem structure is similar to case (ia), and the 36 solutions share the same structure as the PF solutions of the previous case. Therefore, the evolutionary approach works efficiently, providing near optimal solutions in a reasonable time.

The solution with highest profit satisfies the total demand, whereas the most environmentally friendly option only processes the minimum amount of each product (a minimum of 80% of the demand of each product). In any case, the same changeover cleaning method 2 is selected in all solutions, because it is the most economic and environmental advantageous, in spite of the time required, which is not an active

Table 9.13: Case (iv). Utopian, nadir and solutions of compromise according to the different metrics considering profit and environmental impact for Example 2 (* defines utopia and $^-$ defines nadir). Distances are reported normalized.

$z^{profit} \cdot 10^3$ [m.u.]	z^{ei} [Pts]	Sequence	Distance utopian	Distance nadir
153.3351$^-$	160.7842*	$2C^{(6)}_{(0\|5\|0)}2B^{(4)}_{(0\|3\|0)}2A^{(10)}_{(0\|9\|0)}2$	1.000	**1.000**
169.2880	177.2768	$2C^{(6)}_{(0\|5\|0)}2B^{(5)}_{(0\|4\|0)}2A^{(11)}_{(0\|10\|0)}2$	**0.704**	0.711
184.2802 *	193.1041$^-$	$2A^{(12)}_{(0\|11\|0)}2C^{(7)}_{(0\|6\|0)}2B^{(5)}_{(0\|4\|0)}2$	1.000	**1.000**

constraint in this problem. Pareto points are found to be grouped between the two extreme optimal solutions in twelve clusters, whose difference consists of the number of batches of each product. Regarding the most environmentally friendly solution cluster, product C offers more increment in profit and less environmental impact. The following less environmentally advantageous sequence with higher gain in profit includes an additional batch of product B instead of C; and then, a batch of A instead B or C. Next, an additional batch is considered in the production sequence, and finally, the complete fulfillment of demand entails the highest economic profit. In every cluster, solutions differ in the production sequences. To start producing with fiber C is slightly more environmentally friendly and less economically profitable than with fiber A. In all cases, batches are produced in campaigns of products.

Table 9.13 shows that the compromise solution according to the minimum distance to the utopian point consists of sequence $2C^{(6)}_{(0|5|0)}2B^{(5)}_{(0|4|0)}2A^{(11)}_{(0|10|0)}2$ (such string represents the ordered sequence of batches, where the capital letters A, B and C, stand for the product, and the numbers 1, 2 and 3 for the cleaning method being used), which is located approximately in the middle of the whole range of both objective functions. If the maximum distance to the nadir point was selected as decision criterion, there would be two possibilities: either the solution of maximum profit or the solution of minimum environmental impact, since both of them have the same maximum normalized distance to the nadir solution.

Case v uses the parameters shown in Table 9.12 as well. According to the algorithm description, the population should contain 795 individuals. However, due to the strict time horizon limit, there are many individuals at the first generation that are unfeasible (about 80 %). To avoid this, the population has been increased three times with respect to the theoretical value.

Figure 9.12 presents the PF for the two-objective optimization of profit and total environmental impact. The estimated PF in this case contains 14 non dominated solutions (blue circles), using as termination criterion 25 generations, with a total time of $4.32 \cdot 10^4$CPUs. Such Pareto frontier contains 10 Pareto solutions of an improved Pareto frontier estimation, which is obtained from several parameters combinations and contained a total of 17 solutions. Therefore, the evolutionary approach is an efficient method for generating near optimal Pareto solutions in a reasonable time, specially for highly timely constrained problems.

The solutions in the PF make use of cleaning method 1, which is the least time consuming, in all sequences in order to fulfill the time horizon restriction. Cleaning method 2 is combined with method 1 in order to obtain Pareto optimal production sequences in which the economic and environmental performance are simultaneously

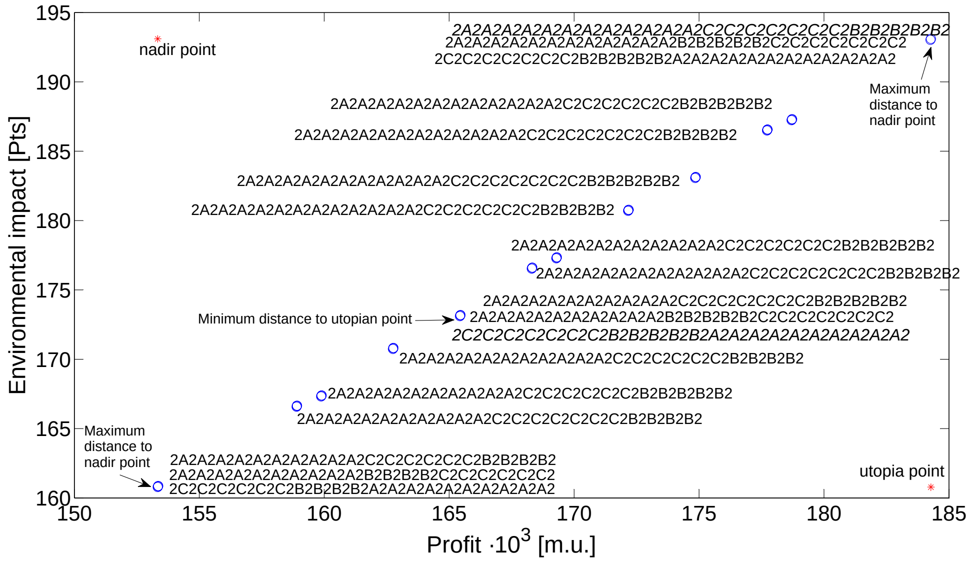


Figure 9.11: Case (iv). Solutions for two-objective optimization considering profit and environmental impact. Non-dominated solutions are encircled in blue (Pareto frontier); red stars are nadir, utopian points; and sequences in italics represent compromise solutions shown in Table 9.13. Note that for the sake of clarity some sequences are not explicitly shown in the Figure.

Table 9.14: Case (v). Utopian, nadir and solutions of compromise according to the different metrics considering profit and environmental impact for Example 2 (* defines utopia and $^-$ defines nadir). Distances are reported normalized.

$z^{profit} \cdot 10^3$ [m.u.]	z^{ei} [Pts]	Sequence	Distance utopian	Distance nadir
152.8718$^-$	161.2775 *	$2A^{(10)}_{(0\|9\|0)}2B^{(4)}_{(0\|3\|0)}2C^{(6)}_{(0\|4\|1)}2$	1.000	**1.000**
166.7887	185.3361	$1A^{(11)}_{(0\|10\|0)}1B^{(5)}_{(3\|1\|0)}2C^{(6)}_{(1\|4\|0)}1$	**0.637**	0.777
177.906 *	213.9418$^-$	$1A^{(12)}_{(6\|5\|0)}1C^{(7)}_{(1\|5\|0)}2B^{(5)}_{(3\|1\|0)}2$	1.000	**1.000**

optimized.

Table 9.14 shows the solution with minimum distance to the utopian point, which entails sequence $1A^{(11)}_{(0|10|0)}1B^{(5)}_{(3|1|0)}2C^{(6)}_{(1|4|0)}1$ (such string represents the ordered sequence of batches, where the capital letters A, B and C, stand for the product, and the numbers 1, 2 and 3 for the cleaning method being used, superscripts indicate the number of batches of each product and subscripts, the number of cleanings of each type:(1|2|3) inside each campaign of products). This solution is located approximately in the middle of the extreme solutions. The maximum distance to the nadir point is attained in the extreme solutions.

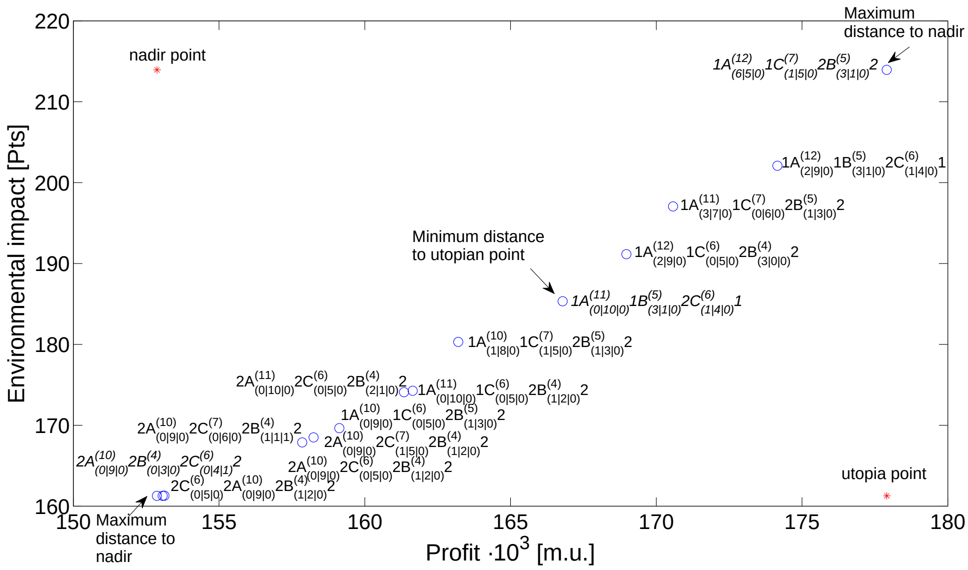


Figure 9.12: Case (v). Solutions for two-objective optimization considering profit and environmental impact. Non-dominated solutions are encircled in blue (Pareto frontier); red stars are nadir, utopian points; and sequences in italics represent compromise solutions shown in Table 9.14. Notation contains the number of batches of each product, the inter-product cleaning method, and the total number of changeovers of each type within each campaign of products (ordered from cleaning method 1 to 3).

9.9 Final remarks

This chapter provides a rigorous framework for the functional integration of the scheduling level. Specifically, the consideration of environmental concerns along with economic criteria in the scheduling of batch plants can be rigorously studied using multiobjective optimization. Although only economic and environmental criteria are posed, the proposed solution procedures are general to include other functional issues.

A strategy consisting of increasing the number of utopian hyperplane divisions has been proposed for reducing the computational burden of the problem.

In this problem, the normal constrained (NC) method provides a high quality description of the Pareto frontier; however, a high number of solutions has to be explored and generated in order to avoid missing Pareto optimal solutions. A strategy consisting of increasing the number of utopian hyperplane divisions has been proposed for reducing the computational burden of the problem. Furthermore, a hybrid evolutionary strategy that combines the strengths of genetic algorithms and local search has been presented to expedite the search for the Pareto set. This method outperforms deterministic global optimization algorithms at the expense of sacrificing theoretical guarantees of global optimality.

Thus, Pareto frontiers provide the decision maker with highly valuable information on the production schedule trade-off that naturally exists between economic and environmental criteria. This information sheds light into production and sequencing relationships that may not be obvious. This information sheds light into the effect of production and sequencing decisions on plant performance that may not be obvious to infer otherwise. In addition, it is highly important to thoroughly consider the decision makers' (e.g. plant managers) preferences to select a final solution to be implemented. In this context and depending on the selected objective functions, solutions were found entailing completely different scheduling decisions but showing the very similar economic and environmental performance. Different Pareto frontiers may be generated with the proposed method for different number and sequence of product batches, cleaning methods and objective functions.

The proposed approach for obtaining a compromise solution, which uses the concept of utopian and nadir points, allows to choose a single solution among the Pareto efficient ones. These solutions are balanced in terms of relative distance to reference points, namely the utopian and nadir of each Pareto frontier.

As for the effect of alternative process changeovers, it results clear that their effect over performance indicators is much smaller than other decisions such as the number of batches to be produced or the actual production sequence. However, the choice of the changeover method is highly important from a functional point of view. Depending on the nature of the objective function and the changeover characteristics, different changeover methods may be selected. As a whole, it is highly important to model all significant scheduling problem features in order to reach adequately integrated solutions.

Finally, from a LCA perspective, an overall assessment of the environmental impact of the production alternatives has been possible; which allows to reach more sensible schedules from an integrative point of view. Moreover, it seems more convenient to work with ratios impact/production rather than total impacts, at least in terms of rational use of resources. the choice of a particular ratio depends on the scenario under study (e.g. demand characteristics), and its type greatly affects the computational performance of the solution method.

9.10 Nomenclature

Sets and subsets

c	Cleaning modes between products.
g	Objective functions.
i	Batches.
p	Products (product S simulates plant 'still' state).

Parameters

N_W	Total number of changes in variable W_i allowed in the local search.
N_X	Total number of changes in variable $X_{ii'c}$ allowed in the local search.

Continuous variables

Mk	Objective function that aims at minimizing the makespan.
z^{ei}	Objective function that aims at minimizing the environmental impact.
z^{prod}	Objective function that aims at maximizing productivity.
z^{profit}	Objective function that aims at maximizing profit.
z^{rei}	Objective function that aims at minimizing the relative environmental impact.
μ^{best}	Vector of objectives for the best compromise solution.
μ^*	Vector of objectives that contains the optimal μ_g^* objectives (utopian point).
μ^0	Vector of objectives that contains the worst μ_g^0 objectives (nadir point).
μ	Vector that contains the μ_g objectives for a Pareto solution.

Binary variables

W_i	Production of batch i.
$X_{ii'c}$	Assignment of cleaning method c to changeover, if batch i is produced immediately before batch i'.

Algorithm notation

j	Iteration counter.
nd_0	Initial number of utopian line divisions.
nd_j	Number of utopian line divisions.
$np_j^{explored}$	Number of explored solutions at iteration j.
np_j^{PF}	Number of solutions that belong to the Pareto frontier at iteration j.
PF_0	Solutions that belong to the Pareto frontier at the first iteration.
PF_j	Solutions that belong to the Pareto frontier at iteration j.
PF^*	Pareto frontier solutions estimated by the proposed algorithm.
S_0	Solutions explored at the first iteration.
S_j	Solutions explored at iteration j.
tol	Tolerance value as termination criterion.

Part V

Conclusions and Outlook

Chapter 10

Conclusions and Future Work

10.1 Conclusions

This thesis stands for a step forward toward the integration of the decision making of batch short-term scheduling level in process industry from a structural and functional point of view. On the one hand, several approaches to the integration of scheduling with the process basic control level have been studied, proposing solution algorithms and assessing the benefits and challenges of the actual integration. On the other hand, the integration of economic and environmental issues in the scheduling problem has been tackled from a general multiobjective perspective, thus enabling the functional integration among the different decision levels. Clearly, the objectives posed in Chapter 1 have been dealt and the work developed has been thoroughly discussed along the different chapters. This section summarizes the most important conclusions derived from the research work.

Part I introduces the short-term scheduling problem and describes the existing integration challenges and the potential benefits of the integration with other decision levels, such as the improvement of overall plant operability and enterprise economic advantage. Thus, the current State-of-the-Art in Chapter 2 sheds light to the complexity and open issues in the scheduling problem focusing on the challenges posed by the aforementioned integration. These include, among others, modeling issues along with algorithmic and optimization developments required for the efficient solution of models defined across different spatial and temporal scales.

In Part II, it has been highlighted the need to improve current *scheduling models* for realistically representing industrial scenarios. These formulations call for the development of faster tailored algorithms capable of exploiting their particular structure. Specifically, emphasis has been placed on the synchronization of operations among stages, the adequate modeling of intermediate storage policies and the use of multiple alternative processing units for batch sequential processes. This approach capitalizes on the immediate and general precedence formulations, which constitute the core of the proposed mathematical models. Moreover, such models have been adequately extended to account for the following modeling aspects: (i) the selection among *alternative batch cleaning operations*, (ii) the *synchronization of equipment transfer operations* in multipurpose plants, (iii) the consideration of *variable batch processing rates* within product campaigns, and (iv) the *introduction of process dynamics* for operation timing purposes. These tools have enabled the integration of decision making levels discussed in the following parts of the thesis.

Structural integration. As a first step toward the introduction of process dynamics in scheduling problems, batch variable processing rates have been considered in a standard scheduling formulation. A single process variable, namely the processing rate, is integrated as a decision variable at the scheduling level, giving rise to linear models. This approach is particularly suited to semicontinuous plants, where the processing rates of the production campaigns can be adapted to the specific production requirements. Numerical examples presented in Chapter 6 have demonstrated that the use of batch variable processing rates within single product campaigns leads to solutions requiring less storage resources, thereby reducing the associated cost.

Part III presents the introduction of process dynamics at the scheduling level via two different approaches: (i) indirectly, that is, by considering cost functions of time; or (ii) directly: by embedding discretized dynamic equations into the scheduling formulation. The former approach is suited when process variables are fixed along the whole batch time interval, and has the advantage of leading to manageable scheduling models. In contrast, to handle time dependent variables, it is necessary to account for the detailed process dynamics, which gives rise to complex mixed integer non-linear scheduling models. Hence, a trade-off exists between model complexity and degree of accuracy, which must be accounted for when pursuing integrated decisions.

Chapter 7 presents an *indirect approach* to manage control decisions at the scheduling level through the characterization of the relationship between values of the free decision variables of the dynamic models, which are fixed along the whole batch time, and their impact over the cost function. An economic objective function enables to assess the influence of *operational variability* on process performance at the scheduling level *in terms of time*. Hence, the integration can be achieved *using economic criteria*. Different strategies can be adopted: i) a direct cost function over time embedded in the optimization model, such as a linear or quadratic regression, or a piecewise linear function; or ii) a heuristic approach that gradually refines the cost approximation by solving the scheduling problem iteratively. The latter strategies proved to be more adequate to deal with large size problems, since they give rise to linear models that are easier to handle than non linear functions (e.g. quadratic regression functions) and highly combinatorial problems (i.e., piecewise linear approximations). Moreover, *variable batch-to-batch times* are introduced at the scheduling level, which *increases process flexibility* and lead to better economic results compared to nominal fixed recipes. From numerical results, it can be observed how such improvement stems from the additional number of batches that can be processed, from the adjustment of operation start-ups and shut-downs, and from the coordination of dynamic stages with the other of process stages.

Moreover, Chapter 8 considers *full process dynamics* at the scheduling level. The process dynamic model is discretized by using orthogonal collocation on finite elements, and introduced at the scheduling level. As a result, a complete integration of process dynamics and scheduling decisions is achieved, but at the expense of formulating a complex mixed integer non-linear dynamic optimization problem which is hard to be solved. Results demonstrate that considering *time variable profiles* for the control variables at the scheduling level *improves the economic performance at the plant level*, affecting both batching and batch processing time decisions. This approach is suited to those problems in which the room for improvement compensates for the additional computational complexity.

On the whole, considering *process dynamics* at the scheduling level offers new opportunities to significantly *improve* the plant *economic performance* compared to the use of fixed recipe conditions, specially in those cases with highly restricted time horizons and tight demand constraints. However, such additional degree of freedom *increases* the *problem complexity*. Accordingly, intermediate strategies which apply simplified models may be also successfully adopted.

Functional integration. Finally, Part IV addresses the integration of the scheduling problem from a functional point of view. Several *economic and environmental objective functions* are considered using a multiobjective optimization approach.

On the one hand, the use of absolute and relative metrics has been investigated in this context, i.e. time or quantity related, demonstrating that they lead to different scheduling solutions. At the scheduling level, time related criteria are usually adopted to quantify the economic performance of the plant. In this sense, as presented in Chapter 4, profitability maximization is only equivalent to makespan minimization under certain conditions, if (i) the produced quantity is fixed, or (ii) all products are equivalent from a profitability point of view, that is, they have the same profit and production time along the different stages. Hence, the decision maker criteria must be well-known in advance and defined according to the overall plant goals and objectives pursued by other hierarchical levels within the company.

On the other hand, the scheduling problem has been further extended to consider environmental issues in addition to traditional economic factors. It has been shown how the decision maker may reach completely *different Pareto frontiers*, in terms of *number and sequence* of product *batches*, as well as in selected *cleaning methods* according to the selected objective functions. Hence, there is a clear need to capture the whole problem complexity and extend the existing scheduling models as proposed in Part II. Specifically, in this thesis the *trade-offs* arising between environmental and economic criteria are studied through the inspection of the Pareto frontier, which provides the decision maker with highly valuable *information about* production *schedule alternatives*. Furthermore, different *selection criteria* have been proposed *to identify a single compromise solution* among all possible points in the Pareto frontier. Such selection metrics are highly important, since only one solution can be finally adopted. In practice, experience may be the main decision driver, but metrics that objectively measure the quality of the Pareto solutions may be helpful.

Finally, a *hybrid optimization strategy* has been developed for the efficient solution of these problems taking advantage of the complementary strengths of metaheuristics and rigorous mathematical local search and is particularly useful for dealing with *large scale problems*. Specifically, a multi-objective genetic algorithm with mathematical programming based local search has been proposed and used to deal with industrial problems. The *reduction in solution time* is significant, and achieved at the expense of loosing theoretical guarantee of attaining the global optimum. Anyhow, the combination of rigorous local search within metaheuristics is a promising framework to generate near optimal solutions for large problems in lower CPU times.

10.2 Future work

This thesis shows promising results stemming from the integration of short-term scheduling decisions, but they are only a hint on the potential improvement which may be actually achieved. Moreover, several issues regarding the complexity of the integration problem require further research. Therefore, this section suggests some potential research lines identified along this work, some of which have even been tackled to some extent.

Scope generalization. Even though general strategies for integration have been proposed, they have been applied to specific problem structures. Therefore, further work is required to deal with the following issues:

- Sequential batch processes have been only studied; however, the conclusions about the benefits of integration could be easily extended to network batch processes. This will require the improvement of adequate mathematical formulations and the analysis of the problem solution with the proposed tools.

- A thorough analysis of the effectiveness of the existing scheduling formulations to deal with dynamic models should be carried out in order to compare their performance when scheduling and process basic control are simultaneously considered.

- Insomuch this work already reports substantial benefits of considering a single dynamic stage within the production process. It is therefore expected that the consideration of several dynamic stages in the same production process will additionally lead to larger profits obtained by further exploiting the flexibility associated with process timing, batch sequencing and resource allocation decisions.

- Further efforts should focus on the effect of the use of a specific economic indicator as objective function to be optimized in the scheduling problem. Specifically, the estimation of the actual costs derived from process conditions should be thoroughly studied.

- Merging the structural and functional integration considered in Part III and Part IV remains a challenge due to the problem scale and complexity.

- The proposed algorithms and integration principles may be extended to further consider other decision levels such as planning, or even in the larger scope of the design stage.

Extension to industrial sized problems. The large problem sizes in industrial practice call for further research on hybrid methods or metaheuristics optimization strategies. These approaches may overcome some of the numerical difficulties derived from the simultaneous decision making in short-term scheduling and process control.

Standardization and applicability. In order to achieve integration among the different decision levels, it is necessary to establish a common modeling framework. In this sense, work related to the use of semantic models, namely ontologies, has been carried out. Such framework shows a high potential allowing for an effective production plant modeling of the scheduling and control levels. The ultimate goal is to facilitate their integration by means of a common model for re-usability, usability and a shared information structure based on the ANSI/ISA 88 standard (Munoz *et al.*, 2011).

Uncertainty. The treatment of uncertainty at the scheduling level can be studied by considering the costs of rescheduling actions. Thus, considering variable recipes at the scheduling level for dealing with plant uncertainty can be clearly beneficial, since the process conditions can be better adapted to meet production targets.

Appendixes

Appendix A

Publications

This is a list of the works carried out so far within the scope of this thesis, in reversed chronological order.

A.1 Journals

A.1.1 Manuscripts published

Capón-García, E.; Moreno-Benito, M.; Espuña, A.; Puigjaner, L. Improved short-term batch scheduling flexibility using variable recipes. *Industrial & Engineering Chemistry Research*, ISSN: 0888-5885, 50 (9): 4983 – 4992 (**2011**).

You, F.; Pinto, J.M.; Capón, E.; Grossmann, I.E.; Arora, N.; Megan, L. Optimal Distribution-Inventory Planning of Industrial Gases: I. Fast Computational Strategies for Large-Scale Problems. *Industrial & Engineering Chemistry Research*, ISSN: 0888-5885, 50 (5):2910 – 2927 (**2011**).

Muñoz, E.; Capón-García, E.; Moreno-Benito, M.; Espuña, A.; Puigjaner, L. Scheduling and control decision-making under an integrated information environment. *Computers & Chemical Engineering*, ISSN: 0098-1354, 35 (5): 774 – 786 (**2011**).

Capón-García, E.; Espuña, A.; Puigjaner, L. Statistical and simulation tools for designing an optimal blanketing system of a multiple-tank facility. *Chemical Engineering Journal*, ISSN: 1385-8947, 152 (1): 122 – 132 (**2009**).

Capón-García, E.; Ferrer-Nadal, S.; Graells, M.; Puigjaner, L. An Extended Formulation for the Flexible Short-term Scheduling of Multiproduct Semicontinuos Plants. *Industrial & Engineering Chemistry Research*, ISSN: 0888-5885, 48 (4): 2009 – 2019 (**2009**).

Kopanos, G.M.; Capón-García, E.; Espuña, A.; Puigjaner, L. Costs for Rescheduling Actions: A Critical Issue for Reducing the Gap between Scheduling Theory and Practice.*Industrial & Engineering Chemistry Research*, ISSN: 0888-5885, 47 (22): 8785 – 8795 (**2008**).

Ferrer-Nadal, S.; Capón-García, E.; Méndez, C. A.; Puigjaner, L. Material Transfer Operations in Batch Scheduling. A Critical Modeling Issue. *Industrial & Engineering Chemistry Research*, ISSN: 0888-5885, 47 (20): 7721 – 7732 (**2008**).

A.1.2 Manuscripts Accepted

Capón-García, E.; Bojarski, A. D.; Espuña, A.; Puigjaner, L. Multiobjective optimisation of multiproduct batch plants scheduling under environmental and economic concerns. *AIChE Journal*, ISSN: 0001-1541, .

A.2 Conference proceeding articles

A.2.1 Articles in conference proceedings

Capón-García, E.; Moreno-Benito, M.; Muñoz, E.; Espuña, A.; Puigjaner, L. Scheduling and control decision-making under an integrated information environment. *European Symposium on Computer Aided Process Engineering* (ESCAPE-20), (S. Pieruzzi and G. Buzzi Ferraris, Eds.), 1195 – 1200, ISBN: 978-0-444-53569-6, 2010.

Capón-García, E.; Rojas, J.; Zhelev, T.; Graells, M. Operation scheduling of batch autothermal thermophilic aerobic digestion processes. *European Symposium on Computer Aided Process Engineering* (ESCAPE-20), (S. Pieruzzi and G. Buzzi Ferraris, Eds.), 1177 – 1182, ISBN: 978-0-444-53569-6, 2010.

Capón-Garcia, E.; Ferrer-Nadal, S.; Méndez, C.A.; Puigjaner, L. Uncovering the relevance of modeling transfer times in the short-term scheduling of multipurpose batch plants. *Foundations of Computer-Aided Process Operations* (FOCAPO), (M. Ierapetritou, M. Bassett and S. Pistikopoulos, Eds), 455 – 458, ISBN: 0965589111, 2008.

Capón, E.; Kopanos, G.; Bonfill, A.; Espuña, A.; Puigjaner, L. A novel proactive-reactive scheduling Approach in chemical multiproduct batch plants. *European Symposium on Computer Aided Process Engineering* (ESCAPE-18), (B. Braunschweig and X. Joulia, Eds.), 435 – 446, ISBN: 978-0-444-53227-5, 2008.

A.2.2 Other congresses and workshops

Capon-García, E.; Guillén-Gosálbez, G.; Espuña, A.; Puigjaner, L. MINLP for Dynamic Optimisation of multiproduct batch plant scheduling. *Exploratory Workshop on MINLP*, Seville, Spain, 2010.

You, F.; Grossmann, I.E.; Capon, E.; Pinto, J.M. Fast Computational Strategies for Large Scale Distribution-Inventory Planning of Industrial Gases Under Demand Uncertainty. *AIChE Annual Meeting 2009*, Nashville, USA, 2009.

Bojarski, A.D.; Capón, E.; Espuña, A.; Puigjaner, L. Batch Process Scheduling Optimization of Multiproduct Plants Under Simultaneous Environmental and Economical Considerations. *AIChE Annual Meeting 2009*, Nashville, USA, 2009.

Capón, E.; Guillén-Gosálbez, G.; Jiménez-Esteller, L.; Espuña, A.; Puigjaner, L. Designing the optimal supply chain for biodiesel production in Spain. *AIChE Annual Meeting 2008*, Philadelphia, USA, 2008.

Pérez-Fortes, M.; Capón-García, E.; Puigjaner, L. Statistical models for pollutants forecasting in urban areas. *11th Mediterranean Congress of Chemical Engineering*, Barcelona, Spain, 2008.

Capón-Garcia, E.; Guillén-Gosálbez, G.; Espuña, A.; Puigjaner, L. Optimizing a Supply Chain for Biofuels in Transport Applications: A case-study. *11th Mediterranean Congress of Chemical Engineering*, Barcelona, Spain, 2008.

Pérez-Moya, M.; Pérez-Fortes, M.; Capón, E.; Calvet, A.; Boada, E,; Graells, M. Competences evaluation criterion in the field of chemical engineering lab course. *11th Mediterranean Congress of Chemical Engineering*, Barcelona, Spain, 2008.

Capón-García, E.; Bojarski, A.; Espuña, A.; Puigjaner, L. Environmentally friendly approach towards batch process scheduling for phosphite products. *11th International Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction* (PRES), Prague, Czech Republic, 2008.

Ferrer-Nadal, S.; Capón-García, E.; Graells, M.; Puigjaner, L. Flexible management for the short-term scheduling of multiproduct semicontinuous plants. *11th International Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction* (PRES), Prague, Czech Republic, 2008.

Graells, M.; Perez, M.; Perez, MM.; Espuña, A.; Capon, E. Microteaching: Flexible training methodology. *New Challenges in Engineering Education and Research* Pécs-Budapest, Hungary, 2008

Pérez-Moya, M.; Pérez-Fortes, M.; Capón, E.; Calvet, A.; Boada, E.; Graells, M. Active Learning Evaluation in the framework of Lab Project Management. *New Challenges in Engineering Education and Research* Pécs-Budapest, Hungary, 2008

Kopanos, G.; Capón, E., Bonfill, A.; Espuña, A.; Puigjaner, L. Novel Proactive-Reactive scheduling approach in chemical multiproduct batch plants. *AIChE Annual Meeting 2007*, Salt Lake City, USA, OMNIPRESS, pp. 372, ISBN: 978-0-8169-1050-2, 2007.

Capón, E.; Parra, C.; Espuña, A.; Puigjaner, L. Optimal Blanketing System of a solvent-storage for multiple tank facility. *AIChE Annual Meeting 2007*, Salt Lake City, USA, OMNIPRESS, pp. 119, ISBN: 978-0-8169-1050-2, 2007.

Capon, E.; Espuña, A.; Puigjaner, L. Enhanced performance of ant colony algorithm compared with other methaurisitics in batch scheduling. *6th European Congress* *of Chemical Engineering* (ECCE-6), Copenhagen, Denmark, Norhaven Books, pp. 511 – 512, 2007.

Capon, E.; Bonfill, A.; Espuña, A.; Puigjaner, L. Scheduling of flexible multipurpose back-up chemical/assemply process systems. *6th European Congress of Chemical Engineering* (ECCE-6), Copenhagen, Denmark, Norhaven Books, pp. 475 – 476, 2007.

Appendix B

Review of short-term scheduling models

This appendix presents a review of three of the most popular continuous time based scheduling formulations available in the literature, namely STN, RTN and unit-specific time event. The objective is to show the mathematical constraints of these models. For the sake of brevity, resource consumption, such as energy or manpower, constraints have been omitted, but they are formulated in the original papers.

B.1 State-Task-Network based continuous time formulation

There have been many approaches to the STN-based continuous models in the last years, such as Mockus and Reklaitis (1999b), Giannelos and Georgiadis (2002) and Maravelias and Grossmann (2003b). The latter 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 n occurring at unknown time T_n, $n = 1, 2, ..., N$, where N is the set of time points. 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 T_n. However, the ending time does not necessarily have to coincide with the occurrence of a time point n, 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 Ws_{in} and Wf_{in}, to denote at which time point a given task i starts and finishes. Moreover, it is necessary to define the batch size of a given task at the beginning Bs_{in}, finishing Bf_{in} and during the processing time of the task Bp_{in}. Variable S_{sn} represents the quantity of resource s available at time point n. Finally, if alternative units can be used to perform a given task, task duplication is needed to take this into account.

B.1.1 Assignment constraints

These constraints control the assignment of the different tasks to the different units. It must be noted that each task i is thought to be able to be produced in a definite equipment j. Constraints B.1 and B.2 impose that for each event point n and equipment j, at most one task i can take place. Moreover, constraint B.3 obliges that any task that is started, must be finished. Nevertheless, it is necessary equation B.4 to define 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, it is assumed that no task can finish at $t = 0$, and no task can start at the last event point.

$$\sum_{i \in I_j} Ws_{in} \leq 1 \qquad \forall j, n \tag{B.1}$$

$$\sum_{i \in I_j} Wf_{in} \leq 1 \qquad \forall j, n \tag{B.2}$$

$$\sum_{n} Ws_{in} = \sum_{n} Wf_{in} \qquad \forall i \tag{B.3}$$

$$\sum_{i \in I_j} \sum_{n' \leq n} (Ws_{in'} - Wf_{in'}) \leq 1 \qquad \forall j, n \tag{B.4}$$

$$Wf_{i0} = 0 \qquad \forall i \tag{B.5}$$

$$Ws_{in} = 0 \qquad \forall i, n = |N| \tag{B.6}$$

B.1.2 Timing constraints

From the idea of event points, the starting time of the horizon takes place at the first event point ($n = 1$). In addition, the finishing time is that corresponding to the last event point ($n = |N|$). In between, it is necessary to order in increasing time the different time points of the formulation. These premises are enforced through equations B.7 to B.9.

$$T_{n=1} = 0 \tag{B.7}$$

$$T_{n=|N|} = z^{Mk} \tag{B.8}$$

$$T_{n+1} \geq T_n \qquad \forall n, n < |N| \tag{B.9}$$

The formulation allows variable time duration tasks, as can be seen from constraint B.10. Moreover, this formulation only defines the duration of a task only at those event points when it starts. The finishing time of a given task starting at event point n is defined through constraints B.11 and B.12.

$$D_{in} = \alpha_i \cdot Ws_{in} + \beta_i \cdot Bs_{in} \qquad \forall i, n \tag{B.10}$$

$$Tf_{in} \leq Ts_{in} + D_{in} + H\,(1 - Ws_{in}) \qquad \forall i, n \tag{B.11}$$

$$Tf_{in} \geq Ts_{in} + D_{in} - H\left(1 - Ws_{in}\right) \qquad \forall i, n \tag{B.12}$$

Constraint B.13 combined with B.9 states that for a given task i, its finishing time at event point n is equal to finishing time at event point $n-1$, unless it starts at event point n, that is, unless $Ws_{in} = 1$. In order to reduce search space and computational times, equation B.14 is introduced. It defines the difference of the finishing time of a task i at event point n with its previous event point, to be greater or equal to the duration of the task at event point n.

$$Tf_{in} - Tf_{in-1} \leq H \cdot Ws_{in} \qquad \forall i, n, n > 1 \tag{B.13}$$

$$Tf_{in} - Tfin - 1 \geq D_{in} \qquad \forall i, n, n > 1 \tag{B.14}$$

Ts_{in} can be eliminated from the formulation through equation B.15 because task starting times fits with event point times. In order to match the finishing time of a task with the event point at which it finishes, constraints B.16 and B.17 are formulated. In general, tasks can end at or before event point n; however, those with zero-wait time intermediate storage policy are enforced to end exactly at event point n, through constraint B.17.

$$Ts_{in} = T_n \qquad \forall \text{i}, \forall \text{n} \tag{B.15}$$

$$Tf_{in-1} \leq T_n + H\left(1 - Wf_{in}\right) \qquad \forall i, n, n > 1 \tag{B.16}$$

$$Tf_{in-1} \geq T_n - H\left(1 - Wf_{in}\right) \qquad \forall i \in I^{ZW}, n, n > 1 \tag{B.17}$$

B.1.3 Batch size constraints

The batch size of a given task in a period time must lie between the upper and lower limits of such task. This is enforced for starting, ending and processing event points of the different tasks through equations B.18 to B.21. Moreover, it is necessary to define the mass balance for batches, that is, the starting and produced amount of processing task i at $n-1$ must be equal to the finishing and produced amount at event point n. This is accomplished by constraint B.22.

$$B_i^{MIN} Ws_{in} \leq Bs_{in} \leq B_i^{MAX} Ws_{in} \qquad \forall i, n \tag{B.18}$$

$$Bi^{MIN} Wf_{in} \leq Bf_{in} \leq B_i^{MAX} Wf_{in} \qquad \forall i, n \tag{B.19}$$

$$B_i^{MIN} \left(\sum_{n'<n} Ws_{in'} - \sum_{n' \leq n} Wf_{in'} \right) \leq Bp_{in} \qquad \forall i, n \tag{B.20}$$

$$Bp_{in} \leq B_i^{MAX} \left(\sum_{n'<n} Ws_{in'} - \sum_{n' \leq n} Wf_{in'} \right) \qquad \forall i, n \tag{B.21}$$

$$Bs_{in-1} + Bp_{in-1} = Bp_{in} + Bfin \qquad \forall i, n \tag{B.22}$$

The produced and consumed amounts of state s in the processing task i at event point n are computed through constraints B.23 to B.26. The first two constraints are aimed at states that are inputs of the processing tasks, whereas the latter are for the outputs. Under no circumstances, the processed tasks must exceed the maximum quantity allowed. This is enforced through constraints B.24 and B.26.

$$B^I_{isn} = \rho_{is} Bs_{in} \qquad \forall i, n, s \in SI_i \tag{B.23}$$

$$B^I_{isn} \leq B^{MAX}_i \rho_{is} Ws_{in} \qquad \forall i, n, s \in SI_i \tag{B.24}$$

$$B^O_{isn} = \rho_{is} Bf_{in} \qquad \forall i, n, s \in SO_i \tag{B.25}$$

$$B^O_{isn} \leq B^{MAX}_i \rho_{is} Wf_{in} \qquad \forall i, n, s \in SO_i \tag{B.26}$$

B.1.4 Mass balance/storage constraints

It is necessary to perform the material balances for the different states in the problem. Constraint B.27 imposes that the amount available at time point n plus the amount sold (SS_{sn}) is equal to the available amount at the previous period adjusted by that amount produced and consumed at the current period.

$$S_{sn} + SS_{sn} = S_{sn-1} + \sum_{i \in I^p_s} B^O_{isn} - \sum_{i \in I^c_s} B^I_{isn} \qquad \forall s, n > 1 \tag{B.27}$$

In addition, it must be taken into account that the available amount of state s must not exceed the established capacity (B.28). This is advantageous for defining different storage policies. However, it is also necessary to define a new binary variable V_{jsn} for those units j that perform as shared-storage tanks. By adding this new variable, it is possible to enforce through constraints B.29 and B.30 that at most one state can be stored at that tank at a time, and the tank capacity for that state cannot be exceeded.

$$S_{sn} \leq C_s \qquad \forall n, s \tag{B.28}$$

$$\sum_{s \in S(j)} V_{jsn} \leq 1 \qquad \forall j \in JT, n \tag{B.29}$$

$$S_{sn} \leq C_j V_{jsn} \qquad \forall j \in JT, n, s \in S_j \tag{B.30}$$

B.1.5 Demand constraints

The aim of the production scheduling is to reach the demand of the different products. This is defined by constraint B.31.

$$\sum_n SS_{sn} = d_s \qquad \forall s \qquad \text{or} \qquad \sum_n SS_{sn} \geq d_s \qquad \forall s \tag{B.31}$$

B.1.6 Tightening constraints

In order to avoid weak relaxations in the solution of this formulation, three different tightening constraints are proposed in the original paper, which allows to achieve faster the solution.

B.1.7 Objective function

The objective function consists of minimizing the makespan of the production schedule.

$$\min \qquad z^{Mk} \tag{B.32}$$

The number of time intervals is a critical issue for all continuous-time models. The most common selected approach consists of increasing the number of time intervals from a relative small number until no improvement in the objective function is achieved.

B.2 Resource-Task-Network based continuous time formulation

The RTN-representation was first introduced by Pantelides (1994). Further improvements were achieved by Castro *et al.* (2001). The considered RTN formulation is based on work developed by Castro *et al.* (2004).

This approach adopts a common time grid for all resources. Event points are numbered from 1 to N, spanning the time from 0 to horizon time H. As other continuous time formulations, the length of each time interval is unknown and is to be determined. In addition, a parameter Δn is used to define the maximum 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 of Δn 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 ending of a given task.

Major assumptions of this approach are: (i) processing units are considered individually, one resource is defined for each available unit, and (ii) only one task can be performed in any given equipment resource at any time.

The resource-task network process representation considers two types of items: resources and tasks. 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). It is assumed that 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 allocation variable is defined as $\bar{\mathrm{N}}_{\mathrm{inn'}}$ which is equal to 1 whenever task i starts at time point n and finishes at or before time point $n' > n$. Therefore, the starting and finishing time points for a given task i 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 R_{rn} represents the excess amount of resource r at time point n. This variable contains the value of a given resource r at a each time point. Finally, if alternative units can be used to perform a given task, task duplication is needed to take this into account.

B.2.1 Timing constraints

A set of global time points is predefined where the first time point takes place at the beginning whereas the last one at the end of the time horizon. This is represented by constraints B.33 and B.34.

$$T_1 = 0 \tag{B.33}$$

$$T_{|N|} = z^{Mk} \tag{B.34}$$

Timing constraints B.35 impose that the difference between the absolute times of any two event points (n and n') is allowed to be either equal to or greater than the processing time of all tasks starting and ending at those same event points. Given that only one task can be executed in any given equipment unit at a certain time, all tasks that take place in the same equipment resource ($r \in R^{EQ}$) are considered in the same constraint, in order to improve the resolution efficiency. Moreover, if there are zero-wait batch tasks, constraint B.36 obliges that the difference between starting and finishing time is exactly equal to processing time.

$$\begin{gathered} T_{n'} - T_n \geq \sum_{i \in I^b} \overline{\mu}_{r,i} \left(\alpha_i \bar{N}_{i,n,n'} \right) \\ \forall r \in R^{EQ}, \forall n, n', n < n' \leq \Delta n + n, n \neq |N| \end{gathered} \tag{B.35}$$

$$\begin{gathered} T_{n'} - T_n \leq H \left(1 - \sum_{i \in I^b, i \in I^{ZW}} \overline{\mu}_{r,i} \bar{N}_{i,n,n'} \right) + \sum_{i \in I^b, i \in I^{ZW}} \overline{\mu}_{r,i} \left(\alpha_i \bar{N}_{i,n,n'} \right) \\ \forall r \in R^{EQ}, \forall n, n', n < n' \leq \Delta n + n, n \neq |N| \end{gathered} \tag{B.36}$$

B.2.2 Balance constraints

The amounts of each resource consumed or produced at the start and end of a task are assumed to be proportional to the binary extent of that task. The total amount of resource r consumed at the start of task i beginning at event point n and ending at event point n' is proportional to $\bar{\mathrm{N}}_{\mathrm{inn}'}$ by μ_{ri}, and the amount produced at its end is proportional by $\overline{\mu}_{ri}$. Therefore, constraint B.37 represents the excess resource balance. The amount of resource r at time period n is equal to the amount at previous period, plus the amounts produced or consumed by tasks ending or starting at time period n.

$$\begin{gathered} R_{rm} = R_r^0\big|_{n=1} + R_{rn-1}\big|_{n>1} + \sum_{i \in I^b} [\sum_{\substack{n' \in N \\ n < n' \leq \Delta n + n}} \left(\mu_{ri} \bar{N}_{inn'} \right) + \\ \sum_{\substack{n' \in N \\ n - \Delta n \leq n' < n}} \left(\overline{\mu}_{ri} \bar{N}_{inn'} \right)] + \sum_{i \in I^s} \left(\mu_{ri} \bar{N}_{inn+1} + \overline{\mu}_{ri} \bar{N}_{in-1n} \right) \quad \forall n, r \in R \end{gathered} \tag{B.37}$$

B.2.3 Storage constraints

Storage tasks are represented by an additional constraint. In order to take into account shared storages, constraint B.38 ensures that for an excess amount of material resource r between a minimum and maximum capacity, the storage task $i \in I^S$ is activated.

$$V_i^{MIN}\bar{N}_{inn+1} \leq \sum_{r \in I_r^s} R_{rn} \leq V_i^{MAX}\bar{N}_{inn+1} \qquad \forall i \in I^s, \forall n, n \neq |N| \tag{B.38}$$

B.2.4 Capacity constraints

Moreover, it is necessary to consider that excess amount for any resource at any time period must lie between its predefined minimum and maximum capacity. This is accomplished through constraint B.39. This constraint is also useful to define different storage policies, namely if a zero value is assigned to material resources, then a no intermediate storage policy is applied, whereas if a big value is assigned, an unlimited intermediate storage is concerned. The limited intermediate storage is represented by assigning R_r^{MAX}, the maximum storage capacity for material resource r.

$$R_r^{MIN} \leq R_{rn} \leq R_r^{MAX} \qquad \forall n, r \in R \tag{B.39}$$

B.2.5 Demand constraints

Given the fact that a definite amount of final products is to be produced, constraint B.40 states that the sum of final product excess resource for horizon time must be equal to or greater than demand.

$$\sum_n R_{rn} = d_r \quad \forall r \in R^{final} \quad \text{or} \quad \sum_n R_{rn} \geq d_r \quad \forall r \in R^{final} \tag{B.40}$$

B.2.6 Objective function

The objective function of reducing makespan is represented by equation B.41.

$$\min z^{Mk} \tag{B.41}$$

B.3 Unit-Specific Time Event

The original idea of unit-specific events was firstly presented by Ierapetritou and Floudas (1998) and then developed 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 events 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 points representation. Although this representation is mainly oriented to batch network process, it can easily deal with sequential processes.

B.3.1 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 point event. Two additional set of binary variables denoting if a task i starts (Ws_{in}) or finishes (Wf_{in}) at an event point n are related to W_{in} by constraint B.43. The fact that every task has to start and finish during the time horizon is ensured by constraint B.44. Constraint B.45 guarantees that one task i may start at event point n if all tasks i beginning earlier have already finished, while constraint B.46 express that a task i may finish at event point n if it has been started at a previous event point n' and has not finished yet.

$$\sum_{i \in I_j} W_{in} \leq 1 \qquad \forall j, n \tag{B.42}$$

$$\sum_{n' \leq n} Ws_{in'} - \sum_{n' < n} Wf_{in'} = W_{in} \qquad \forall i, n \tag{B.43}$$

$$\sum_{n} Ws_{in} = \sum_{n} Wf_{in} \qquad \forall i \tag{B.44}$$

$$Ws_{in} \leq 1 - \sum_{n' < n} Ws_{in'} + \sum_{n' < n} Wf_{in'} \qquad \forall i, n \tag{B.45}$$

$$Wf_{in} \leq \sum_{n' < n} Ws_{in'} - \sum_{n' < n} Wf_{in'} \qquad \forall i, n \tag{B.46}$$

B.3.2 Batch size constraints

The model also takes into account the batching constraints. Therefore, equation B.47 establishes the maximum and minimum batch size. In addition, equation B.47 gives the maximum available storage capacity for storage tasks.

$$B_i^{MIN} W_{in} \leq B_{in} \leq B_i^{MAX} W_{in} \qquad \forall i, n \tag{B.47}$$

$$B_{i^{st}n}^{st} \leq V_{i^{st}}^{MAX} \qquad \forall i^{st}, n \tag{B.48}$$

Constraints B.47 and B.48 define the batch size for those tasks that take more than one time period to be completed.

$$B_{in} \leq B_{in-1} + B_i^{MAX} \left(1 - W_{in-1} + Wf_{in-1}\right) \qquad \forall i, n > 1 \tag{B.49}$$

$$B_{in} \geq B_{in-1} - B_i^{MAX} \left(1 - W_{in-1} + Wf_{in-1}\right) \qquad \forall i, n > 1 \tag{B.50}$$

On the other hand, from constraint B.49 to B.51 to B.56, the initial and final batch sizes are defined, in order to accurately perform the mass balances.

$$Bs_{in} \leq B_{in} \qquad \forall i, n \tag{B.51}$$

$$Bs_{in} \geq B_{in} - B_i^{MAX} \left(1 - Ws_{in}\right) \qquad \forall i, n \tag{B.52}$$

$$Bs_{in} \leq B_i^{MAX} Ws_{in} \qquad \forall i, n \tag{B.53}$$

$$Bf_{in} \leq B_{in} \qquad \forall i, n \tag{B.54}$$

$$Bf_{in} \geq B_{in} - B_i^{MAX} \left(1 - Wf_{in}\right) \qquad \forall i, n \tag{B.55}$$

$$Bf_{in} \leq B_i^{MAX} Wf_{in} \qquad \forall i, n \tag{B.56}$$

B.3.3 Mass balance constraints

Constraints B.57 defines the mass balance. That is, the available quantity of state s at event point n is equal to that corresponding to the previous time point, less the quantity sold at the current time period, as well as the difference in storages, and the quantities produced and consumed for that state at the previous and current time points respectively.

$$\begin{aligned} S_{sn} = S_{sn-1} - SS_{sn} + \sum_{i \in I_s^p} \rho_{is} Bf_{in-1} - \sum_{i \in I_s^c} \rho_{is} Bs_{in} + \sum_{i^{st} \in I_s^{st}} B_{in-1}^{st} - \\ - \sum_{i^{st} \in I_s^{st}} B_{in}^{st} \forall s, \forall n, n > 1 \end{aligned} \tag{B.57}$$

Finally, it is also necessary to impose that demand requirements must be accomplished. It can be expressed by any expression of constraint B.58.

$$\sum_n SS_{sn} = d_s \quad \forall s \quad or \quad \sum_n SS_{sn} \geq d_s \quad \forall s \tag{B.58}$$

B.3.4 Timing constraints

Two new sets of continuous variables are defined within the duration constraints applied to the processing tasks. These variables are Ts_{in} and Tf_{in} which denote respectively the starting and finishing time of a task i executed at unit j at time event n. The finishing time has to be greater than or equal to the starting time of a task i at unit j at time event n (constraint B.59). Alternatively, constraint B.60 makes equal starting and finishing time if the same task i does not take place at event point n, otherwise it is relaxed. The combination of constraint B.61 and sequencing constraints B.62 and B.63 ensures the finish time at $n-1$ to be equal to the starting time at n if task i is active and must extend to the following event n, otherwise these constraints are relaxed.

$$Tf_{in} \geq Ts_{in} \qquad \forall i, n \tag{B.59}$$

$$Tf_{in} \leq Ts_{in} + HW_{in} \qquad \forall i, n \tag{B.60}$$

$$Ts_{in} \leq Tf_{i(n-1)} + H\left(1 - W_{i(n-1)} + Wf_{i(n-1)}\right) \qquad \forall i, n > 1 \tag{B.61}$$

Constraint B.62 works together with constraint 50 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 the tasks which can not both process and store material.

$$\begin{aligned} Tf_{in'} - Ts_{in} \geq pt_i Ws_{in} - H\left(1 - Ws_{in}\right) - H\left(1 - Wf_{in'}\right) - \\ -H\left(\sum_{n \leq n'' \leq n'} Wf_{in''}\right) \qquad \forall i, (n, n')|n \leq n' \end{aligned} \tag{B.62}$$

$$\begin{aligned} Tf_{in'} - Ts_{in} \leq pt_i Ws_{in} + H\left(1 - Ws_{in}\right) + H\left(1 - Wf_{in'}\right) + \\ +H\left(\sum_{n \leq n'' \leq n'} Wf_{in''}\right) \qquad \forall i \notin I^{ps}, (n, n')|n \leq n' \end{aligned} \tag{B.63}$$

Finally, constraint B.64 is applied to tasks performing as storage.

$$Tf^{st}_{i^{st}n} \geq Ts^{st}_{i^{st}n} \qquad \forall i^{st}, n \tag{B.64}$$

B.3.5 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.

$$Ts_{in} \geq Tf_{i(n-1)} \qquad \forall i, n > 1 \tag{B.65}$$

$$Ts_{in} \geq Tf_{i'(n-1)} - H(1 - W_{i'(n-1)}) \qquad \forall i \neq i', j|(J_i, J_{i'}), n > 1 \tag{B.66}$$

$$\begin{aligned} Ts_{in} \geq Tf_{i'(n-1)} - H(1 - Wf_{i'(n-1)}) \\ \forall s, i \in I^c_s, i' \in I^p_s, (j, j')|J_i, J_{i'}, j \neq j', n > 1 \end{aligned} \tag{B.67}$$

Consecutive tasks i and i' under zero-wait transfer policy are sequenced by constraint B.68.

$$\begin{aligned} Ts_{in} \leq Tf_{i'(n-1)} - H(2 - Wf_{i'(n-1)} - Ws_{in}) \\ \forall s \in S, \forall i \in I^c_s, \forall i' \in I^p_s, (j, j')|J_i, J_{i'}, j \neq j', n > 1 \end{aligned} \tag{B.68}$$

B.3.6 Sequencing constraint: Storage tasks

The following constraints (B.69 to B.73) are stated for the additional sequencing storage tasks i^{st} that have to be separately defined from the processing tasks. Specific variables $Ts^{st}_{i^{st}n}$ and $Tf^{st}_{i^{st}n}$ are used to denote the time at which storage tasks i^{st} start and finish. Shared storage is modelled by specifying the set of storage tasks so that multiple states s are linked to each storage task.

$$Ts_{in} \geq Tf^{st}_{i^{st}n-1} \qquad \forall s, \forall i \in I^c_s, i^{st}, \forall n > 1 \tag{B.69}$$

$$Ts_{in} \leq Tf^{st}_{i^{st}n-1} + H(1 - Ws_{in}) \qquad \forall s, \forall i \in I^c_s, \forall i^{st} \in I^{st}_s, n > 1 \tag{B.70}$$

$$Ts^{st}_{i^{st}n} \geq Tf_{i'(n-1)} - H(1 - Wf_{i'(n-1)}) \qquad \forall s, i' \in I^p_s, i^{st} \in I^{st}_s, n > 1 \tag{B.71}$$

$$Ts^{st}_{i^{st}n} \leq Tf_{i'(n-1)} + H(1 - Wf_{i'(n-1)}) \qquad \forall s, i' \in I^p_s, i^{st} \in I^{st}_s, n > 1 \tag{B.72}$$

$$Ts^{st}_{i^{st}n} = Tf^{st}_{i^{st}(n-1)} \qquad \forall i^{st} \in I^{st}_s, n > 1 \tag{B.73}$$

B.3.7 Tightening constraints

Tighter relaxed solutions of this formulation can be obtained by applying the tightening constraints introduced by Maravelias and Grossmann (2003b) which enhance also the performance of this formulation.

B.3.8 Objective function

$$\min z^{Mk} \geq Tf_{in} \qquad \forall i, n \tag{B.74}$$

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 in face of the processing times.

B.4 Nomenclature

Subscripts

i, i'	Tasks
i^{st}	Storage tasks
j, j'	Equipment units
n, n', n''	Event points
r	Resources
s	States

Sets

I_j	Tasks i that can be scheduled in equipment unit j.
I^b	Batch tasks.
I^s	Storage tasks.
I^{ZW}	Tasks with zero-wait storage policy.
SI_i	States consumed in task i.
SO_i	States produced from task i.
JT	Shared storage tanks.
S_j	States that can be stored in shared storage tank j.
I_s^{st}	Storage tasks for state s.
I^{ps}	Tasks that are processing or storing.
I_s^p	Tasks that produce state s.
I_s^c	Tasks that consume state s.
J_i	Units that can perform task i.
R_r^{EQ}	Equipment resources (storage tanks not included).
R_r^{INT}	Resources r corresponding to intermediate products.
R_r^{final}	Resources r corresponding to final products.
Δn	Maximum number of event points between the beginning and ending of a given task.
N	Total number of event points.

Parameters

pt_i	Processing time of task i.
H	Time horizon.
M	Big-M value.
α_i	Fixed duration of a task i.
β_i	Variable duration of a task i.
ρ_i	Mass balance coefficient fort the consumption/production 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.
$\bar{\mu}_{ri}, \mu_{ri}$	Amount produced/consumed of resource r in task i.
R_r^{MIN}, R_r^{MAX}	Minimum,maximum availability of resource r.
B_i^{MIN}, B_i^{MAX}	Lower/upper bounds on the batch size of task i.
V_i^{MIN}, V_i^{MAX}	Lower/upper bounds on storage capacity for task i.

Continuous variables

D_{in}	Duration of task i starting at 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_n	Absolute time of time point n (starting of time point n and ending of $n-1$).

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.
$Ts^{st}_{i^{st}n}$	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.
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.
S_{sn}	Amount of state s available at time point n.
SS_{sn}	Sales of state s at time point n.
R_{rn}	Excess amount of resource r at time point n.
R^0_r	Initial amount of resource r.
z^{Mk}	Makespan.

Binary variables

Ws_{in}	Equals to 1 if task i starts at time point n, and 0 otherwise.
Wf_{in}	Equals to 1 if task i ends at time point n, and 0 otherwise.
Wp_{in}	Equals to 1 if task i is processed at time point n, and 0 otherwise.
W_{in}	Equals to 1 if task i activated at time point n, and 0 otherwise.
V_{jsn}	Equals to 1 if state s is stored in shared tank j during time period n, and 0 otherwise.
$\bar{\mathrm{N}}_{inn'}$	Equals to 1 if task i starts at time point n and finishes at time point n', and 0 otherwise.

Appendix C

Data for Examples 1 and 2 in Chapter 5

This appendix presents the data used for different Examples presented in Chapter 5 of this thesis. The motivating example of a multiproduct batch plant producing acrylic fibers is thoroughly described in Appendix D.

Table C.1: Procesing times and routes for Example 1 in Chapter 5

Products	Stage 1		Stage 2		Stage 3		Stage 4	
	Unit	Time [h]	Unit	Time [h]	Unit	Time [h]	Unit	Time [h]
A	U1	15	U3	8	U4	12		
B	U1	10	U2	20	U3	5	U4	13
C	U3	9	U2	7	U1	20		
D	U4	5	U3	17	U2	7		

Table C.2: Procesing times and routes for Example 2 in Chapter 5

Products	Stage 1		Stage 2		Stage 3		Stage 4		Stage 5		Number of batches
	Unit	Time [h]	Unit	Time[h]	Unit	Time [h]	Unit	Time [h]	Unit	Time [h]	
A	R1	3.5	F	2.5							2
	R2	7									
B	G	3.9	R1	4.1	P	2.9					2
			R2	8.2	Z	3.2					
C	F	4	R1	3.8	G	4.5	P	3	Z	2.9	2
			R2	7.6							
D	Z	5.7	F	3							2
E	P	2.5	F	3							2

Appendix D

Data for the multiproduct batch plant producing acrylic fibers

This appendix presents the data used for an illustrative case study consisting a multi-product fiber batch plant, which was originally posed by Grau *et al.* (1996). The example is introduced in Chapter 4, and further examined along Chapters 9 of this thesis.

D.1 Plant description and products recipes

The case study consists of a multi-product batch process plant that produces three acrylic fiber formulations by a suspension polymerization process (Figure D.1) requiring 14 processing stages. Due to minimisation of inventory costs, the possible storage of polymer (considered as intermediate product) after stages deaeration (stages 11, 12) has been disregarded and polymer extrusion (stage 13) is performed right after polymer deareation is done. Production recipes contain a detailed description of the product batch sizes (Grau *et al.*, 1996) , as well as operational times (Table D.1) and energy demands (Grau *et al.*, 1996) of each production stage. Production costs and sales prices are shown in Figure D.2(a). Batch sizes and an example of an industrial demand is given in Table D.2.

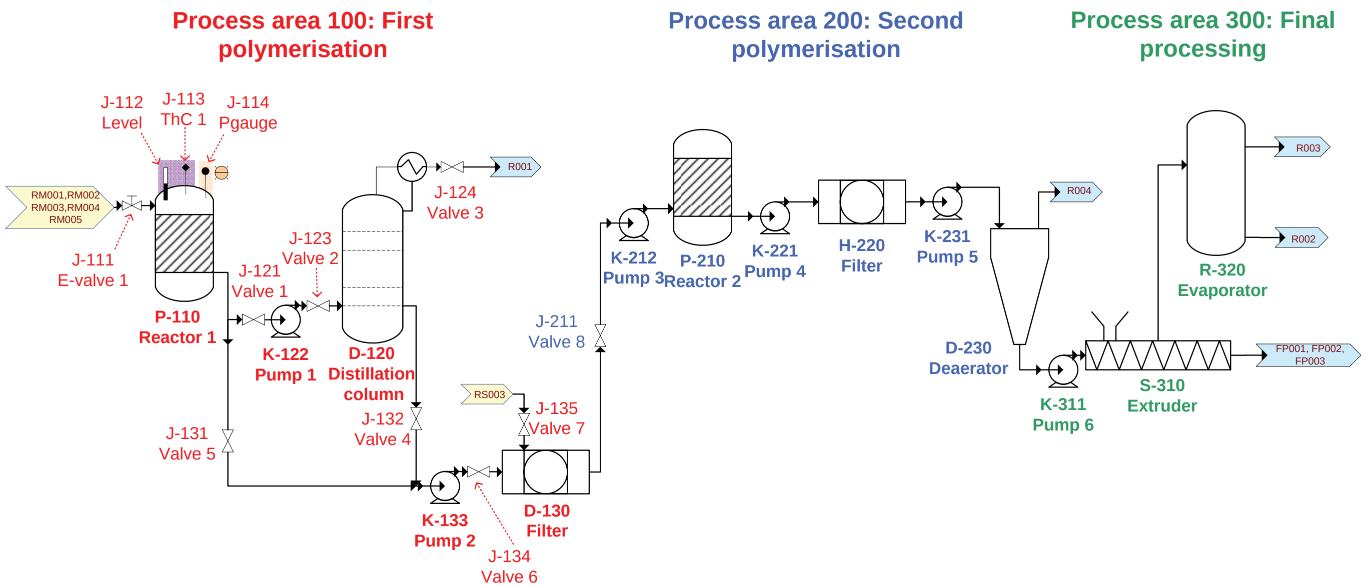


Figure D.1: Detailed flowsheet of the production process of acrylic fibers manufacturing.

Table D.1: Operation times and equipment associated with each stage for all possible produced products [h].

		Product A					Product B					Product C				
Stage	Equipment	P	L	O	U	TOT	P	L	O	U	TOT	P	L	O	U	TOT
1	R1	0.2	0	2	0.3	2.5	0.2	0	3	0.75	3.95	0.2	0	1	0.3	1.5
2	P1	0.2	0	0.3	0	0.5	0	0	0	0	0	0.2	0	0.3	0	0.5
3	C1	0.5	0.3	2.5	0.75	4.05	0	0	0	0	0	0.5	0.3	2	0.75	3.55
4	P2	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95
5	F1	0.5	0	0.75	0	1.25	0.5	0	0.75	0	1.25	0.5	0	0.75	0	1.25
6	P3	0.2	0	0.75	0	1.25	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95
7	R2	0.3	0.75	1	0.75	2.8	0.3	0.75	0.75	0.75	2.55	0.3	0.75	0.5	0.75	2.3
8	P4	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95
9	F2	0.5	0	0.75	0	1.25	0.5	0	0.75	0	1.25	0.5	0	0.75	0	1.25
10	P5	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95
11	D1	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95
12	P6	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95
13	E1	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95	0.2	0	0.75	0	0.95
14	V1	0.3	0.75	3.5	0	4.55	0.3	0.75	3	0	4.05	0.3	0.75	1.5	0	2.55

Table D.2: Product batch sizes and example of industrial demand.

Fiber	Batch size [ton/batch]	Demand [ton]
A	2.5	30
B	1.8	9
C	1.5	10.5

D.2 Data for Chapter 7: Detailed model of the polymerization stage

The detailed model of the polymerization stage is considered in Chapter 7 in order to study the effects of introducing at the scheduling level process dynamics considering several free decision variables.

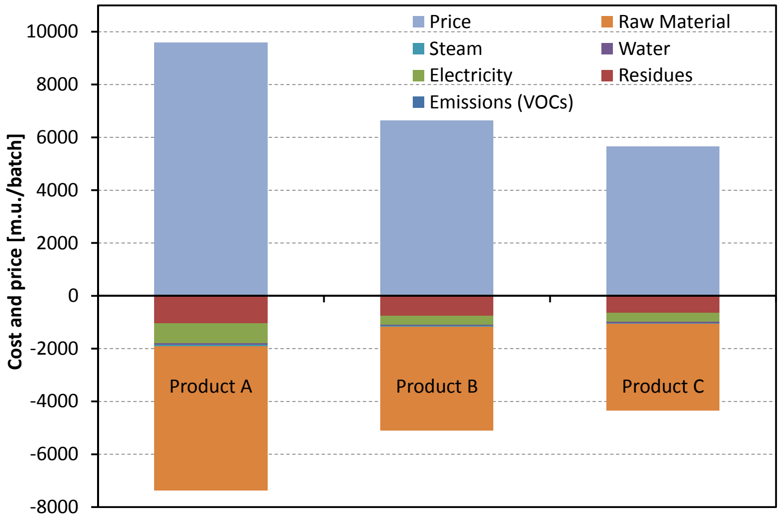
Acrylic fibers are composed of at least 85 percent in weight of acrylonitrile monomer. The remaining composition of the fiber typically includes at least one of other monomers, such as methyl methacrylate (MMA), methyl acrylate, vinyl acetate (VAc), vinyl chloride, or vinylidene chloride. Polyacrylonitrile fiber polymers are produced by the industry using two methods, namely suspension polymerization and solution polymerization. Either batch or continuous reaction modes may be employed. Suspension polymerization processes are advantageous because of the easy separation of the final polymer material, the easy removal of heat, the simple temperature control, and the relatively low levels of impurities and additives in the end product (Silva *et al.*, 2004a).

The polymerization reaction mechanism is based on the classical free-radical polymerization, which is composed by the steps of initiation, propagation, chain transfer to monomer and termination. The most fundamental step of the reaction mechanism for the description of the polymer composition and the composition distribution is the propagation step (Silva *et al.*, 2004a). As for the termination step, it is described by the terminal model (Silva *et al.*, 2004c). This model assumes that the reactivity of the propagation reaction is governed only by the nature of the monomer and of the terminal unit of the growing polymer chain (Silva *et al.*, 2004a).

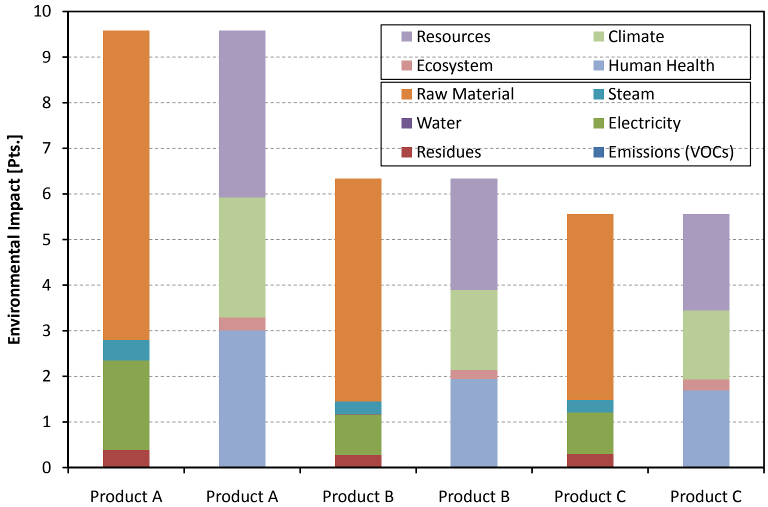
In general, polymerization processes are characterized by the simultaneous occurrence of several complex nonlinear phenomena. However, an approximated model of the reaction stage can allow the improvement of the plant operability and the optimization of the process as a whole. Hence, this work models the suspension copolymerization of acrylonitrile with two different monomers, namely vinyl acetate, methyl methacrylate. Next, the hypothesis and parameters of the model are described:

- The suspension copolymerization model of acrylonitrile (AN) with other organic monomers (VAc and MMA) has been adapted from VAc/AA copolymerization model proposed by Silva *et al.* (2004a), due to the similarities in process characteristics, reaction mechanism and physical properties of the involved monomers. All kinetic parameters and reactivity ratios are adapted to the components of this case.

- Glass and gel effects should be taken into account since they introduce relevant non-linear behavior in free-radical polymerization models (Silva *et al.*, 2004a). They are typical kinetic phenomena induced by the increase of the system viscosity during polymer formation (Silva *et al.*, 2004a).



(a) Costs and price.



(b) Environmental impact distributed along different items, left column operation related, while right column in different end point categories.

Figure D.2: Batch cost and price, and environmental impact for the three acrylic fibers.

- In this system, the AN monomer is soluble in both aqueous and organic phases. Consequently, the distribution coefficient should be considered for this component. A negligible transport gradient between both phases is considered, assuming equilibrium conditions at every moment. On the contrary, the other monomers are considered to be completely insoluble in the aqueous phase. Due to lack of specific data for AN equilibrium in such a system, given the behavior similitudes between AA and AN, and processing conditions, the AA equilibrium data published in Silva *et al.* (2004b) are employed.

- Semibatch operation policies are able to successfully control the copolymer composition, by adding fresh monomer during semibatch reactions (Silva *et al.*, 2004c). The less reactive monomer is fully loaded at the beginning of the process, while the most reactive one is added throughout the process.

- The sets of controlled and manipulated variables are selected based on both the impact on final polymer quality and the possibility of manipulation in real time (Machado *et al.*, 2010). In order to control quality indicators such as copolymer composition and average molecular weight, typical manipulated variables are reactor temperature, reactant concentrations and reactant feed flow rates (Machado *et al.*, 2010). In our case, instant copolymer composition is controlled employing reactant feed rate as manipulated variable. A feedback PI controller is suitable to this purpose, using as feed monomer the most the most reactive one.

- Temperature is considered constant, and ranges between 50 and 70ºC. It is very important to note that the thermodynamic partition coefficients depend very strongly and nonlinearly on this process variable, as well as on AN aqueous concentration.

- Typically, molecular characteristics of the final product, as well as reaction time, are the processing objectives and restrictions that are taking into account. For example, in the case of Machado *et al.* (2010) based on AA/VAc, it was much more important to control polymer properties rather than control polymer productivity. However, in this example, we are going to analyze costs, and production issues, while quality properties and conversion are restrictions that should be accomplished.

- The addition of fresh monomer diluted in water in semi-batch co-polymerization processes is encouraged, since this complementary amount of water may exert a strong positive influence on the stabilization of the reaction system without significant loss of productivity (Machado *et al.*, 2010). For example, it helps to minimize mixing problems by keeping the suspension viscosity under control. In addition, it avoids the increase in polymer concentration and maintains the polymer holdup at safe levels, preventing from particle agglomeration and new organic droplets formation in the medium.

The products to be processed are described in Table D.3. The properties of the monomers, polymers and initiators, as well as the kinetic parameters for each system, are given in Tables D.4 and D.5. Recipe parameters, fixed variables and boundaries for free decision variables are given in Table D.6.

Table D.3: Products

Product	M1	M2	Mass fraction P2
1	VAc	AN	85%
2			95%
3	MMA	AN	85%
4			95%

Table D.4: Kinetic parameters and physical properties in the acrylonitrile copolimerization.

Parameter	M1		M2	Units
	VAc	MMA	AN	
α_M	0.001	0.001	0.001	-
α_P	$7{\cdot}10^{-4}$	$1.96{\cdot}10^{-4}$	$1.5{\cdot}10^{-4}$	-
$T_{g,M}$	173	225	189	K
$T_{g,P}$	305	433	358	K
ρ_M	0.925	0.939	0.810	g/cm^3
ρ_P	1.190	1.190	1.184	g/cm^3
MW	86.09	100.12	53.06	g/mol
$k_{P,0}$	$2.34{\cdot}10^3$	$8.44{\cdot}10^2$	$6.11{\cdot}10^4$	$m^3/(s \cdot kmol)$
$k_{T,0}$	$2.94{\cdot}10^7$	$2.70{\cdot}10^7$	$9.42{\cdot}10^8$	$m^3/(s \cdot kmol)$
ρ_{water}	1.000			g/cm^3
k_D	$3.46{\cdot}10^6$			s^{-1}
f	0.8			-

Table D.5: Reactivity ratio of monomers 1 and 2 for each copolymerization system.

Parameter	Copolymer	
	VAc/AN	MMA/AN
r_1	0.061	1.04
r_2	4.05	0.15

Table D.6: Kinetic parameters and physical properties for the acrylonitrile copolimerization.

T	323.15	K
I_{ini}	0.062	kmol
$M1_{ini}$	5 - 40	kmol
$M2_{ini}$	10 - 80	kmol
$P1_{ini}$	0	kmol
$P2_{ini}$	0	kmol
V^I	200	m^3
V^{II}	400	m^3
F_2	0.015 - 0.005	kmol/s

Equations D.1 to D.5 represent the initiator and monomer reaction rate, and polymer formation rate (P1 polymer is derived from monomer M1 and polymer P2 from monomer M2). Glass and gel effects are represented by equations D.9 to D.12, while equations D.19 and D.20 define the feedback PI control loop.

$$\frac{dI}{dt} = -k_D I \tag{D.1}$$

$$\frac{dM_1}{dt} = -\left(k_{P11}P^I\right)\left[M_1 + \left(\frac{1}{\varphi+1}\right)\frac{M_2}{r_1}\right] \tag{D.2}$$

$$\frac{dM_2}{dt} = -\left(k_{P11}P^I\right)\left[\left(\frac{1}{\varphi+1}\right)\frac{M_2}{r_1} + \frac{r_2}{r_1}\left(\frac{1}{\varphi+1}\right)^2\frac{M_2^2}{M_1}\right] + F_2 \tag{D.3}$$

$$\frac{d\varrho_1}{dt} = \left(k_{P11}P^I\right)\left[M_1 + \left(\frac{1}{\varphi+1}\right)\frac{M_2}{r_1}\right] \tag{D.4}$$

$$\frac{d\varrho_2}{dt} = \left(k_{P11}P^I\right)\left[\left(\frac{1}{\varphi+1}\right)\frac{M_2}{r_1} + \frac{r_2}{r_1}\left(\frac{1}{\varphi+1}\right)^2\frac{M_2^2}{M_1}\right] \tag{D.5}$$

$$\left(k_{P11}P^I\right) = \sqrt{\frac{2fk_DI}{\frac{1}{\psi_1} + 2\zeta\frac{1}{(\psi_1\psi_2)^{1/2}}\frac{r_2}{r_1}\left(\frac{1}{\varphi+1}\right)\frac{M_2}{M_1} + \frac{1}{\psi_2}\left(\frac{1}{\varphi+1}\right)^2\left(\frac{r_2}{r_1}\frac{M_2}{M_1}\right)^2}} \tag{D.6}$$

$$\zeta = 107.84\exp\left[-63.64\frac{M_2}{M_1(\varphi+1)+M_2}\right] \tag{D.7}$$

$$\psi_i = \frac{k_{P_{ii}}^2}{k_{T_{ii}}} = \frac{k_{P_{ii0}}^2}{k_{T_{ii}0}}g(T,x_M) \qquad i=1,2 \tag{D.8}$$

$$g(T,x_M) = \exp\left\{-146.8(\nu_f-\nu_0) - 1076.3(\nu_f-\nu_0)^2\right\} + 92.9(\nu_f-\nu_0) \tag{D.9}$$

$$\nu_f = \sum_{i\in\{1,4\}}\nu_{f_i}\phi_i \tag{D.10}$$

$$\nu_{f_i} = 0.025 + \alpha_i(T-T_{gi}) \tag{D.11}$$

$$\phi_i = \frac{\frac{\rho_i}{m_i}}{\sum\limits_{i\in\{1,4\}}\frac{\rho_i}{m_i}} \tag{D.12}$$

$$\varphi = K\frac{V^{II}}{V^I} \tag{D.13}$$

$$[M_2]^{II} = \left(\frac{\varphi}{\varphi+1}\right)\frac{M_2}{V^{II}} \tag{D.14}$$

$$K\left(T,[M_2]^{II}\right) = \left[A + B\,[M_2]^{II} + \frac{C}{\left([M_2]^{II}\right)^2}\right]^{-1} \tag{D.15}$$

$$A = -16.67 + 0.455\,(T - 273.15) - 2.92{\cdot}10^{-3}\,(T - 273.15)^2 \tag{D.16}$$

$$B = 23.02 - 0.594\,(T - 273.15) + 3.96{\cdot}10^{-3}\,(T - 273.15)^2 \tag{D.17}$$

$$C = 0.317 - 9.02{\cdot}10^{-3}\,(T - 273.15) + 6.04{\cdot}10^{-5}\,(T - 273.15)^2 \tag{D.18}$$

$$Error = x_{P2,SP} - x_{P2,PV} \tag{D.19}$$

$$F = K_P\;Error + \int Error\;dt \tag{D.20}$$

D.3 Data for Chapter 9: Environmental Impact Assessment of Changeover Operations

In Chapter 9, trade-offs between economic and environmental objectives are studied when changeover operations between batches are considered. Specifically, a changeover operation must be carried out between any two batches. Three different changeover cleaning methods, which differ in time, cost and environmental impact, may be adopted.

Next, the assumptions for calculating the LCI of the products are detailed. Specifically, the data regarding residues, non-controlled emissions, raw materials, steam, water, and electricity consumption are based on good engineering practices and the data available in the literature. In addition, the results regarding costs and environmental impact of the three produced fibers and the proposed changeover methods are given.

Raw materials consumption estimation. Raw materials (solvent, monomers and initiators) addition for fiber production is considered at stage 1 (polymerization). An overall reaction yield of 95% is assumed. In addition, a 40% of the total initial amount introduced in the reactor is solvent, and the remaining 60% is monomer mixture, which is composed by 85% acrylonitrile, 10% methyl metacrylate and 5% vynil chloride. The solvent is considered to be pure acetone, while vynil chloride, styrene, acrylonitrile and methyl metacrylate are the possible co-monomers. Each one of the former raw materials LCI data has been retrieved from their corresponding Ecoinvent LCI (EcoinventV2.0, 2008).

Residues generation. The remaining quantity of each batch (5% in mass) is released in the last stage (evaporation), and treated as production waste. A certain percentage of consumed water (30%) is also considered as residue to be treated. The LCI associated with its treatment as waste has been related to treatment of "heat carrier liquid, 40% $C_3H_8O_2$, to waste water treatment, class 2/CH S" in Ecoinvent.

Non-controlled emissions. According to US-EPA (1984) (pg. 33), acrylonitrile emissions in this production process occur at the pelletizer (repulping) and polymer dryer (deaeration) (stages 7 and 11 of the recipe) and estimates an air emission of 18.75 kg/Mg product released in acrylic wet spun homopolymer manufacturing. In this case, these emissions are considered as air emissions of pure acetone, disregarding any monomer emission.

Electricity consumption. Electricity consumption includes pumping required for product movement between stages that are not gravity driven and also for pumping cooling water and steam compression. In the case of pumping cooling water, a pumping $\Delta P = 1 \cdot 10^5$ Pa and a flow of 20 m^3/h, which requires and approximate power of 1.5 kW, is considered. On the other hand, for compressing heating steam, a yield which represents 0.6 GJ useful heat of steam/GJ electricity is used. In all cases, the LCI information for electricity consumption is considered as "Electricity, medium voltage, at grid/ES U".

Heating and cooling needs. In the case of heating, it is considered to be supplied using steam, the LCI has been gathered using the "Steam, for chemical processes, at plant/RER U" Ecoinvent unit. It is a medium-low pressure saturated steam, at $9 \cdot 10^5$ Pa (2029,45 kJ/kg steam). Steam is used to heat streams according to the recipe provided in Grau *et al.* (1996). For the estimation of cooling needs, water is used to cool down the streams. All cooling requirements are computed as water cooling and assuming no electrical refrigeration required. Cooling water consumption is computed by taking into account its specific heat (liquid water is 4.18 kJ/kg), and an average ΔT for water of about 20ºC.

Water consumption. Process water is considered to require softening, consequently the Ecoinvent LCI "Water, completely softened, at plant/RER U" is used. Process water is required in some recipe stages besides cooling. The filtering stages require a water flow of 40 m^3/h, and for the cleaning of these units a water flow of 10 m^3/h is needed.

Changeover characterization. Despite the fact that product changeover involves different operations, in this paper we focused on cleaning operations. According to Allen *et al.* (2002), the nature of the cleaning process should be considered taking into account several aspects: (i) nature of the vessels to be cleaned (capacities, materials of construction and shape), (ii) the cleaning schedule, (iii) the residual quantity of chemical left to be cleaned in the vessel, (iv) the cleaning agent (aqueous/organic, chemical solubility/miscibility), and (v) the requirements of waste treatment for the used cleaning agent. Mainly in the batch industries where individual unit operations are utilized for multiple products, many pieces of equipment are subject to long clean-out periods using large solvent volumes and/or aqueous detergents. It is current practice to try to use clean-in-place (CIP) procedures instead of break down and rebuild approaches where unit operation allows it (Constable *et al.*, 2009) . Although in some cases the unit operation requires its break down and rebuild (e.g. plate filtration) most vessel cleaning is performed using CIP.

Regarding clean up scheduling (ii), it depends on the process or product and cleaning between batches could be due to product requirements (color changes in paint manufacturing), or process requirements (solidification of product in a filter requires its clean up). Estimation of point (iii) requires knowing vessel characteristics and some rough estimate of the viscosity and surface tension of the liquid to be cleaned; however, as a rule of thumb, the amount in weight percent left in vessels ranges from 3 to 0.03% (Allen *et al.*, 2002). With regard to (iv) in the case of aqueous cleaning agents, these are sent to waste water treatment (WWT) plants, while organic solvents are recycled or incinerated. In general, the actual amount of clean up agent will depend on the amount of this agent that can be recycled/reused in other cleaning operations.

In the case study, three different product changeovers are possible. Each of them has associated different costs, inventory/impact and duration (Table 9.1). Since cleaning options are very different, a comparison based on used volume or energy would be too simplistic, and we have decided to use the environmental impact and cost of those stages to select among them by including such aspects in the objective function calculations. A few assumptions have been made regarding the LCI for each of the three available changeover policies.

- Regarding costs, they have been assigned according to the cleaning requirements and general engineering principles used for the estimation of former production costs.
- Electricity consumption [GJ] has been considered to be a function of changeover time ($ChanT$), it is calculated considering the $ChanT$ [h] multiplied by the power of a pump with a flow of 20 m^3/h and a ΔP of $2 \cdot 10^5$ Pa, which is nearly 1.5kW. Electricity consumption also includes electricity requirements for steam compression.
- As for water consumption, a pump of 20 m^3/h is considered in the water cleaning method; so the changeover time multiplied by the pump capacity is approximately the water consumption in that operation.
- Similarly to the estimation of water consumption, solvent is estimated considering a pump capacity and the required changeover time. Solvent recycle has been disregarded.

Figure D.2 presents the batch cost and environmental impact for the production a batch of each product. Raw materials represent the most important operating cost for all products, followed by residues treatment and electricity. However, there are no great differences in production costs among products because their recipe is similar in terms of raw materials and processing stages. In the case of Figure D.2(b), environmental impacts for each product are shown in two different columns distributed in different items. One of them in terms of raw materials, utilities consumption, residues treatment and emissions and the other column using the different end point environmental impact categories that IMPACT 2002 implements (resource usage, global climate change, damage to ecosystem and human health impacts). In the first case, the highest contribution to environmental impact is due to raw materials production, followed by electricity and thirdly water consumption and residues which have approximately the same impact. The distribution along end point categories shows similar impacts to resource use, climate change and human health, while smaller effects to ecosystem quality.

Figures D.3 to D.5 show the changeover costs, environmental impacts and time for each pair of products using the three available cleaning methods. The differences

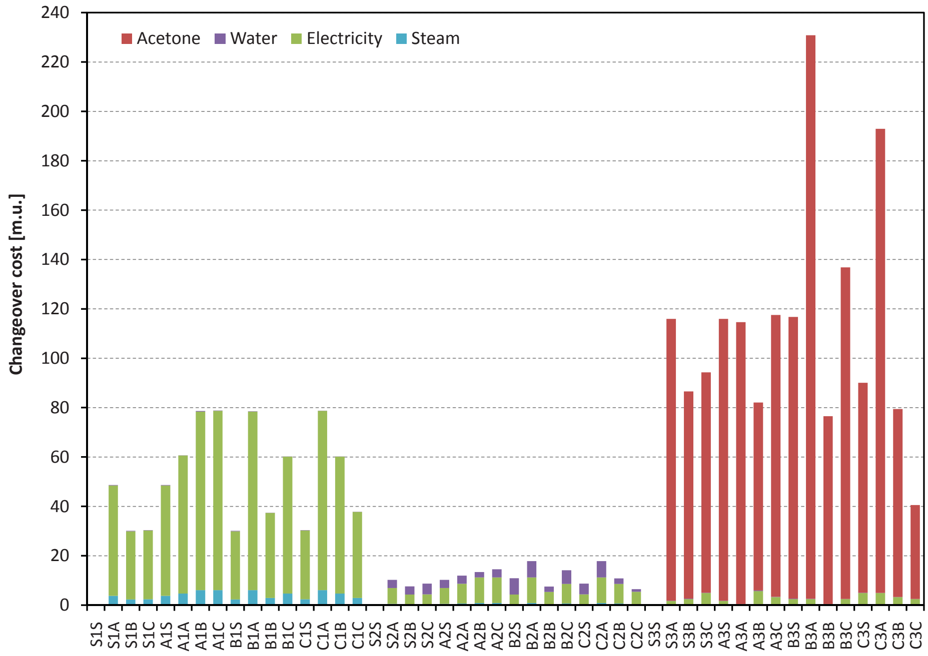


Figure D.3: Changeover costs between pairs of products (S-still state, A, B, C) for the three methods (1, 2, 3).

briefly outlined in Table 9.1 can be appreciated, and the contribution of each operating resource to the total cost is unveiled. Therefore, the high operating cost of method 3 is basically due to fresh acetone consumption. In the case of changeover 1, cost is basically due to electricity consumption, whereas steam represents a smaller fraction of total cost, and electricity and water are the main costs of cleaning method 2.

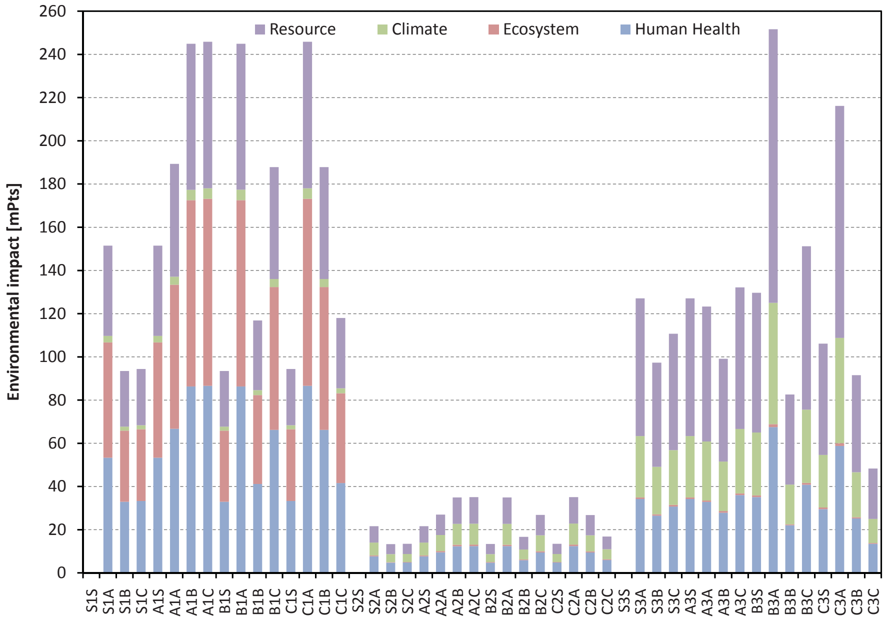


Figure D.4: Changeover environmental impacts between pairs of products (S-still state, A, B, C) for the three methods (1, 2, 3).

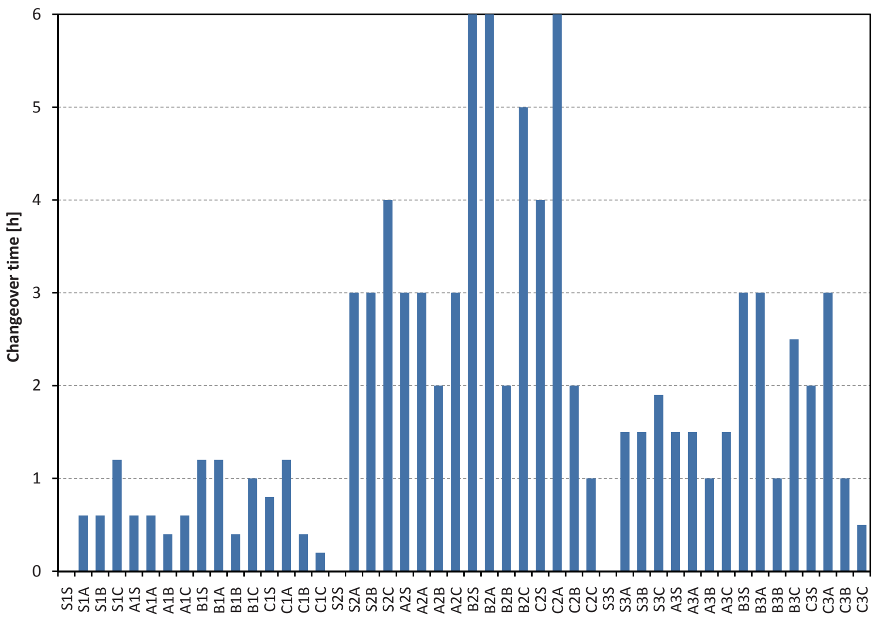


Figure D.5: Changeover time between pairs of products (S-still state, A, B, C) for the three methods (1, 2, 3).

D.4 Nomenclature

Parameters

f	Initiator efficiency.
F_m	Feed flow rate of monomer m.
I	Concentration of initiator.
K	Partition coefficient.
k_D	Kinetic constant for initiator decomposition.
k_P^{ij}	Kinetic constant for propagation of radical i with radical j.
k_T^{ij}	Kinetic constant for termination of radical i with monomer j.
M_m	Concentration of monomer m.
MW_p	Molecular weight of product p.
P_{ij}	Radical chain containing i mers of species 1 and j mers of species 2 in the chain and the species 1 at the active site (radical 1).
Q_{ij}	Radical chain containing i mers of species 2 and j mers of species 1 in the chain and the species 2 at the active site (radical 2).
r_m	Reactivity ratio of monomers m.
T_0	Initial temperature of the reaction stage [ºC].
T	Reaction temperature.
T_p^g	Glass transition temperature of product p.
$t_p^{reaction}$	Processing time in reaction stage of product p .
t	Time.
V^i	Volume of phase i.
μ_f	Free volume of reaction system.
μ_{f0}	Free volume of reaction system at zero conversion.
μ_{fi}	Free volume of species i.
μ_{fM}	Free volume of the reaction system at zero conversion.
x_M	Monomer conversion.

Greek symbols

α_p	Expansion coefficient of product p.
Φ_i	Volume fraction of the species i in the reactor.
ρ_p	Pure density of product p.
φ	Global partition coefficient.
ζ	Cross termination constant.
Γ_{ij}	Dead polymer chain containing i mers of the species 1 and j mers of the species 2.

Superscripts

I	Organic phase.
II	Aqueous phase.

Subscripts

M	Monomer.
P	Polymer.
p	Product (either polymer or monomer).

Bibliography

Aguilar-Lasserre, A, L Pibouleau, C Azzaro-Pantel, & S Domenech. "Enhanced genetic algorithm-based fuzzy multiobjective strategy to multiproduct batch plant design." *Applied Soft Computing* 9: (2009) 1321–1330.

Al-Mutairi, E, & M. M El-Halwagi. "Environmental-impact reduction through simultaneous design, scheduling, and operation." *Clean Technologies And Environmental Policy* 12, no. 5: (2010) 537–545.

Alle, A, & J. M Pinto. "Mixed-integer programming models for the scheduling and operational optimization of multiproduct continuous plants." *Industrial & Engineering Chemistry Research* 41, no. 11: (2002) 2689–2704.

Allen, D. T, D. R Shonnard, & S Prothero, Eds. *Evaluating Environmental Performance during process synthesis In: Green Engineering: Environmentally Conscious Design Of Chemical Processes*, Prentice Hall PTR, New Jersey, 2002, Ch. 8, 199–249.

Amaro, A, & A Barbosa-Póvoa. "Supply Chain Management with Optimal Scheduling." *Industrial & Engineering Chemistry Research* 47: (2008) 116–132.

Arnaout, J.-P, G Rabadi, & R Musa. "A two-stage Ant Colony Optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times." *Journal of Intelligent Manufacture* 21: (2010) 693–701.

Azapagic, A, A Emsley, & I Hamerton. *Polymers, The Environment and Sustainable Development*. John Wiley & Sons, 2003.

Azzaro-Pantel, C, L Bernal-Haro, P Baudet, S Domenech, & L Pibouleau. "A two-stage methodology for short-term batch plant scheduling: discrete-event simulation and genetic algorithm." *Computers & Chemical Engineering* 22, no. 10: (1998) 1461–1481.

Baker, K. R. *Introduction to Sequencing and Scheduling*. Wiley and Sons, 1997.

Barbosa-Povoa, A. P. "A critical review on the design and retrofit of batch plants." *Computers & Chemical Engineering* 31, no. 7: (2007) 833–855.

Barker, M, & J Rawtani. *Practical Batch Process Management*. Elsevier, 2005.

Baudin, M. *Manufacturing Systems Analysis with Applications to Production Scheduling*. Yourdon Press, 1990.

Berlin, J, & U Sonesson. "Minimising environmental impact by sequencing cultured dairy products: two case studies." *Journal Of Cleaner Production* 16, no. 4: (2008) 483–498.

Berlin, J, U Sonesson, & A Tillman. "A life cycle based method to minimise environmental impact of dairy production through product sequencing." *Journal Of Cleaner Production* 15: (2007) 347–356.

Bhatia, T, & L. T Biegler. "Dynamic optimization in the design and scheduling of multiproduct batch plants." *Industrial & Engineering Chemistry Research* 35, no. 7: (1996) 2234–2246.

Bhatia, T. K, & L. T Biegler. "Dynamic optimization for batch design and scheduling with process model uncertainty." *Industrial & Engineering Chemistry Research* 36, no. 9: (1997) 3708–3717.

Biegler, L. T. *Nonlinear Programming. Concepts, Algorithms, and Applications to Chemical Processes.* Society for Industrial and Applied Mathematics and Mathematical Optimization Society, 2010.

Biegler, L. T, & I. E Grossmann. "Retrospective on optimization." *Computers & Chemical Engineering* 28, no. 8: (2004) 1169–1192.

Biegler, L. T, I. E Grossmann, & A. W Westerberg. *Systematic Methods of Chemical Process Design.* Prentice Hall, 1999.

Blomer, F, & H. O Gunther. "LP-based heuristics for scheduling chemical batch processes." *International Journal Of Production Research* 38, no. 5: (2000) 1029–1051.

Blum, C, & A Roli. "Metaheuristics in combinatorial optimization: Overview and conceptual comparison." *Acm Computing Surveys* 35, no. 3: (2003) 268–308.

Bonami, P, M Kilinc, & J Linderoth. "Algorithms and Software for Convex Mixed Integer Nonlinear Programs." Technical Report 1664, University of Wisconsin, 2009.

Bonfill, A, A Espuña, & L Puigjaner. "Decision support framework for coordinated production and transport scheduling in SCM." *Computers & Chemical Engineering* 32: (2008) 1206–1224.

Burkard, R. E, & J Hatzl. "A complex time based construction heuristic for batch scheduling problems in the chemical industry." *European Journal Of Operational Research* 174, no. 2: (2006) 1162–1183.

Busch, J, J Oldenburg, M Santos, A Cruse, & W Marquardt. "Dynamic predictive scheduling of operational strategies for continuous processes using mixed-logic dynamic optimization." *Computers & Chemical Engineering* 31, no. 5-6: (2007) 574–587.

Bussieck, M. R, & S Vigerske. "MINLP Solver Software.", 2010. Accepted for publication in Wiley Encyclopedia of Operations Research and Management Science.

Caballero, J. A, & I. E Grossmann. "A review of the state of the art in optimization." *Revista Iberoamericana De Automatica E Informatica Industrial* 4, no. 1: (2007) 5–+.

Cabezas, H, J Bare, & S Mallick. "Pollution prevention with chemical process simulators: the generalized waste reduction (WAR) algorithm - full version." *Computers and Chemical Engineering* 23: (1999) 623–634.

Cano-Ruiz, J, & G McRae. "Environmentally Conscious Chemical Process Design." *Annual Review of Energy and the Environment* 23: (1998) 499–536.

Cao, Y. "Pareto Front: Two efficient algorithms to find Pareto Front." Technical report, http://www.mathworks.com/matlabcentral/fileexchange/17251-pareto-front, on 16 March 2010, 2009.

Castells, F, V Aelion, K Abeliotis, & D Petrides. "An Algorithm for Life Cycle inventory Analysis." *AIChE Symposium Series* 90, no. 303: (1994) 151–160.

Castro, P, A. P. F. D Barbosa-Povoa, & H Matos. "An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants." *Industrial & Engineering Chemistry Research* 40, no. 9: (2001) 2059–2068.

Castro, P, H Matos, & A. P. F. D Barbosa-Povoa. "Dynamic modelling and scheduling of an industrial batch system." *Computers & Chemical Engineering* 26, no. 4-5: (2002) 671–686.

Castro, P. M. "Optimal Scheduling of Pipeline Systems with a Resource-Task Network Continuous-Time Formulation." *Industrial & Engineering Chemistry Research* 49, no. 22: (2010) 11,491–11,505.

Castro, P. M, A. P Barbosa-Povoa, & H. A Matos. "Optimal periodic scheduling of batch plants using RTN-based discrete and continuous-time formulations: A case study approach." *Industrial & Engineering Chemistry Research* 42, no. 14: (2003) 3346–3360.

Castro, P. M, & I. E Grossmann. "New Continuous-Time MILP Model for the Short-Term Scheduling of Multistage Batch Plants." *Industrial & Engineering Chemistry Research* 44, no. 24: (2005) 9175–9190.

Castro, P. M, I Harjunkoski, & I. E Grossmann. "Greedy algorithm for scheduling batch plants with sequence-dependent changeovers." *Aiche Journal* 57, no. 2: (2011) 373–387.

Castro, P. M, & A. Q Novais. "Scheduling Multistage Batch Plants with Sequence-Dependent Changeovers." *Aiche Journal* 55, no. 8: (2009) 2122–2137.

Castro, P. M, A. P Barbosa-Povoa, H. A Matos, & A. Q Novais. "Simple Continuous-Time Formulation for Short-Term Scheduling of Batch and Continuous Processes." *Industrial & Engineering Chemistry Research* 43: (2004) 105–118.

Castro, P. M, A. Q Novais, & A Carvalho. "Optimal Equipment Allocation for High Plant Flexibility: An Industrial Case Study." *Industrial & Engineering Chemistry Research* 48: (2008) 2742–2761.

Cerdá, J, G. P Henning, & I. E Grossmann. "A mixed integer linear programming model for short-term scheduling of single-stage multiproduct batch plants with parallel lines." *Industrial & Engineering Chemistry Research* 36, no. 5: (1997) 1695–1707.

Chachuat, B, A. B Singer, , & P. I Barton. "Global Methods for Dynamic Optimization and Mixed-Integer Dynamic Optimization." *Industrial & Engineering Chemistry Research* 45 (25): (2006) 8373–8392.

Chakraborty, A, R Colberg, & A Linninger. "Plant-Wide Waste Management. 3. Long-Term Operation and Investment Planning under Uncertainty." *Industrial & Engineering Chemistry Research* 42: (2003) 4772–4788.

Chakraborty, A, & A Linninger. "Plant-Wide Waste Management. 1. Synthesis and Multiobjective Design." *Industrial & Engineering Chemistry Research* 41: (2002) 4591–4604.

Charpentier, J. C. "Modern chemical engineering in the framework of globalization, sustainability, and technical innovation." *Industrial & Engineering Chemistry Research* 46, no. 11: (2007) 3465–3485.

Chatzidoukas, C, C Kiparissides, J. D Perkins, & E. N Pistikopoulos. "Optimal grade transition campaign scheduling in a gas-phase polyolefin FBR using mixed integer dynamic optimization." *Process Systems Engineering 2003, Pts A And B* 15: (2003) 744–747.

Chen, H, & D. R Shonnard. "Systematic Framework for Environmentally Conscious Chemical Process Design: Early and Detailed Design Stages." *Industrial & Engineering Chemistry Research* 43: (2004) 535–552.

Chyu, C.-C, & W.-S Chang. "A Pareto evolutionary algorithm approach to bi-objective unrelated parallel machine scheduling problems." *International Journal of Advanced Manufacturing Technologies* 49: (2010) 697–708.

Cizniar, M, D Salhi, M Fikar, & M Latifi. "A Matlab package for orthogonal collocations on finite elements in dynamic optimisation." In *15th Int. Conference Process Control*. 2005.

Clift, R, & A Azapagic. "The application of Life Cycle Assessment to process selection, design and operation." *Tools And Methods For Pollution Prevention* 62: (1999) 69–84.

Coello, C. A. C, G. B Lamont, & D. A. V Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Springer, 2007, 2nd Edition.

Coello-Coello, C, & R Landa-Becerra. "Evolutionary Multiobjective Optimization in Materials Science and Engineering." *Materials and Manufacturing Processes* 24: (2009) 119–129.

Conejo, A, E Castillo, R Mínguez, & R Bertrand. *Decomposition Techniques in Mathematical*

Programming. Berlin: Springer-Verlag, 2006.

Conn, A. R, K Scheinberg, & L. N Vicente. *Introduction To Derivative-Free Optimization.* MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics and Mathematical Programming Society, 2009, 1st Edition.

Constable, D. J, C Jimenez-Gonzalez, & A Lapkin. *Process metrics In: Green Chemistry Metrics: Measuring and Monitoring Sustainable Processes*, New York, USA: John Wiley and Sons, Inc., 2009, Ch. 6, 228–247.

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

Council, T. E. C. I. "Cefic review 2009-2010. Sustainability and innovation driving chemistry solutions for the future." Technical report, CEFIC, 2010.

Cuchí, X. "Informe anual sobre la indústria a Catalunya 2009." Report, Generalitat de Catalunya; Departament dInnovació, Universitats i Empresa, 2010.

Cuthrell, J. E, & L. T Biegler. "On The Optimization Of Differential-Algebraic Process Systems." *Aiche Journal* 33, no. 8: (1987) 1257–1270.

———. "Simultaneous-Optimization And Solution Methods For Batch Reactor Control Profiles." *Computers & Chemical Engineering* 13, no. 1-2: (1989) 49–62.

Das, I, & J. E Dennis. "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems." *Siam Journal On Optimization* 8, no. 3: (1998) 631–657.

Dayou, L, Y Pu, & Y Ji. "Development of a multiobjective GA for advanced planning and scheduling problem." *The International Journal of Advanced Manufacturing Technologies* 42: (2009) 974–992.

Dietz, A, C Azzaro-Pantel, L Pibouleau, & S Domenech. "Multiobjective optimization for multiproduct batch plant design under economic and environmental considerations." *Computers and Chemical Engineering* 30: (2006) 599–613.

Diwekar, U, & Y Shastri. "Design for environment: a state-of-the-art review." *Clean Technologies And Environmental Policy* 13, no. 2: (2011) 227–240.

Durand, G. A, & J. A Bandoni. "Improving the Computational Performance of STN/RTN Formulations for Short-Term Scheduling." *Industrial & Engineering Chemistry Research* 49, no. 24: (2010) 12,477–12,495.

EcoinventV2.0. "The Ecoinvent database V2.0." Technical report, Swiss Centre for Life Cycle Inventories, 2008.

Edgar, T. F. "Control and operations: when does controllability equal profitability?" *Computers & Chemical Engineering* 29, no. 1: (2004) 41–49.

Edwards, M. F. "Product engineering - Some challenges for chemical engineers." *Chemical Engineering Research & Design* 84, no. A4: (2006) 255–260.

Erdirik-Dogan, M, & I. E Grossmann. "Slot-based formulation for the short-term scheduling of multistage, multiproduct batch plants with sequence-dependent changeovers." *Industrial & Engineering Chemistry Research* 47, no. 4: (2008) 1159–1183.

Ferrer-Nadal, S, T Holczinger, C. A Méndez, F Friedler, & L Puigjaner. "Rigorous scheduling resolution of complex multipurpose batch plants: S-Graph vs. MILP." In *ESCAPE-16*, edited by W. Marquardt, & C. Pantelides. 2006, 2033–2039.

Ferrer-Nadal, S, C. A Mendez, M Graells, & L Puigjaner. "Optimal reactive scheduling of manufacturing plants with flexible batch recipes." *Industrial & Engineering Chemistry Research* 46, no. 19: (2007) 6273–6283.

Ferrer-Nadal, S, L Puigjaner, & G Guillen-Gosalbez. "Managing risk through a flexible recipe framework." *Aiche Journal* 54, no. 3: (2008) 728–740.

Ferris, M. "MATLAB and GAMS: Interfacing Optimization and Visualization Software." Technical report, Computer Sciences Department, University of Wisconsin, 2005.

http://www.cs.wisc.edu/math-prog/matlab.html.

Flores-Tlacuahuac, A, & I. E Grossmann. "Simultaneous cyclic scheduling and control of a multiproduct CSTR." *Industrial & Engineering Chemistry Research* 45, no. 20: (2006) 6698–6712.

Floudas, C. A, & X. X Lin. "Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review." *Computers & Chemical Engineering* 28, no. 11: (2004) 2109–2129.

———. "Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications." *Annals Of Operations Research* 139, no. 1: (2005) 131–162.

French, S. *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop.* Ellis Horwood Limited, 1982.

Gandibleux, X. *Metaheuristics for multiobjective optimisation.* Xavier Gandibleux, 2004.

Giannelos, N. F, & M. C Georgiadis. "A Novel Event-Driven Formulation for Short-Term Scheduling of Multipurpose Continuous Processes." *Industrial & Engineering Chemistry Research* 41, no. 10: (2002) 2431–2431.

Grau, R, M Graells, J Corominas, A Espuña, & L Puigjaner. "Global strategy for energy and waste analysis in scheduling and planning of multiproduct batch chemical processes." *Computers & Chemical Engineering* 20: (1996) 853–868.

Grossmann, I. "Enterprise-wide optimization: A new frontier in process systems engineering." *Aiche Journal* 51, no. 7: (2005) 1846–1857.

Grossmann, I, M Erdirik-Dogan, & R Karuppiah. "Overview of planning and scheduling for enterprise-wide optimization of process industries." *At-Automatisierungstechnik* 56, no. 2: (2008) 64–79.

Grossmann, I. E. "Challenges in the new millennium: product discovery and design, enterprise and supply chain optimization, global life cycle assessment." *Computers & Chemical Engineering* 29, no. 1: (2004) 29–39.

Grossmann, I. E, & L. T Biegler. "Part II. Future perspective on optimization." *Computers & Chemical Engineering* 28, no. 8: (2004) 1193–1218.

Guo, Y, W Li, A Mileham, & G Owen. "Applications of particle swarm optimisation in integrated process planning and scheduling." *Robotics and Computer - Integrated Manufacturing* 25: (2009) 280–288.

Gupta, S, & I. A Karimi. "An improved MILP formulation for scheduling multiproduct, multistage batch plants." *Industrial & Engineering Chemistry Research* 42, no. 11: (2003) 2365–2380.

Ha, J. H, H. K Chang, E. S Lee, I. B Lee, B. S Lee, & G Yi. "Intermediate storage tank operation strategies in the production scheduling of multi-product batch processes." *Computers & Chemical Engineering* 24, no. 2–7: (2000) 1633–1640.

Halim, I, & R Srinivasan. "Sequential Methodology for Scheduling of Heat-Integrated Batch Plants." *Industrial & Engineering Chemistry Research* 48, no. 18: (2009) 8551–8565.

Harjunkoski, I, & I. E Grossmann. "Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods." *Computers & Chemical Engineering* 26, no. 11: (2002) 1533–1552.

Harjunkoski, I, R Nystrom, & A Horch. "Integration of scheduling and control-Theory or practice?" *Computers & Chemical Engineering* 33, no. 12: (2009) 1909–1918.

He, Y, & C. W Hui. "Genetic algorithm for large-size multi-stage batch plant scheduling." *Chemical Engineering Science* 62, no. 5: (2007) 1504–1523.

He, Y. H, & C. W Hui. "A binary coding genetic algorithm for multi-purpose process scheduling: A case study." *Chemical Engineering Science* 65, no. 16: (2010) 4816–4828.

Hegyháti, M, T.Majozi, T.Holczinger, & F.Friedler. "Practical infeasibility of cross-transfer in batch plants with complex recipes: S-graph vs MILPmethods." *Chemical Engineering*

Science 64: (2009) 605–610.

Heo, S. K, K. H Lee, H. K Lee, I Lee, & J. H Park. "A New Algorithm for Cyclic Scheduling and Design of Multipurpose Batch Plants." *Industrial & Engineering Chemistry Research* 42, no. 4: (2003) 836–846.

Hoogeveen, H. "Multicriteria scheduling." *European Journal Of Operational Research* 167, no. 3: (2005) 592–623.

Humbert, S, M Margni, & O Jolliet. "IMPACT 2002+: User Guide Draft for version 2.1." Technical report, Industrial Ecology & Life Cycle Systems Group, GECOS, Swiss Federal Institute of Technology Lausanne (EPFL), Lausanne, Switzerland, 2005.

Hwang, C, & K Yoon. *Multiple attribute decision making.* Berlin: Springer-Verlag, 1981.

Ierapetritou, M. G, & C. A Floudas. "Effective continuous-time formulation for short-term scheduling. 2. Continuous and semicontinuous processes." *Industrial & Engineering Chemistry Research* 37, no. 11: (1998) 4360–4374.

International Society for Measurement and Control "ANSI/ISA-88. Batch Control Part 2: Data Structures and Guidelines for Languages." International standard, ISA Society, 2001.

ISO14001. "Environmental management systems - Requirements with guidance for use." Technical report, ISO, 2004.

Janak, S. L, & C. A Floudas. "Improving unit-specific event based continuous-time approaches for batch processes: Integrality gap and task splitting." *Computers & Chemical Engineering* 32, no. 4-5: (2008) 913–955.

Janak, S. L, X Lin, & 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." *Industrial & Engineering Chemistry Research* 43: (2004) 2516–2533.

Jayaraman, V. K, B. D Kulkarni, S Karale, & P Shelokar. "Ant colony framework for optimal design and scheduling of batch plants." *Computers & Chemical Engineering* 24, no. 8: (2000) 1901–1912.

Kallrath, J. "Planning and scheduling in the process industry." *Or Spectrum* 24, no. 3: (2002) 219–250.

———. "Solving planning and design problems in the process industry using mixed integer and global optimization." *Annals Of Operations Research* 140, no. 1: (2005) 339–373.

Kim, M. S, J. H Jung, & I. B Lee. "Optimal scheduling of multiproduct batch processes for various inter-stage storage policies." *Industrial & Engineering Chemistry Research* 35, no. 11: (1996) 4058–4066.

Kim, S. B, H. K Lee, I. B Lee, E. S Lee, & B Lee. "Scheduling of non-sequential multipurpose batch processes under finite intermediate storage policy." *Computers & Chemical Engineering* 24, no. 2-7: (2000) 1603–1610.

Klatt, K. U, & W Marquardt. "Perspectives for process systems engineering-Personal views from academia and industry." *Computers & Chemical Engineering* 33, no. 3: (2009) 536–550.

Kondili, E, C Pantelides, & R Sargent. "A general algorithm for short term scheduling of batch operations." *Computers & Chemical Engineering* 17: (1993) 211–227.

Korovessi, E, & A. A Linninger. *Batch processes.* Taylor & Francis Group, 2006.

Kotecha, P. R, M Bhushan, & R. D Gudi. "Efficient Optimization Strategies with Constraint Programming." *Aiche Journal* 56, no. 2: (2010) 387–404.

Lasschuit, W, & N Thijssen. "Supporting supply chain planning and scheduling decisions in the oil and chemical industry." *Computers & Chemical Engineering* 28, no. 6-7: (2004) 863–870.

Li, Z, & M Ierapetritou. "Process Scheduling Under Uncertainty Using Multiparametric Programming." *AIChE Journal* 53: (2007) 3183–3203.

Lim, M, & Z Zhang. "A multi-agent based manufacturing control strategy for responsive manufacturing." *Journal of Materials Processing Technology* 139: (2003) 379 – 384.

Lin, X, C. A Floudas, S Modi, & N. M Juhasz. "Continuous-Time Optimization Approach for Medium-Range Production Scheduling of a Multiproduct Batch Plant." *Industrial & Engineering Chemistry Research* 41, no. 16: (2002) 3884–3906.

Liu, B, L Wang, Y Liu, B Qian, & Y. H Jin. "An effective hybrid particle swarm optimization for batch scheduling of polypropylene processes." *Computers & Chemical Engineering* 34, no. 4: (2010) 518–528.

Machado, F, E. L Lima, & J. C Pinto. "Dynamic Optimization of Semibatch Vinyl Acetate/Acrylic Acid Suspension Copolymerizations." *Polymer Engineering And Science* 50, no. 4: (2010) 697–708.

Majozi, T. "Minimization of energy use in multipurpose batch plants using heat storage: an aspect of cleaner production." *Journal Of Cleaner Production* 17, no. 10: (2009) 945–950.

Maravelias, C. T. "Mixed-time representation for state-task network models." *Industrial & Engineering Chemistry Research* 44, no. 24: (2005) 9129–9145.

Maravelias, C. T, & I. E Grossmann. "Minimization of the makespan with a discrete-time state-task network formulation." *Industrial & Engineering Chemistry Research* 42, no. 24: (2003a) 6252–6257.

———. "A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants." *Computers & Chemical Engineering* 28, no. 10: (2004) 1921–1949.

———. "On the relation of continuous- and discrete-time state-task network formulations." *Aiche Journal* 52, no. 2: (2006) 843–849.

Maravelias, C. T, & C Sung. "Integration of production planning and scheduling: Overview, challenges and opportunities." *Computers & Chemical Engineering* 33, no. 12: (2009) 1919–1930.

Maravelias, C. T, & I. E Grossmann. "New General Continuous-Time State-Task Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants." *Industrial & Engineering Chemistry Research* 42: (2003b) 3056–3074.

Mathworks. "The Mathworks Corporation. Matlab 7.2." Technical report, http://www.mathworks.com, 2009.

Melnyk, S, R Sroufe, F Montabon, & T Hinds. "Green MRP: identifying the material and environmental impacts of production schedules." *International Journal Of Production Research* 39: (2001) 1559–1573.

Mendez, C. A, & J Cerda. "An efficient MILP continuous-time formulation for short-term scheduling of multiproduct continuous facilities." *Computers & Chemical Engineering* 26, no. 4-5: (2002) 687–695.

———. "Dynamic scheduling in multiproduct batch plants." *Computers & Chemical Engineering* 27, no. 8-9: (2003) 1247–1259.

———. "An MILP framework for batch reactive scheduling with limited discrete resources." *Computers & Chemical Engineering* 28, no. 6-7: (2004) 1059–1068.

Mendez, C. A, G. P Henning, & J Cerda. "An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities." *Computers & Chemical Engineering* 25, no. 4-6: (2001) 701–711.

Messac, A, A Ismail-Yahaya, & C Mattson. "The Normalized Normal Constraint Method for Generating the Pareto Frontier." *Structural and Multidisciplinary Optimization* 25: (2003) 86–98.

Metaxiotis, K. S, D Askounis, & J Psarras. "Expert systems in production planning and scheduling: A state-of-the-art survey." *Journal Of Intelligent Manufacturing* 13, no. 4: (2002) 253–260.

Mishra, B. V, E Mayer, J Raisch, & A Kienle. "Short-term scheduling of batch processes. A comparative study of different approaches." *Industrial & Engineering Chemistry Research* 44, no. 11: (2005) 4022–4034.

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

Mockus, L, & G. V Reklaitis. "Continuous time representation approach to batch and continuous process scheduling. 1. MINLP formulation." *Industrial & Engineering Chemistry Research* 38: (1999a) 197–509.

———. "Continuous time representation approach to batch and continuous process scheduling. 2. Computational issues." *Industrial & Engineering Chemistry Research* 38, no. 1: (1999b) 204–210.

Munoz, E, E Capon-Garcia, M Moreno-Benito, A Espuna, & L Puigjaner. "Scheduling and control decision-making under an integrated information environment." *Computers and Chemical Engineering* 35: (2011) 774 – 786.

Munoz, E, A Espuna, & L Puigjaner. "Towards an ontological infrastructure for chemical batch process management." *Computers & Chemical Engineering* 34, no. 5: (2010) 668–682.

Musulin, E, M Arbiza, A Bonfill, L Puigjaner, R Olsson, & K Arzén. "Closing the information loop in recipe-based batch production." *Computers and Chemical Engineering* 20, no. 2: (2005) 2163 – 2179.

Nystrom, R. H, R Franke, I Harjunkoski, & A Kroll. "Production campaign planning including grade transition sequencing and dynamic optimization." *Computers & Chemical Engineering* 29, no. 10: (2005) 2163–2179.

Ottino, J. M. "New tools, new outlooks, new opportunities." *Aiche Journal* 51, no. 7: (2005) 1840–1845.

Pan, M, X. X Li, & Y Qian. "Continuous-time approaches for short-term scheduling of network batch processes: Small-scale and medium-scale problems." *Chemical Engineering Research & Design* 87, no. 8A: (2009) 1037–1058.

Pan, M, Y Qian, & X. X Li. "A novel precedence-based and heuristic approach for short-term scheduling of multipurpose batch plants." *Chemical Engineering Science* 63, no. 17: (2008) 4313–4332.

Pantelides, C. "Unified frameworks for optimal process planning and scheduling." In *Proceedings on the Conference on Foundations of Computer-Aided Operations*, edited by D. W. Rippin, J.C. Hale, & J. Davis. 1994, 253–274.

Panwalkar, S. S, & W Iskander. "Survey Of Scheduling Rules." *Operations Research* 25, no. 1: (1977) 45–61.

Papageorgaki, S, & G Reklaitis. "Optimal Design of Multipurpose Batch Plants. 2. A Decomposition Solution Strategy." *Industrial & Engineering Chemistry Research* 29, no. 10: (1990) 2062–2073.

Papageorgiou, L. G, & C. C Pantelides. "Optimal Campaign Planning/Scheduling of Multipurpose Batch/Semicontinuous Plants. 1. Mathematical Formulation." *Industrial & Engineering Chemistry Research* 35: (1996) 488–509.

Park, M, D Kim, D Ko, & I Moon. "Multiobjective optimisation for environment-related decision making in paper mill processes." *International Journal Of Environment And Pollution* 29, no. 1-3: (2007) 127–143.

Pawlewski, P, P Golinska, M Fertsch, J. A Trujillo, & Z. J Pasek. "Multiagent Approach for Supply Chain Integration by Distributed Production Planning, Scheduling and Control System." *International Symposium On Distributed Computing And Artificial Intelligence 2008* 50: (2009) 29–37.

Perry, R. H, D. W Green, & J. O Maloney. *Perry's chemical engineers' handbook*. McGraw-Hill, 1997.

Pinto, J, & I. E Grossmann. "Optimal cyclic scheduling of multistage continuous multiproduct plants." *Computers & Chemical Engineering* 18, no. 9: (1994) 797–816.

Pinto, J. M, & I. E Grossmann. "A Continuous Time Mixed Integer Linear Programming Model for Short Term Schedulin of Multistage Batch Plants." *Industrial & Engineering Chemistry Research* 34: (1995) 3037–3051.

Ponnambalam, S, & M Mohan-Reddy. "A GA-SA Multiobjective Hybrid Search Algorithm for Integrating Lot Sizing and Sequencing in Flow-Line Scheduling." *The International Journal of Advanced Manufacturing Technologies* 21: (2003) 126–137.

Prata, A, J Oldenburg, A Kroll, & W Marquardt. "Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor." *Computers & Chemical Engineering* 32, no. 3: (2008) 463–476.

Rardin, R. *Optimization in Operations Research*. Prentice Hall, Upper Saddle River, NJ, 2000.

Ravindran, A. R. *Operations Research and Management Science Handbook*. Taylor and Francis, 2008, 1st Edition.

Reklaitis, G. V, A. K Sunol, D. W Rippin, & O Hortacsu, Eds. *Batch Processing Systems Engineering*. Springer, 1996.

Romero, J, A Espuna, F Friedler, & L Puigjaner. "A new framework for batch process optimization using the flexible recipe." *Industrial & Engineering Chemistry Research* 42, no. 2: (2003) 370–379.

Romero, J, L Puigjaner, T Holczinger, & F Friedler. "Scheduling Intermediate Storage Multipurpose Batch Plants Using the S-Graph." *AIChE Journal* 50: (2004) 403–417.

Sahinidis, N, & I Grossmann. "Multiperiod investment model for processing networks with dedicated and flexible plants." *Industrial & Engineering Chemistry Research* 30: (1991) 1165–1171.

Sanmartí, E, L Puigjaner, T Holczinger, & F Friedler. "Combinatorial Framework for Effective Scheduling of Multipurpose Batch Plants." *AIChE Journal* 48: (2002) 2557–2570.

de Schryver, A, M Goedkoop, & M Oele. "Introduction to LCA with SimaPro 7." Technical report, Pre-Product Ecology Consultants, 2006.

Seider, W, J Seader, & D Lewin. *Product and Process Design Principles. Synthesis, Analysis and Evaluation*. John Wiley and Sons, Inc., 2004.

Senties, O. B, C Azzaro-Pantel, L Pibouleau, & S Domenech. "A Neural Network and a Genetic Algorithm for Multiobjective Scheduling of Semiconductor Manufacturing Plants." *Industrial and Chemistry Research* 48: (2009) 9546–9555.

———. "Multiobjective scheduling for semiconductor manufacturing plants." *Computers & Chemical Engineering* 34, no. 4: (2010) 555–566.

Shafeeq, A, M. I. A Mutalib, K. A Amminudin, & A Muhammad. "More on Completion Time Algorithms for Intermediate Storage Tanks in Multiproduct Batch Process Scheduling Using Matrix Representation." *Industrial & Engineering Chemistry Research* 47, no. 24: (2008a) 9957–9970.

———. "New completion time algorithms for sequence based scheduling in multiproduct batch processes using matrix." *Chemical Engineering Research & Design* 86, no. 10A: (2008b) 1167–1181.

Shah, N. "Process industry supply chains: Advances and challenges." *Computers & Chemical Engineering* 29, no. 6: (2005) 1225–1235.

Shah, N, C Pantelides, & W Sargent. "A general algorithm for short-term scheduling of batch operations-II. Computational issues." *Computers & Chemical Engineering* 2: (1993) 229–244.

Shaik, M. A, & C. A Floudas. "Improved unit-specific event-based continuous-time model for short-term scheduling of continuous processes: Rigorous treatment of storage requirements." *Industrial & Engineering Chemistry Research* 46, no. 6: (2007) 1764–1779.

———. "Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using RTN framework." *Computers & Chemical Engineering* 32, no. 1-2: (2008) 260–274.

Shaik, M. A, S. L Janak, & C. A Floudas. "Continuous-time models for short-term scheduling of multipurpose batch plants: A comparative study." *Industrial & Engineering Chemistry Research* 45, no. 18: (2006) 6190–6209.

Shobrys, D. E, & D. C White. "Planning, scheduling and control systems: why cannot they work together." *Computers & Chemical Engineering* 26, no. 2: (2002) 149–160.

Silva, F. M, E. L Lima, & J. C Pinto. "Acrylic acid/vinyl acetate suspension copolymerizations. 2. Modeling and experimental results." *Industrial & Engineering Chemistry Research* 43, no. 23: (2004a) 7324–7342.

———. "Acrylic acid/vinyl acetate suspension copolymerizations. I. Partition coefficients for acrylic acid." *Journal Of Applied Polymer Science* 93, no. 3: (2004b) 1077–1088.

———. "Control of the copolymer composition in suspension copolymerization reactions." *Industrial & Engineering Chemistry Research* 43, no. 23: (2004c) 7312–7323.

Smith, R. *Chemical process design and integration*. John Wiley & Sons, 2005.

Song, J, H Park, D Lee, & S Park. "Scheduling of actual size refinery processes considering environmental impacts with multiobjective optimization." *Industrial & Engineering Chemistry Research* 41: (2002) 4794–4806.

Srinivasan, B, S Palanki, & D Bonvin. "Dynamic optimization of batch processes - I. Characterization of the nominal solution." *Computers & Chemical Engineering* 27, no. 1: (2003) 1–26.

Stefanis, S. K, A. G Livingston, & E. N Pistikopoulos. "Environmental impact considerations in the optimal design and scheduling of batch processes." *Computers & Chemical Engineering* 21, no. 10: (1997) 1073–1094.

Steuer, R. E. *Multiple criteria optimization: Theory computation and application*, Chichester, UK: John Wiley & sons, 1986, Ch. 14. Wiley series in probability and mathematical statistics - applied.

Stoer, J, & R Bulirsch. *Introduction to numerical analysis*. Springer-Verlag, 1993.

Strojny, G, T Witkowski, & P Antczak. "The implementation of neural networks with heuristics for the optimization of the production scheduling." *2006 IEEE International Conference on Industrial Technology* 16: (2006) 2058–2063.

Subbiah, S, T Tometzki, S Panek, & S Engell. "Multi-product batch scheduling with intermediate due dates using priced timed automata models." *Computers & Chemical Engineering* 33: (2009) 1661–1676.

Sundaramoorthy, A, & I Karimi. "A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants." *Chemical Engineering Science* 60: (2005) 2679–2702.

Sundaramoorthy, A, & C. T Maravelias. "Modeling of storage in batching and scheduling of multistage processes." *Industrial & Engineering Chemistry Research* 47, no. 17: (2008) 6648–6660.

Sundaramoorthy, A, & C Maravelias. "A General Framework for Process Scheduling." *Aiche Journal* 57, no. 3: (2011) 695–710.

Susarla, N, J Li, & I. A Karimi. "A Novel Approach to Scheduling Multipurpose Batch Plants Using Unit-Slots." *Aiche Journal* 56, no. 7: (2010) 1859–1879.

Terrazas-Moreno, S, A Flores-Tlacuahuac, & I. E Grossmann. "Simultaneous cyclic scheduling and optimal control of polymerization reactors." *Aiche Journal* 53, no. 9: (2007) 2301–

2315.

US-EPA. "Locating and estimating air emissions from sources of acrylonitrile." Technical report, Office Of Air Quality Planning And Standards, United States Environmental Protection Agency (USEPA), 1984.

Varma, V. A, G. V Reklaitis, G. E Blau, & J. F Pekny. "Enterprise-wide modeling & optimization - An overview of emerging research challenges and opportunities." *Computers & Chemical Engineering* 31, no. 5-6: (2007) 692–711.

Verderame, P. M, J. A Elia, J Li, & C. A Floudas. "Planning and Scheduling under Uncertainty: A Review Across Multiple Sectors." *Industrial & Engineering Chemistry Research* 49, no. 9: (2010) 3993–4017.

Vin, J. P, & M. G Ierapetritou. "A new approach for efficient rescheduling of multiproduct batch plants." *Industrial & Engineering Chemistry Research* 39, no. 11: (2000) 4228–4238.

Westerlund, J, M Hastbacka, S Forssell, & T Westerlund. "Mixed-time mixed-integer linear programming scheduling model." *Industrial & Engineering Chemistry Research* 46, no. 9: (2007) 2781–2796.

Wiecek, M. M, M Ehrgott, G Fadel, & J. R Figueira. "Multiple criteria decision making for engineering." *Omega-International Journal Of Management Science* 36, no. 3: (2008) 337–339.

Wu, D, & M. G Ierapetritou. "Decomposition approaches for the efficient solution of short-term scheduling problems." *Computers & Chemical Engineering* 27, no. 8-9: (2003) 1261–1276.

Xhafa, F, & A Abraham. *Metaheuristics for scheduling in industrial and manufacturing applications*. Springer, 2008, 1st Edition.

Yao, Z, & X Yuan. "An approach to optimal design of batch processes with waste minimization." *Computers and Chemical Engineering* 24: (2000) 1437–1444.

You, F. Q, P. M Castro, & I. E Grossmann. "Dinkelbach's algorithm as an efficient method to solve a class of MINLP models for large-scale cyclic scheduling problems." *Computers & Chemical Engineering* 33, no. 11: (2009) 1879–1889.

Young, D. M, & H Cabezas. "Designing sustainable processes with simulation: the waste reduction (WAR) algorithm." *Computers and Chemical Engineering* 23: (1999) 1477–1491.

Zeballos, L. J, P. M Castro, & C. A Mendez. "Integrated Constraint Programming Scheduling Approach for Automated Wet-Etch Stations in Semiconductor Manufacturing." *Industrial & Engineering Chemistry Research* 50, no. 3: (2011) 1705–1715.

Zentner, M. G, A Elkamel, J. F Pekny, & G. V Reklaitis. "A language for describing process scheduling problems." *Computers & Chemical Engineering* 22, no. 1–2: (1998) 125–145.

Zhang, Z, & R. W. H Sargent. "Optimal operation of mixed production facilities-a general formulation and some approaches for the solution." *Computers & Chemical Engineering* 20, no. 6–7: (1996) 897–904.