**ADVERTIMENT**. La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX (www.tesisenxarxa.net) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

**ADVERTENCIA**. La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR (www.tesisenred.net) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

**WARNING**. On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX (www.tesisenxarxa.net) service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the TDX service. Introducing its content in a window or frame foreign to the TDX service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author

# Techniques for the Efficient Solution of Large-scale Production Scheduling & Planning Problems in the Process Industries

# Techniques for the Efficient Solution of Large-scale Production Scheduling & Planning Problems in the Process Industries

Georgios M. Kopanos

A Thesis presented for the degree of
Doctor of Philosophy

Director Prof. Dr. Luis Puigjaner
Co-director Prof. Dr. Michael C. Georgiadis

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

December 2010

**Copyright © 2010 by Georgios M. Kopanos**
The copyright of this thesis rest with the author. No quotations of it should be published without the author's prior written consent and information derived from it should be acknowledged.
Trademarked names are used in this book without the inclusion of a trademark symbol. These names are used in an editorial context only; no infringement of trademark is intended. All the trademarked names cited in this thesis are © of their respective owners.
This document has been prepared using LaTeX.

*Dedicated to my daughter, Elisabet*

*You see things; and you say, " Why? ".*
*But I dream things that never were; and I say, " Why not? ".*

George Bernard Shaw (1856 – 1950)

# Summary

Nowadays, due to the rapidly changing economic and political conditions, global companies face a continuous challenge to constantly re-evaluate and optimally configure the operations of their Supply Chain (SC) for achieving key performance indices, such as profitability, cost reduction and customer service. Companies seek to optimize their global SCs in response to competitive pressures or to acquire advantage of new flexibility in the restrictions on world trade. Process industries also follow this trend. The process systems engineering community has been aware of this change and, today, is playing a key role in expanding the system boundaries from chemical process systems to business process systems. The global optimization of a SC network is an extremely complex task. For this reason, SC decisions are typically divided into three decision levels: the operational (scheduling), the tactical (planning), and the strategic (design).

Production planning and scheduling constitute a crucial part of the overall SC decision level pyramid. Planning and scheduling activities are concerned with the allocation over time of scarce resources between competing activities to meet these requirements in an efficient fashion. More specifically, the planning function aims to optimize the economic performance of the enterprise, as it matches production to demand in the best possible way. The production scheduling component is of vital importance as it is the layer which translates the economic imperatives of the plan into a sequence of actions to be executed on the plant floor, so as to deliver the optimized economic performance predicted by the higher-level plan. Overall, recent research is directed toward finding solutions that enable efficient and accurate handling of problems of large size and increasing complexity. However, there remains significant work to be done on both model enhancements and improvements in solution algorithms, if industrially-relevant problems are to be tackled routinely, and software based on these are to be used on a regular basis by practitioners in the field. In addition, new academic developments are mostly tested on complex but relatively small- to medium-size problems. Therefore, the implementation of new production and scheduling approaches in real-life industrial case studies constitutes a challenging task.

Since most academic developments are too distant from industrial environments, the aim of this thesis is to be a step forward in narrowing the gap between planning and scheduling theory and practice by devising efficient mathematical approaches for solving real-life industrial scheduling and planning problems.

An overview of production planning and scheduling, an analysis of existing approaches, methods and tools used throughout this study are first presented. The second part of this thesis is focused on the development of mathematical models for production processes with continuous parallel units. This problem arises in a number of different production environments. In this part, a novel mathematical programming framework is developed based on elegant modeling of the underlying problem. This work addresses challenging problems in a highly complex real-life bottling facility. The proposed framework addresses appropriately important changeover aspects such as changeover carryover and crossover, thereby leading to solutions resulting in higher utilization of resources.

The third part is focused on semicontinuous industries, which combine continuous and batch operation modes in their overall production process. First, a mathematical programming framework and a solution strategy are presented for the optimal production scheduling of multiproduct multistage semicontinuous process industries. A real-life ice-cream production facility has been considered. Second, a general mathematical programming approach is developed for the resource-constrained production planning problem in semicontinuous processes. This work has been motivated by a challenging problem in food processing industries related to yogurt production lines, where labor (i.e., the number of available workers) constitutes the limited resource constraint. Third, a novel mathematical formulation for the simultaneous optimization of production and logistics operations planning in large-scale single- or multi-site semicontinuous process industries is proposed. Alternative transportation modes are considered for the delivery of final products from production sites to distribution centers, a reality that most of the current approaches neglect. Two industrial-size case studies for a real-life dairy industry have been solved to optimality.

The forth part of the thesis deals with scheduling in batch processes. First, a real-life multiproduct multistage pharmaceuticals production facility is considered. A systematic two-stage iterative solution strategy, based on mathematical programming, has been developed to address this problem. Additionally, a new precedence concept have been developed in order to cope with objectives containing changeover issues. A salient feature of the proposed approach is that the scheduler can maintain the number of decisions at a reasonable level, thus reducing appropriately the solution search space. This usually results in manageable model sizes that often ensures a more stable and predictable optimization model behavior. Finally, a preliminary two-layered decomposition method to the batch process scheduling problem in multipurpose production plants is developed. The procedure is tested on published problem instances of a broadly-studied benchmark scheduling problem that considers a polymers production plant.

# Resumen

Hoy en día, debido a que las condiciones económicas y políticas cambian rápidamente, las empresas globales se enfrentan a un desafío continuo para reevaluar constantemente y configurar de forma óptima las operaciones de su cadena de suministro (CS) para alcanzar los índices de rendimiento clave, tales como la reducción de costes de rentabilidad y servicio al cliente. Las empresas buscan optimizar sus CSs en respuesta a presiones de la competencia o para adquirir ventaja de una mayor flexibilidad en las restricciones sobre todo en el comercio mundial. Las industrias de proceso también siguen esta tendencia. La comunidad que investiga la ingeniería de los sistemas de procesos ha sido consciente de este cambio y, hoy en día, está jugando un papel clave en la expansión de los límites de los sistemas más allá de los procesos químicos para incluir también los sistemas de negocio. La optimización global de una red CS es una tarea extremadamente compleja. Por esta razón, las decisiones CS por lo general contemplan tres niveles de decisión: operativo (programación de operaciones), táctico (planificación de la producción) y estratégico (diseño).

La planificación de la producción y la programación de operaciones constituyen una parte crucial de los niveles de decisión jerarquizados de la CS completa. Las actividades de planificación y programación tratan de la asignación en el tiempo de los recursos escasos entre actividades que compiten para satisfacer de forma eficiente dichas necesidades. Más concretamente, la función de planificación tiene como objetivo optimizar el rendimiento económico de la empresa, ya que debe hacer coincidir la producción con la demanda de la mejor manera posible. El componente de programación de la producción es de vital importancia ya que es la capa que traduce los imperativos económicos del plan en una secuencia de acciones a ser ejecutadas en la planta, con el fin de ofrecer el rendimiento económico optimizado previsto por el plan de alto nivel. En general, las investigaciones recientes se dirigen a la búsqueda de soluciones que permitan un manejo eficiente y preciso de problemas de gran tamaño y de complejidad cada vez mayor.

Sin embargo, queda mucho trabajo por hacer tanto en las mejoras del modelo como en las mejoras en los algoritmos de solución del problema, cuando se trata de abordar de manera rutinaria problemas relevantes para la industria, donde el software producido debe ser utilizado de manera regular por los profesionales en el campo. Además, los nuevos desarrollos académicos son en su mayoría de cierta complejidad, pero relativamente de pequeño tamaño comparados con los problemas industriales incluso de mediano tamaño. Por lo tanto, la aplicación de nuevas estrategias de producción y nuevos enfoques de programación en los estudios industriales en la vida real constituye un reto difícil.

Como la mayoría de los desarrollos académicos están demasiado lejos del entorno de aplicabilidad industrial, el objetivo de esta tesis es dar un paso significativo en la reducción del salto existente entre la teoría y la práctica de la planificación y programación mediante la elaboración de enfoques eficaces de programación matemática para la solución de los problemas de planificación y programación en la vida real industrial.

Primero se presenta una perspectiva de la programación y la planificación de producción, un análisis de los enfoques actuales y los métodos usados en esta tesis. La segunda parte de la tesis se enfoca en el desarrollo de modelos matemáticos de los procesos de producción continuos con unidades en paralelo. Este problema se suscita en una variedad de entornos de producción diferentes. En este apartado, se desarrolla un novedoso marco de programación matemática basado en una modelización elegante del problema subyacente. El trabajo realizado se centra en la problemática difícil que plantea una planta embotelladora de alta complejidad en la vida real donde se producen cientos de productos finales. El entorno de solución propuesto aborda adecuadamente aspectos importantes tales como la prórroga de cambio y el cruce, lo que conduce a soluciones con una mayor utilización de los recursos.

La tercera parte se centra en las industrias semicontinuas, que combinan los modos de funcionamiento continuo y por lotes en su proceso de producción global. En primer lugar se presentan, un marco de programación matemática y una estrategia de solución para la programación de la producción óptima de la industria de proceso semicontinuo multiproducto y multi-etapa. Se ha considerado el problema de una instalación real destinada a la producción de helados, el cual se ha resuelto satisfactoriamente. En segundo lugar, se ha desarrollado un enfoque general de programación matemática para el problema de planificación de la producción con recursos limitados compartidos en el caso de los procesos semicontinuos. El trabajo ha sido motivado por un problema de difícil solución en las industrias alimentarias relacionados con las líneas de producción de yogur, donde la mano de obra (es decir, el número de trabajadores disponibles) constituye la restricción de recursos limitados. En tercer lugar, se propone una nueva formulación matemática para la optimización simultánea de las operaciones de producción y logística de planificación a gran escala en industrias de proceso semicontinuo que contemplan uno o varios centros de producción. Se consideran modos alternativos de transporte para la entrega de los productos finales desde los sitios de producción a los centros de distribución, una realidad que la mayoría de los enfoques actuales se

ignora totalmente. Los resultados alcanzados se aplican a dos estudios, sacados de la realidad, de tamaño industrial relacionados con la industria láctea.

La cuarta parte de la tesis se trata de la programación de operaciones en los procesos por lotes. En primer lugar, se estudia una planta farmacéutica real multiproducto y multi-etapa. Para hacer frente a este problema se ha desarrollado una estrategia sistemática iterativa en dos etapas, basada en programación matemática. Además, se ha desarrollado un nuevo concepto de precedencia con el fin de hacer frente a los objetivos que contienen temas de cambio de producto. Una característica sobresaliente del enfoque propuesto es que el programador puede mantener el número de decisiones a un nivel razonable, lo que reduce adecuadamente el espacio de búsqueda de la solución. De esta forma resultan usualmente modelos de tamaño manejable que suele asegurar un comportamiento más estable y predecible del procedimiento de optimización. Por último, se desarrolla un método de descomposición preliminar de dos capas para el problema de programación de procesos por lotes en las plantas de producción de usos múltiples (multipropósito). El procedimiento desarrollado se prueba en diversos casos, utilizando un estudio publicado, referenciado como banco de pruebas (benchmark), que trata de una planta real para la producción de polímeros.

# Acknowledgments

Undoubtedly, I am truly indebted to my thesis supervisor, Prof. Luis Puigjaner, for his mentorship and encouragement along these last four years, since he has supported me with his patience and knowledge whilst allowing me the room to work in my own way. I was delighted to have Prof. Michael C. Georgiadis as my co-advisor because he directed me to focus my research on industrial scheduling and planning problems. His ability to question every aspect of any idea I had was of the utmost importance making me think of new ways of approaching a problem. Also, I do owe my deepest gratitude to my family, my friends and colleagues.

I gratefully acknowledge the financial support received from the Spanish Ministry of Education that awarded me the FPU Grant (Becas del Programa de Formación del Profesorado Universitario) to carry out this thesis, and granted me with three additional Mobility Grants (Ayudas para Estancias Breves) allowing me to realize part of my studies in foreign universities.

I would also like to express my gratitude to Prof. Christos T. Maravelias (University of Wisconsin-Madison, the USA), to Prof. Tapio Westerlund (Åbo Akademi University, Finland), and to Prof. Nilay Shah (Imperial College London, UK) for their hospitality and the fruitful discussions and collaborations made during my secondment to their research groups. I also thank Prof. Carlos A. Méndez (Universidad Nacional del Litoral, Argentina) for the collaboration in one of the chapters of this thesis.

Last, but foremost, I offer my sincerest gratitude to my wife, Alexia, who has fully supported me throughout my research with her love, encouragement, patience and apprehension. Without her it would have been impossible for me to finish this work.

# Contents

# List of Tables

# List of Figures

Part I

# Overview

# Chapter 1

# Introduction

## 1.1 Introduction to Manufacturing Types

Nowadays, the two major manufacturing disciplines are process manufacturing, and discrete manufacturing. In contrast with discrete manufacturing, process manufacturing results in final (and often intermediate) products that cannot be disassembled back into their original components. For instance, a can of soda cannot be returned to its basic components such as carbonated water, citric acid, potassium benzoate, aspartame, and other ingredients. Orange juice cannot be put back into an orange. However, a car or computer, on the other hand, can be disassembled and its components, to a large extent, returned to stock. In discrete manufacturing the manufacturing floor works off shop orders to build something and the individual products are easily identifiable. The automobile, the computer, and the aerospace industry are some representative discrete industries. Process manufacturing is common in the food, beverages, chemicals, pharmaceuticals, petroleum, ceramics and paper industries.

Roughly speaking, machines that do the main processing operations typically have very high startup and shutdown costs and usually work around the clock. A machine in the process industries also incurs a high changeover cost when it has to switch over from one product to another. In process manufacturing, the products are undifferentiated. Process manufacturing is the branch of manufacturing that is associated with formulas and manufacturing recipes, and can be contrasted with discrete manufacturing, which is concerned with bills of material and routing. This is more than a subtle difference in terminology; the terms characterize distinct manufacturing approaches. This thesis is focused on process manufacturing with emphasis on food, pharmaceutical and chemical products.

## 1.2 Overview of Process Industry and Motivation

According to the European Chemical Industry Council (2010), the European Union (EU) is the second largest chemicals producing area in the world. The EU accounts for 24%, worth 449 billion €, of the total global chemical sales in 2009. Moreover, the EU is the world's top exporter and importer of chemicals, accounting for more than 40% of total global trade in 2009. Of the thirty largest chemical companies in the world, twelve are headquartered in Europe (e.g., BASF, Bayer, Shell, Ineos, Total) representing around 10% of world chemical sales. Pharmaceuticals and chemicals form the first- and second-leading EU manufacturing sector in terms of value-added per employee in 2006, respectively. It is also noteworthy that food and beverages industry hold the seventh place in this list.

The significant role of the process industry in the EU's economic status is evident. EU's process industry ought to be active and improve its operational and functional performance through its entire supply chain network, in order to maintain its leading position in the world's highly competitive market. The faster growth rhythm of Asian countries, especially China and Japan, has created a strong competitor for EU process industries, thus making indispensable the enhancement of the production management and the overall supply chain network.

### 1.2.1 Supply chain management

According to Min and Zhou (2002), Supply Chain (SC) is referred to as an integrated system which synchronizes a series of inter-related business processes in order to: (i) acquire raw materials and parts, (ii) transform these raw materials and parts into final products, (iii) add value to theses products, (iv) distribute and promote final products to either retailers or customers, and (v) facilitate information exchange among various business entities (i.e., vendors, manufacturers, distributors, third-party logistics providers, retailers). SC networks consist of a number of echelons (e.g., suppliers, production plants, warehouses, distribution centers, markets) and are described by a forward flow of materials and a backward flow of information. The SC is often represented as a network similar to the one illustrated in Figure 1.1.

According to the Council of Supply Chain Management Professionals (2007), Supply Chain Management (SCM) is the process of planning, implementing and controlling the SC operations in an efficient way. SCM spans all movements (mainly logistics operations) and storage of raw materials, work-in-process inventory, intermediate products and finished goods from the point-of-origin to the point-of-consumption (Simchi-Levi, Kaminsky, & Simchi-Levi, 2004). SCM crystallizes about integrated business planning that have been espoused by logistics experts, strategists, and operations research practitioners as far back as 1950s. In the last decade, they have been a few changes in business environment that have contributed to the development of SC networks. SCM may be considered as an outcome of globalization and the proliferation of multi-national companies, joint ventures, strategic alliances and business partnerships, there were found to be signifi-

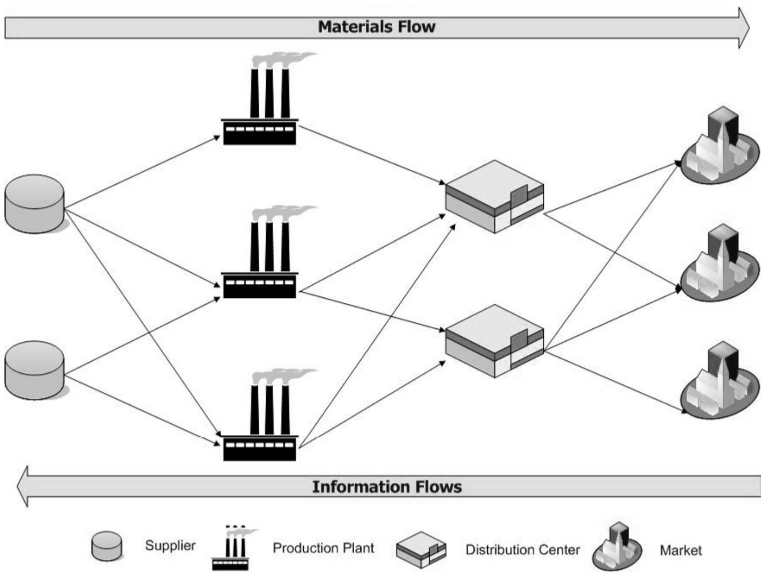



**Figure 1.1:** A typical supply chain network.

cant success factors, following the earlier "Just-In-Time", "Lean Management" and "Agile Manufacturing" practices. SC constitutes a special case of the value chain, a term first introduced by Porter (1985), for those companies that produce and distribute physical products. Competitive advantage accrue to those companies that control their value chain costs better than their competitors. An alternative way to gain competitive advantage is by differentiating their products by providing some combination of superior quality, customer service, product variety, unique market present, and so on (Shapiro, 2007).

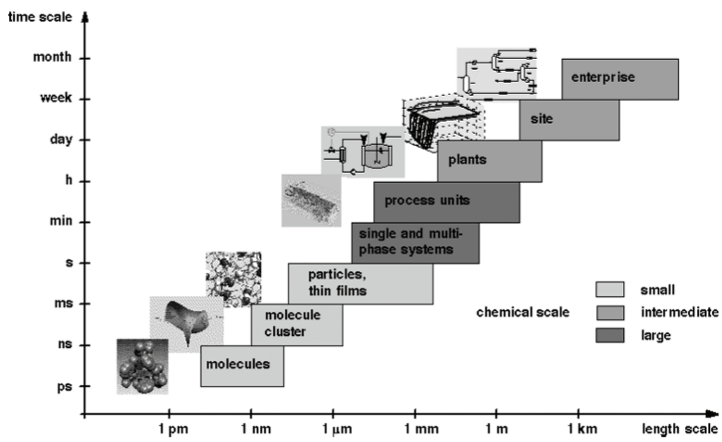



**Figure 1.2:** The "chemical supply chain" (Marquardt *et al.*, 2000).

In today's rapidly changing economic and political conditions global corporations face a continuous challenge to constantly evaluate and optimally configure their SC operations to achieve key performance indices, either it is profitability, cost reduction or customer service (Tsiakis & Papageorgiou, 2008). The Process Systems Engineering (PSE) community is performing a key role in extending system boundaries from chemical process systems to business process systems. PSE has traditionally been concerned with the understanding and development of systematic procedures for the design, control and operation of chemical process systems (Sargent, 1991). The scope of PSE can be broadened by making use of the concept of the "chemical supply chain" (see Figure 1.2). According to Grossmann and Westerberg (2000), PSE may be defined as the field of study that is concerned with the improvement of decision-making processes for the creation and operation of the "chemical supply chain".

### 1.2.2 Planning and scheduling

There are three SC decision-making levels: the operational, the tactical, and the strategic (Shapiro, 2007). More specifically, the operational level deals with short-term scheduling problems, the tactical level involves medium-term planning decisions, and the strategic level corresponds to the SC design problem. This thesis is primarily focused on scheduling and planning problems. However, a contribution on SC design has been also realized during the Ph.D. studies (see Appendix A).

Kreipl and Pinedo (2004) presented a thorough overview in production planning and scheduling. According to this, a *production planning model* typically optimizes several consecutive stages in a SC (i.e., a multi-echelon model), with each stage having one or more facilities. Such a model is designed to allocate the production of the different products to the various facilities in each time period, while taking into account inventory holding costs and transportation costs. A planning model may make a distinction between different product families, but usually does not make a distinction between different products within a family. It may determine the optimal production run (or, equivalently, the batch size or lot size) of a given product family when a decision has been made to produce such a family at a given facility. If there are multiple families produced at the same facility, then there may be setup costs and setup times. The optimal production run of a product family is a function of the trade-off between the setup cost and/or setup time and the inventory cost. Generally speaking, the main objectives in planning involves inventory costs, transportation costs, tardiness costs, and the major setup costs. However, in a medium-term planning model, it is typically not customary to take the sequence dependency of setup times and setup costs into account. The sequence dependency of setups is difficult to incorporate in an integer programming model because it can increase significantly the complexity of the formulation.

A detailed *scheduling model* is typically concerned with a single facility. Such a model usually takes more detailed information into account than a planning model. It is typically assumed that there are a given number of jobs and each one has its own parameters; including sequence-dependent setup times and sequence-dependent setup costs. The jobs have to be scheduled in such a way that one or more objectives are optimized (e.g., minimization of: makespan, weighted lateness, or, total setup costs).

Planning models differ from scheduling models in a number of ways. First, planning models often cover multiple stages and optimize over a medium-term horizon (e.g., weeks, or months), whereas scheduling models are usually designed for a single facility and optimize over a short-term horizon (e.g., hours, or days). Second, planning models use more aggregate information, whereas scheduling models use more detailed information. Third, the objective to be minimized in a planning model is typically a total cost objective and the unit in which this is measured is a monetary unit; the objective to be minimized in a scheduling model is typically a function of the completion times of the jobs and the unit in which this is measured is often a time unit. Nevertheless, even though there are fundamental differences between these two types of models, they often have to be incorporated into a single framework, share information, and interact extensively with one another (Kreipl & Pinedo, 2004).

## 1.3 Thesis Outline

The general structure of this thesis has been devised bearing in mind the different types of production processes in the process industry sector. Figure 1.3 illustrates the outline of the thesis.

Part I, in addition to this introductory chapter, presents, in Chapter 2, a state-of-the-art review, which finally allows identifying some production scheduling and planning trends and challenges. Afterwards, in the subsequent chapter, the methods and tools used throughout this thesis are briefly outlined.

The main body of this thesis has been divided in three parts. Part II deals with continuous processes. Specifically, in Chapter 4, a novel mathematical approach to the simultaneous production planning and scheduling of continuous parallel units producing a large number of final products that can be classified into product families has been developed. This problem appears in many stages of operation in the process industries, including packing in batch and continuous production facilities. Thus, it is quite important since it arises in a number of different production environments (e.g., food and beverage industry, consumer packed goods). The proposed approach has been used to solve a complex real-world problem in the continuous bottling stage of the Cervecería Cuauhtémoc Moctezuma beer production facility, located in Mexico.

Part III focuses on food process industries that combine batch and continuous operation modes in their overall production plant. In Chapter 5, a mixed integer programming framework and a solution strategy are presented for the optimal production scheduling of multiproduct multistage semicontinuous process industries. An ice-cream production facility (UNILEVER, the Netherlands) is studied in detail. The overall mathematical framework relies on an efficient modeling approach of the sequencing decisions, the integrated modeling of all production stages and the

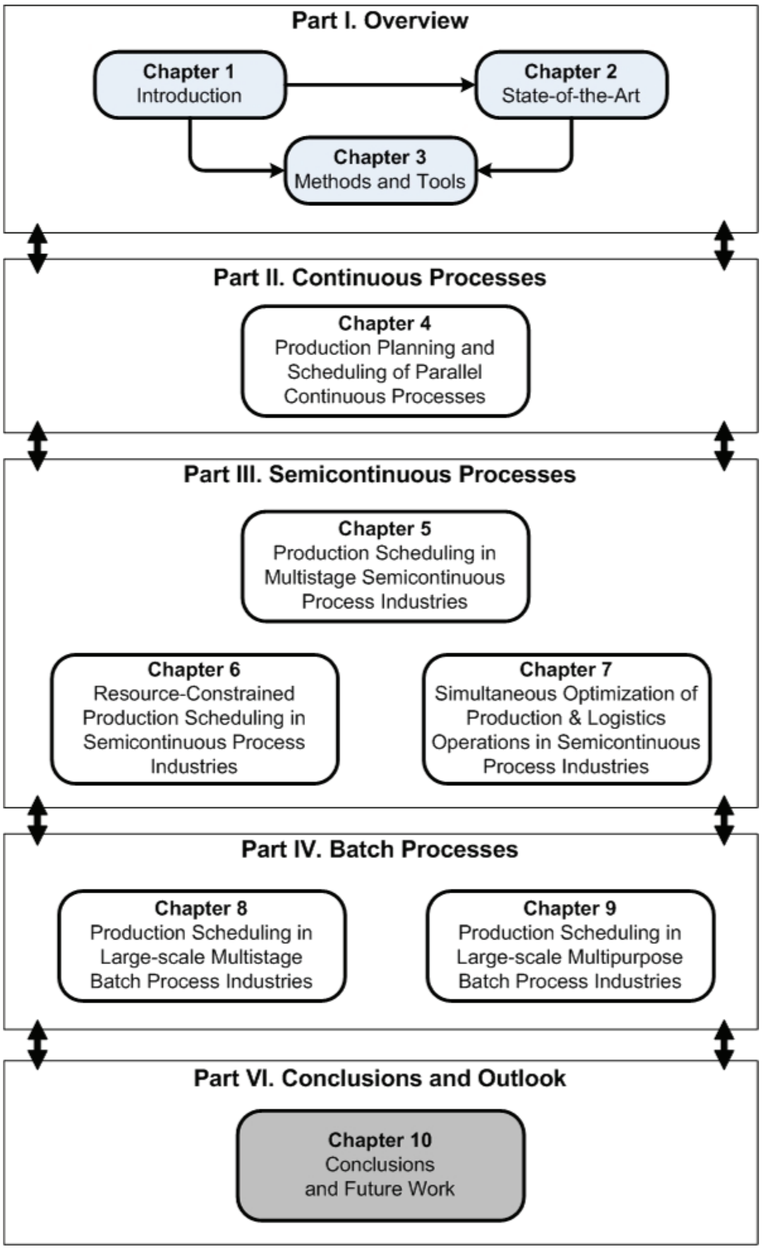



**Figure 1.3:** Thesis outline.

inclusion of strong valid integer cuts in the formulation. The simultaneous optimization of all processing stages increases the plant production capacity, reduces the production cost for final products, and facilitates the interaction among the different departments of the production facility. The proposed mathematical formulation is well-suited to the ice-cream production facility considered, however it could be also used, with minor modifications, in scheduling problems arising in other semicontinuous industries with similar processing features. Then, in Chapter 6, a general mathematical programming approach is presented for the resource-constrained production problem in semicontinuous processes. This work has been motivated by a challenging problem in food processing industries related to yogurt production lines (KRI-KRI dairy industry, Greece), where labor (i.e., the number of available workers) constitutes the limited resource constraint. The proposed mathematical approach can also cope with unexpected events such as workers absence, and products orders modifications. Finally, Chapter 7 presents the simultaneous production and logistics operations planning in large-scale single- or multi-site semicontinuous process industries. A novel mixed discrete/continuous-time mixed integer programming model for the problem under consideration has been developed. A remarkable feature of the proposed approach is that in the production planning problem timing and sequencing decisions are taken for product families rather than for products. However, material balances are realized for every specific product, thus permitting the detailed optimization of production, inventory, and transportation costs. Moreover, alternative transportation modes are considered for the delivery of final products from production sites to distribution centers. The proposed approach has been used to solve two industrial-size case studies, for an emerging real-life dairy industry (KRI-KRI dairy industry, Greece).

Part IV deals with scheduling in batch processes. Chapter 8 presents an efficient systematic iterative solution strategy, based on mathematical programming, for the efficient solution of real-world scheduling problems in multiproduct multistage batch plants. The proposed strategy consists of a constructive step, wherein a feasible and initial solution is rapidly generated by following an iterative insertion procedure, and an improvement step, wherein the initial solution is systematically enhanced by implementing iteratively several rescheduling techniques; based on the mathematical model. A salient feature of the proposed approach is that the scheduler can maintain the number of decisions at a reasonable level thus reducing appropriately the search space. This usually results in manageable model sizes that often guarantees a more stable and predictable optimization model behavior. Several challenging large-scale problem instances, considering alternative optimization goals, of a pharmaceuticals production facility (ABB, Germany) have been solved. In Chapter 9, a new two-layered decomposition methodology to the batch process scheduling problem in multipurpose production plants is proposed. In the first level, an approximate scheduling model derived from the detailed STN-based time-indexed scheduling formulation is solved. The model partially relaxes the allocation of task instances to processing units details of the full scheduling formulation. In the second level, the output of the approximate scheduling problem is used to provide batching targets for the detailed scheduling model within an iterative decomposition scheme. The applicability and efficiency of the proposed approach is tested on published problem instances of the Westenberger-Kallrath (W-K) benchmark scheduling problem.

Finally, Chapter 10 summarizes the main contribution of this thesis and draws up concluding remarks for future work.

# Chapter 2

# State-of-the-Art

## 2.1 Introduction

Planning and scheduling techniques have been employed in the process industries since the early 1940s (Shobrys, 2001), even before the first computers appeared. Indeed the oil industry was one of the early adopters of planning methods based on the formulation and solution of linear programming models, with applications already reported in the late 1950s. But, it is since the early 1980s in particular that the theme of production planning and scheduling for the process industries has received significant attention. Initially, from the early 1980s to the early 2000s, this was due to the resurgence in interest in flexible processing either as a means of ensuring responsiveness or adapting to the trends in process industries towards lower volume, higher value-added materials in the developed economies (Reklaitis, 1982). More recently, the topic has received a new impetus as enterprises attempt to optimize their overall Supply Chain (SC) in response to competitive pressures or to take advantage of recent relaxations in restrictions on global trade. The planning and/or scheduling problem at a single site is usually concerned with meeting fairly specific production requirements. Customer orders, stock imperatives or higher-level SC or long-term planning would usually set these. Then, the planning/scheduling activity is concerned with the allocation over time of scarce resources between competing activities to meet these requirements in an efficient fashion.

The objective in production planning is to determine the production and inventory levels that will allow to fulfill given customer demand at the minimum cost (including processing, holding and backlog, and switchover costs) subject to (typically aggregate) production capacity constraints. Thus, a production planning solution consists of production targets and inventory levels over a number of periods into which the planning horizon under consideration is partitioned. On the other hand, the objective in scheduling is the allocation of limited resources (e.g., equipment units, utilities, manpower) to competing tasks and the sequencing and timing of tasks on units, given a set of production targets and subject to detailed production constraints. The production scheduling component is of vital importance as it is the layer which translates the economic imperatives of the plan into a sequence of actions to be executed on the plant, so as to deliver the optimized economic performance predicted by the higher-level plan. Clearly, the two problems are interdependent since the solution of production planning (production targets) is input to scheduling, and the production capacity constraints in production planning depend on the scheduling solution.

The key components of the planning and/or scheduling problem are: *resources*, *tasks* and *time*. The resources need not be limited to processing equipment items, but may include material storage equipment, transportation equipment (intra- and inter-plant), operators, utilities (e.g., steam, electricity, cooling water), auxiliary devices and so on. The tasks typically comprise processing operations (e.g., reaction, separation, blending, packing) as well as other activities which change the nature of materials and other resources such as transportation, quality control, cleaning, changeovers, etc. There are both external and internal elements to the time component. The external element arises out of the need to co-ordinate manufacturing and inventory with expected product liftings or demands, as well as scheduled raw material receipts and even service outages. The internal element relates to executing the tasks in an appropriate sequence and at right times, taking account of the external time events and resource availabilities. Overall, this arrangement of tasks over time and the assignment of appropriate resources to the tasks in a resource-constrained framework must be performed in an efficient fashion, which implies the optimization, as far as possible, of some objective. Typical objectives include the minimization of cost or maximization of profit, maximization of customer satisfaction, minimization of deviation from target performance, etc.

## 2.2 Types of Production Processes

In this section, the major production process patterns found in the process industries are presented. Production process can be classified as *continuous*, *semicontinuous* and *batch*. A brief description of these processes follows.

### 2.2.1 Continuous processes

In the continuous processing mode, units are continuously fed and yield a constant product flow. For mass production of similar products, continuous processes can achieve higher consistent product quality, taking advantage of the economies of scale and reducing manufacturing costs and waste. Processes from the petrochemical industry are usually good examples of continuous production processes. The most important difference between batch production and continuous production is that any changes in the product's properties such as color, dimensions, or quality needs to be done online. And whenever it is effected, the results can be seen only after a fixed period which can extend from a few hours to days. Moreover, maintenance in case of continuous process plants calls for online maintenance which requires very high alertness and quick response times from dedicated technicians.

### 2.2.2 Semicontinuous processes

Semicontinuous processing offers a more customized operation for highly dynamic and uncertain environments. Semicontinuous operations are characterized by their processing rate, running continuously with periodic startups and shutdowns for frequent product transition. The processing times of semicontinuous processes are relatively long periods of time called campaigns, each dedicated to the production of a single product. Typical campaign lengths range from a few hours to several days. Most process plants in the process industry combine continuous operations and batch processes in their product processing routes thus working in semicontinuous mode, since production is more flexible and equipment can be more efficiently utilized.

### 2.2.3 Batch processes

The primary characteristic of the batch production process is that all components are completed at a workstation before they move to the next one. Batch production is popular in the manufacture of pharmaceutical ingredients, inks, paints, adhesives and a plethora of contemporary commodities. Batch production is useful for a factory that makes seasonal items or products for which it is difficult to forecast demand. There are several advantages of batch production; for instance, it can reduce initial capital outlay because a single production line can be used to produce several products. Batch production can be useful for small businesses who cannot afford to run continuous production lines. It is worth mentioning that companies may use batch production as a trial run.

Despite the fact that batch processing has been traditionally associated with specialty chemicals and products of high-added value (e.g., pharmaceuticals), the demand patterns can be so unpredictable that profitability may only be achieved by taking full advantage of the inherent flexibility of a batch production facility. Therefore, in order to reduce the risk of new investment, batch plants are preferred as an adequate and flexible answer to the variability in the supply of raw materials, the manufacturing of diverse products and the instability of product demands.

At this point, it is worth mentioning that processing times constitute one of the major differences between scheduling of batch and continuous processes. On the one hand, in batch plants the processing times are typically fixed and known a priori. Moreover, the production amount depends on the capacity of the batch processing unit. On the other hand, in continuous plants the processing times are a function of unit-dependent processing rates, final product demand and storage limitations. Additionally, in continuous plants, the production amount is available continuously while it is being produced, unlike in batch plants, where the produced quantity is available only after the end time of the batch that is being processed.

### Batch Process Types

According to Rippin (1993), traditional batch (and therefore semicontinuous) production facilities can be classified into *multiproduct* and *multipurpose*.

In multiproduct plants, each product has the same processing network. That means that each product requires the same sequence of processing tasks (often known as stages) and thus there is only one way to produce a specific product; although some products may skip some task in the sequence. Due to the historic association between the work on batch plant scheduling and that on discrete parts manufacturing, these plants are sometimes called flowshops in the Operational Research (OR) literature. In multipurpose plants, the products are manufactured via different processing networks, and there may be more than one way in which to manufacture the same product. In general, a number of products undergo manufacture at any given time. Flow patterns are not straight lines, as in the multiproduct case, and some units may be used to perform non-consecutive operations for the same product. Multipurpose plants result in more flexible operation, which can be optimized to decrease equipment idle time to more efficiently utilize critical equipment units. In the OR literature, multipurpose plants are sometimes referred to as jobshops. It should be emphasized that these batch processes do not fall under the usual flowshop or jobshop in the OR, because the number of jobs (tasks) to be executed is not known a priori, processing times can depend on job size, and jobs are linked to each other through material balance constraints and intermediate storage requirements.

It is worth mentioning that *pipeless plants* have been discussed in the literature (Takahashi & Fujii, 1991) as potential alternatives to traditional batch plants. Their main distinguishing feature is that material is transported from one processing stage to another in transferable vessels. Processing takes place at a number of processing stations, and normally the same vessels used for transferring material also hold the material while it is being processed at each station. When necessary, cleaning of the vessels takes place at specialized cleaning stations. The elimination of fixed piping networks for material transfer enables pipeless plants to be considerably more flexible than their conventional counterparts. Nowadays, pipeless plants are still scarce in the process industries and adapted to every particular case. For this reason such plants are not further discussed in the current thesis.

### Intermediates Storage Policies

Storage for intermediate products plays a significant role in exploiting the inherent flexibility of the batch process. During the operation of batch plants several material storage policies may be implemented. A list and a short description of the main storage tactics follows.

- *Unlimited Intermediate Storage (UIS)*
  This case corresponds to the unrestricted storage case. The material is stable, and one or more dedicated storage vessels are available, the total capacity of which is unlimited. UIS is considered as the best case scenario upper limiting bound for all other solutions since the best attainable solution with the shortest production times is obtained using this policy.
- *No Intermediate Storage (NIS)*
  In this case, the material is stable and there are no storage tanks available for intermediate materials; however, materials can be stored temporarily inside the processing unit, waiting to be transferred to the next unit once it has been empty (this is also true for UIS policy).
- *Zero Wait (ZW)*
  This policy restricts the NIS policy by avoiding the alternative use of processing units as storage facilities for intermediate materials. This policy is usually used in cases where the materials are unstable products that must be transferred to the next processing unit immediately after completion. This is the most restrictive policy and constitutes a lower limiting bound.
- *Finite Intermediate Storage (FIS)*
  Limited storage capacity is available in terms of the number of storage units, their capacities and connections between processing units and tanks. The material is stable, and one or more dedicated storage vessels are available, all of which may be subject to optimization.
- *Shared Intermediate Storage (SIS)*
  The material is stable, and may be stored in one or more storage vessels that may also be used to store other materials (though not at the same time).

## 2.3 Optimization Methods

All but the most trivial scheduling problems belong to the class of NP-hard problems, thus there are no known solution algorithms that are of polynomial complexity in the problem size. This has posed a great challenge to the research community, and a large body of work has arisen aiming to develop either tailored algorithms for specific problem instances or efficient general-purpose methods.

### 2.3.1 Mathematical programming approaches

The application of mathematical programming approaches implies the development of a mathematical framework and the use of an optimization algorithm. Most mathematical approaches aim to develop models that are of a standard form (from linear programming models for refinery planning to mixed integer non-linear programming models for multipurpose batch plant scheduling). These models are then usually solved by commercial software or specialized algorithms that take account of the problem structure (for more details see Chapter 3).

The main decision variables of the mathematical models usually include some or all of the following:

- selection of resources (e.g., units, utilities) to execute tasks at the appropriate times;
- sequence of tasks;
- timing of tasks;
- amounts processed in each task; and
- inventory levels of all materials over time.

The Boolean nature of some of the decisions (e.g., sequencing and resource selection) implies the utilization of binary variables. All variables values will be subject to some or all of the following constraints:

- non-preemptive processing once started (i.e., processing activities must proceed until completion);
- resource constraints at any time (i.e., the utilization of a resource must not exceed its availability);
- material balances;
- capacity constraints for processing and storage; and
- full demand satisfaction for orders by their due-dates (if backlogs are not allowed).

The mathematical programming modeling of scheduling and planning in process industries is focused on four key elements: (i) the time representation, (ii) the event representation, (iii) the material balance approach, and (iv) the objective function.

**Time representation**

The representation of the time horizon is an essential feature of mathematical programming approaches, because processing tasks interact through the use of shared resources and, therefore, the discontinuities in the overall resource utilization profiles must be tracked over time, in order to ensure feasibility by not exceeding the available resource capacities. The complexity arises from the fact that these discontinuities are functions of any schedule proposed and are not known in advance. The three time representation approaches are:

✦ *Discrete-time*: the horizon is divided a priori possibly into a number of equally spaced intervals so that any event that introduces such discontinuities (e.g., the starting of a task or a due-date for an order) can only take place at an interval boundary. This implies a relatively fine division of the time grid, so as to capture all the possible event times, and in the solution to the problem it is likely that many grid points will not actually exhibit resource utilization discontinuities.

- *Continuous-time*: the horizon is divided into fewer intervals, the spacing of which will be determined as part of the solution to the problem (i.e., the time horizon is partitioned as part of the optimization). The number of intervals will correspond more closely to the number of resource utilization discontinuities in the solution.
- *Mixed-time*: the time grid is fixed but the durations of the tasks are variable.

As discussed in Méndez *et al.* (2006), in discrete-time models, constraints have only to be monitored at certain known time points, a fact that reduces the problem complexity and makes the model structure simpler and easier to solve, especially when resource and inventory limitations are considered. On the other hand, this type of problem simplification has two main drawbacks. First, the size of the mathematical model as well as its computational efficiency strongly depend on the number of time intervals postulated, which is defined as a function of the problem data and the desired accuracy of the solution. Second, suboptimal or even infeasible schedules may be generated because of the reduction of the domain of the timing decisions. In order to overcome the previous limitations and generate data-independent models, a continuous-time representation may be employed. In typical continuous-time formulations, a variable time handling allows obtaining a significant reduction of the number of variables and at the same time, more flexible solutions in terms of time can be generated. However, because of the modeling of variable processing times, resource and inventory limitations usually need the definition of more complicated constraints involving many big-M terms, which tends to increase the model complexity and the integrality gap. Discrete-time formulations, despite being a simplified version of the original scheduling problem, have proven to be efficient, adaptable and convenient for some industrial applications, especially in those cases where a reasonable number of intervals is sufficient to obtain the desired problem representation. However, discrete-time models suffer from a number of drawbacks: (i) the discretization interval must be fine enough to capture all significant events, a fact that may result in large model sizes, (ii) it is difficult to model operations where the processing time is dependent on the batch size, and (iii) the modeling of continuous and semicontinuous operations must be approximated, and minimum run lengths give rise to complicated constraints. Concluding, the appropriate selection of the time representation mainly depends on: the production process, the resource limitations, and the objective function of the scheduling and/or planning problem under consideration.

### Material balance

Depending on the handling of batches and batch sizes, scheduling formulations can be broadly classified into: *network-based formulations* for general processes, and *batch-based formulations* for sequential processes. The first category refers to monolithic approaches, which simultaneously deal with the lot-sizing and scheduling problem. These methods are able to deal with arbitrary network processes involving complex product recipes. However, their generality usually implies large model sizes and consequently their application is currently restricted to processes involving a small number of processing tasks and rather short scheduling horizons. Batch-based formulations are used for single-stage, multistage and multipurpose processes where batches are processed sequentially and where batch splitting/mixing are not allowed and there are no recycle streams. In these approaches, the number and the size of batches are known in advance. In other words, the lot-sizing (or batching) problem have already been solved. The batching problem converts the demand of products into individual batches aiming at optimizing some criterion. Afterwards, the available manufacturing resources are allocated to the batches over time. This approximate two-stage approach, widely used in industry, can address much larger practical problems than monolithic methods, especially those involving a quite large number of batch tasks related to different intermediates or final products (Méndez *et al.*, 2006). Network-based formulations typically employ the State-Task Network or the Resource-Task Network process representation in order to formally represent the problem. A description of these process representations follows.

**State-Task Network (STN).** Kondili, Pantelides, and Sargent (1988) introduced the STN process representation, by presenting a discrete-time Mixed Integer Programming (MIP) model. The STN is a directed graph that consists of three key elements: (i) *state nodes* representing feeds, intermediates and final products, (ii) *task nodes* representing the process operations which transform material from one or more input states into one or more output states, and (iii) *arcs* that link states and tasks indicating the flow of materials. State and task nodes are denoted by circles and rectangles, respectively (see Figure 2.1).

The three main advantages of the STN representation are that:

(i) It distinguishes the process operations from the resources that may be used to execute them, and therefore provides a conceptual platform from which to relax the unique assignment assumption and optimize unit-to-task allocation.

(ii) It avoids the use of task precedence relations which become very complicated in multipurpose plants. A task can be scheduled to begin if its input materials are available in the correct amounts and other resources (e.g., processing equipment and utilities) are also available, regardless of the plant history.

(iii) It provides a means of describing very general process recipes, involving batch splitting and mixing, material recycles, and intermediate storage.

As argued by Pantelides (1994) STN, despite its advantages, suffers from a number of drawbacks:

(i) The model of plant operation is somewhat restricted, since each task is assumed to use exactly one major item of equipment during its operation.

(ii) Tasks are always assumed to be processing activities which change material states, therefore changeovers or transportation activities have to be treated as special cases.

(iii) Each item of equipment is treated as a distinct entity, a fact that introduces solution degeneracy if multiple equivalent items exist.

(iv) Different resources (e.g., materials, units, utilities) are treated differently, giving rise to many different types of constraints, each of which must be formulated carefully to avoid unnecessarily increasing the integrality gap.

**Resource-Task Network (RTN).** Pantelides (1994) proposed the RTN process representation, which is based on a uniform description of all resources, as a more general case of the STN representation. In the RTN representation, a task is assumed only to consume and produce resources, in contrast to the STN, where a task consumes and produces materials while using equipment and utilities during its execution. Processing items are treated as though consumed at the beginning of a task and produced at the end. In the RTN representation, circles represent not only states but also other resources required in the batch process such as processing units and vessels (see Figure 2.1). A special feature of the RTN is that processing equipment in different conditions (e.g., "clean" or "dirty") can be treated as different resources, with different activities (e.g., "processing" or "cleaning") consuming and generating them, thus achieving a simpler representation of changeovers. The RTN main advantage over the STN representation lies in its conceptual simplicity and its direct applicability to a large number of complex process scheduling problems. In scheduling problems involving identical equipments, RTN-based formulations overwhelm STN-based models, since they introduce just a single binary variable instead of the multiple variables used by the STN. In few words, RTN-

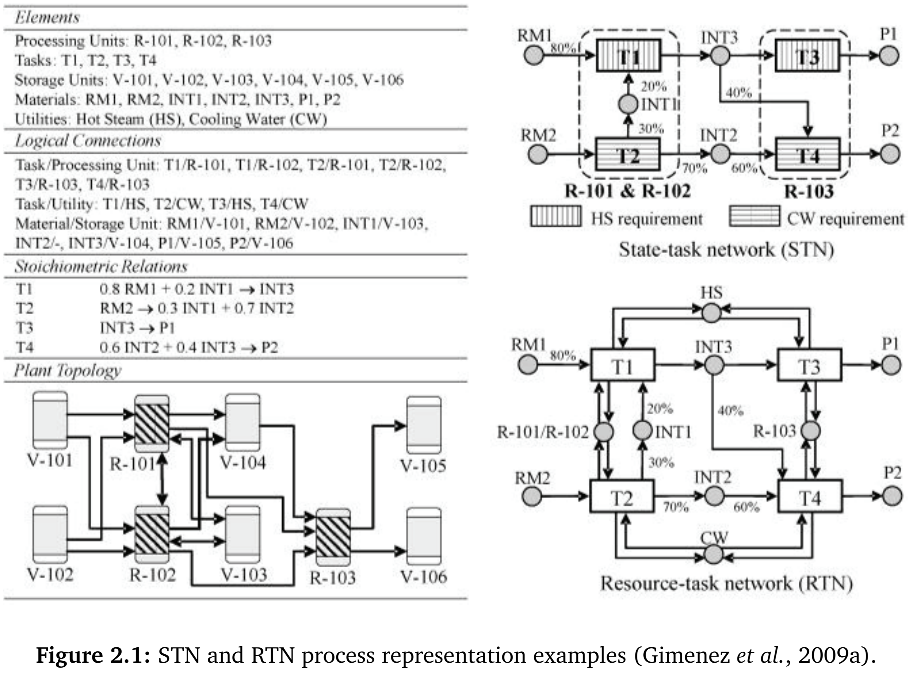



**Figure 2.1:** STN and RTN process representation examples (Gimenez *et al.*, 2009a).

based models reduce the batch scheduling problem to a simple resource balance problem carried out in each predefined time period (Méndez *et al.*, 2006). The ability to capture additional problem features in a straightforward fashion made the RTN representation a promising framework for future research.

**Event representation**

In addition to the time representation and material balances, scheduling models are based on different concepts or basic ideas that arrange the events of the schedule over time with the main purpose of guaranteeing that the maximum capacity of the shared resources is never exceeded (Méndez *et al.*, 2006). In Figure 2.2 are illustrated the five basic event representation concepts. Namely, they are: (a) *global time intervals*, (b) *global time points*, (c) *unit-specific time events*, (d) *time slots*, and (e) *precedence-based*.

Global time intervals are used in discrete-time models. In this event representation fixed time grids are predefined and the tasks are forced to begin and finish exactly at a point of the grid (see Figure 2.2a). Consequently, the original scheduling problem is reduced to an allocation problem where the main model decisions denote the assignment of the time interval at which every task begins. The contributions of Kondili *et al.* (1988); Kondili, Pantelides, and Sargent (1993) and Shah, Pantelides, and Sargent (1993a) are based on this concept.

To continue with, in contrast with discrete-time models, there is a variety of event representations in continuous-time domain formulations. More specifically, network-based models for general processes use global time points or unit-specific time events, while batch-based formulations employ time slots or precedence-based relationships. In the global time points concept the timing of time intervals is treated as a new model variable. Thus, a common and a variable time grid is defined for all shared resources while the starting and the finishing times are linked to specific time points through key discrete variables (see Figure 2.2b). Some of

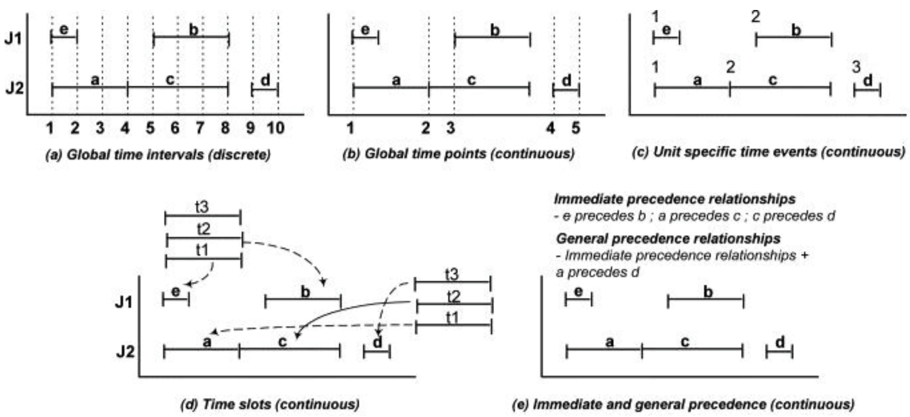



**Figure 2.2:** Types of event representation (Méndez *et al.*, 2006).

the most important works that use this concept are the ones of Castro, Barbosa-Póvoa, and Matos (2001), where an RTN representation is used, and Maravelias and Grossmann (2003). The unit-specific time events concept defines a different variable time grid for each shared resource, allowing different tasks to start at different moments for the same event point (see Figure 2.2c). This concept results to more complicated models compared with the global time points concept, since, due to lack of references points, additional constraints and variables need to be defined for dealing with shared resources. Representative works of the unit-specific time events concept are the ones reported by Ierapetritou and Floudas (1998) and Giannelos and Georgiadis (2002).

The previous event representations are used in network-based formulation for general processes. For sequential processes, time slots and precedence-based formulations have been developed. Indeed some of them have been recently extended to also consider general processes. In the time slots concept a set of an appropriate number of predefined time intervals for each processing unit with unknown durations is firstly postulated in order to allocate them to the tasks (see Figure 2.2d). The choice of the number of time slots required represents an important trade-off between optimality and computational performance. The work of Sundaramoorthy and Karimi (2005) has been based on this concept.

At this point, it should be emphasized that in all the aforementioned event representation concepts a predefined number of time points or slots is needed; a fact that may affect the problem optimal solution if the right number of them is not considered. For this reason, alternative approaches that are based on the concept of task precedence have been emerged (see Figure 2.2e). In these formulations, sequencing binary variables, through big-M constraints, enforcing the sequential use of shared resources are explicitly employed. As a result, sequence-dependent changeovers can be treated in a straightforward manner.

Three precedence concepts have been reported: (i) the immediate precedence, (ii) the unit-specific immediate precedence, and (iii) the general precedence. The immediate precedence concept explores the relation between each pair of consecutive tasks without considering if the orders are assigned or not into the same unit. A representative example of an immediate precedence formulation can be found in Méndez, Henning, and Cerdá (2000). The unit-specific immediate precedence is based on the immediate precedence concept. The difference is that it takes into account only the immediate precedence of the tasks that are assigned into the same unit. The formulation presented by Cerdá, Henning, and Grossmann (1997) is a representative example of unit-specific immediate precedence models. The general precedence generalizes the precedence concept by exploring the precedence relations of every task regarding all the remaining tasks and not only the immediate predecessor. This approach results to a lower number of binary variables, comparing it with the other two approaches, reducing significantly the computational effort on average. The work of Méndez and Cerdá (2003a) is based on the general precedence concept. Concluding, it should be mentioned that a common weakness of precedence-based formulations is that the number of sequencing variables scales in the number of batches to be scheduled, which may result in significant model sizes for real-world applications.

#### Objective function

Different measures of the quality of the solution can be used for scheduling and planning problems. The selection of the optimization goal directly affects the solution quality as well as the model computational performance. Typical objective functions include the optimization of: makespan, weighted lateness, production costs, inventory costs, total cost, revenue, and profit. It should be noted that some objective functions can be very hard to implement for some event representations, requiring additional variables and complex constraints.

### 2.3.2 Alternative solution approaches

Although this thesis has been focused on solution approaches based on mathematical programming techniques, it is important to note that there are other solution methods for dealing with process scheduling and/or planning problems. These methods can be used either as alternative methods, or as methods that can be combined with mathematical programming models. The major alternative solution methods for solving scheduling problems are: heuristics, metaheuristics, artificial intelligence, constraint programming, and hybrid methods. In addition to these methods, Process Systems Engineering (PSE) research community developed two elaborate approaches to deal with process scheduling problems: the event operation network representation, and the S-graph representation.

#### Heuristics

Most scheduling heuristics, also called dispatching rules, are concerned with formulating rules for determining sequences of activities. They are therefore best suited to processes where the production of a product involves a pre-specified sequence of tasks with fixed batch sizes; in other words variants of multiproduct processes. Often, it is assumed that fixing the front-end product sequence will fix the sequence of activities in the plant. Generally, the processing of a product is broken down into a sequence of jobs that queue for machines, and the rules dictate the priority order of the jobs. Dannenbring (1977), Kuriyan and Reklaitis (1989), and Pinedo (1995) give a good exposition on the kinds of heuristics that may be used for different plant structures. It is worth pointing out that *most of the heuristic methods originated in the discrete manufacturing industries, and might sometimes be expected to perform poorly in process industry contexts*. In process scheduling problems, most of the concerns with these approaches are associated with the divisibility of material in practice, which implies variable batch sizes, and batch splitting and mixing. In fact the last two activities are becoming increasingly popular as a means of effecting late product differentiation. Some of the most broadly used dispatching rules are: FCFS (first come first served), EDD (earliest due date), SPT (shortest processing time), LPT (longest processing time), ERD (earliest release date), and WSPT (weighted shortest processing time). Often, composite dispatching rules, involving a combination of basic rules, can perform significantly better. Besides, a special feature of heuristics is that they can be easily embedded in mathematical models to generate more efficient hybrid approaches for large-scale scheduling problems. An extensive review and a classification of various heuristics can be found in Panwalkar and Iskander (1977) and Blackstone, Phillips, and Hogg (1982).

There is a lack of works that developed heuristics for process scheduling problems, since it is difficult to devise a series of rules to solve such complex processes. Kudva, Elkamel, Pekny, and Reklaitis (1994) addressed the special case of linear multipurpose plants where products flow through the plant in a similar fashion, but potentially using different stages and with no recycling of material. They took account of limited intermediate storage, material receipts at any stage, soft order deadlines, changeover costs and pre-specified equipment maintenance times. A rule-based constructive heuristic was used, which required the maintenance of a status sheet on each unit and material type for each time instance on a discrete-time grid. The algorithm used this status sheet with a sorted list of orders and developed a schedule for each order by backwards recursive propagation. The schedule derived depended strongly on the order sequence. Solutions were found to be within acceptable bounds of optimality when compared with those derived through formal optimization procedures. Graells, Espuña, and Puigjaner (1996) presented a heuristic strategy for the scheduling of multipurpose batch plants with mixed intermediate storage policies. A decomposition procedure was employed where subschedules were generated for the production of intermediate materials. Each subschedule consisted of a mini production path determined through a branch-and-cut enumeration of possible unit-to-task allocations. The mini-paths were then combined to form the overall schedule. The overall schedule was checked for feasibility with respect to material balances and storage capacities. Improvements to the schedules may be effected manually through an electronic Gantt chart.

It should be emphasized that the implementation of heuristics to scheduling problems in the process industries is not straightforward. For this reason, most academic research has been directed towards the development of mathematical programming approaches for process scheduling and planning, since these approaches are capable of representing all the complex interactions in such complex processing networks.

**Metaheuristics**

Metaheuristics are often inspired by moves arising in natural phenomena. Metaheuristics optimize a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Metaheuristics such as genetic algorithms, graphs theory, simulated annealing, tabu search, particle swarm and ant colony optimization methods have been widely used in a variety of scheduling problems. These techniques have become popular for optimizing certain types of scheduling problems, however, they also have significant drawbacks such as that they do not provide any guarantee on the quality of the solution obtained, and it is often impossible to tell how far the current solution is from optimality. Furthermore, these methods do not formulate the problem as a mathematical program, since they involve procedural search techniques that in turn require some type of discretization or graph representation, and the violation of constraints is handled through ad hoc penalty functions. For this reason, the use of metaheuristics might be problematic for problems involving general processes, complex inequality constraints and continuous decision variables. In this case, the set of feasible solutions might lack nice properties and it might even be difficult to find a feasible solution (Burkard, Hujter, Klinz, Rudolf, & Wennink, 1998).

Some excellent contributions in this direction can be found in Kirkpatrick, Gelatt, and Vechi (1983), Glover (1990), Ku and Karimi (1991), Xia and Macchietto (1994), Franca, Gendreau, Laporte, and Muller (1996), Murakami, Uchiyama, Hasebe, and Hashimoto (1997), Raaymakers and Hoogeveen (2000), Pacciarelli (2002), Cavin, Fischer, Glover, and Hungerbhüler (2004), Ruiz and Maroto (2006), Ruiz and Stutzle (2008) and Venditti, Pacciarelli, and Meloni (2010), among many others. Despite the fact that the aforementioned methods may generate fast and effective solutions for complex problems, they are usually tailor-made and cannot systematically estimate the degree of quality of the solution generated. Moreover, the efficiency of these techniques strongly depends on the proper implementation and fine tuning of parameters since they combine the problem representation and the solution strategy into the same optimization framework.

#### Artificial intelligence methods

Artificial intelligence (AI) is the mimicking of human thought and cognitive processes to solve complex problems automatically. AI uses techniques for writing computer code to represent and manipulate knowledge. There are different techniques that mimic the different ways that people think and reason. The main AI techniques are: rule-based methods, agent-based methods, and expert systems. Rule-based methods can be distinguished into case-based reasoning and model-based reasoning techniques. Case-based reasoning is based on previous experiences and patterns of previous experiences while model-based reasoning concentrates on reasoning about a system's behavior from an explicit model of the mechanisms underlying that behavior. 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. Expert systems, also known as knowledge-based approaches, encapsulate the specialist knowledge gained from a human expert and apply that knowledge automatically to make decisions. Some interesting implementations of AI technologies into real-world scheduling problems can be found in Zweben and Fox (1994), Sauer and Bruns (1997), and Henning and Cerdá (2000).

### Constraint programming

Constraint programming is a programming paradigm that was originally developed to solve feasibility problems (Van Hentenryck, 1989, 2002), but it has been extended to solve optimization problems, particularly scheduling problems. Constraint programming is very expressive since continuous, integer, and Boolean variables are permitted; moreover, variables can be indexed by other variables. Furthermore, a number of constructs and global constraints have also been developed to efficiently model and solve specific problems, and constraints need neither be linear nor convex. The solution of constraint programming models is based on performing constraint propagation at each node by reducing the domains of the variables. If an empty domain is found the node is pruned. Branching is performed whenever a domain of an integer, binary or Boolean variable has more than one element, or when the bounds of the domain of a continuous variable do not lie within a tolerance. Whenever a solution is found, or a domain of a variable is reduced, new constraints are added. The search terminates when no further nodes must be examined. The effectiveness of constraint programming depends on the propagation mechanism behind constraints. Thus, even though many constructs and constraints are available, not all of them have efficient propagation mechanisms. For some problems, such as scheduling, propagation mechanisms have been proven to be very effective Constraint programming methods have proved to be quite effective in solving certain types of scheduling problems, especially those that involve sequencing and resource constraints. However, they are not always effective for solving more general optimal scheduling problems that involve assignments (Méndez *et al.*, 2006).

Some of the most common propagation rules for scheduling are the "timetable" constraint (Le Pape, 1998), the "disjunctive constraint" propagation (Baptiste, Le Pape, & Nuijten, 2001), the "edge-finding", and the "not-first, not-last" (Baptiste *et al.*, 2001). Finally, Laborie (2003) summarized the main approaches to propagate resource constraints in constraint-based scheduling and identified some of their limitations for using them in an integrated planning and scheduling framework.

### Hybrid methods

In this paragraph, some important hybrid solution techniques, applied in process scheduling, based on exact solution methods (i.e., mathematical programming) are presented. It should be emphasized that although small- and medium-size models can be usually solved to optimality by using default values in code parameters, large size problems are generally unmanageable by mathematical formulations. Therefore, in order to make the use of exact methods more attractive in real-world applications, increasing effort has been oriented towards the development of systematic techniques that allow maintaining the number of decisions at a reasonable level, even for large-scale problems. A reduced search space usually results in manageable model sizes that often guarantee a more stable and predictable optimization model behavior. Furthermore, once the best possible feasible solution has been generated in a short time, optimization-based methods could be employed to gradually enhance a non-optimal solution with low computational effort. Following this trend, the work of Castro, Harjunkoski, and Grossmann (2009) have been recently emerged as alternative solution strategies to these challenging problems. An apparent drawback of these techniques is that optimality can no longer be assured. Nevertheless, from a practical point of view, guaranteeing global optimality may not be relevant in many industrial scenarios mainly due to the following features: (i) a very short time is just available to generate a solution and send it to the plant floor, (ii) optimality is easily lost because of the highly dynamic nature of industrial environments, (iii) implementing the schedule as such is limited by the real process, and (iv) only a part of the real scheduling goals are generally taken into account in the model since not all scheduling objectives can be quantified. Heuristic model reduction methods, decomposition/aggregation techniques, and improvement optimization-based techniques constitute the principal methods that are embedded in exact mathematical models to face large-scale scheduling problems. A detailed state-of-the-art of these techniques can be found in Méndez *et al.* (2006). A brief description of the aforementioned methods follows.

- ❏ Heuristic model reduction methods usually take into advantage an empirical solution tactic or a particular problem feature and incorporate this knowledge into the mathematical problem representation. As a result, good solutions can be generated in a reasonable time. Simple or combined dispatching rules are usually adopted. The contributions by Lázaro and Puigjaner (1985), Pinto and Grossmann (1995), Cerdá *et al.* (1997), Blömer and Günther (2000a), and Méndez, Henning, and Cerdá (2001) are some representative works of heuristic model reduction methods.

- ❏ Approaches based on spatial or temporal decomposition, such as the works by Graves (1982), Lázaro and Puigjaner (1988) and Gupta and Maranas (1999), usually rely on Lagrangian decomposition. Aggregation techniques aggregate later time periods within the specified time horizon in order to reduce the dimensionality of the problem, or to aggregate the scheduling problem so that it can be considered as part of a planning problem (refer to the works of Birewar and Grossmann (1990) and Bassett, Pekny, and Reklaitis (1997)).

- ❏ Improvement optimization-based techniques can be interpreted as a special case of rescheduling where an initial solution is partially adjusted with the only goal of enhancing a particular scheduling criterion. These techniques use the current schedule as the initial point of a procedure that iteratively enhances the existing solution in a systematic manner. The works by Röslof, Harjunkoski, Björkqvist, Karlsson, and Westerlund (2001) and Méndez and Cerdá (2003a), which followed this direction, have shown promising results with relatively low computational cost.

**Event operation network presentation**

Graells, Cantón, Peschaud, and Puigjaner (1998) provided a realistic representation of complex recipes that uses a flexible modeling environment to schedule batch chemical processes. The process structure (individual tasks, entire subtrains or complex structures of manufacturing activities) and related materials (e.g., raw materials, intermediate or final products) are characterized by a processing network that describes the material balance. In the most general case, the activity that is carried out in each process constituted a general activity network. Manufacturing activities are considered at three levels of abstraction: the *process level*, the *stage level*, and the *operation level*. This hierarchical approach considers material states (subject to material balance and precedence constraints) and temporal states (subject to time constraints) at different levels.

At the process level, the process and materials network provides a general description of production structures (such as synthesis and separation processes) and of the materials involved, including intermediates and recycled materials. An explicit material balance is specified for each of the processes in terms of a stoichiometric-like equation that relates raw materials, intermediates and final products (see Figure 2.3). Each process may represent any kind of activity that is required to transform the input materials into the derived outputs.

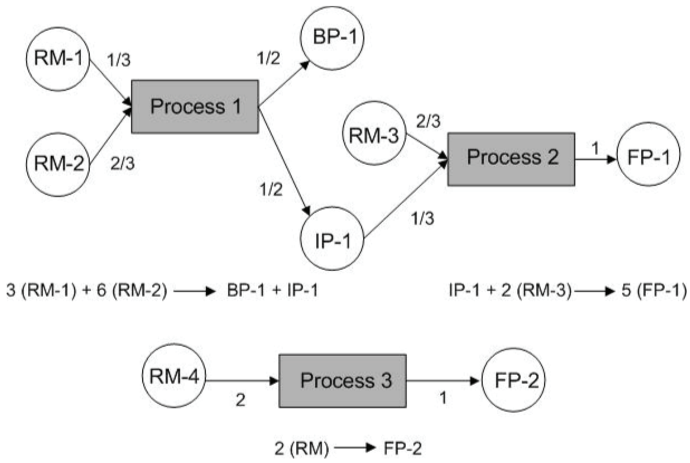



**Figure 2.3:** Process and materials network describing the processing of two products. RM, IP and FP are raw materials, intermediate products, and final products, respectively.

The stage level lies between the process level and the detailed description of the activities involved at the operation level. At this level, the block of operations that are executed in the same equipment is described. Hence, at the stage level each process is split into a set of the blocks (see Figure 2.4). Each stage involves the following constraints:

- The sequence of operations that are involved requires a set of implicit constraints (links).
- Unit assignment is defined at this level. Thus, for all the operations in the same stage, the same unit assignment must be made.
- A common size factor is attributed to each stage. This size factor summarizes the contribution of all the operations involved.

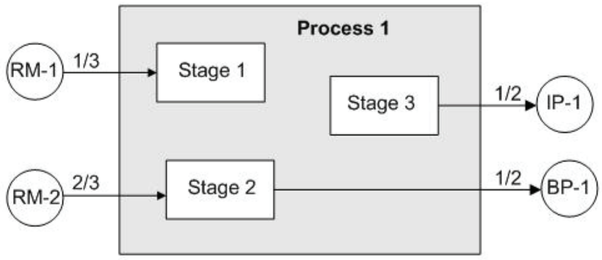



**Figure 2.4:** Stage level. Each stage involves different unit assignment.

The operation level contains a detailed description of all the activities considered in the network. Implicit time constraints (links) must also be met at this level. The detailed representation of the structure of activities that define the different processes is called the Event Operation Network (EON). The general utility requirements (e.g., renewable, non-renewable, storage) are also represented at this level. The EON representation model describes the appropriate timing of process operations. A continuous-time representation of process activities is made using three basic elements: events, operations and links (Puigjaner, 1999). Events designate the time instants in which some change occurs. They are represented by nodes in the EON graph, and may be linked to operations or other events. Each event is associated with a time value and a lower bound.

Operations comprise the time intervals between events (see Figure 2.5). Each operation m is represented by a box linked with solid arrows to its associated nodes: initial $NI_m$ and final $NF_m$ nodes. Operations establish the equality links between nodes, in terms of the characteristic properties of each operation: the operation time (TOP), and the waiting time (TW). The operation time will depend on the amount of materials to be processed, the unit model and the product changeover. The waiting time is the lag time between operations, which is bounded. Finally, precedence constraints are used to establish links between events.


**Figure 2.5:** The time description for operations. TOP = operation time, TW = waiting time, $NI_m$ = initial node, and $NF_m$ = final node of operation.

A dashed arrow represents each link K from its node of origin $NO_k$ to its destiny node $ND_k$ and an associated offset time $\Delta T_K$ (see Figure 2.6).

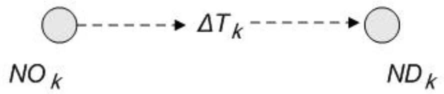



**Figure 2.6:** Event to event link and associated offset time.

Despite its simplicity, the EON representation is very general and flexible and it allows complex recipes to be handled (see Figure 2.7). The corresponding TOP, according to the batch size and material flow rate, also represents transfer operations between production stages. The necessary time overlap of semicontinuous operations with batch units is also considered in this representation by means of appropriate links.

Plant operation can be simulated by means of the EON representation using the following information, which is contained in the process recipe and production structure characteristics:

- A sequence of production runs or jobs associated with a process or recipe.
- A set of assignments that is associated with each job and consistent with the process.
- A batch size that is associated with each job and is consistent with the process.
- A set of shifting times for all of the operations involved.

These decisions may be generated automatically using diverse procedures to determine an initial feasible solution. Hence, simulation may be executed by solving the corresponding EON to determine the timing of the operations and other resource requirements.

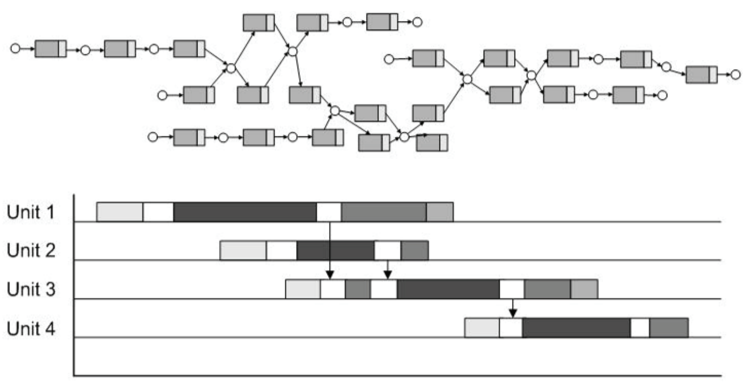



**Figure 2.7:** EON representation of a branched complex recipe. The Gantt chart is given below.

### S-graph representation

Sanmartí, Friedler, and Puigjaner (1998) and Sanmartí, Holczinger, Friedler, and Puigjaner (2002) introduced a graph representation for solving process scheduling problems. This scheduling graph, called S-graph, takes into consideration the specific characteristics of chemical processes in the scheduling. It allows scheduling problems to be formulated using similar graph representations to those used to solve the job-shop problem. However, it takes into account the higher complexity of chemical multipurpose batch scheduling.

The master recipes are represented as a directed conjunctive graph, in which the nodes represent the production tasks and the arcs are the precedence relationships among tasks. The number above the arrows represents the task processing times. An additional node is associated with each product: the last task or tasks of the production are connected to the corresponding node by an arc. Thus, for each product, the number of nodes in the graph is the number of tasks in the recipe plus one. For instance, in Figure 2.8, the sequence of tasks A1 → A2 → A3 is displayed using a graph of this type. The graph consists of four nodes instead of three. The fourth node is required to represent the end of the last task A3, since the completion of task A1 coincides with the start of task A2, and the completion of A2 with the start of A3.

Complex recipes can be represented in this way. Figure 2.9a illustrates the conventional representation of the recipes of three products, in which two intermediates are produced, mixed and further processed in the production of the first product A. Figure 2.9b shows the graph representation of the recipes given in Figure 2.9a, where Ei denotes the set of equipment units that can perform the task represented by node i (e.g., E1={1,2}). Furthermore, an additional node (the product node) is introduced for each product. In this representation, the value assigned to an arc expresses a lower bound for the difference between the start times of the two related tasks. The processing time of a task may vary for different equipment units. In this case, the weight of the arc is the minimum of the processing times of the plausible equipment units. The S-graph uses the recipe representation described above to find a single solution for a scheduling problem. There is one schedule graph for each feasible schedule of the problem. The S-graph $G'(N,_1,A_2)$ is called a schedule graph of the recipe graph $G(N,_1,\varnothing)$, if all the tasks represented in the recipe graph have been scheduled by taking into

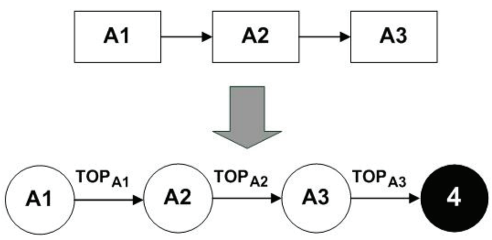
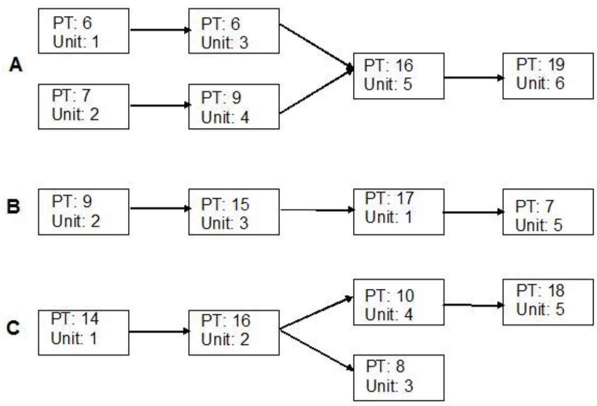


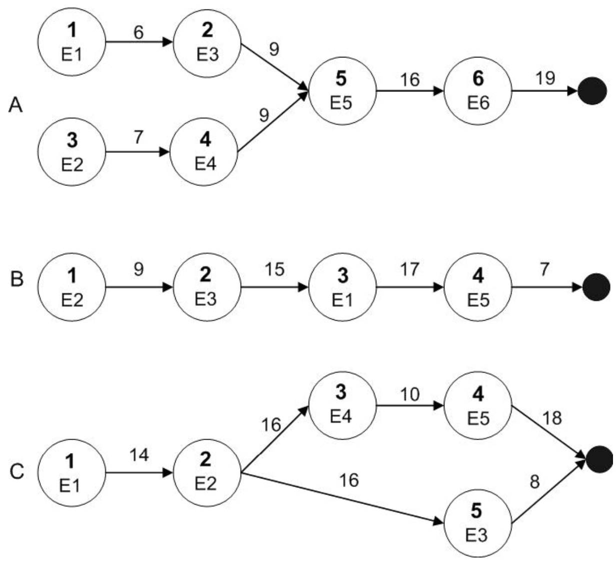

**Figure 2.8:** S-graph representation of a sequence of tasks.

account equipment-task assignments. The schedule graph of the optimal schedule can be effectively generated by an appropriate search strategy, which enables early detection of infeasible schedules (Sanmartí *et al.*, 1998, 2002).

Initially, S-graph was only applied to problems considering minimization the makespan. The problem was solved using a branch-and-bound and an efficient graph algorithm to evaluate the makespan. Afterwards, S-graph capabilities were further extended so that there is now an effective search algorithm for determining schedules that optimize throughput, revenue, or profit over a predefined time horizon in multipurpose batch plants (Majozi & Friedler, 2006), and taking into account aspects of uncertainty (Laínez, Hegyháti, Friedler, & Puigjaner, 2010a).




(a) Conventional representation of master recipes of three products.




(b) RTN process representation.

**Figure 2.9:** S-graph representation of recipes shown.

## 2.4 Uncertainty

Process industries are dynamic in nature and, therefore, different kinds of unexpected events occur quite frequently. Unexpected disturbances affect the nominal operating conditions and the, now out-of-day, schedule of the production facility. A lack of appropriate procedures for tackling disruptions caused by uncertain events yields into significant performance deterioration. Despite the fact that the study of uncertainty is out of the scope of the current thesis (however, in Chapter 6 some unexpected scenarios are considered, and efficiently tackled on-line), in this section a brief description of the major methods for managing uncertainty is given. More details regarding optimization under uncertainty in process industries can be found in the neatly written state-of-the-art review by Sahinidis (2004).

### 2.4.1 Uncertainty sources

A taxonomy of the main sources of uncertainty in each SC decision-making level is given in Figure 2.10. It is important to notice that most sources of uncertainty do not fit totally within one of these categories, but the boundaries are somehow diffuse. Besides, because of the interactions between the different levels of decision making, uncertainties from one level may affect decisions made in other levels. For instance, variable demands do not only alter tactical planning decisions, but also the process scheduling in the operational level (Bonfill, 2006).

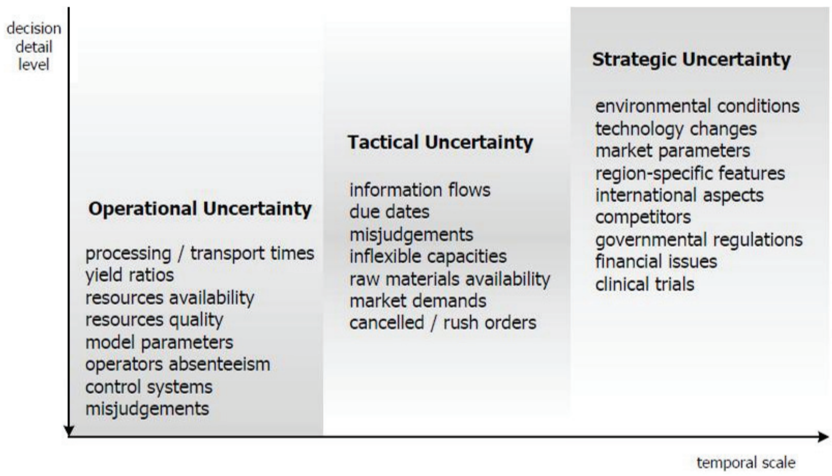



**Figure 2.10:** A taxonomy of uncertainty sources (Bonfill, 2006).

### 2.4.2 Managing uncertainty

The methods for managing uncertainty can be mainly distinguished into *proactive (off-line)* and *reactive (on-line)* approaches. Figure 2.11 depicts the different techniques for each approach of dealing with uncertainty.

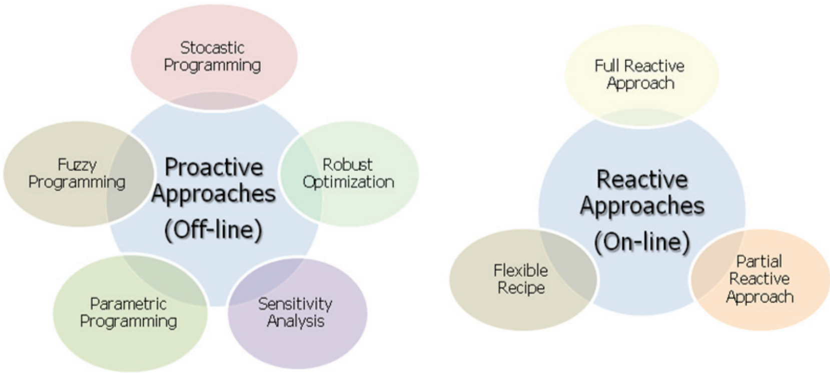



**Figure 2.11:** Methods for managing uncertainty.

### Proactive Approaches

Proactive methods, and mainly proactive scheduling approaches, can be viewed as sub-optimization strategies that provide visibility for future actions to achieve a greater system's performance. If the uncertainty occurs as predicted, the loss of opportunities and reschedule requirements are reduced, whereas the full force of the perturbation affects the expected results if the uncertainty is neglected (Aytug, Lawley, McKay, Mohan, & Uzsoy, 2005).

**Stochastic-based approaches.** Stochastic-based approaches is the most commonly used approach in the literature. The original deterministic mathematical model is transformed into a stochastic model treating the uncertainties as random variables. Stochastic approaches are mainly divided into the following categories:

- two-stage, where variables are separated to first stage (or here-and-now) decisions and to second stage (or wait-and-see) decisions, or multistage stochastic programming:
  (i) scenario-based,
  (ii) probabilistic distribution.
- chance constraint programming based approach.

**Fuzzy programming methods.** The principal difference between the stochastic and fuzzy optimization approaches is in the way uncertainty that is modeled. Here, fuzzy programming considers random parameters as fuzzy numbers and constraints are treated as fuzzy sets. Some constraint violation is allowed and the degree of satisfaction of a constraint is defined as the membership function of the constraint. Objective functions in fuzzy mathematical programming are treated as constraints with the lower and upper bounds of these constraints defining the decision-makers expectations. Fuzzy logic and probability are different ways of expressing uncertainty. While both fuzzy logic and probability theory can be used to represent subjective belief, fuzzy set theory uses the concept of fuzzy set membership (i.e., how much a variable is in a set), probability theory uses the concept of subjective probability (i.e., how probable do I think that a variable is in a set).

**Robust optimization methods.** These methods focus on building the preventive scheduling and/or planning to minimize the effects of disruptions on the performance measure. They also try to ensure that the predictive and realized schedule and/or planning do not differ drastically, while maintaining a high level of schedule and/or performance. In mathematics, robust optimization is an approach in optimization to deal with uncertainty. It is similar to the recourse model of stochastic programming, in that some of the parameters are random variables, except that feasibility for all possible realizations (called scenarios) is replaced by a penalty function in the objective. As such, the approach integrates goal programming with a scenario-based description of problem data.

**Sensitivity analysis.** It is used to ascertain how a given model output depends upon the input parameters. This is an important method for checking the quality of a given model, as well as a powerful tool for checking the robustness and reliability of any solution (Li & Ierapetritou, 2008). Sensitivity analysis determines, on individual parameters of the model, the range in which the solution remains optimal provided all other parameters are fixed at their given values. Although valuable knowledge can be obtained, sensitivity analysis is usually considered as a post-optimization approach that does not provide any mechanism to control and improve the robustness of a given proposed solution (Mulvey, Vanderbei, & Zenios, 1995).

**Parametric programming methods.** Parametric optimization serves as an analytic tool in process synthesis under uncertainty mapping the uncertainties in the definition of the synthesis problem to optimal design alternatives. From this point of view, it is the exact mathematical solution of the uncertainty problem. Parametric-programming can be used into a model predictive control framework for on-line optimization via off-line (parametric) techniques. Parametric programming techniques have been developed and proposed as a means of reducing computational effort associated to optimization problems regarding uncertainty. To address such problems by using the aforementioned techniques, it is obtained a complete map of all the optimal solutions. As a result, as the operating conditions fluctuate, one does not have to re-optimize for the new set of conditions since the optimal solution as a function of parameters (or the new set of conditions) is already available (Pistikopoulos, Dua, Bozinis, Bemporad, & Morari, 2002).

#### Reactive Approaches

Reactive methods deal with uncertainty after the occurrence of the unexpected events. Since they tackle unforeseen events on-line, they should be fast enough, computationally speaking, in order to be applicable into the industrial environment. Despite the fact that the study of uncertainty is out of the scope of this thesis, a contribution on reactive scheduling, where the importance of considering rescheduling costs is highlighted, has been also realized during the Ph.D. studies (see Appendix A).

**Full reactive approaches.** They make decisions dynamically when some event occurs by permitting full alterations of the current schedule/planning. It is the most computationally expensive approach. Additionally, its proposed solution is usually very difficult to be applied to the real industrial scenario because of the large number of modifications that these approaches propose.

**Partial reactive approaches.** They are based on the modification of the predictive schedule and/or planning to update the decisions according to the actual situation by allowing a limited number of modifications. Computational effort is moderated. However, the problem may be over-restricted, thus disregarding potential optimal solutions.

**Flexible recipe.** Process operating conditions are modified in order to adjust the processing times so as to return to the original requirements. The major drawback of this procedure is that there may be little flexibility for the modification of these conditions to guarantee the quality of the products.

## 2.5 Literature Review

In this section, a literature review in the research field of scheduling and/or planning problems by mathematical programming approaches in the process industry is presented. A separate literature review for the food process industry is also included because it corresponds to an emerging, promising and challenging field of research that has received little attention despite its significant role in contemporary markets.

### 2.5.1 Process industry

The literature review of the most important mathematical approaches addressing scheduling and/or planning problems in the process industry have been classified into: discrete-time and continuous-time models, and multisite and resource-constrained production.

#### Discrete-time models

Kondili *et al.* (1988, 1993) introduced the STN process representation and they presented a discrete-time MIP model. This model was based on the definition of binary variables that indicate whether tasks start in specific units at the beginning of each time interval, together with associated continuous batch sizes. Other key variables were the amount of material in each state held in dedicated storage over each time interval, and the amount of each utility required for processing tasks over each time interval. The key constraints were related to resources (i.e., processing units and utilities), material balances and capacity. The use of a discrete time grid captured all the plant resource utilizations in a straightforward manner; discontinuities in these were forced to occur at the predefined interval boundaries. However, this approach was hindered in its ability to handle large problems by the weakness of the allocation constraints and the general limitations of discrete-time approaches such as the need for relatively large numbers of grid points to represent activities with significantly different durations.

The emergence of the STN representation formed the basis of many other works aiming to take into advantage of the representational capabilities of the formulation while enhancing its computational performance. More specifically, Sahinidis and Grossmann (1991) disaggregated the allocation constraints and exploited the embedded lot-sizing nature of the model where relatively small demands were distributed throughout the scheduling horizon. The computational performance of the model was improved, despite the larger nature of the disaggregated model. In addition, the formulation of Sahinidis and Grossmann (1991) had a much smaller integrality gap than the original STN model. Afterwards, Shah *et al.* (1993a) modified the allocation constraints even further to generate the smallest possible integrality gap for this type of formulations. They also devised a tailored branch-and-bound solution procedure which utilizes a much smaller LP relaxation and solution processing to improve integrality at each node. The same authors Shah, Pantelides, and Sargent (1993b) considered the extension to cyclic scheduling, where the same schedule was repeated at a frequency to be determined as part of the optimization. This was then extended by Papageorgiou and Pantelides (1996a,b) to cover the case of multiple campaigns, each with a cyclic schedule to be determined.

Elkamel (1993) also proposed a number of measures to improve the performance of the STN-based discrete-time scheduling model. A heuristic decomposition method was proposed, which solves separate scheduling problems for parts of the overall scheduling problem. The decomposition may be based on the resources (longitudinal decomposition) or on time (axial decomposition). In the former, the recipes and suitable equipment for each task were examined for the possible formation of unique task-unit subgroups which can be scheduled separately. Axial decomposition was based on grouping products by due dates and decomposing the horizon into a series of smaller time periods, each concerned with the satisfaction of demands falling due within it. In addition, a perturbation heuristic was described, which actually was a form of local search around the relaxation.

Yee and Shah (1997, 1998) also considered various manipulations to improve the performance of general discrete-time scheduling models. An important feature of their work was the variable elimination, since they recognized that in such models, only about 5-15% of the variables reflecting task-to-unit allocations were active at the integer solution, and it would be beneficial to identify as far as possible inactive variables prior to solution. For this reason, they proposed an LP-based heuristic, a flexibility and sequence reduction technique, and a formal branch-and-price method. They also recognized that some problem instances resulted in poor relaxations and propose valid inequalities and a disaggregation procedure similar to that of Sahinidis and Grossmann (1991) for particular data instances.

Gooding (1994) considers a special case of the problem with fixed demands and dedicated storage. The scheduling model is described in a digraph form where nodes corresponded to possible task-unit-time allocations, and arcs corresponded to the possible sequences of tasks. The explicit description of the sequence in this form addressed one of the major weaknesses of the discrete-time model of Kondili *et al.* (1993). The formulation of Gooding, Pekny, and McCroskey (1994) performed relatively well in problems with a strong sequencing component, but suffered from model complexity in that all possible sequences must be accounted for directly.

Pantelides, Realff, and Shah (1995) reported a STN-based approach to the scheduling of pipeless plants, where material is conveyed between processing stations in movable vessels, and thus requiring the simultaneous scheduling of the movement and processing operations.

Blömer and Günther (1998) proposed a series of LP-based heuristics that can reduce solution times considerably, without compromising the quality of the solution obtained. Rodrigues, Latre, and Rodrigues (2000) addressed the short-term planning/scheduling problems when the product demands are driven by customer orders. They proposed a multi-level decomposition procedure, containing at least two levels. At the planning level, demands were adjusted, a raw material delivery plan were defined and a capacity analysis was performed. Therefore, time windows for each operation were defined. At the scheduling level, an STN-based MIP model was developed. Grunow, Günther, and Lehmann (2002) show how the STN tasks could be aggregated into higher level processes for the purposes of longer-term campaign planning.

Pantelides (1994) presented a critique of the STN and associated scheduling formulations, and he introduced the RTN process representation in order to overcome the drawbacks of the STN. He developed a discrete-time model based on the RTN which, due to the uniform treatment of resources, only required the description of three types of constraint (i.e., task allocation, batch size, and resource availability), and does not distinguish between identical units; a fact that resulted in more compact and less degenerate optimization models. He also demonstrated that the integrality gap could not be worse than the most efficient form of STN formulation.

At this point, it should be emphasized that while the discrete-time STN and RTN models are quite general and effective in monitoring the level of limited resources at the fixed times, their major weakness is the handling of long time horizons and relatively small processing and changeover times. Regarding the objective function, these models can easily handle profit maximization (cost minimization) for a fixed time horizon. Other objectives such as makespan minimization are more complex to implement since the time horizon and, in consequence, the number of time intervals required, are unknown a priori (Maravelias & Grossmann, 2003). For these reasons, the more recent research have been focused mainly on developing scheduling models based on a continuous representation of time, where fewer grid points are required as they will be placed at the appropriate resource utilization discontinuities during problem solution.

### Continuous-time models

A number of mathematical programming approaches have been developed for the scheduling of multiproduct batch plants. All are based (either explicitly or implicitly) on a continuous representation of time.

Pekny, Miller, and McCrae (1988) addressed the scheduling problem in a multiproduct plant with no storage (i.e., ZW policy), and they show that the problem in question has the same structure as the asymmetric traveling salesman problem. They applied an exact parallel computation technique employing a tailor-made branch-and-bound procedure which used an assignment problem to provide problem relaxations. Then, Pekny, Miller, and McCrae (1990) extended this work in order to account for product transition costs. Linear programming relaxations were used, and large-scale problems were solved to optimality with relatively modest computational effort. Finally, Gooding *et al.* (1994) further extended this work to cover the case of multiple units at each processing stage.

Birewar and Grossmann (1989) developed a MIP formulation for a similar type of plant, and they demonstrated that a straightforward LP model could be used to minimize the makespan: (i) through careful modeling of slack times, and (ii) by exploiting the fact that relatively large numbers of batches of relatively few products will be produced. The result was a family of schedules, from which an individual schedule may be extracted. Birewar and Grossmann (1990) further extended this work for simultaneous long-term planning and scheduling, where the planning function took account of scheduling limitations.

Pinto and Grossmann (1995) proposed a MIP model for the minimization of earliness in a multiproduct plant with multiple units at each processing stage. Two types of individual time grids were used: one for units and one for orders. For each unit, a number of intervals of unknown duration were defined, which represented the possible sequence of tasks (one per interval). For each order, the time interval corresponded to a processing stage. These interval durations were also unknown, since processing times were unit-dependent. In order to ensure that, when a stage of an order was assigned to a unit, the starting times on both grids were equal, a set of mixed integer constraints were used. Precedence relations were employed for the material balances. Afterwards, Pinto and Grossmann (1997) extended this model to take account of interactions between processing stages and shared resources (e.g., steam). They retained the individual grids, and account for the resource discontinuities through complex mixed integer constraints, which weakened the model and resulted in large computational times. For this reason, they developed a hybrid logic-based/MIP algorithm, where the disjunctions related to the relative timing of orders, in order to reduce the computational cost.

Moon, Park, and Lee (1996) developed a MIP formulation for ZW multiproduct plants. The objective was to assign tasks to sequence positions so as to minimize the makespan, with non-zero transfer and setup times being included. Afterwards, Kim, Jung, and Lee (1996) further extended this work for more general intermediate storage policies. They developed several formulations based on completion time relations.

Cerdá *et al.* (1997) and Karimi and McDonald (1997) addressed the case of single-stage processes with multiple units per processing stage. Cerdá *et al.* (1997) focused on changeovers and order fulfillment, while Karimi and McDonald (1997) focused on semicontinuous processes and total cost (i.e., transition, shortage and inventory) with the complication of minimum run lengths. A characteristic of both approaches is that discrete demands must be captured on the continuous time grid. Méndez *et al.* (2000) developed a continuous-time precedence-based MIP model for a process with a single production stage with parallel units followed by a storage stage with multiple units, with restricted connectivity between the stages. Afterwards, Méndez *et al.* (2001) further extended this work to the multistage case with discrete shared resources. In common with other models, there were no explicit time slots in the model, and the key variables were allocations of activities to units and the relative orderings of activities.

Hui and Gupta (2001) presented a MIP formulation for the short-term scheduling of multiproduct batch plants with parallel nonidentical production units. They used bi-index, instead of typical tri-index, discrete decision variables. As a result the number of discrete variables were decreased, however the number of constraints were increased. Lee, Heo, Lee, and Lee (2002) developed a MIP model for scheduling problems in single-stage and continuous multiproduct processes on parallel lines with intermediate due dates and restrictions on minimum run lengths. Chen, Liu, Feng, and Shao (2002) proposed a continuous-time MIP model for the short-term scheduling of multiproduct single-stage batch plants with parallel lines involving constraints concerning release times and due dates of orders, as well as the sequence-dependent setup times and forbidden subsequences of production orders and the ready times of units. They also introduced some heuristic rules, and they demonstrated that the rational employment of these heuristic rules could cut down the size of the model and had no effect on the optimality of the scheduling problem.

Chen, Papageorgiou, and Pinto (2008) studied the medium-term planning problem of a single-stage single-unit continuous multiproduct polymer plant. They proposed a slot-based MIP model based on a hybrid discrete/continuous time representation, where the production planning horizon was divided into several discrete weeks, and each week was formulated with a continuous time representation. Liu, Pinto, and Papageorgiou (2008) further improved the work of Chen *et al.* (2008) by presenting a MIP formulation based on the classic traveling salesman problem formulation. Their proposed model, without time slots, was computationally more effective. Recently, Liu, Pinto, and Papageorgiou (2009) extended that work for medium-term planning of a single-stage multiproduct continuous plant to the case with parallel units. Erdirik-Dogan and Grossmann (2006) proposed a multiperiod slot-based MIP model for the simultaneous planning and scheduling of single-stage single-unit multiproduct continuous plants. A bilevel decomposition algorithm in which the original problem is decomposed into an upper level planning and a lower level scheduling problem was also developed in order to deal with complex problems. Erdirik-Dogan and Grossmann (2008) later extended their work to address parallel units. Sung and Maravelias (2008) presented a MIP formulation for the production planning of single-stage multiproduct processes. The problem was formulated as a multi-item capacitated lot-sizing problem in which: (i) multiple items can be produced in each planning period, (ii) sequence-independent setups can carryover from previous periods, (iii) setups can crossover planning period boundaries, and (iv) setups can be longer than one period.

In the literature some RTN-based continuous formulations have been reported. Specifically, Castro *et al.* (2001) developed a MIP formulation for the optimal scheduling of batch processes. Their formulation used a continuous time representation and is based on the RTN representation. Castro and Grossmann (2006) presented a multiple time grid RTN-based continuous-time MIP model for the short-term scheduling of single stage multiproduct batch plants, which was based on the general formulation proposed by Castro, Barbosa-Póvoa, Matos, and Novais (2004). The most important difference was that a different time grid was used for each machine of the process instead of a single time grid for all events taking place. Their model can handle both release and due dates while the objective can be either the minimization of total cost or total earliness. Castro, Grossmann, and Novais (2006) developed two multiple-time grid continuous-time MIP models for the scheduling of multiproduct multistage plants featuring equipment units with sequence-dependent changeovers. The performance of both formulations was compared to other MIP models and constraint programming models. The results show: (i) that multiple-time grid models were better suited for single-stage problems or, when minimizing total earliness, (ii) that the constraint programming model was the best approach for makespan minimization, and (iii) that the continuous-time model with global precedence variables was the best overall performer.

A number of continuous-time formulations based both on the STN or the RTN representation and the definition of global time points have been developed. Mockus and Reklaitis (1999a,b) presented a general STN-based mathematical framework for describing scheduling problems arising in multipurpose batch and continuous chemical plants. The problem was formulated as a large nonlinear MIP model. A technique that exploits the characteristics of the problem in order to reduce the amount of required computation was also reported. Schilling and Pantelides (1996) presented a general MIP formulation for optimal scheduling of processes. Their continuous-time formulation was based on the RTN representation. In common with other continuous-time scheduling formulations, this exhibited a large integrality gap that rendered its solution using standard branch-and-bound algorithms highly problematic. Therefore, a branch-and-bound algorithm that branched on both discrete and continuous variables was proposed to address this complication. Zhang and Sargent (1996) extended the RTN concept in order to provide a unified mathematical formulation of the problem of determining the optimal operating conditions of a mixed production facility, comprising multipurpose plant for both batch and continuous operations. Their formulation used a variable event-time sequence common to all system events. This resulted into a large nonlinear MIP problem. However, for batch processes with fixed recipes the problem was linear and can be solved by existing techniques.

Giannelos and Georgiadis (2002) developed a STN-based MIP formulation for scheduling multipurpose batch processes. A number of event points was prepostulated, which was the same for all tasks in the process. Event times were defined by the ends of task execution, and they are generally different for different tasks of the process, giving rise to a nonuniform time grid. The necessary time monotonicity for single tasks was ensured by means of simple duration constraints. Suitable sequencing constraints, applicable to batch tasks involving the same state, were also introduced, so that state balances were properly posed in the context of the nonuniform time grid. The expression of duration and sequencing constraints was greatly simplified by hiding all unit information within the task data. Their model were less computationally expensive from the previous reported models mainly due to smaller model size.

Maravelias and Grossmann (2003) developed a STN-based continuous-time MIP model for the scheduling of multipurpose batch plants. Their model addressed the general problem of batch scheduling, accounting for resource constraints, variable batch sizes and processing times, various storage policies (i.e., UIS, FIS, NIS, and ZW), batch mixing/splitting, and sequence-dependent changeover times. The key features of their model were: (i) a continuous-time representation was used, common for all units, (ii) assignment constraints were expressed using binary variables that were defined only for tasks, not for units, (iii) start times of tasks were eliminated, so that time-matching constraints were used only for the completion times of tasks, and (iv) a new class of valid inequalities that improved the LP relaxation was added to the MIP formulation. Maravelias (2005) proposed a mixed-time representation for STN-based scheduling models, where the time grid was fixed, but processing times were allowed to be variable and span an unknown number of time periods. The proposed representation was able to handle batch and continuous processes, and optimized holding, backlog, and utility costs. It also dealt with release and due dates at no additional computational cost, and coped with variable processing times. It is also worth mentioning a recent study by Ferrer-Nadal, Capón-García, Méndez, and Puigjaner (2008) that incorporated the representation of transfer times, which had been ignored in STN- and RTN-based formulations thus not guaranteeing the generation of feasible solutions. By considering transfer times the generation of infeasible solutions, previously reported in the literature, was avoided.

Prasad and Maravelias (2008) developed a MIP formulation that involved three levels of discrete decisions, i.e., selection of batches, assignment of batches to units, and sequencing of batches in each unit. Continuous decision variables included sizing and timing of batches. They considered various objective functions: minimization of makespan, earliness, lateness and production cost, as well as maximization of profit, an objective not addressed by previous multistage scheduling methods. In addition, in order to enhance the solution of their model, they proposed symmetry breaking constraints, developed a preprocessing algorithm for the generation of constraints that reduced the number of feasible solutions, and fixed sequencing variables based upon time window information. Sundaramoorthy and Maravelias (2008b) extended the work of Prasad and Maravelias (2008) to account for variable processing times. To account for batching decisions, they used additional batch-selection and batch-size variables and introduce demand-satisfaction and unit-capacity constraints. Assignment constraints were active only for the subset of batches that were selected, and sequencing was carried out between batches that were assigned on the same processing unit. They also proposed an alternate formulation to handle sequence-dependent changeover costs. Finally, they presented methods that allowed to fix a subset of sequencing variables as well as they developed a set of tightening inequalities based on time windows, in order to enhance the computational performance of their model.

The works of Prasad and Maravelias (2008) and Sundaramoorthy and Maravelias (2008b) assumed unlimited storage. Méndez and Cerdá (2003b) and Wu and He (2004) considered storage constraints for scheduling in the more general multipurpose batch processes, however did not consider batching decisions. Sundaramoorthy and Maravelias (2008a) proposed a precedence-based MIP formulation for the simultaneous batching and scheduling in multiproduct multistage processes with storage constraints, and they showed how their model can be modified to address all storage policies. They also discussed a class of tightening constraints, and they presented an extension for the modeling of changeover costs.

Gimenez *et al.* (2009a) presented a network-based MIP framework for the short-term scheduling of multipurpose batch processes. Their approach was based on five key concept: (i) a new continuous-time representation is developed that does not require tasks to start (end) exactly at a time point; thus reducing the number of time points needed to represent a solution, (ii) processing units were modeled as being in different activity states to allow storage of input/output materials, (iii) time variables for "idle" and "storage" periods of a unit were introduced to enable the matching between tasks and time points without big-M constraints, (iv) material transfer variables were added to explicitly account for unit connectivity, and (iv) inventory variables for storage in processing units were incorporated to model non-simultaneous and partial material transfers. Afterwards, Gimenez, Henning, and Maravelias (2009b) extended this work to address aspects such as: (i) preventive maintenance activities on unary resources (e.g., processing and storage units) that were planned ahead of time, (ii) resource-constrained changeover activities on processing and shared storage units, (iii) non-instantaneous resource-constrained material transfer activities, (iv) intermediate deliveries of raw materials and shipments of finished products at predefined times, and (v) scenarios where part of the schedule was fixed because it had been programmed in the previous scheduling horizon.

Marchetti and Cerdá (2009a) presented a MIP continuous-time approach for the scheduling of single-stage multiproduct batch plants with parallel units and sequence-dependent changeovers. Their formulation was based on a unit-specific precedence-based representation. By explicitly including the equipment index in the domain of the sequencing variables, additional nontrivial tightening constraints producing better lower bounds on the optimal values of alternative objective functions (i.e., makespan or overall earliness) or key variables (i.e., task starting and completion times) were developed. Marchetti, Méndez, and Cerdá (2010) proposed two precedence-based MIP continuous-time formulations (i.e., a rigorous and a cluster-based MIP) for the simultaneous lot-sizing and scheduling of single-stage multiproduct batch facilities. Both approaches can handle multiple customer orders per product at different due dates as well as variable processing times. The two proposed models differ in the way that sequencing decisions were taken. The rigorous approach dealt with the sequencing of individual batches processed in the same unit, while the approximate cluster-based method arranged groups of batches, each one featuring the same product, due date, and assigned unit. Since cluster members were often consecutively processed, each cluster can be treated and assigned to units as a single entity for sequencing purpose. It should be noted that the cluster-based model may result in suboptimal solutions.

### Multisite production

Much of the research effort to date has focused on the planning and scheduling of production for individual plants situated at a single geographical site and involving a set of batch, semicontinuous or even continuous unit operations. As is well known, this is in itself a complex problem, optimal or even feasible solutions to which are often notoriously difficult to obtain. However, it must also be recognized that production scheduling is only one aspect of the wider problem of process scheduling. For instance, the scheduling of plant maintenance operations, the co-ordinated planning of the production at a number of distinct geographical locations, and the management of distribution and SCs, all lead to important scheduling problems that interact strongly with production scheduling at individual production plants. It might be expected that large benefits would ensue from co-ordinated planning across sites, in terms of costs and market effectiveness. Most business processes dictate that a degree of autonomy is required at each manufacturing and distribution site, but pressures to co-ordinate responses to global demand while minimizing cost imply that simultaneous planning of production and distribution across plants and warehouses should be undertaken. This would result in the most efficient utilization of all resources. A target-setting approach, where central plans set achievable production targets without imposing operational details is compatible with operational details being determined at each site.

Wilkinson, Cortier, Shah, and Pantelides (1996) showed how the RTN representation of Pantelides (1994) can be used to represent a variety of distribution options. The multisite planning problem can therefore be directly posed using the RTN representation and the discrete-time model of Pantelides (1994). Wilkinson *et al.* (1996) recognized that a potential problem with this approach is the very large model sizes that will ensue. A secondary issue is that the development of a central plan to a very fine level of detail is probably unnecessary. This led to the development of an aggregation procedure by Wilkinson, Shah, and Pantelides (1995). The aim was to capture production and distribution capacities accurately without considering detailed scheduling. The same authors applied this technique to a continent-wide industrial case study. This involved optimally planning the production and distribution of a system with three factories and fourteen market warehouses and over a hundred products.

Karimi and McDonald (1997) described a similar problem for multiple facilities which effectively produced products on single-stage continuous lines for a number of geographically distributed customers. Their basic model was of multiperiod LP form, and took account of available processing time on all lines, transportation costs and shortage costs.

Timpe and Kallrath (2000) and Kallrath (2002b) described a general MIP model based on a time-indexed formulation covering the relevant features required for the complete Supply Chain Management (SCM) of a multisite production network. The model combined aspects related to production, distribution and marketing and involves production sites and sales points. Besides standard features of lot-sizing problems (raw materials, production, inventories, demands) further aspects, e.g., different time scales attached to production and distribution, the use of periods with different lengths, the modeling of batch and campaign production need to be considered. While the actual application was taken from the chemical industry, the model provided a starting point for many applications in the chemical process industry, food or consumer goods industry.

Verderame and Floudas (2009) presented a multisite operational planning model that provided daily production and shipment profiles which represented a tight upper bound on the true capacity of the SC under investigation. The proposed scheme effectively modeled the production capacity of each production facility within the SC. The proposed planning model was to an industrial case and favorably compared to an existing planning model.

The SCM problem is particularly challenging because it not only encompasses the decisions of the planning/scheduling levels described about but also distribution logistics, market and price uncertainties as well as financial aspects. A substantial amount of work is appearing in this respect: detailed scheduling considerations in SC design (Puigjaner, Laínez, & Álvarez, 2009; Li & Ierapetritou, 2010), embedded financial issues and environmental aspects (Laínez, Guillén-Gonsálbez, Badell, Espuña, & Puigjaner, 2007; Puigjaner & Guillém-Gosálbez, 2008; Bojarski, Laínez, Espuña, & Puigjaner, 2009) and the linking of marketing and SC models (Laínez, Reklaitis, & Puigjaner, 2010b).

**Resource-constrained production**

Manufacturing resources are generally grouped into two types: *renewable* and *non-renewable* resources. A renewable resource is one that is recovered when the task to which it was allocated has concluded. Renewable resources can be discrete (e.g., tools, manpower) or continuous (e.g., heating, refrigeration, electricity). In contrast, non-renewable resources, like intermediates or raw materials, are consumed by tasks and every resource capacity allocated to them is no longer recovered at their completion. The literature in resource-constrained scheduling and planning problems in process industries is rather poor.

Pinto and Grossmann (1997) presented a MIP sequential approach based on a slot-based continuous-time representation that extended a former mathematical formulation for unconstrained multistage batch plants (Pinto & Grossmann, 1995). As the number of binary variables and big-M constraints substantially increased, the general MIP resource-constrained model became almost computationally unsolvable. Consequently, the authors developed a problem solution methodology that combined a branch-and-bound MIP algorithm with disjunctive programming. Slot-based representations were also presented by Lamba and Karimi (2002) and Lim and Karimi (2003) to tackle semicontinuous scheduling problems of single-stage parallel production lines with resource constraints. Lamba and Karimi (2002) used identical slots across all processors while Lim and Karimi (2003) employed asynchronous slots. Since the underlying idea of an asynchronous slot is similar to the unit-specific time event, checkpoints for resource utilization were placed at the start of each slot and additional variables and constraints should be included to establish the slot relative positions.

Méndez and Cerdá (2002a) developed a precedence-based MIP continuous-time representation that independently handles unit allocation and task sequencing decisions through different sets of binary variables. Sequencing variables allowed to order the tasks allocated either to the same equipment unit or to another discrete resource. In this way, an important saving in binary variables was achieved. Afterwards, Méndez and Cerdá (2002b) reported a more general MIP formulation to deal with both continuous and discrete finite renewable resources. Each continuous resource was divided into a discrete number of subsources or pieces that were assigned to tasks through new allocation variables. Then, sequencing variables were still used to ordering tasks allocated to the same discrete or continuous resource item. The maximum number of pieces into which a continuous renewable can be divided was a model parameter, while each piece capacity was a non-negative variable selected by the model. However, the proposed resource representation may sometimes exclude the problem optimum from the feasible space and, consequently, optimality was not guaranteed. It should be pointed out that the models of Méndez and Cerdá (2002a) and Méndez and Cerdá (2002b) can potentially lead to overestimation of utility levels.

Sundaramoorthy, Maravelias, and Prasad (2009) proposed a discrete-time MIP model for the simultaneous batching and scheduling in multiproduct multistage processes under utility (e.g., cooling water, steam, and electricity) constraints. Since different tasks often share the limited utilities at the same time, they used a common time-grid approach. Further, the proposed method handles the batching decisions (i.e., the number and sizes of batches) seamlessly without the usage of explicit batch-selection variables. Finally, they introduce a new class of inventory variables and constraints, in order to preserve batch identity in storage vessels.

Marchetti and Cerdá (2009b) presented a general precedence-based MIP framework to the short-term scheduling of multistage batch plants that accounted for sequence-dependent changeover times, intermediate due dates and limited availability of discrete and continuous renewable resources. Their formulation relied on a continuous-time formulation based on the general precedence notion that uses different sets of binary variables to handle allocation and sequencing decisions. To avoid resource overloading, additional constraints in terms of sequencing variables and a new set of 0–1 overlapping variables were presented. Finally, preordering rules can be easily implemented in the MIP model.

### 2.5.2 Food process industry

A plethora of contributions addressing production scheduling and planning problems can be found in the OR and PSE communities literature. However, the use of optimization-based techniques for scheduling food process industries is still in its infancy. This can be mainly attributed to the complex production recipes, the large number of products to be produced under tight operating and quality constraints and the existence of mixed batch and semicontinuous production modes.

The literature in the field of single-site production scheduling and planning of food processing industries is rather poor. Entrup, Günther, Van Beek, Grunow, and Seiler (2005) presented three different MIP model formulations, which employed a combination of a discrete- and continuous-time representation, for scheduling and planning problems in the packing stage of stirred yogurt production. They accounted for shelf-life issues and fermentation capacity limitations. However, product changeover times and production costs were ignored. The latter makes the proposed models more appropriate to cope with planning rather than scheduling problems, where products changeovers details are crucial. The data set used to demonstrate the practical applicability of their models consisted of 30 products based on 11 recipes that could be processed on four packing lines. They reported near-optimal solutions within reasonable computational time for the case study solved.

Marinelli, Nenni, and Sforza (2007) addressed the planning problem of 17 products in 5 parallel packing machines, which share resources, in a packing line producing yogurt. Their optimization goal was the minimization of inventory, production and machines setup cost. Sequence-dependent costs and times were not considered. They presented a discrete mathematical planning model which failed to obtain the optimal solution of the real application in an acceptable computation time. For this reason, they proposed a two-stage heuristic for obtaining near-optimal solutions for the problem under study.

Doganis and Sarimveis (2008) studied the scheduling problem at a yogurt packing line of a dairy company in Greece. Their objective was to optimally schedule two (or three) parallel conjoined (coupled) packing machines over a 5-day production horizon in order to meet the weekly demand for 25 different products. Each one of the identical machines could produce any of the 25 products. Products changeover times and costs were considered and total demand satisfaction was imposed. Simultaneous packing of multiple products was not allowed since the parallel machines shared the same feeding line. The latter restriction as well as the limited number of products considered greatly simplified the problem under question. The apparent reduction of changeover times was transformed into additional machine idle time. Finally, potential limitations of the fermentation stage were completely ignored.

The food production and distribution networks show a number of distinct features, such a sensitive quality of the products, production processes with both continuous and batch characteristics, the generation of by-products, and severe food safety and hygienic requirements (Grunow & van der Vorst, 2010). Akkerman, Farahani, and Grunow (2010) recently presented an excellent review of quantitative operations management approaches to food distribution management, and relate this to challenges faced by the industry. A number of research challenges in strategic network design, tactical network planning, and operational transportation planning were highlighted with emphasis on food quality, food safety, and sustainability.

Brown, Keegan, Vigus, and Wood (2001) presented a large-scale linear program that modeled the production and distribution network of the Kellogg Company, a large producer of breakfast cereals and other foods. A salient aspect of the proposed model is that it was functioning on different time scales, using weeks or, months as time units.

Higgins, Beashel, and Harrison (2006) presented a model to schedule the shipment of sugar from production sites to ports from which ships were used to export sugar internationally. The main objective of this approach was to support rescheduling activities during the season to account for changing production rates. Eksioglu and Jin (2006) developed a general MIP approach for network planning of perishable products. Perishability was modeled by a maximum number of periods over which the product could be stored. A constraint was added in the formulation to make sure that product inventory in distribution centers was not used to cover the demand after having been stored beyond the specified maximum number of periods.

Ahumada and Villalobos (2009) critically reviewed the main contributions in the field of production and distribution planning for agri-foods based on agricultural crops. They focused on models that have been successfully implemented in problems of industrial interest. The models were classified according to relevant features, such as the optimization approaches used, the type of crops modeled and the scope of the plans, among many others.

Bilgen and Günther (2010) presented, a so-called block planning approach which established cyclical production patterns based on the definition of setup families. Two transportation modes were considered for the delivery of final goods from the plants to distribution centers, full truckload and less than truckload. The proposed MIP model minimized total production and transportation costs. A number of example problems illustrated the applicability of the proposed planning approach. Rong, Akkerman, and Grunow (2010) described a MIP model which cleverly integrated food quality degradation in decision-making on production and distribution in food SCs.

Melo, Nickel, and Saldanha-da-Gama (2009) presented an excellent and comprehensive review of the most recent literature contributions on facility location analysis within the context of SCM. They discussed the general relation between facility location models and strategic SC planning. A number of separate sections were dedicated to the relation between facility location and SCM as well as efficient solution methods and applications studies.

Manzini and Bindi (2009) presented an integrated framework for the design and optimization of a multi-echelon, multi-level production/distribution system. The framework relies on MIP techniques combined with cluster analysis, heuristic algorithms, and optimal transportation rules.

Recently Moula, Peidro, Díaz-Madroñero, and Vicens (2010) presented a review of mathematical programming models for SC production and transport planning. The review critically identified current and future research in this field and proposed a taxonomy framework based on a number of elements such as SC structure, decision level, modeling approach, purpose, novelty and applications.

### 2.5.3 Industrial applications

The vast literature in the scheduling and planning area highlights the successful application of different optimization approaches to an extensive variety of challenging problems. As the economic advantages of implementing scheduling and planning tools became evident, BASF, DOW and Du Pont began more intensive use of in-house developed tools for their planning and scheduling (Hess, 2002). Nowadays, more difficult and larger problems than those studied years ago can be solved, sometimes even to optimality, in a reasonable time by using more efficient integrated mathematical frameworks. This important achievement comes mainly from the remarkable advances in modeling techniques, algorithmic solutions and computational technologies that have been made in the last few years. Although a promising near future in the area can be predicted from this optimistic current situation, it is also well-known that the actual gap between practice and theory is still evident. New academic developments are mostly tested on complex but relatively small problems whereas current real-world applications consist of hundreds of batches, dozens of pieces of equipment and long scheduling periods, usually ranging from one to several weeks (Méndez *et al.*, 2006).

Honkomp, Lombardo, Rosen, and Pekny (2000) gave a list of reasons why the practical implementation of scheduling tools based on optimization is fraught with difficulty. These include:

- The large amount of user defined input for testing purposes.
- The difficulty in capturing all the different types of operational constraints within a general framework, and the associated difficulty in defining an appropriate objective function.
- The large amounts of data required.
- Computational difficulties associated with the large problem sizes found in practice.

- Optimality gaps arising out of many shared resources.
- Intermediate storage and material stability constraints.
- Non-productive activities (e.g., setup times, and cleaning).
- Effective treatment of uncertainties in demands and equipment effectiveness.

However, there have been some success stories in the application of state-of-the-art scheduling and planning methods in the process industry. An early success story was already reported in a complex problem in the textile sector by Espuña. and Puigjaner (1989). The detailed mathematical model built included specific ad-hoc rules that permitted a fast simulation to obtain a feasible solution to start optimization. This strategy was applied to deal with the problem encountered in a large textile factory. Different fabrics and designs were produced using diverse machines, which could be shared by some products. The manufacturing of a total of 35,000 articles was considered. This situation resulted to a very complex problem of task assignment and optimization of production lines for the best use of the existing equipment to meet specified orders. Typical figures indicated that even qualified and experienced personnel found that drawing up the production plans for this kind of industrial facility was a burdensome job if it had to be done manually. The high-performance simulation module enabled the production manager to easily modify long-term and short-term production plans, and to evaluate objectively the consequences of such modifications (or decide to implement the changes suggested by the short-term production planning module). Hence, the production manager could cope with engineering decisions that require immediate attention. This feature was very useful when market pressures lead to the need for unexpected changes in a long-term production planning policy.

Schnelle (2000) applied MIP-based scheduling and design techniques for an agrochemical facility. The results indicated that sharing of equipment items between different products was a good idea, and the process reduced the number of alternatives to consider to a manageable number. Berning, Brandenburg, Gürsoy, Mehta, and Tölle (2002) described a large-scale planning/scheduling application which uses genetic algorithms for detailed scheduling at each site and a collaborative planning tool to coordinate plans across sites. The plants all operate batchwise, and may supply each other with intermediates, thus creating interdependencies in the plan. The scale of the problem was large, involving 600 different process recipes, and 1000 resources. Kallrath (2002a) presented the successful application of MIP methods for planning and scheduling in BASF. He described a software tool for simultaneous strategic and operational planning in a multisite production network. The total net profit of a global network was optimized, where key decisions included: (i) operating modes of equipment in each time period, (ii) production and supply of products, (iii) minor changes to the infrastructure (e.g., addition and removal of equipment from sites), and (iv) raw material purchases and contracts. A multiperiod model was formulated where equipment may undergo one mode change per period. The standard material balance equations were adjusted to account for the fact that transportation times are much shorter than the period durations. Counter-intuitive but credible plans were obtained which resulted in cost savings of several millions of dollars. Keskinocak, Wu, Goodwin, Murthy, Akkiraju, Kumaran, and Derebail (2002) described the application of a combined agent- and optimization-based framework for the scheduling of paper products manufacturing. The framework solved the problems of order allocation, run formation, trimming and trim loss minimization and load planning. The deployment of the system was claimed to save millions of dollars per year. Their approach used constructor and improver agents to generate candidate solutions which are evaluated against multiple criteria. Wang, Löhl, Stobbe, and Engell (2000) and Harjunkoski and Grossmann (2001) addressed complex real-world scheduling problems in the polymer and the steel-making casting industry, respectively.

## 2.6 Trends and Challenges

A considerable amount of fruitful research work has already been carried out on scheduling and/or planning in process industries. Mathematical optimization can provide a quantitative basis for decisions and allow to cope most successfully with complex problems, and it has proven itself as a useful technique to reduce costs and to support other objectives. Despite that, this technology has not yet found its way into many commercial software packages. For scheduling problems, there is not yet a commonly accepted state-of-the-art technology although some promising approaches have been developed, especially for job shop problems. Nevertheless,the majority of software packages is still based on pure heuristics (Kallrath, 2002b).

A number of issues can provide interesting future research challenges in the research field of scheduling and/or planning in process industry. Based on the literature review, it is foreseen the need to devote further research efforts in order to meet the following trend and challenges:

- ❏ The recent research is all about solution efficiency and techniques to render ever larger problems tractable. There remains work to be done on both model enhancements and improvements in solution algorithms if industrially-relevant problems are to be tackled routinely, and software based on these are to be used on a regular basis by practitioners in the field.
- ❏ Much of the more recent research has focussed on continuous-time formulations, but little technology has been developed based on these. The main challenge here is in continual improvement in problem formulation and pre-processing to improve relaxation characteristics, and tailored solution procedures for problems with relatively large integrality gaps.
- ❏ The mathematical models developed should be implemented into industrial or industrially-based studies, in order to demonstrate to industrial practitioners the potential benefits for adopting mathematical programming methods in managing scheduling and/or planning problem in industrial environments.
- ❏ The multisite problem has received relatively little attention, and is likely to be a candidate for significant research in the near future. A major challenge

is to develop planning approaches that are consistent with detailed production scheduling at each site and distribution scheduling across sites. An obvious obstacle is the problem size, therefore appropriate modeling frameworks should be devised in order to tackle rigorously and efficiently these highly complicated optimization problems.

- There are some process industries that have received little attention; regarding scheduling and/or planning research. On of the most emerging and challenging industry of this type, is the food process industry. Scheduling and planning approaches in the food process industry is rather poor, despite the fact that there are many optimization challenges. There is a need for optimization frameworks able to cope with scheduling/planning problem under the complex semicontinuous process mode of these industries.
- Another focus of modeling, which is possible now due to increased computer power available, is the opportunity to solve design and operational planning problems, or strategic and operational planning problems simultaneously in one model.
- Another challenge relates to the seamless integration of the activities at different decision levels; this is of a much broader and more interdisciplinary nature. The financial aspects will require more rigorous treatment, as scheduling and planning become integrated.
- Another major issue is the handling of uncertainty (e.g, in terms of processing times, and availability of equipment, modification of orders). A major challenge here is how to best formulate a stochastic optimization model that is meaningful and whose results are easy to interpret and implement.

# Chapter 3

# Methods and Tools

## 3.1 Introduction

In this chapter, the background of the methods and tools used in the development and implementation of the different mathematical models and solution approaches devised in this thesis is described. Mathematical programming constitutes the main optimization approach for dealing with the several industrial case studies considered. First, some general principles of mathematical programming are discussed. Afterwards, some theoretical concepts and solution techniques for linear and mixed integer programming problems are briefly presented. Finally, a short description of the commercial tools utilized to solve the problems under study is given.

## 3.2 Mathematical Programming

Mathematical programming is the use of mathematical models, particularly optimizing models, to assist in taking decisions. The term "Programming" antedates computers and means "preparing a schedule of activities". It is still used, for instance, in oil refineries, where the refinery programmers prepare detailed schedules of how the various process units will be operated and the products blended. Mathematical programming is, therefore, the use of mathematics to assist in these activities. Mathematical Programming is one of a number of Operational Research (OR) techniques. Its particular characteristic is that the best solution to a model is found automatically by optimization software. A mathematical programming model answers the question "What's best?" rather than "What happened?" (statistics), "What if?" (simulation), "What will happen?" (forecasting), or "What would an expert do and why?" (expert systems).

Being so ambitious does have its disadvantages. Mathematical programming is more restrictive in what it can represent than other techniques. **Nor should it be imagined that it really does find the best solution to the real-world problem. It finds the best solution to the problem as modeled.** If the model has been built well, this solution should translate back into the real world as a good solution to the real-world problem. If it does not, analysis of why it is no good leads to greater understanding of the real-world problem.

Whatever the real-world problem is, it is usually possible to formulate the optimization problem in a generic form. All optimization problems with explicit objectives can in general be expressed as nonlinearly constrained optimization problems in the following generic form:

$$\left.\begin{array}{lll} & \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{maximize/minimize}} \quad f(\boldsymbol{x}) & \\ \text{subject to} & & \\ & \phi_m(\boldsymbol{x}) \quad = \mathbf{0} & (m = 1, ..., M) \\ & \psi_k(\boldsymbol{x}) \quad \leq \mathbf{0} & (k = 1, ..., K) \\ \text{where } \boldsymbol{x} = (x_1, x_2, ..., x_n)^T \in \mathbb{R}^n & & \end{array}\right\} \tag{3.1}$$

where $f(\boldsymbol{x})$, $\phi_m(\boldsymbol{x})$, and $\psi_k(\boldsymbol{x})$ are scalar functions of the real column vector $\boldsymbol{x}$. The function $f(\boldsymbol{x})$ is called *objective function*, and is a quantitative measure of the performance of the system in question. The components $x_i$ of vector $\boldsymbol{x}$ are called *decision variables*, or simply variables, and they can be either continuous, discrete or a mixed of these two. The variables are the unknowns whose values are to be determined such that the objective function is optimized. Additionally, $\phi_m(\boldsymbol{x})$ are constraints in terms of M equalities, and $\psi_k(\boldsymbol{x})$ are constraints written as K inequalities. Therefore, there are M+K constraints in total. Constraints represent any restrictions that the decision variables must satisfy.

The procedure of identifying the aforementioned components is known as modeling. Depending on the properties of the functions $f$, $\phi$, $\psi$, and the vector $\boldsymbol{x}$, the mathematical program (3.1) is called:

- *Linear:* If $\boldsymbol{x}$ is continuous and the functions $f$, $\phi$, and $\psi$ are all linear.
- *Nonlinear:* If $\boldsymbol{x}$ is continuous and at least one of the functions $f$, $\phi$, and $\psi$ is nonlinear.
- *Mixed integer linear:* If $\boldsymbol{x}$ requires at least some of the variables $x_i$ to take integer (or binary) values only; and the functions $f$, $\phi$, and $\psi$ are linear.
- *Mixed integer nonlinear:* If $\boldsymbol{x}$ requires at least some of the variables $x_i$ to take integer (or binary) values only; and at least one of the functions $f$, $\phi$, and $\psi$ is nonlinear.

### 3.2.1 Optimality criteria

A point $x$ which satisfies all the constraints is called a feasible point and therefore is a *feasible solution* to the problem. The set of all feasible points is called the *feasible region*. A point $x_*$ is called a strong local maximum of the optimization problem if $f(\boldsymbol{x})$ is defined in a $\delta$-neighborhood $N(x_*, \delta)$ and satisfies $f(x_*) > f(\boldsymbol{u})$ for $\forall u \in N(x_*, \delta)$ where $\delta > 0$ and $u \neq x_*$. If $x_*$ is not a strong local maximum, the inclusion of equality in the condition $f(x_*) \geq f(\boldsymbol{u})$ for $\forall u \in N(x_*, \delta)$ defines the point $x_*$ as a weak local maximum (see Figure 3.1). The local minima can be defined in the similar manner when $>$ and $\geq$ are replaced by $<$ and $\leq$, respectively. Figure 3.1 illustrates several local maxima and minima. Point A is a strong local maximum, and point B is a weak local maximum since there exist many different values of $x$ which will lead to the same value of $f(x_*)$. Finally, point C is a global maximum.

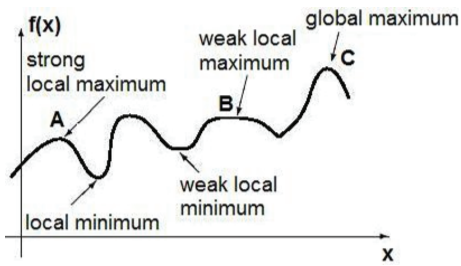
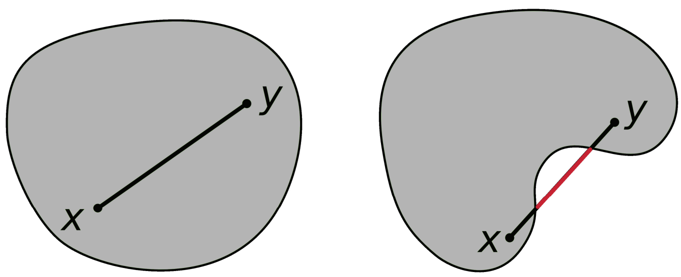



**Figure 3.1:** Illustrate example for strong and weak maxima and minima.

### 3.2.2 Convexity

Let $\mathscr{C}$ be a set in a real or complex vector space. Set $\mathscr{C}$ is *convex* if, for every pair of points $x$ and $y$ belonging within the set, every point on the straight line segment that connects them is also within the set $\mathscr{C}$, as illustrated in Figure 3.2. This definition is can be mathematically expressed as:




(a) A convex set. (b) A noncovex set.

**Figure 3.2:** Graphical representation for convexity.

$$\mathscr{C} \text{ is convex} \quad \Longleftrightarrow \quad \forall (x,y) \in \mathscr{C} \wedge \theta \in [1,0] : \big((1-\theta)x + \theta y\big) \in \mathscr{C}$$

A function $f(\boldsymbol{x})$ is convex if its epigraph (i.e., the set of points lying on or above its graph) is a convex set, as shown in Figure 3.3. Convexity plays a significant role in mathematical programming due to the following theorem:

**Theorem 3.1** *If a mathematical program is convex then any local (i.e., relative) minimum is a global minimum.*

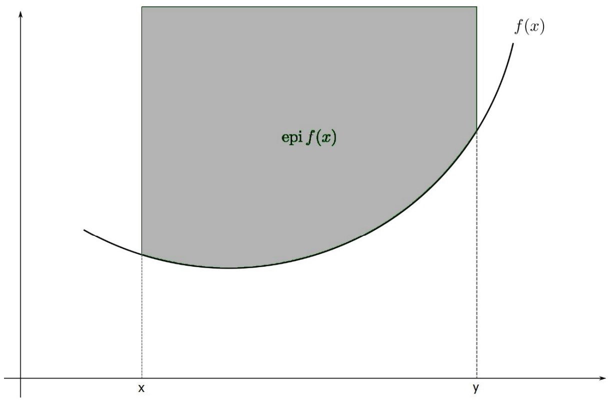



**Figure 3.3:** Graphical representation for a convex function.

The research subfield that deals with nonconvex programs is referred to as global optimization, which aims finding the globally best solution of models in the potential presence of multiple local optima.

### 3.2.3 Duality

Duality is one of the most fundamental concepts in mathematical programming and establishes a connection between two "symmetric" programs, namely, the primal and dual problem. Duality is a powerful and widely employed tool in applied mathematics for a number of reasons. First, the dual program is always convex even if the primal is not. Second, the number of variables in the dual is equal to the number of constraints in the primal which is often less than the number of variables in the primal program. Third, the maximum value achieved by the dual problem is often equal to the minimum of the primal.

The dual function is introduced as:

$$\xi(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \underset{\boldsymbol{x}}{\text{Infimum}} \quad \{f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \phi(\boldsymbol{x}) + \boldsymbol{\mu}^T \psi(\boldsymbol{x})\} \tag{3.2}$$

Then, the dual problem of the primal problem (3.1) is defined as follows:

$$\begin{aligned} &\underset{\boldsymbol{\lambda},\boldsymbol{\mu}}{\text{maximize}} \quad \xi(\boldsymbol{\lambda},\boldsymbol{\mu}) \\ &\text{subject to} \quad \boldsymbol{\mu} \geq \mathbf{0} \end{aligned} \tag{3.3}$$

Hence, using the Lagrangian function, the dual problem can also be rewritten as:

$$\underset{\boldsymbol{\lambda},\boldsymbol{\mu};\boldsymbol{\mu}\geq\mathbf{0}}{\text{maximize}} \left\{ \underset{\boldsymbol{x}}{\text{Infimum}} \quad \mathscr{L}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu}) \right\} \tag{3.4}$$

where the vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are called Lagrangian multipliers, and the Lagrangian function is defined by:

$$\mathscr{L}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{\mu}^T \boldsymbol{g}(\boldsymbol{x})$$

The theorem 3.2 establishes an important relationship between the dual and primal problems.

**Theorem 3.2** ***Weak duality*** *For any feasible solution $\boldsymbol{x}$ of the primal problem (3.1) and for any feasible solution $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, of the dual problem (3.3), the following holds*

$$f(\boldsymbol{x}) \geq \xi(\boldsymbol{\lambda},\boldsymbol{\mu}) \tag{3.5}$$

In addition, the theorem 3.3 is of relevant importance in mathematical programming. It shows that for convex programs the primal problem solution can be obtained by solving the dual problem.

**Theorem 3.3** *If the primal problem is convex, then $f(\boldsymbol{x}^*) = \xi(\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)$. Otherwise, one or both of the two sets of feasible solutions is empty.*

Note that $\boldsymbol{x}^*$ represents the optimal solution of the primal problem, and $\boldsymbol{\lambda}^*$, $\boldsymbol{\mu}^*$ are the optimal solutions of the dual problem.

In nonconvex programs, there is a difference between the optimal objective function values of the dual and primal problems ($\xi(\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*) - f(\boldsymbol{x}^*)$) which is called *duality gap*. In convex programs duality gap is zero. According to Conejo, Nogales, and Prieto (2002), for nonconvex programs of engineering applications, the duality gap is usually relatively small.

## 3.3 Linear Programming

Linear Programming (LP) is a technique for the optimization of a linear objective function, subject to linear equality and/or linear inequality constraints. Given a polytope and a real-valued affine function defined on this polytope, a LP method will find a point on the polytope where this function has the optimal value if such point exists, by searching through the polytope vertices.

LP problems can be expressed in standard form, as follows:

$$\text{maximize}\, \{c^T \boldsymbol{x} : A\boldsymbol{x} \leq b, \boldsymbol{x} \geq 0\} \tag{3.6}$$

where $\boldsymbol{x}$ represents the vector of decision variables (to be determined), $c$ and $b$ are vectors of (known) coefficients and $A$ is a (known) matrix of coefficients. The expression to be optimized is called the objective function ($c^T\boldsymbol{x}$ in this case). The equations $A\boldsymbol{x} \leq b$ are the constraints which specify a convex polytope over which the objective function is to be optimized.

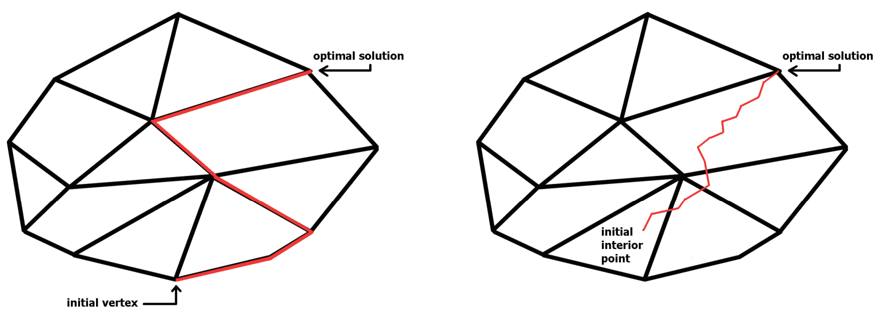
The linear programming optimization and relevant solution algorithms, such as Simplex and interior-point methods, are principally based on the following fundamental theorem:

**Theorem 3.4** *If an LP has an optimal solution, there is a vertex (i.e., extreme point) of the feasible polytope that is optimal.*

### 3.3.1 The Simplex method

The Simplex algorithm, which was first developed by G. B. Dantzig in 1947, solves linear programs by moving along the boundaries from one vertex (extreme point) to the next. The algorithm starts with an initial vertex *basic feasible* solution and tests its optimality. The algorithm terminates, if some optimality condition is verified, otherwise, the algorithm identifies an adjacent vertex, with a better objective value. The optimality of this new solution is tested again, and the entire scheme is repeated, until an optimal vertex is finally found. Since every time a new vertex is identified the objective value is improved (except from a certain pathological case), and the set of vertices is finite, it follows that the algorithm will terminate in a finite number of iterations. Given the above description of the algorithm, it is inferred that the Simplex essentially starts from some initial extreme point, and follows a path along the edges of the feasible region towards an optimal extreme point, such that all the intermediate extreme points visited are not worsening the objective function (see Figure 3.4a).

It is worth mentioning that in 1953 Dantzig and Orchard-Hays proposed the Revised Simplex method, which actually is not a different method but is a different (more efficient) way to carry out each computational step of the Simplex method.




(a) Simplex methods. (b) Interior point methods.

**Figure 3.4:** Graphical interpretation of linear programming methods.

### 3.3.2 Interior-point methods

During the period 1979 – 1996, there has been intensive interest in the development of interior-point methods. A theoretical breakthrough came in 1979, when L. G. Khachian discovered an ellipsoid algorithm whose running time in its worst case was significantly lower than that of the Simplex algorithm. Other theoretical results quickly followed, notably that of N. Karmarkar who discovered an interior-point algorithm whose running time performance in its worst case was significantly lower than that of Kachiyan's. This in turn was followed by more theoretical results by others improving on the worst-case performance. In a nutshell, an interior-point algorithm is one that improves a feasible interior solution point of the linear program by steps through the interior, rather than one that improves by steps around the boundary of the feasible region, as the Simplex algorithm does (see Figure 3.4).

Assuming an initial feasible interior point is available and that all moves satisfy the whole set of constraints, the key ideas behind interior-point methods are as follows:

- Try to move through the interior in directions that show promise of moving quickly to the optimal solution.
- Recognize that if we move in a direction that sets the new point too "close" to the boundary, this will be an obstacle that will impede our moving quickly to an optimal solution. One way around this is to transform the feasible region so that the current feasible interior point is at the center of the transformed feasible region. Once a movement has been made, the new interior point is transformed back to the original space, and the whole process is repeated with the new point as the center.
- The simple stopping rule typically followed is to stop with an approximate optimal solution when the difference between iterates "deemed" sufficiently small in the original space.

The interested reader is referred to Dantzig and Thapa (1997, 2003) for a detailed description of the basic principles, the theory and extensions of linear programming algorithms.

## 3.4 Mixed Integer Programming

Mathematical programs, which some of its (decision) variables are integer and/or binary, are called *mixed integer programs*. Integer variables appear when modeling indivisible entities, while a very common use of binary (0–1) variables is to represent binary choice. Consider an event that may or may not occur, and suppose that it is part of the problem to decide between these possibilities. In order to model such a dichotomy, a binary variable, which typically equals 1 if the event occurs otherwise is set to zero, can be used. The event itself can be almost anything. Depending on the specific problem the event may represent yes/no decisions, logical conditions, fixed costs or piecewise linear functions.

(Linear) Mixed Integer Programming (MIP) problems can be expressed in standard form, as follows:

$$\begin{aligned} \text{maximize}\, c^T\boldsymbol{x} + h\boldsymbol{y} \\ A\boldsymbol{x} + G\boldsymbol{y} \le b \\ \boldsymbol{x} \ge 0 \\ \boldsymbol{y} \ge 0 \text{ and integer or binary} \end{aligned} \tag{3.7}$$

where $\boldsymbol{x}$ represents the vector of non-negative variables, $\boldsymbol{y}$ represents the vector of integer and/or binary variables, $c$ and $b$ are vectors of coefficients, and $A$ and $G$ are matrices of coefficients. In this case the objective function is $c^T\boldsymbol{x} + h\boldsymbol{y}$.

Principally, there are three methodologies for solving this type of programs: the branch-and-bound, the cutting-plane, and the branch-and-cut methods. A brief description of those methods follows.

### 3.4.1 Branch-and-bound methods

The branch-and-bound method is the basic workhorse technique for solving integer and discrete programming problems. The idea of branch-and-bound was introduced by Land and Doig (1960), and actually is based on the observation that the enumeration of integer solutions has a tree structure. More specifically, the solution of a problem with a branch-and-bound algorithm is described as a search through a tree, wherein the root node corresponds to the relaxed original problem, and each other node corresponds to a subproblem of the original problem. In MIP problems, the branch-and-bound algorithm only branches on the integer variables, therefore the discussion can be restricted to a purely integer problem without loss of generality.

For instance, consider the complete enumeration of a MIP model having one integer variable $1 \le x_1 \le 3$, and two binary variables $x_2$ and $x_3$. Figure 3.5 illustrates the complete enumeration of all solutions for these variables, even those which might be infeasible due to other constraints on the model. The structure in Figure 3.5 looks like a tree lying on its side with the root node on the left, and the leaf nodes on the right. The leaf nodes represent the actual enumerated complete solutions; so there are 12 of them. For example, the node at the upper right represents the solution in which $x_1 = 1$, $x_2 = 0$, and $x_3 = 0$. The other nodes can be thought of as representing sets of possible solutions. For example, the root node represents all solutions that can be generated by growing the tree. Another intermediate bud node, e.g., the first node directly to the right of the root node, represents another subset of all of the possible solutions, in this case, all of the solutions in which $x_1 = 2$ and the other two variables can take any of their possible values. For any two directly connected nodes in the tree, the parent node is the one closer to the root, and the child node is the one closer to the leaves.

The main idea in branch-and-bound method is to avoid growing the whole tree as much as possible, because the entire tree is just too big in any real problem. Instead branch-and-bound grows the tree in stages, and grows only the most

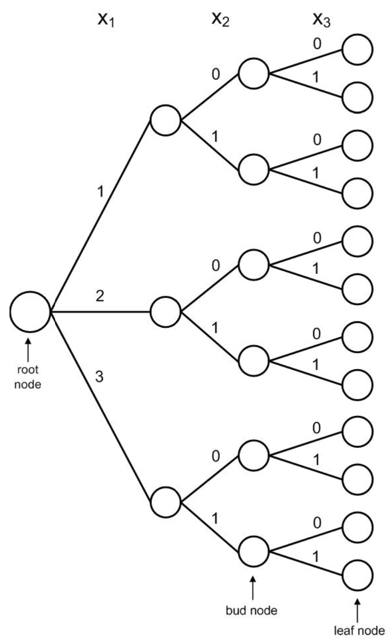



**Figure 3.5:** An illustrative example of branch-and-bound enumeration tree.

promising nodes (i.e., partial or complete solutions) at any stage. It determines which node is the most promising by estimating a bound on the best value of the objective function that can be obtained by growing that node to later stages. The name of the method comes from the *branching* that happens when a bud node (i.e., partial solution, either feasible or infeasible) is selected for further growth and the next generation of children of that node is created. The *bounding* comes in when the bound on the best value attained by growing a node is estimated. Hopefully, in the end branch-and-bound will have grown only a very small fraction of the full enumeration tree.

Additionally, the branch-and-bound algorithm attempts to reduce the amount of enumeration by *pruning* branches from the enumeration tree of all possible solution by applying two simple maxims:

- A branch can be eliminated (pruned), if it can be shown to contain no integer feasible solutions with a better value than the incumbent solution (i.e., the best complete feasible solution found so far).

- An upper bound for the integer solutions, on any branch, is always the relaxed LP solution, which ignores the integer requirements.

The order in which the branch-and-bound algorithm proceeds after the first branch is governed by the *branching rules* adopted. Branching rules can range from *breadth-first*, which expand all possible branches from a tree node before going deeper in the tree, to *depth-first*, that expand the deepest node first.

Finally, the branch-and-bound algorithm terminates when the incumbent solution's objective function value is better than or equal to the bounding function value associated with all of the bud nodes. This means that none of the bud nodes could possibly develop into a better solution than the complete feasible solution already have in hand, so there is no point in expanding the tree any further. Of course, according to the pruning policies, all bud nodes in this condition will already have been pruned, so this terminating rule amounts to saying that branch-and-bound stops when there are no more bud nodes left to consider for further growth. This also proves that the incumbent solution is optimum.

### 3.4.2 Cutting-plane methods

Cutting planes were introduced by R. E. Gomory in the 1950s as a method for solving integer programming and MIP problems. However most experts, including Gomory himself, considered them to be impractical due to numerical instability, as well as ineffective because many rounds of cuts were needed to make progress towards the solution. Things turned around in the mid-1990s when Cornuejols and co-workers showed them to be very effective in combination with branch-and-cut and ways to overcome numerical instabilities. Nowadays, all commercial MIP solvers use Gomory cuts in one way or another. Gomory cuts, however, are very efficiently generated from a simplex tableau, whereas many other types of cuts are either expensive or even NP-hard to separate. Among other general cuts for MIP, most notably lift-and-project dominates Gomory cuts. Other well-known cutting-plane methods include the Kelley's method, and the Kelley-Cheney-Goldstein method.

The basic idea of cutting-plane methods is to alter the convex set of solutions to the related continuous LP problem (i.e., the LP problem that results by dropping the integer constraints) so that the optimal extreme point to the changed continuous problem is integer-valued. This is accomplished by systematically adding additional constraints (cutting planes) that cut off parts of the convex set that do not contain any feasible integer points and solving the resultant problems by the simplex algorithm. Note that an adding cut to a current fractional (i.e., not satisfying integrality) solution must assure that every feasible integer solution of the actual program is feasible for the cut, and the current fractional solution is not feasible for the cut.

### 3.4.3 Branch-and-cut methods

Branch-and-cut method is a hybrid of branch-and-bound and cutting-plane methods. The method solves the LP without the integer constraint using the regular simplex algorithm. When an optimal solution is obtained, and this solution has a non-integer value for a variable that is supposed to be integer, a cutting-plane algorithm is used to find further linear constraints which are satisfied by all feasible integer points but violated by the current fractional solution. If such an inequality is found, it is added to the LP, such that resolving it will yield a different solution which is hopefully "less fractional". This process is repeated until either an integer solution is found (which is then known to be optimal) or until no more cutting planes are found.

At this point, the branch-and-bound part of the algorithm begins. The problem is split into two versions, one with the additional constraint that the variable is greater than or equal to the next integer greater than the intermediate result, and one where this variable is less than or equal to the next lesser integer. In this way new variables are introduced in the basis according to the number of basic variables that are non-integers in the intermediate solution but which are integers according to the original constraints. The new LPs are then solved using the simplex method and the process repeats until a solution satisfying all the integer constraints is found. During the branch-and-bound process, further cutting planes can be separated, which may be either global cuts (i.e., valid for all feasible integer solutions) or local cuts (i.e., satisfied by all solutions fulfilling the side constraints from the currently considered branch-and-bound subtree).

### 3.4.4 Other methods

It is worth mentioning a special set of integer programs called disjunctive programs. Roughly speaking, disjunctive programs comprised a logical system of conjunctive and disjunctive statements, where each statement is defined by a constraint. The basic theory of disjunctive programming can be found in the contribution works of Raman and Grossmann (1994) and Lee and Grossmann (2000). A distinctive methodology for solving mixed integer nonlinear programs is the outer-approximation algorithm developed by Duran and Grossmann (1986). Finally, for more details about MIP algorithms the reader is referred to Wolsey (1998), Nemhauser and Wolsey (1999) and Gass (2003).

## 3.5 Software

In this section, a brief description of the commercial software used to solve the optimization models developed throughout this thesis. There exist a number of commercial tools for general modeling and optimization purposes such as GAMS, AIMMS, AMPL, and ILOG, which render very similar characteristics. In this thesis, GAMS has been used since the CEPIMA research group, where this thesis has been realized, is familiar with this program as well as owns a commercial license. Additionally, GAMS is the most widely used modeling and optimization software in the PSE community. CPLEX solver has been selected for solving the MIP problems addressed throughout the thesis.

### 3.5.1 GAMS – General Algebraic Modeling System

GAMS was the first algebraic modeling language and is formally similar to commonly used fourth-generation programming languages. GAMS contains an integrated development environment (i.e., a language compiler) and is connected to a group of integrated high-performance third-party optimization solvers, such as CPLEX, BARON, GUROBI, CONOPT, and XPRESS. GAMS is tailored for complex, large-scale modeling applications, and allows to build large maintainable models that can be adapted quickly to new situations.

According to Rosenthal (2010) and Castillo, Conejo, Pedregal, García, and Alguaci (2001), some of the more remarkable features of GAMS algebraic modeling language are:

- The model representation is analogous to the mathematical description of the problem. Therefore, learning GAMS programming language is almost natural for those working in the optimization field. Additionally, GAMS is formally similar to commonly used programming languages.
- Models are described in compact and concise algebraic statements which are easy for both humans and machines to read.
- The modeling task is completely apart from the solving procedure. Once the model of the system in question has been built, one can choose among the diverse solvers available to optimize the problem.
- Allows changes to be made in model specifications simply and safely.
- Allows unambiguous statements of algebraic relationships.
- Permits model descriptions that are independent of solution algorithms.
- All data transformations are specified concisely and algebraically. This means that all data can be entered in their most elemental form and that all transformations made in constructing the model and in reporting are available for inspection.
- The ability to model small size problems and afterwards transform them into large-scale problems without significantly varying the code.
- Decomposition algorithms can be programmed in GAMS by using specific commands, thus not requiring additional software.
- GAMS imports/exports data from/to Microsoft EXCEL. Additionally, GAMS can be easily linked with MATLAB (The Mathworks, 1998) using the matgams library (Ferris, 1999) if some special data manipulation is needed.

### 3.5.2 CPLEX solver

IBM ILOG CPLEX, often informally referred to simply as CPLEX, is an optimization solver package. It is named for the Simplex method and the C programming language, although today it contains interior point methods and interfaces in the C++ , C#, and Java programming languages. GAMS/CPLEX is a GAMS solver that allows users to combine the high level modeling capabilities of GAMS with the power of CPLEX optimizers. CPLEX optimizers are designed to solve large, difficult problems quickly and with minimal user intervention. Access is provided (subject to proper licensing) to CPLEX solution algorithms for linear, quadratically constrained and mixed integer programming problems. While numerous solving options are available, GAMS/CPLEX automatically calculates and sets most options at the best values for specific problems. It is worth mentioning that for problems with integer variables CPLEX uses a branch-and-cut algorithm which solves a series of LP subproblems. Because a single mixed integer problem generates many subproblems, even small MIP problems can be very compute intensive and require significant amounts of physical memory.

## 3.6 Final Remarks

In this chapter, the major optimization techniques and tools used throughout this thesis have been presented. The main concepts beneath each method have been briefly described in order to provide the reader with a general understanding of the theory involved into the solution approaches.

Indeed, the mathematical model of a system is the collection of mathematical relationships which, for the purpose of developing a design or plan, characterize the set of feasible solutions of the system. Precisely, being the scope of this thesis the development of techniques for the efficient solution of large-scale production scheduling and planning problems in the process industries, it becomes of utmost importance the discovery of tailored strategies for specific industrial sectors that conform specific MIP modeling techniques (Chapters 4, 5, 6 and 7) and MIP-based specific solution approaches (decomposition techniques in Chapters 8 and 9). At this point, it is worth noticing that *the process of building a mathematical model is often considered to be as important as solving it* because this process provides insight about how the system works and helps organize essential information about it. Models of the real world are not always easy to formulate because of the richness, variety, and ambiguity that exists in the real world or because of our ambiguous understanding of it. As a result, building up concise, useful and efficient mathematical models/approaches is a very difficult and challenging task.

Part II

# Continuous Processes

Chapter 4

# Production Planning and Scheduling of Parallel Continuous Processes

## 4.1 Introduction

In this chapter we focus on production processes with continuous parallel units, often also termed as single-stage continuous processes. Given the prominence of this class of problems, a number of standalone scheduling as well as integrated production planning and scheduling approaches have been proposed in the literature for similar problems, though no methods have been reported for the specific problem discussed here. Specifically, we developed a mathematical approach to the simultaneous production planning and scheduling of continuous parallel units producing a large number of final products that can be classified into product families. The problem under consideration appears in many stages of operation in process industries, including packing in batch and continuous production facilities. Thus, it is quite important since it arises in a number of different production environments (e.g., food and beverage industry, consumer products, etc.).

In contrast with previous research works, a more general case has been considered based on: (i) product families, (ii) short planning periods that may lead to idle units for entire periods, (iii) changeovers spanning multiple periods, and (iv) maintenance activities. The motivation to consider product families comes from the fact that in many production environments, there exist products that share many characteristics. In fact, the current work was first developed to address problems in a highly complex real-life bottling facility producing hundreds of final products. The grouping into families is based on various criteria, including product similarities, processing similarities, or changeover considerations. The goal of the aforementioned grouping is to lead to computationally tractable optimization models without compromising the quality of solution. Furthermore, the use of product families reflects managerial practice prevalent in many production systems (Inman & Jones, 1993; Grunow *et al.*, 2002; Günther, Grunow, & Neuhaus, 2006; Kopanos, Puigjaner, & Georgiadis, 2010).

## 4.2 Problem Statement

The production planning and scheduling of continuous parallel units is considered here. The problem is defined in terms of the following items:

(i) A known planning horizon divided into a set of periods, $n \in N$.

(ii) A set of parallel processing units, $j \in J$, with available production time in period $n$ equal to $\omega_{jn}$.

(iii) A set of product families or simply *families*, $f \in F$, wherein all products are grouped into; $f \in F_j$ is the subset of families that can be assigned to unit $j$, and $j \in J_f$ is the subset of units that can process family $f$.

(iv) A set of products $i \in I$ with demand $\zeta_{in}$ at the end of time period $n$, backlog $\psi_{in}$ and inventory $\xi_{in}$ costs, minimum and maximum production rates ($\rho_{ijn}^{min}$, $\rho_{ijn}^{max}$), minimum processing times $\tau_{ijn}^{min}$, and processing cost $\lambda_{ij}$; the subset of products in family $f$ is denoted by $i \in I_f$; $i \in I_j$ is the subset of products that can be assigned to unit $j$, and $j \in J_i$ is the subset of units that can produce product $i$.

(v) A sequence-dependent switchover operation, or simply *changeover*, is required on each processing unit whenever the production is changed between two different product families; the required changeover time is $\gamma_{ff'j}$, while the changeover cost is $\phi_{ff'j}$; and

(vi) A sequence-independent switchover operation, henceforth referred to as *setup*, is required whenever a product $i$ is assigned to a processing unit $j$; the setup time is $\delta_{ij}$ and the setup cost is $\theta_{ij}$.

We assume a non-preemptive operation mode, no utility restrictions, and that changeover times are not greater than a planning period.

The objective is to determine:

(i) the assignment ($Y_{fjn}^F$) of product families on each unit in every production period;

(ii) the sequencing ($X_{ff'jn}$) between families on each unit in every period;

(iii) the assignment of products to processing units in every period ($Y_{ijn}$);

(iv) the production amount for every product in every period ($P_{in}$); and,

(v) the inventory ($S_{in}$) and backlog ($B_{in}$) profiles for all products.

So as to satisfy customer demand at the minimum total cost, including operating, changeover and setup costs, as well as inventory and backlog costs.

## 4.3 Proposed Approach

The novelty of the proposed formulation lies in the integration of three different modeling approaches. In particular, we use: (a) a discrete-time approach for inventory and backlog costs calculation for production planning, (b) a continuous-time approach with sequencing using immediate precedence variables for the scheduling of families, and (c) lot-sizing type of capacity constraints for the scheduling of products. Figure 4.1 illustrates the proposed modeling approach.

For the production planning subproblem we employ a time grid with fixed, though not necessarily equal, production periods. The planning horizon is therefore divided into $n \in N = \{1, 2, ...\}$ periods; period $n$ starts at time point $n - 1$ and finishes at time point $n$. Note that the use of a discrete-time approach at the planning level (big-bucket) enables the correct calculation of holding and backlog costs. Material balances for each product are expressed at the end of each planning period in terms of: total production level, $P_{in}$, inventory level, $S_{in}$, and backlog level, $B_{in}$. We assume that the amounts produced during a planning period become available at the end of this period. The *communication* between the production planning and scheduling subproblems is accomplished via the amount $Q_{ijn}$ of product $i$ produced in unit $j$ during period $n$. Variable $Q_{ijn}$ is used by the production planning problem for the calculation of variables $P_{in}$ for the material balances, while at the same time variables $Q_{ijn}$ are subject to detailed sequencing and capacity constraints of the scheduling subproblem.

The scheduling subproblem has two levels. At the first level, we schedule product families on units using an immediate precedence approach for sequencing, while at the second level, we employ a lot-sizing-based approach to express capacity constraints for products. In particular, we define assignment binary variable $Y^F_{fjn}$ to denote the assignment of family $f$ in unit $j$ during period $n$, and sequencing binary variable $X_{ff'jn}$ ($\bar{X}_{f'fjn}$) to denote an immediate precedence $f \rightarrow f'$ in unit $j$ within period $n$ (across periods $n - 1$ and $n$). This allows us to account for changeover times between families and correctly calculate changeover costs. The activation of the sequencing variables is achieved using a modification of the network formulation of Karmarkar and Schrage (1985) and Sahinidis and Grossmann (1991). For the timing of processing of family $f$ in unit $j$ during period $n$ we introduce variable $C_{fjn}$. Individual product setups and capacity constraints are modeled using lot-sizing-type setup binary variables: $Y_{ijn} = 1$ if product $i$ is produced in unit $j$ during period $n$. Variables $Y_{ijn}$ are used to constrain the processing time $T_{ijn}$, which is in turn used to constrain the production amount $Q_{ijn}$.

In addition to this novel integration of modeling approaches, our formulation can accurately account for changeovers in the presence of fixed planning periods. First, it allows us to track the last family produced in each period, and therefore account correctly for the first changeover in the following period. If the last family in period $n - 1$ and the first in period $n$ are the same, then no changeover time/cost is added (changeover carryover). Second, it allows changeover operations to cross planning period boundaries (changeover crossover), thereby allowing higher utilization of resources and obtaining better solutions (see Sung and

Maravelias (2008) for a discussion of these aspects in the context of lot-sizing problems). Finally, an extension of our approach allows the seamless modeling of idle planning periods, another aspect that has often been neglected, though it can appear in optimal solutions when the length of the planning period is short and/or a unit is not heavily loaded.

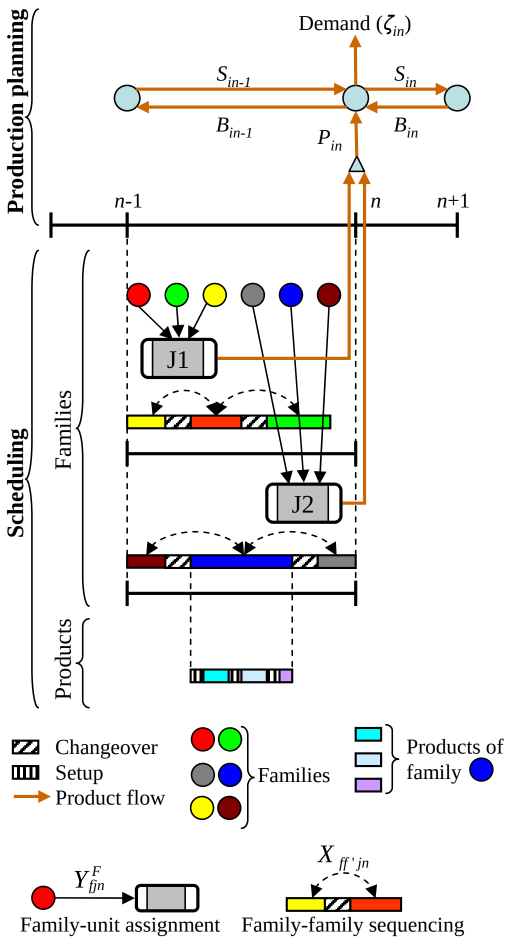



**Figure 4.1:** Proposed modeling approach.

## 4.4 Mathematical Formulation

In this section, we present a MIP formulation for the production planning and scheduling of a parallel-unit (single-stage) facility described in Section 4.2. Constraints are grouped according to the type of decision (e.g., assignment, timing, sequencing, etc.). To facilitate the presentation of the model, we use uppercase

Latin letters for optimization variables and sets, and lowercase Greek letters for parameters.

**Material Balance Constraints.** The total amount $P_{in}$ of product $i$ produced in period $n$ is the summation of the produced quantities $Q_{ijn}$:

$$P_{in} = \sum_{j \in J_i} Q_{ijn} \qquad \forall i, n \tag{4.1}$$

Mass balances for every product $i$ are expressed at the end of each production period $n$, where initial backlogs ($B_{in=0}$) and initial inventories ($S_{in=0}$) are given:

$$S_{in} - B_{in} = S_{in-1} - B_{in-1} + P_{in} - \zeta_{in} \quad \forall i, n \tag{4.2}$$

Inventory capacity constraints can be enforced via constraints similar to:

$$\begin{aligned} S_{in} &\leq \text{product storage capacity} \quad \forall i, n \quad \text{or} \\ \sum_i S_{in} &\leq \text{plant storage capacity} \quad \forall n \end{aligned} \tag{4.3}$$

**Family Allocation Constraints.** Obviously, a family $f$ is assigned to a processing unit $j \in J_f$ during a production period $n$ if at least one product that belongs to this family, $i \in I_f$, is produced on this processing unit at the same period:

$$Y^F_{fjn} \geq Y_{ijn} \qquad \forall f, i \in I_f, j \in J_f, n \tag{4.4}$$

where $Y^F_{fjn}$ denotes a family-to-unit assignment, and $Y_{ijn}$ denotes a product-to-unit assignment. Moreover, constraint set (4.5) enforces the binary $Y^F_{fjn}$ to be zero when no product $i \in I_f$ is produced on unit $j$ in period $n$:

$$Y^F_{fjn} \leq \sum_{i \in I_f} Y_{ijn} \qquad \forall f, j \in J_f, n \tag{4.5}$$

Note that If we do not include constraint set (4.5), we may obtain solutions where $Y^F_{fjn} = 1$ for a family that $\sum_{i \in I_f} Y_{ijn} = 0$; that also means $T^F_{fjn} = 0$. Note that this undesired case could be observed if the changeover cost/time for $f \to f'$ is higher than the sum of costs/times for changeovers $f \to f''$ and $f'' \to f'$.

**Family Sequencing and Timing Constraints.** We introduce binary variable $X_{ff'jn}$ to define the local precedence between two families $f$ and $f'$: $X_{ff'jn} = 1$ if family $f'$ is processed immediately after family $f$ in unit $j$. Constraints (4.6) and (4.7) ensure that, if family $f$ is allocated on the processing unit $j$ at period $n$ (i.e., $Y^F_{fjn} = 1$), at most one family $f'$ is processed before and after it, respectively. If family $f$ is assigned first on a processing unit $j$ (i.e., $WF_{fjn} = 1$), then it has no predecessor. Similarly, if family $f$ is assigned last on a processing unit $j$ (i.e., $WL_{fjn} = 1$), then has no successor.

$$\sum_{f' \neq f, f' \in F_j} X_{f'fjn} + WF_{fjn} = Y^F_{fjn} \qquad \forall f, j \in J_f, n \tag{4.6}$$

$$\sum_{f' \neq f, f' \in F_j} X_{ff'jn} + WL_{fjn} = Y^F_{fjn} \qquad \forall f, j \in J_f, n \tag{4.7}$$

The correct number of immediate precedence variables is activated through constraint set (4.8), which enforces the total number of sequenced pairs within a period to be equal to the total number of active assignments during this period minus one:

$$\sum_{f \in F_j} \sum_{f' \neq f, f' \in F_j} X_{ff'jn} + 1 = \sum_{f \in F_j} Y^F_{fjn} \qquad \forall j, n \tag{4.8}$$

To avoid sequence subcycles, we also include constraint set (4.9), which ensures a feasible timing of the families assigned to the same processing unit:

$$\begin{aligned} C_{f'jn} \geq C_{fjn} + T^F_{f'jn} + \gamma_{ff'j} X_{ff'jn} - \omega_{jn}(1 - X_{ff'jn}) \\ \forall f, f' \neq f, j \in (J_f \cap J_{f'}), n \end{aligned} \tag{4.9}$$

**Family Changeovers Across Adjacent Periods.** We introduce binary variable $\bar{X}_{ff'jn}$ to denote a changeover from family $f$ to $f'$ in unit $j$ taking place at the beginning of period $n$. This binary variable is active only for the family $f$ processed last in period $n-1$ (i.e., $WL_{fjn-1} = 1$) and the family $f'$ that is processed first in period $n$ (i.e., $WF_{f'jn} = 1$), according to constraints (4.10) and (4.11) (see Figure 4.2).

$$WF_{fjn} = \sum_{f' \in F_j} \bar{X}_{f'fjn} \qquad \forall f, j \in J_f, n > 1 \tag{4.10}$$

$$WL_{fjn-1} = \sum_{f' \in F_j} \bar{X}_{ff'jn} \qquad \forall f, j \in J_f, n > 1 \tag{4.11}$$

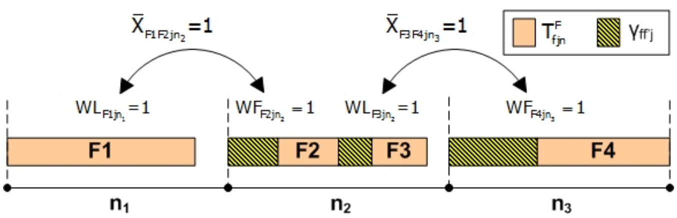



**Figure 4.2:** Family changeovers between adjacent periods.

**Changeover Crossover Constraints.** In most existing approaches, changeover times have to begin and finish within the same period. In other words, crossovers of changeover times are not allowed. This restriction may result in suboptimal solutions. For example, in Figure 4.3 a better solution can be obtained if the changeover from F1 to F2 starts in period $n_1$ and finishes in $n_2$.

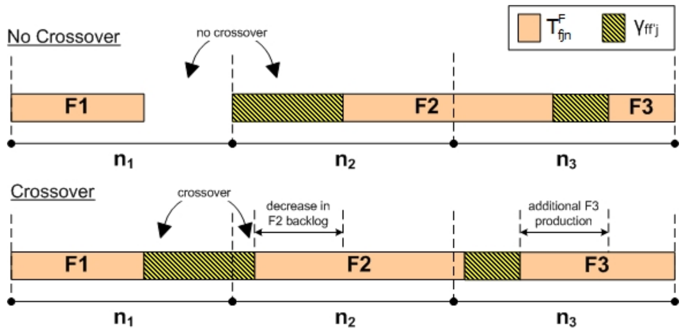



**Figure 4.3:** Crossover of changeover times.

To model changeover crossovers, nonegative variables $\bar{U}_{jn}$ and $U_{jn}$ are introduced. If a changeover operation with a duration equal to $\gamma_{ff'j}$ starts in period $n-1$ and is continued in period $n$, then: $U_{jn-1}$ represents the fraction of time of the changeover operation that takes place in period $n-1$; and $\bar{U}_{jn}$ represents the fraction of time of the operation that is performed in period $n$ (see Figure 4.4).

$$\bar{U}_{jn} + U_{jn-1} = \sum_{f \in F_j} \sum_{f' \in F_j, f' \neq f} \gamma_{ff'j} \bar{X}_{ff'jn} \qquad \forall j, n > 1 \tag{4.12}$$

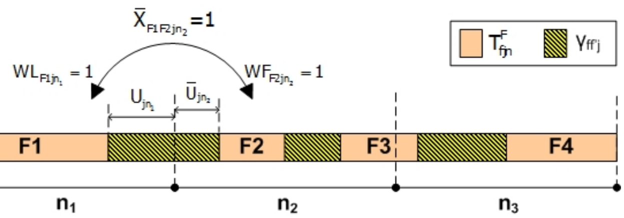



**Figure 4.4:** Modeling of crossover of changeover times.

**Unit Production Time.** The summation of the production times of families $f \in F_j$ that are processed on unit $j$ plus the total changeover times within period $n$ (including variables $\bar{U}_{jn}$ and $U_{jn}$) is constrained by the available production time $\omega_{jn}$:

$$\bar{U}_{jn} + U_{jn} + \sum_{f \in F_j} T^F_{fjn} + \sum_{f \in F_j} \sum_{f' \in F_j, f' \neq f} \gamma_{ff'j} X_{ff'jn} \leq \omega_{jn} \qquad \forall j, n \tag{4.13}$$

where $\bar{U}_{jn=1}$ corresponds to the time point that unit $j$ is available to begin processing any task in the first period. Notice that $\bar{X}_{ff'jn=1} = 0$ by definition.

**Product Lot-Sizing Constraints.** Upper and lower bounds on the production, $Q_{ijn}$, of product $i$ on unit $j$ during period $n$ are enforced by constraint (4.14), where variable $T_{ijn}$ denotes the processing time of product $i$ on unit $j$ in period $n$,

$$\rho_{ij}^{min} T_{ijn} \leq Q_{ijn} \leq \rho_{ij}^{max} T_{ijn} \qquad \forall i, j \in J_i, n \tag{4.14}$$

Upper and lower bounds on the processing time $T_{ijn}$ are enforced by:

$$\tau_{ijn}^{min} Y_{ijn} \leq T_{ijn} \leq \tau_{ijn}^{max} Y_{ijn} \qquad \forall i, j \in J_i, n \tag{4.15}$$

Note that a tight processing time upper bound $\tau_{ijn}^{max}$ can be estimated as follows:

$$\tau_{ijn}^{max} = \begin{cases} \omega_{jn} & \text{if } \sum\limits_{n'=1}^{N} \zeta_{in'}/\rho_{ij}^{min} \geq \omega_{jn}, \\ \sum\limits_{n'=1}^{N} \zeta_{in'}/\rho_{ij}^{min} & \text{if } \sum\limits_{n'=1}^{N} \zeta_{in'}/\rho_{ij}^{min} < \omega_{jn}. \end{cases}$$

Since, the switchovers between products that belong to the same family ($i \in I_f$) are sequence-independent, sequencing and timing decisions regarding products can be made post optimization without affecting the quality of solution. For example, the sequencing can be determined depending on the characteristics of the production process. The family processing time $T_{fjn}^F$ is defined by (see Figure 4.5):

$$T_{fjn}^F = \sum_{i \in I_f} (T_{ijn} + \delta_{ij} Y_{ijn}) \qquad \forall f, j \in J_f, n \tag{4.16}$$

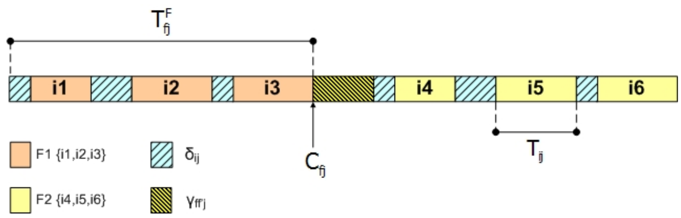



**Figure 4.5:** Family and product processing times.

**Objective Function.** The optimization goal is the minimization of total inventory, backlog, changeover (inside and across periods), setup, and operating costs:

$$\begin{aligned} \min \quad & \sum_i \sum_n (\xi_{in} S_{in} + \psi_{in} B_{in}) \\ + & \sum_f \sum_{f' \neq f} \sum_{j \in (J_f \cap J_{f'})} \sum_n \phi_{ff'j} (X_{ff'jn} + \bar{X}_{ff'jn}) \\ + & \sum_i \sum_{j \in J_i} \sum_n (\theta_{ij} Y_{ijn} + \lambda_{ij} Q_{ijn}) \end{aligned} \tag{4.17}$$

**Integrality and Nonegativity Constraints.** The domains of decision variables are defined as follows:

$$\begin{aligned}
Y_{ijn} \in \{0,1\} \quad \text{and} \quad Q_{ijn}, T_{ijn} \geq 0 \qquad & \forall i, j \in J_i, n \\
WF_{fjn}, WL_{fjn}, Y^F_{fjn} \in \{0,1\} \quad \text{and} \quad T^F_{fjn}, C_{fjn} \geq 0 \qquad & \forall f, j \in J_f, n \\
X_{ff'jn}, \bar{X}_{ff'jn} \in \{0,1\} \quad & \forall f, f', j \in (J_f \cap J'_f), n \\
B_{in}, S_{in}, P_{in} \geq 0 \qquad & \forall i, n \\
U_{jn}, \bar{U}_{jn} \geq 0 \qquad & \forall j, n
\end{aligned} \tag{4.18}$$

The proposed MIP model, CR, consists of constraints (4.1) – (4.18).

**Extension I: Idle Units.** Note that model CR, similarly to most existing approaches, is based on the assumption that units do not remain completely idle in any period. In other words, processing units produce at least one product in each period, except maintenance periods. Generally speaking, this assumption is valid for medium to long planning periods (e.g., a production week). However, if short periods are used (e.g., a production day) to accurately model frequent intermediate due dates, idle periods may be present in an optimal solution.

To model unit idle periods, we define a dummy product $i \in I^{idle}_f$ for each family, with zero setup time and cost (i.e., $\delta_{ij} = 0, \theta_{ij} = 0 \quad \forall i \in I^{idle}_f$). The processing times of dummy products are then constrained by (4.15) with $\tau^{min}_{ijn} = 0$ and $\tau^{max}_{ijn} = \omega_{jn}$. Note that if a processing unit produces only a dummy product in a production period then this actually means that the unit remains idle during that period. Having defined a variable for idle time, we can now express constraint (4.13) as equality, where $T_{ijn} \forall i \in I^{idle}_f$ now act as slack variables. The new MIP model for idle units is named CR-D.

Figure 4.6 shows an illustrative example of a single-unit production plan over three production periods, where the unit produces family F1 in period $n_1$, remains idle during period $n_2$, and produces family F2 in period $n_3$ (white boxes denote the imaginary production of dummy product $i \in I^{idle}_f$). Note that the two solutions are equivalent.

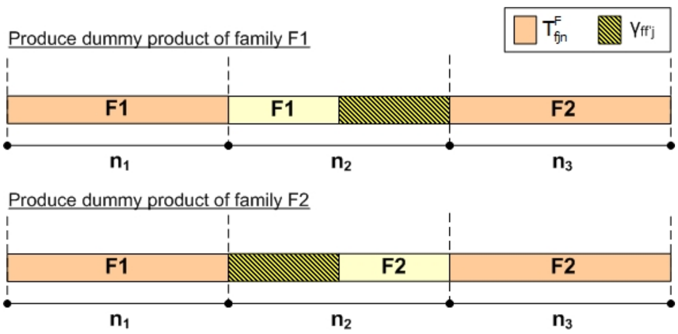



**Figure 4.6:** Modeling of changeover crossover through idle periods.

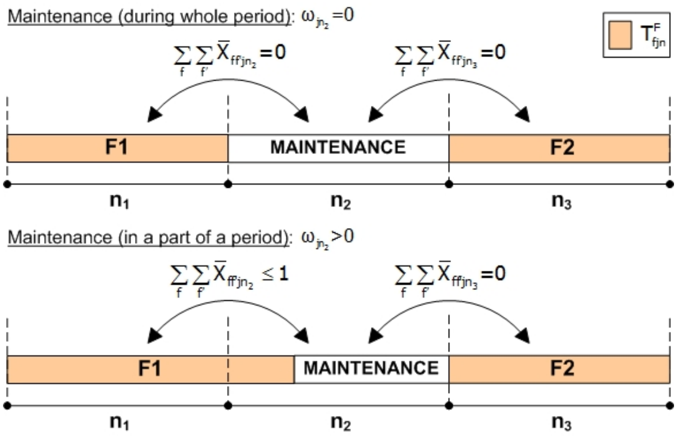



**Figure 4.7:** Modeling of maintenance activities.

**Extension II: Maintenance Activities.** Maintenance activities in a given period can be readily addressed by fixing the corresponding changeover crossover variables to zero and modifying the available production time $\omega_{jn}$ accordingly. We assume that the maintenance activity is carried out at the end of the production period $n$. Our model can accommodate cases where the duration of a maintenance task is equal to the available production time (by setting $\omega_{jn} = 0$) or cases where the duration of the maintenance activity is smaller than the length of the period $n$. Note that if a maintenance task is performed between the production of two different families, there is no need for a changeover operation. Figure 4.7 illustrates our approach. Finally, note that we can also cope with maintenance tasks whose duration is greater than a planning period.

## 4.5 Applications

In this section, we discuss the application of the proposed model to an illustrative example and a large-scale industrial case study. All problem formulations were solved on a Sun Ultra 4.0 Workstation with 8 GB RAM using CPLEX 11 via a GAMS 22.9 interface (Rosenthal, 2010). A maximum time limit of 300 CPU s was used. It represents the amount of time practitioners are willing to wait for a solution. This is because different scenarios have to be tested before a solution is dispatched, and rescheduling solutions should be generated routinely and in a timely fashion. Furthermore, we selected this short time limit to make a fair comparison of our method to commercial tools that typically yield solutions within few minutes.

### 4.5.1 Illustrative example

We consider a simple example with 15 products (I01 – I15), grouped into 5 families (F01 – F05), and 3 processing units (J01 – J03). All products can be produced

**Table 4.1:** Illustrative Example: changeover times (costs).

| Family | F01 | F02 | F03 | F04 | F05 |
|---|---|---|---|---|---|
| F01 | - | 3.0 (50) | 3.0 (40) | 5.0 (60) | 1.5 (50) |
| F02 | 5.3 (40) | - | 3.0 (50) | 3.0 (80) | 2.0 (90) |
| F03 | 2.8 (70) | 4.0 (30) | - | 2.5 (80) | 4.0 (30) |
| F04 | 2.4 (100) | 4.0 (100) | 3.0 (90) | - | 3.0 (60) |
| F05 | 3.2 (30) | 4.0 (50) | 2.0 (50) | 4.0 (70) | - |

**Table 4.2:** Illustrative Example: product demands per period (kg).

| Product | day 1 ($n_1$) | day 2 ($n_2$) | day 3 ($n_3$) | day 4 ($n_4$) |
|---|---|---|---|---|
| I01 | 50 | 0 | 70 | 20 |
| I02 | 0 | 80 | 10 | 50 |
| I03 | 30 | 50 | 20 | 30 |
| I04 | 0 | 10 | 75 | 10 |
| I05 | 70 | 90 | 10 | 20 |
| I06 | 65 | 0 | 75 | 0 |
| I07 | 40 | 50 | 0 | 30 |
| I08 | 0 | 45 | 0 | 0 |
| I09 | 55 | 0 | 45 | 15 |
| I10 | 10 | 100 | 30 | 50 |
| I11 | 40 | 15 | 20 | 30 |
| I12 | 0 | 95 | 40 | 30 |
| I13 | 80 | 0 | 40 | 30 |
| I14 | 0 | 50 | 0 | 0 |
| I15 | 0 | 0 | 0 | 60 |

on any unit. Products are grouped into families as follows: $I_{F01} = \{I01, I02, I03\}$, $I_{F02} = \{I04, I05, I06\}$, $I_{F03} = \{I07, I08, I09\}$, $I_{F04} = \{I10, I11, I12\}$, and $I_{F05} = \{I13, I14, I15\}$. The total production horizon is 4 days and is divided into four 24-h periods. Maintenance is scheduled on units J01, J02, and J03, in periods $n_2$, $n_3$, and $n_4$, respectively. Processing data include: setup time $\delta_{ij} = 0.5$; setup cost $\theta_{ij} = 50$; operating cost $\lambda_{ij} = 0.1$; production rate $\rho_{ij}^{max} = 10$; minimum processing time $\tau_{ij}^{min} = 0.2$; inventory cost $\xi_{in} = 1$; and backlog cost $\psi_{in} = 3$ for all products. Changeover times $\gamma_{ff'j}$ and costs $\phi_{ff'j}$ between product families are given in Table 4.1. Product demands can be found in Table 4.2. The processing sequence of products that belong to the same family during every period $n$ is predetermined; specifically, it is in ascending index order.

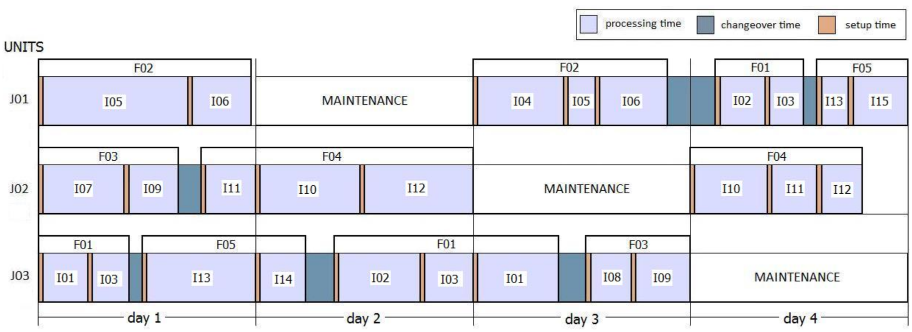



**Figure 4.8:** Illustrative Example: Gantt chart of optimal solution.

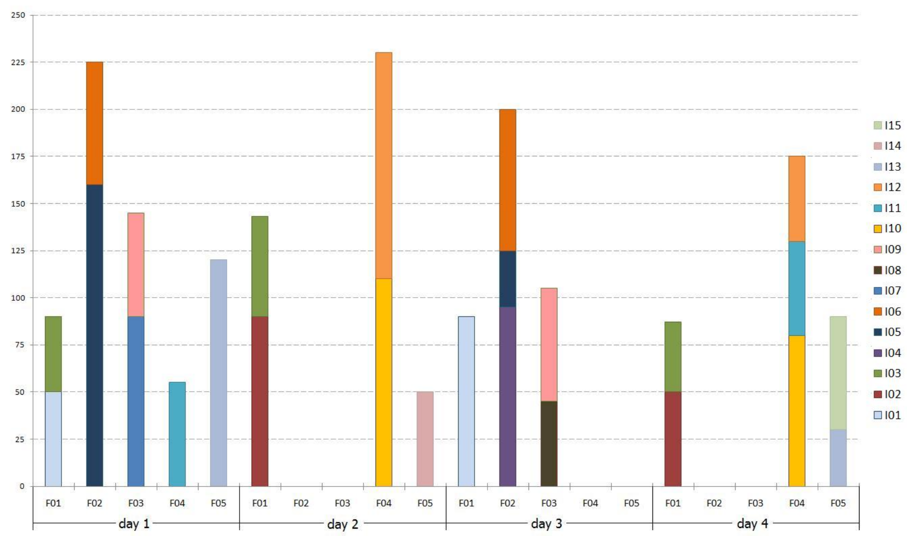



**Figure 4.9:** Illustrative Example: Production profiles of families and products (kg).

The optimization goal is the minimization of the total cost, as defined in equation (4.17). The proposed MIP model obtained the optimal solution ($2,630) in 30 CPU s (see Table 4.3). The optimal production schedule for products and product families is shown in Figure 4.8. Notice the changeover crossover between family F02 and F01 on unit J01 across the boundary between the third and fourth day. Figure 4.9 presents the production profiles for product families and products, while Figure 4.10 shows the inventory and backlog profiles for every product. High inventories are observed in the first day; especially for product I05 (90 kg), I07 (50 kg), and I13 (40 kg). High backlogs are observed in the second day, for product I08 (45 kg) and I04 (10 kg), and in the third day, for product I10 (30 kg), I11 (20 kg), and I12 (15 kg).

**Table 4.3:** Computational results for all problem instances.

| Problem Instance | cons-traints | continuous variables | binary variables | nodes | objective function ($) | CPU s |
|---|---|---|---|---|---|---|
| Illustrative Example | 1,147 | 708 | 510 | 10,501 | 2,630 | 30 |
| Case Study: Instance I | 5,192 | 3,730 | 1,890 | 97,017 | 235,577 | 193 |
| Case Study: Instance II | 8,882 | 6,269 | 3,264 | 59,573 | 155,629 | 300 |

## 4.5.2 Industrial case study

In this subsection, we consider a complex real-world problem in the continuous bottling stage of the Cervecería Cuauhtémoc Moctezuma beer production facility, situated in Mexico. The facility under study consists of eight processing units (J01 – J08), working in parallel and producing a total of 162 products which are grouped into 22 families (F01 – F22). Process data are not provided due to confidentiality issues.

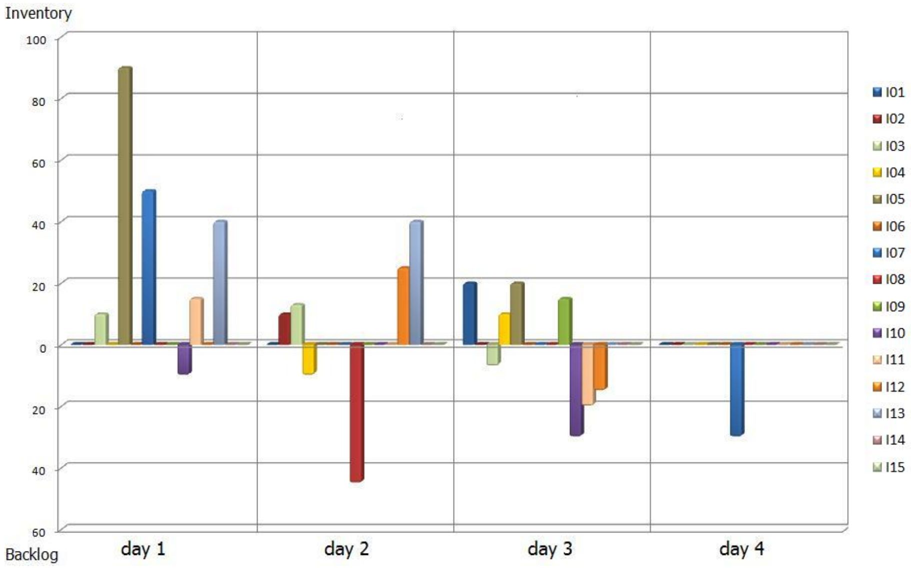



**Figure 4.10:** Illustrative Example: Product inventory and backlog profiles (kg).

**Instance I.** In this instance, we study the planning-scheduling problem over a 6-week production horizon divided into six 168-h periods. The problem was solved to optimality using model CR in less that 200 CPU s. Model and solution statistics can be found in Table 4.3. Figure 4.11 shows the Gantt chart of the optimal solution ($235,577) for product families. Note that each family block is subdivided into a number of smaller product blocks separated by setup times. The production profiles of all families in the eight units of the facility are presented in Figure 4.12, while Figure 4.13 shows the total inventory and backlog cost profiles. In the first week, we observe a high inventory cost, while backlog cost remains low during all periods. Inventories represent 34% of the total cost.

Furthermore, the solution obtained using the approach presented in this study was compared against the solution that was found, dispatched, and executed in practice using a combination of in-house and commercial tools. The main cost components of the two solutions are compared in Figure 4.14. Clearly, our solution is substantially better. In particular, inventory cost is 78% lower than the inventory cost of the executed solution, and backlog cost is less than 5% of the executed solution. Also, changeover and setup costs have been reduced by more than 80% and 25%, respectively (see the table in the bottom of Figure 4.14). The results of this case study indicate that the proposed framework can in fact be used to address real-world problems. It is computationally efficient and yields solutions of very good quality. Given the significant improvement over the practiced methods, our formulation is currently incorporated into and tested using the tools currently employed to generate detailed production plans.

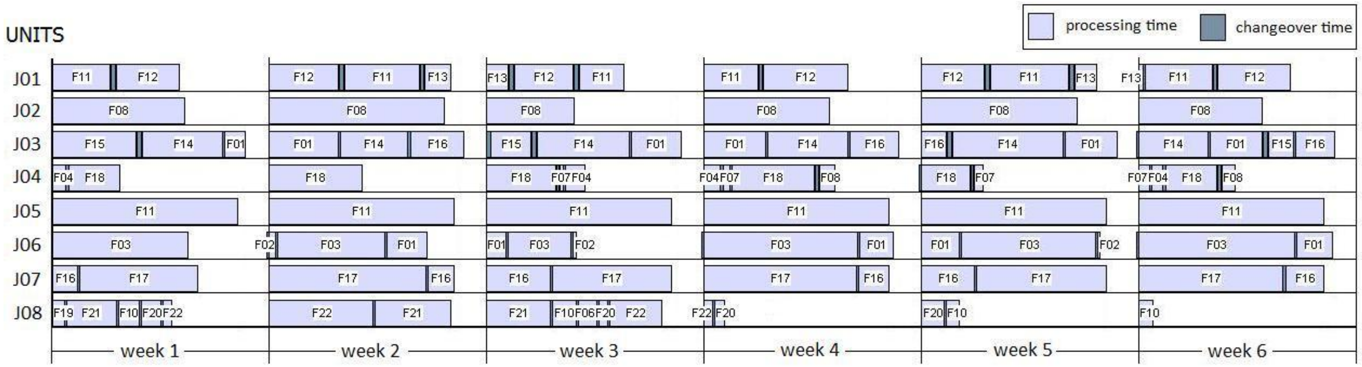



**Figure 4.11:** Industrial Case Study - Instance I: Family Gantt chart of optimal solution.

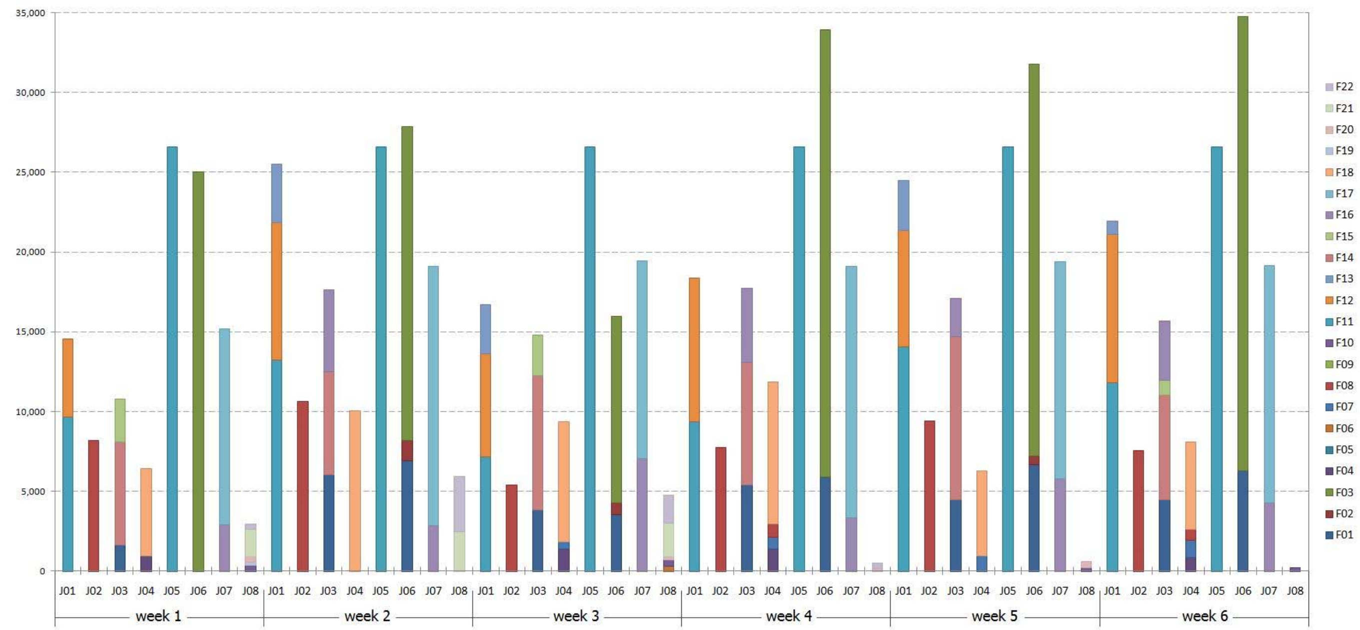



**Figure 4.12:** Industrial Case Study - Instance I: Production profile for families (kg).

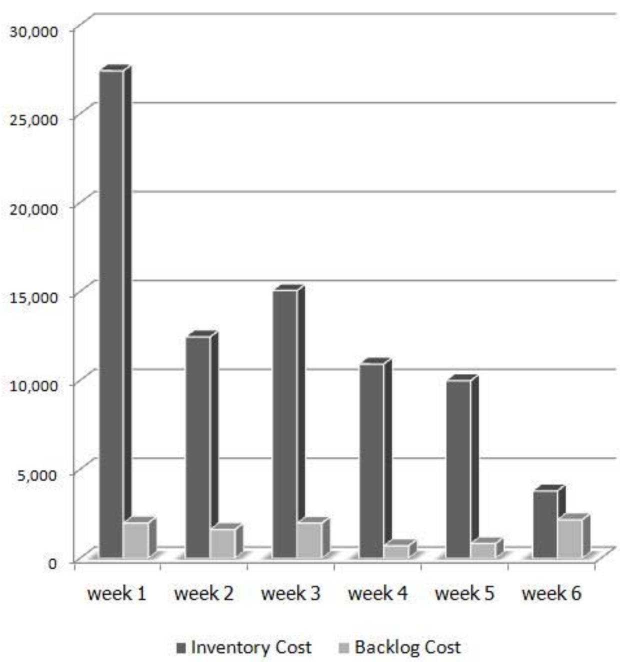
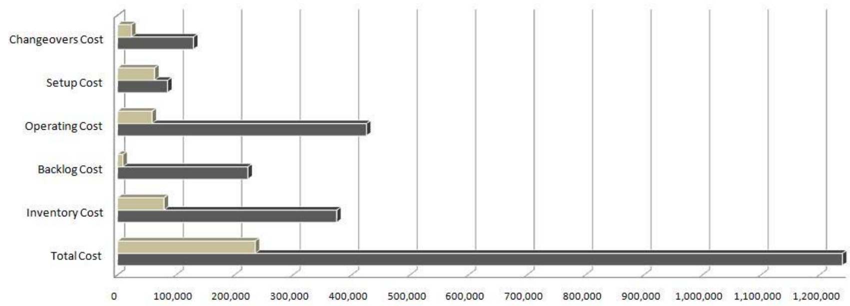



**Figure 4.13:** Industrial Case Study - Instance I: Total inventory & backlog cost profiles ($).




| | Total Cost | Inventory Cost | Backlog Cost | Operating Cost | Setup Cost | Changeovers Cost |
|---|---|---|---|---|---|---|
| CR Model Solution | 235,577 | 79,637 | 9,250 | 59,185 | 63,371 | 24,133 |
| Executed Solution | 1,239,615 | 375,026 | 223,085 | 425,714 | 85,788 | 130,002 |

**Figure 4.14:** Industrial Case Study - Instance I: Comparison of solutions obtained by the proposed MIP model and the tools currently used in practice ($).

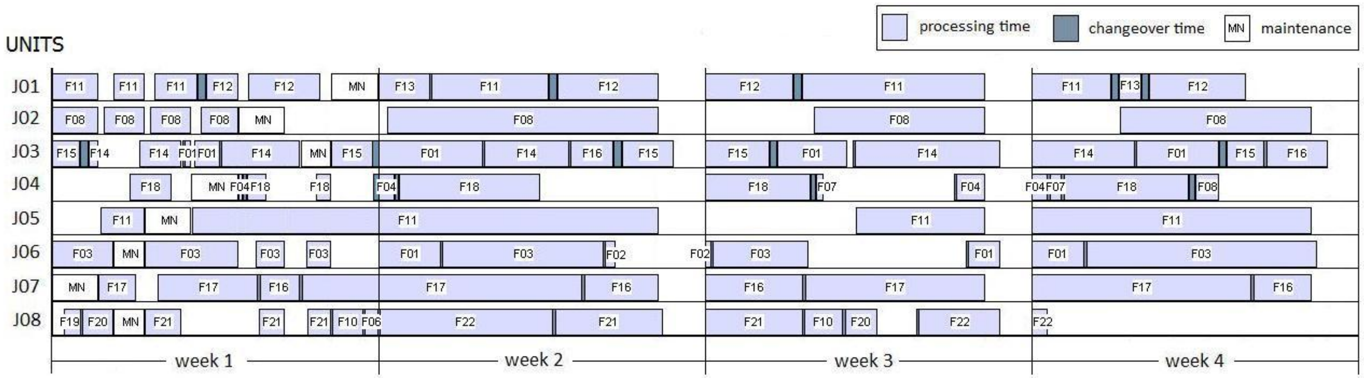



**Figure 4.15:** Industrial Case Study - Instance II: Family Gantt chart of best solution.

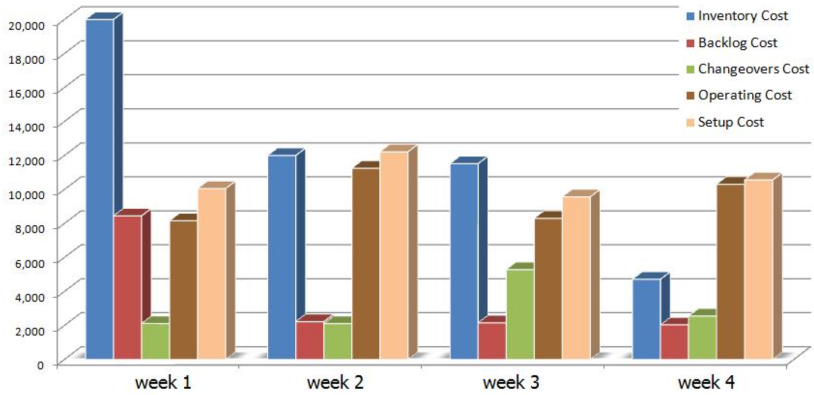



**Figure 4.16:** Industrial Case Study - Instance II: Cost analysis ($).

**Instance II.** In this instance, we discuss a problem on the same processing facility but over a planning horizon of four weeks partitioned into seven 24-h (week 1) and three 168-h planning periods (weeks 2 – 4). In other words, we consider daily orders during the first week, and weekly orders during the next three weeks. This partitioning represents industrial practice, where orders in the near future are known with certainty and are treated separately, while orders in later periods are aggregated. The consideration of such small planning periods leads to larger formulations that are harder to solve to optimality, but are necessary to avoid unmet demand. Furthermore, good solutions to these problems often require changeover crossovers, as well as idling of units for more than one (short) planning period. To address this instance, we developed a formulation that combines model CR-D for the first seven days, allowing units to remain idle over a day; and model CR for the three 7-day production periods.

The model and solution statistics of the resulting MIP model can be found in Table 4.3. The best solution obtained within 300 CPU s has a total cost of $155,629 and has an optimality gap equal to 1%. The solution obtained using the proposed framework is again significantly better than the solution found by commercial tools. The production profile of all families is shown in Figure 4.15, while a cost analysis for every week can be found in Figure 4.16, where for week 1 we present the aggregated costs for day periods 1 – 7. It is worth noting that the consideration of 24-h planning periods during the first week results into higher inventory and backlog costs because daily demands are harder to meet. Inventories represent 31% of the total cost (vs. 34% in Instance I), but backlog costs have increased to 10% from 4% in Instance I. Finally, note that we do observe idle 24-h periods during the first week.

## 4.6 Concluding Remarks

In this chapter, a novel MIP formulation was presented for the production planning and scheduling of single-stage continuous processes with product families, a type of production facility that appears in a number of different production environments and industrial sectors. Our approach combines a discrete-time partitioning of the planning horizon to account for the major planning decisions (production targets, shipments, and inventory levels) with a continuous-time treatment of detailed scheduling decisions within each planning period. Furthermore, at the scheduling level, it combines a *precedence-based* approach to correctly enforce sequencing constraints among product families with a *lot-sizing-like* approach to account for production time constraints for individual products. Our approach addresses appropriately aspects such as changeover carryover and crossover, thereby leading to solutions with higher utilization of resources. Also, it is not based upon the assumption that processing units cannot remain idle during a production period, thereby allowing us to partition the planning horizon into smaller periods, which in turn results in better solutions. Importantly, the integration of these approaches leads to computationally effective MIP models. Very good solutions to problems with hundreds of products can be obtained within 5 CPU min, while optimal solutions can also be found in reasonable time. Furthermore, the proposed formulation yields solutions which are substantially better than the ones obtained using commercial tools, suggesting that MIP methods can be used to address large-scale problems of practical interest.

## 4.7 Nomenclature

**Indices / Sets**

| | |
|---|---|
| $i \in I$ | products |
| $f, f' \in F$ | product families (families) |
| $j \in J$ | processing units (units) |
| $n \in N$ | production periods (periods) |

**Subsets**

| | |
|---|---|
| $F_j$ | families that can be processed in unit $j$ |
| $I_f$ | products that belong to family $f$ |
| $I_f^{idle}$ | dummy product for family $f$ (one per family) |
| $I_j$ | products that can be processed in unit $j$ |
| $J_f$ | processing units that can process family $f$ |
| $J_i$ | processing units that can process product $i$ |

**Parameters**

| | |
|---|---|
| $\gamma_{ff'j}$ | changeover time between family $f$ and $f'$ in unit $j$ |
| $\delta_{ij}$ | setup time of product $i$ in unit $j$ |
| $\zeta_{in}$ | demand of product $i$ at time $n$ |
| $\theta_{ij}$ | setup cost of product $i$ in unit $j$ |
| $\lambda_{ij}$ | operating cost of product $i$ in unit $j$ |
| $\xi_{in}$ | holding cost of product $i$ in period $n$ |
| $\rho_{ij}^{max}$ | maximum production rate of product $i$ in unit $j$ |
| $\rho_{ij}^{min}$ | minimum production rate of product $i$ in unit $j$ |

| | |
|---|---|
| $\tau_{ij}^{max}$ | maximum processing time of product $i$ in unit $j$ |
| $\tau_{ij}^{min}$ | minimum processing time of product $i$ in unit $j$ |
| $\phi_{ff'j}$ | changeover cost between family $f$ and $f'$ in unit $j$ |
| $\psi_{in}$ | backlog cost of product $i$ in period $n$ |
| $\omega_{jn}$ | available production time in unit $j$ in period $n$ |

**Continuous Variables**

| | |
|---|---|
| $B_{in}$ | backlog of product $i$ at time $n$ |
| $C_{fjn}$ | completion time for family $f$ in unit $j$ in period $n$ |
| $P_{in}$ | total produced amount of product $i$ in period $n$ |
| $Q_{ijn}$ | produced amount of product $i$ in unit $j$ during period $n$ |
| $S_{in}$ | inventory of product $i$ at time $n$ |
| $T_{ijn}$ | processing time for product $i$ in unit $j$ in period $n$ |
| $T_{fjn}^{F}$ | processing time for family $f$ in unit $j$ in period $n$ |
| $U_{jn}$ | time within period $n$ consumed by a changeover operation that will be completed in the next period on unit $j$ |
| $\bar{U}_{jn}$ | time within period $n$ consumed by a changeover operation that started in the previous period on unit $j$ |

**Binary Variables**

| | |
|---|---|
| $WF_{fjn}$ | $= 1$ if family $f$ is assigned first to unit $j$ in period $n$ |
| $WL_{fjn}$ | $= 1$ if family $f$ is assigned last to unit $j$ in period $n$ |
| $X_{ff'jn}$ | $= 1$ if family $f$ is processed exactly before $f'$ in period $n$ in unit $j$ |
| $\bar{X}_{ff'jn}$ | $= 1$ if family $f$ in period $n-1$ is followed from family $f'$ in period $n$ on unit $j$ |
| $Y_{ijn}$ | $= 1$ if product $i$ is assigned to unit $j$ in period $n$ |
| $Y_{fjn}^{F}$ | $= 1$ if family $f$ is assigned to unit $j$ in period $n$ |

Part III

# Semicontinuous Processes

Chapter 5

# Production Scheduling in Multistage Semicontinuous Process Industries

## 5.1 Introduction

Production scheduling is certainly important in the process industries including production of specialty chemicals, pharmaceuticals, food and paper in which material is often produced in campaigns using various batch sizes in shared equipments. Determining the timing and sequence of production campaigns becomes increasingly difficult as manufacturers strive to achieve increased production rates while minimizing total costs. In this chapter, a real-life multiproduct multistage ice-cream production facility is considered as a representative semicontinuous process industry. Most production plants in the food industry sector combine continuous operations and batch processes in their product processing routes, thus working in semicontinuous production mode, since production is more flexible and equipment can be more efficiently utilized.

A novel Mixed Integer Programming (MIP) framework and a solution strategy are presented for the optimal production scheduling of multiproduct multistage semicontinuous process industries, such as the ice-cream production facility studied in details. The overall mathematical framework relies on an efficient modeling approach of the sequencing decisions, the integrated modeling of all production stages and the inclusion of strong valid integer cuts in the formulation. The simultaneous optimization of all processing stages increases the plant production capacity, reduces the production cost for final products, and facilitates the interaction among the different departments of the production facility. To the best of our knowledge, there is no previous work in the literature presenting an exact method for addressing the challenges of the underlying food process scheduling problem.

## 5.2 The Ice-cream Production Facility

The ice-cream production facility under study, which represents a typical ice-cream factory, was first described by Bongers and Bakker (2006). The plant is based on a three-stage production process, as shown in Figure 5.1. More specifically, the basic mixes are produced in the main process line (PROC), followed by storage in the ageing vessels (V1 – V6). After a minimum ageing time, the mixes are used for the production of final products (A – H) in two packing lines (PACK1, PACK2). The main process line has a processing rate of 4.5 tons/h, and can feed all ageing vessels; one at a time. Packing line 1 is supplied by two ageing vessels (V1 and V2) of 8 tons capacity each and can accommodate products A to D, whereas packing line 2 can pack products E to H and it is supplied by four ageing vessels (V3 – V6) of 4 tons capacity each. Minimum ageing times and packing rates for all products can be found in Figure 5.1. The maximum shelf life for all intermediate mixes in ageing stage is 72 h. Sequence-dependent changeover, or simply changeover, operations (mainly cleaning and sterilizing tasks) are performed both in the process and the packing lines whenever the production is changed from a product to a different one. Table 5.1 provides the necessary changeover times for performing these operations in the process and the packing lines. Moreover, a cleaning time of 2 h is needed before shutting down the process line and the packing machines. Finally, the production facility is available for 120 h a week (a 48-h weekend).

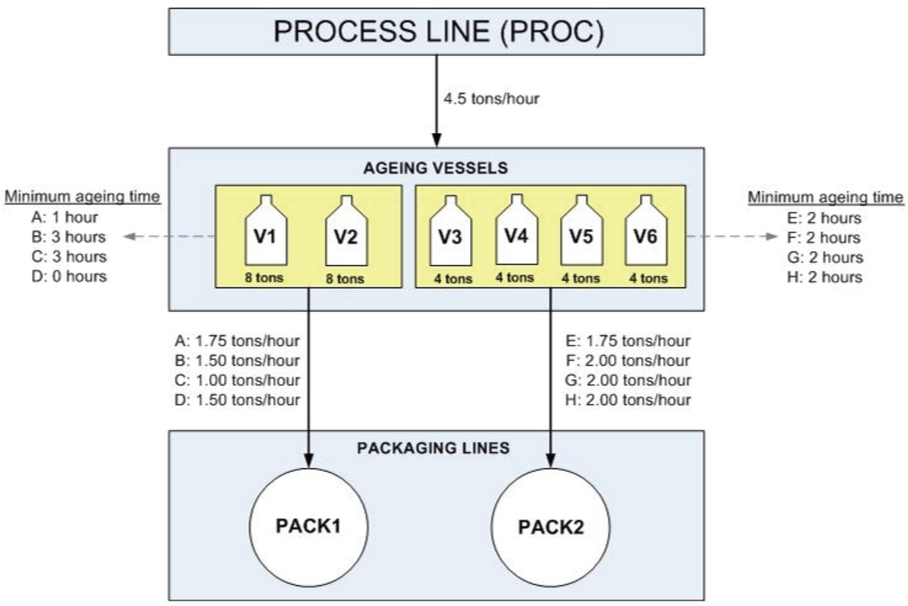



**Figure 5.1:** Ice-cream production facility.

**Table 5.1:** Changeover times in the process line and the packing lines (minutes).

| | process line | | | | | | | | packing lines | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Product | A | B | C | D | E | F | G | H | A | B | C | D | E | F | G | H |
| A | 0 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 0 | 60 | 60 | 60 | 0 | 0 | 0 | 0 |
| B | 30 | 0 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 0 | 60 | 60 | 0 | 0 | 0 | 0 |
| C | 30 | 30 | 0 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 0 | 60 | 0 | 0 | 0 | 0 |
| D | 30 | 30 | 30 | 0 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 0 | 0 | 0 | 0 | 0 |
| E | 30 | 30 | 30 | 30 | 0 | 15 | 15 | 15 | 0 | 0 | 0 | 0 | 0 | 60 | 60 | 60 |
| F | 30 | 30 | 30 | 30 | 5 | 0 | 15 | 15 | 0 | 0 | 0 | 0 | 30 | 0 | 60 | 60 |
| G | 30 | 30 | 30 | 30 | 5 | 5 | 0 | 15 | 0 | 0 | 0 | 0 | 30 | 30 | 0 | 60 |
| H | 30 | 30 | 30 | 30 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 | 30 | 30 | 30 | 0 |

## 5.3 Typical Scheduling Practice in Food Industries

As Bongers and Bakker (2006) pointed out, the practical scheduling inside the vast majority of food factories is focused on just scheduling the packing lines. Afterwards, the packing lines schedule is "thrown over the wall" to the process department, in which a schedule should be made to meet the packing demand. To go further, this schedule is also "thrown over the wall" to the incoming materials department, in which a schedule is made to order/receive the materials. The way food industries are being scheduled nowadays is posing two major problems:

- Each department will strive to ensure that is not to blame for not delivering packing products according to the schedule, while less available production capacity will be communicated to the plant management.

- Any change in the packing schedule might lead to an infeasible schedule in the upstream departments. For instance, packing lines may not run due to lack of intermediate products or unnecessary intermediate products being made in the process plant floor.

Since the above problems are frequently met in relevant industrial environments, the challenge is to appropriately tackle them in an integrated way in order to increase the plant production capacity, and reduce the production cost for final products. The gap between scheduling theory and practice is still evident, since most academic developments are too distant from industrial environments. This industrial reality drove Bongers and Bakker (2006) to characterize as a challenging problem the simultaneous scheduling of all processing stages (i.e., the process line, the ageing vessels, and the packing lines) in typical food processing industries, such as the ice-cream production facility studied in details.

## 5.4 Problem Statement

This study considers the production scheduling problem of industrial-scale multiproduct multistage semicontinuous processes, similar to the previously described ice-cream production process, with the following features:

(i) A set of product orders $i \in I$ should be processed by following a predefined sequence of processing stages $s \in S$ with processing units $j \in J$ working in parallel.

(ii) The total demand $\zeta_i$ for each product order $i$ is divided into a number of batches $b \in B$ that must follow a specific set of processing stages $s$.

(iii) Product order $i$ can be processed in a specific subset of units $j \in J_i$. Similarly, processing stage $s$ can be processed in a specific subset of units $j \in J_s$.

(iv) Every ageing vessel $j \in J_{s_2}$ has a maximum capacity $\mu_j^{max}$. In ageing vessels, a product batch should remain for a minimum ageing time $\tau_i^{age}$ and no longer than its corresponding shelf life $\varepsilon_i^{life}$.

(v) Parameter $\rho_{ij}$ denotes the processing and packing rate for every product $i$ in unit $j \in J_i$.

(vi) Sequence-dependent changeover times $\gamma_{ii'j}$ between consecutive product orders are present in the process (S1) and the packing (S3) stage.

(vii) All model parameters are deterministic.

(viii) Once the processing of an order in a given stage is started, it should be carried out until completion without interruption (i.e., non-preemptive mode).

The key decision variables are:

(i) the allocation of batch $b$ of product $i$ to units $j \in J_i$ per stage, $Y_{ibsj}$;

(ii) the relative sequence for any pair of product batches $i, b$ and $i', b'$ in the process line (i.e., stage S1), $\bar{X}_{ibi'b'}$;

(iii) the relative sequence for any pair of products $i$ and $i'$ in ageing vessels (i.e., stage S2) and packing lines (i.e., stage S3) for $j \in (J_i \cap J_{i'} \cap J_s)$, $X_{ii'}$;

(iv) the starting and completion time for batch $b$ of product $i$ in stage $s$; $L_{ibs}$ and $C_{ibs}$, respectively.

The minimization of makespan constitutes the optimization goal in this work.

### 5.4.1 Industrial production policy

Generally speaking, in most food processing industries, such as the one studied here, in order to achieve high production levels and minimize switchovers of products, the industrial practice imposes operations of the intermediate storage/processing vessels (e.g., ageing vessels, fermentation tanks, etc.) in their maximum capacity. Coming back to the ice-cream production facility under study, the fact that the ageing vessels operate in maximum capacity allows us to solve the batching problem beforehand, and afterwards solve the scheduling problem for the predefined number of product batches.

**Minimum number of batches.** The minimum number of batches $\beta_i^{min}$ to satisfy the demand for each specific product order $i$ depends on the capacity of the storing/ageing vessels in which it can be stored. In the case that these ageing vessels have the same capacity $\mu_j^{max}$, the minimum number of batches is given by:

$$\beta_i^{min} = \frac{\zeta_i}{\mu_j^{max}} \qquad \text{where } j \in (J_i \cap J_{s_2})$$

**Filling time for ageing vessels (processing time in the process line).** The time $\tau_i^{fill}$ to fill an ageing vessel with a product $i$ is calculated by:

$$\tau_i^{fill} = \frac{\mu_j^{max}}{\rho_{ij'}} \qquad \text{where } j \in (J_i \cap J_{s_2}), j' \in (J_i \cap J_{s_1})$$

Note that unit $j' \in J_{s_1}$ corresponds to the process line and unit $j \in J_{s_2}$ to the ageing vessels. It should be also noticed that the above equation is valid *if and only if*: (i) the product $i$ can be stored into a number of *equal-capacity* ageing vessels $\mu_j^{max}$, and (ii) the ageing vessels are supplied by the process lines that have the same processing rate $\rho_{ij'}$. Obviously, the ageing vessels filling time equals to the processing time in the process line due to the continuous process mode.

**Emptying time for ageing vessels (packing time in the packing lines).** The time $\tau_i^{empty}$ to empty an ageing vessel from a product $i$ is calculated as follows:

$$\tau_i^{empt} = \frac{\mu_j^{max}}{\rho_{ij'}} \qquad \text{where } j \in (J_i \cap J_{s_2}), j' \in (J_i \cap J_{s_3})$$

Note that unit $j' \in (J_i \cap J_{s_3})$ corresponds to the packing line where product $i$ can be packed. Once again, this expression is valid *if and only if* the product $i$ can be stored into a number of *equal-capacity* ageing vessels $\mu_j^{max}$. Obviously, the ageing vessels emptying time is equal to the packing time of the packing lines.

## 5.5 Mathematical Formulation

In this section, the proposed MIP formulation is presented for the production scheduling of the multiproduct multistage production facility described above. Constraints are grouped according to the type of decision (e.g., assignment, timing, sequencing, etc.). To facilitate the presentation of the model, we use uppercase Latin letters for optimization variables and sets, and lowercase Greek letters for parameters.

**Unit allocation constraints for any product batch in every processing stage.** Constraints (5.1) guarantee that each product batch $i, b$ goes through one unit $j \in (J_i \cap J_s)$ in each stage $s$.

$$\sum_{j \in (J_i \cap J_s)} Y_{ibsj} = 1 \qquad \forall i, b \leq \beta_i^{min}, s \tag{5.1}$$

**Timing constraints for a product batch in the same processing stage.** The timing for a batch $b$ of product $i$ in each stage $s$ is defined by constraint sets (5.2) to (5.5) (see Figure 5.2). In the process stage, the completion time $C_{ibs}$ for a batch $b$ of product $i$ equals to its starting time $L_{ibs}$ plus the necessary ageing vessel filling time $\tau_i^{fill}$, according to constraints (5.2). In the ageing stage, the timing for each product batch $i, b$ is given by constraints (5.3). The standing (waiting) time for a product batch $i, b$ in ageing stage is denoted by $W_{ibs}$. This standing time plus the minimum ageing time $\tau_i^{age}$ should not exceed the product shelf life, as constraint set (5.4) ensures. Finally, constraints (5.5) calculate the timing for any batch $b$ of product $i$ in the packing stage.

$$L_{ibs} + \tau_i^{fill} = C_{ibs} \qquad \forall i, b \leq \beta_i^{min}, s = 1 \tag{5.2}$$

$$L_{ibs} + \tau_i^{fill} + \tau_i^{age} + W_{ibs} + \tau_i^{empt} = C_{ibs} \qquad \forall i, b \leq \beta_i^{min}, s = 2 \tag{5.3}$$

$$W_{ibs} \leq \varepsilon_i^{life} - \tau_i^{age} \qquad \forall i, b \leq \beta_i^{min}, s = 2 \tag{5.4}$$

$$L_{ibs} + \tau_i^{empt} = C_{ibs} \qquad \forall i, b \leq \beta_i^{min}, s = 3 \tag{5.5}$$

**Timing constraints for a product batch between consecutive processing stages.** Constraints (5.6) and (5.7) define the timing for every product batch $i, b$ between two consecutive processing stages (see Figure 5.2). Constraints (5.6) state that the starting time $L_{ibs}$, for any product batch $i, b$, in the ageing stage is equal to the starting time in the process stage due to the continuous nature of the process

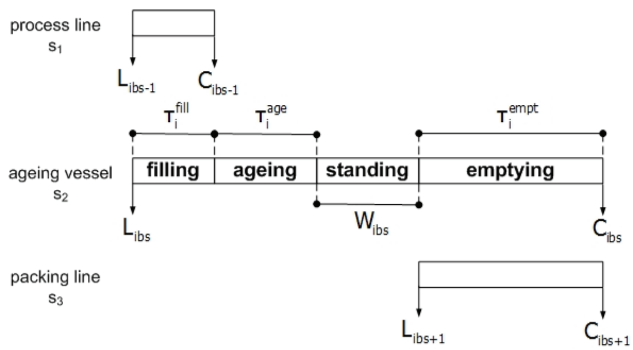



**Figure 5.2:** Timing decisions for a product batch $i, b$ for every processing stage.

stage. Moreover, an ageing vessel is free for processing a product only when it is completely empty, therefore the completion time of a product batch $i, b$ stored in an ageing vessel equals the completion time of this batch in the packing line, according to constraints (5.7).

$$L_{ibs} = L_{ibs-1} \qquad \forall i, b \leq \beta_i^{min}, s = 2 \tag{5.6}$$

$$C_{ibs} = C_{ibs-1} \qquad \forall i, b \leq \beta_i^{min}, s = 3 \tag{5.7}$$

**Timing constraints for two batches of the same product in the packing stage.** If the underlying industrial policy requires a single production campaign in the packing stage, without allowing a waiting time between batches of the same product, constraints (5.8) must be added into the MIP formulation. According to these constraints, the completion time for a product batch $i, b$ should be equal to the starting time for the next indexed product batch $i, b+1$.

$$C_{ibs} = L_{ib+1s} \qquad \forall i, b < \beta_i^{min}, s = 3 \tag{5.8}$$

**Sequencing constraints between product batches in all processing stages.** Constraints (5.9) to (5.13) define the relative sequencing between two product batches. Constraints (5.9) to (5.12) have been formulated as big-M constraints, where the available scheduling horizon $\omega$ plays the role of the M parameter. In addition, our mathematical formulation uses global precedence sequencing variables: (i) for any pair of product batches $i, b$ and $i', b'$ ($i < i'$) in the process stage $\bar{X}_{ibi'b'}$, and (ii) for any pair of different products $i$ and $i'$ ($i < i'$) both in the ageing and the packing stage $X_{ii'}$. Note that the sequencing decisions are the same for the ageing and packing stage, and they are modeled for both stages through a single binary variable $X_{ii'}$ for a given pair of products. Constraints (5.9) enforce the starting time of a product batch $i', b'$ to be greater than the completion time of any product batch $i, b$ processed beforehand plus the corresponding sequence-dependent changeover time $\gamma_{ii'j}$, when both batches are assigned to the same process line. Similarly, constraint set (5.10) describes the opposite case. In a similar manner, constraints (5.11) and (5.12) define the sequencing between any pair of different products $i$ and $i' > i$ in the ageing and the packing stage. Finally, in order to avoid symmetric solutions, if two batches $b$ and $b' > b$ of the same product $i$ are assigned to the same unit, we assume that the lower indexed batch $b$ is performed first, according to constraints (5.13).

$$\begin{gathered} L_{i'b's} \geq C_{ibs} + \gamma_{ii'j} - \omega(1 - \bar{X}_{ibi'b'}) - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \\ \forall i, b \leq \beta_i^{min}, i', b' \leq \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s = 1 \end{gathered} \tag{5.9}$$

$$\begin{gathered} L_{ibs} \geq C_{i'b's} + \gamma_{i'ij} - \omega \bar{X}_{ibi'b'} - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \\ \forall i, b \leq \beta_i^{min}, i', b' \leq \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s = 1 \end{gathered} \tag{5.10}$$

$$L_{i'b's} \geq C_{ibs} + \gamma_{ii'j} - \omega(1 - X_{ii'}) - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \quad \forall i, b \leq \beta_i^{min}, i', b' \leq \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s > 1 \tag{5.11}$$

$$L_{ibs} \geq C_{i'b's} + \gamma_{i'ij} - \omega X_{ii'} - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \quad \forall i, b \leq \beta_i^{min}, i', b' \leq \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s > 1 \tag{5.12}$$

$$L_{ib's} \geq C_{ibs} - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \quad \forall i, b \leq \beta_i^{min}, b' \leq \beta_i^{min}, s, j \in (J_i \cap J_s) : b < b' \tag{5.13}$$

**Objective function: Makespan.** The time point at which all product orders are accomplished corresponds to the makespan. The makespan objective is closely related to the throughput objective. For instance, minimizing the makespan in a parallel-machine environment with changeover times forces the scheduler to balance the load over the various machines and to minimize the sum of all the setup times in the critical bottleneck path. Moreover, the minimization of makespan probably leads to a maximization of production at a mid-term planning level.

$$\min \quad C_{max} \geq C_{ibs} \qquad \forall i, b \leq \beta_i^{min}, s = 3 \tag{5.14}$$

**Tightening constraints.** To reduce the computational effort, constraints (5.15) can further tighten the mathematical formulation by imposing a lower bound on the makespan objective. Note that parameter $\phi_j^{min}$ represents the minimum waiting time to begin using packing line $j \in J_{s_3}$. Obviously, $\phi_j^{min}$ depends on the minimum filling time for ageing vessels $j' \in J_{s_2}$ that are connected to packing line $j$. Additionally, parameter $\alpha_j^{min}$ stands for the minimum number of products that should be assigned to packing line $j$ to ensure full demand satisfaction. Finally, parameter $\gamma_j^{min}$ denotes the minimum changeover time between two different products in packing line $j$.

$$C_{max} \geq \phi_j^{min} + (\alpha_j^{min} - 1)\gamma_j^{min} + \sum_{i \in I_j} \tau_i^{empt} \beta_i^{min} \quad \forall s, j \in J_s : s = 3 \tag{5.15}$$

The proposed MIP model can be further tightened by correlating the relative sequence variables of the process stage (S1) and the packing stage (S3). Constraints (5.16) describe these valid integer cuts by forcing the relative sequence between products $i$ and $i' > i$ in packing and ageing stages to maintain the same for the product batches $i, b$ and $i', b'$ in the process stage; for products $i$ and $i'$ that share the same packing line. In other words, if product $i$ is assigned before product $i'$ to packing unit $j \in (J_i \cap J_{i'} \cap J_{s_3})$, constraint set (5.16) drives all the batches of product $i$ to be allocated to the process line before any batch of product $i'$. Figure 5.3 illustrates graphically the role of these constraints.

$$\bar{X}_{ibi'b'} = X_{ii'} \qquad \forall i, b \leq \beta_i^{min}, i', b' \leq \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s), j' \in (J_i \cap J_{i'} \cap J_{s+2}) : i < i', s = 1 \tag{5.16}$$

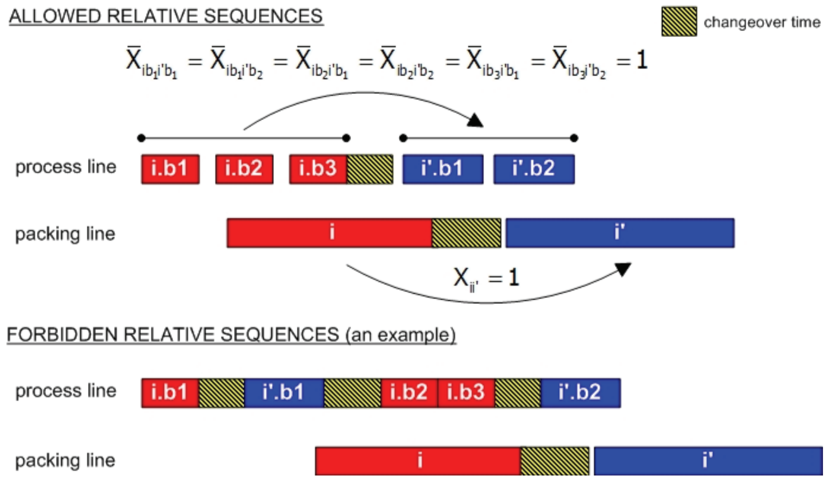



**Figure 5.3:** Illustrative example: relative sequences according to constraints (5.16).

**Integrality and nonegativity constraints.** The domains of all decision variables are defined as follows:

$$\begin{aligned}
Y_{ibsj} \in \{0,1\} \quad & \forall i, b \leq \beta_i^{min}, s, j \in (J_i \cap J_s) \\
\bar{X}_{ibi'b'} \in \{0,1\} \quad & \forall i, b \leq \beta_i^{min}, i', b' \leq \beta_{i'}^{min}, \\
& s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s = 1 \\
X_{ii'} \in \{0,1\} \quad & \forall i, i', s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s > 2 \\
L_{ibs}, C_{ibs} \geq 0 \quad & \forall i, b \leq \beta_i^{min}, s \\
W_{ibs} \geq 0 \quad & \forall i, b \leq \beta_i^{min}, s = 2
\end{aligned} \tag{5.17}$$

The overall MIP formulation consists of constraint sets (5.1) – (5.17).

## 5.6 Proposed Solution Methodology

In this section, a solution methodology is presented for solving efficiently the scheduling problem under study. Before explaining the proposed solution technique, the following points should be taken into consideration: (i) final products can be packed into a specific packing line, (ii) the intermediates of final products that are packed into the same packing line can be stored into the same equal-capacity ageing vessels, and (iii) full demand satisfaction is imposed.

In accordance with the abovementioned points, a lower bound for the makespan for every packing line $\xi_j^{min}$ is calculated as follows:

$$\xi_j^{min} = \phi_j^{min} + \gamma_j^{total} + \sum_{i \in I_j} \tau_i^{empt} \beta_i^{min} \qquad \forall s, j \in J_s : s = 3$$

where parameter $\gamma_j^{total}$ represents the minimum total changeover time in packing line $j$. Additionally, we define the subset $J^{min}$ which contains the packing line that appears the highest $\xi_j^{min}$ value. By doing this, it can be safely assumed that the makespan $C_{max}$ is equal to the makespan of the packing line $j \in J^{min}$, as follows:

$$C_{max} = \phi_j^{min} + \gamma_j^{total} + \sum_{i \in I_j} \tau_i^{empt}\, \beta_i^{min} \qquad \forall s, j \in (J_s \cap J^{min}) : s = 3 \tag{5.18}$$

Therefore, constraints (5.15) can be placed by constraints (5.18), which are tighter. Note that constraints (5.18) provide the minimum possible makespan, and their incorporation into the MIP model may violate some of the remaining constraints, thus leading to infeasible solutions. This may happen when the packing line $j \in J^{min}$ is not the bottleneck and the timing decisions depend on the previous processing stages. In this case, only constraints (5.15) should be used.

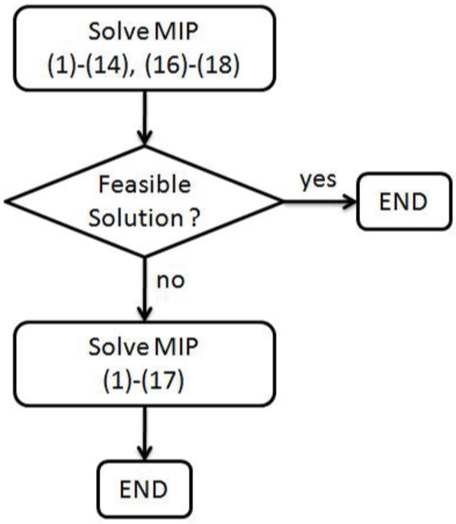



**Figure 5.4:** Proposed solution methodology.

As Figure 5.4 demonstrates, the proposed solution methodology can be distinguished into two steps:

1. Solve the MIP model consisting of constraints (5.1) – (5.14), and (5.16) – (5.18). The solution method terminates, if a feasible solution is obtained, otherwise go to step 2.
2. Solve the MIP formulation consisting of constraints (5.1) – (5.17).

## 5.7 Industrial Case Study

A real-life industrial case study, as described in Section 5.2, is used to illustrate the applicability and the efficiency of the proposed scheduling approach and solution strategy. A total set of ten different problem instances, regarding the demands of final products, have been solved. Roughly speaking, final products are characterized by very high demands given in Table 5.2. All problem instances have been solved in a Dell Inspiron 1520 2.0 GHz with 2 GB RAM using CPLEX 11 via a GAMS 22.8 interface (Rosenthal, 2010).

**Problem Instance PI.01.** Bongers and Bakker (2006) made the first attempt to solve this scheduling problem by using an advanced commercial scheduling software. As they have reported, a feasible schedule on all stages could not be derived automatically by applying the available solvers. They finally obtained a feasible schedule by manual interventions. Recently, Subbiah and Engell (2010) studied the same ice-cream production plant. They used the framework of timed automata, and they solved the optimization problem using reachability analysis (Abdeddaïm, Asarin, & Maler, 2006). A heuristic methodology was implemented to reduced the model size. A feasible solution was found in few CPU s, however it cannot be ruled out that the heuristics employed pruned the optimal solution.

In this problem instance some decisions in the beginning of the production week of interest have been taken at the end of the previous production week. More specifically, product batches D.b1, G.b1, G.b2, and G.b3 have already passed from the process line and assigned to ageing vessels V1, V3, V4, and V5, respectively. Moreover, these product batches have already allocated to the ageing process at the beginning of the time horizon (t=0), and as such they are ready for passing to the packing stage again at t=0. For this reason, in this example parameter $\phi_j^{min} = 0$.

The MIP model consists of 15,848 equations, 491 continuous variables, and 2,024 binary variables. The optimal solution was reached in just 1.83 CPU s despite the fact of the challenging (very high) total demand for final products. The optimal production schedule, which is illustrated in Figure 5.5, results into a makespan of 118.55 h. Table 5.3 shows the breakdown of the utilization of the available scheduling time in the process and packing lines. The process line is utilized for both processing and cleaning 80.89% of the available time compared to a food industry standard of 70%. Packing lines 1 and 2 operate at 98.79% and 92.08% of the total available time, respectively, including both packing and cleaning. The high total demand explains the high utilization in the process and the packing lines. Packing lines illustrate low total changeover times. As expected, in the pro-

**Table 5.2:** Demands for final products for all problem instances (kg).

| Products | PI.01 | PI.02 | PI.03 | PI.04 | PI.05 | PI.06 | PI.07 | PI.08 | PI.09 | PI.10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | 80,000 | 48,000 | 32,000 | 8,000 | 88,000 | 16,000 | 8,000 | 16,000 | 48,000 | 8,000 |
| B | 48,000 | 56,000 | 32,000 | 32,000 | 16,000 | 16,000 | 8,000 | 40,000 | 24,000 | 72,000 |
| C | 32,000 | 16,000 | 40,000 | 64,000 | 24,000 | 16,000 | 96,000 | 32,000 | 56,000 | 8,000 |
| D | 8,000 | 48,000 | 32,000 | 24,000 | 40,000 | 88,000 | 8,000 | 56,000 | 16,000 | 72,000 |
| E | 112,000 | 80,000 | 32,000 | 52,000 | 12,000 | 24,000 | 116,000 | 36,000 | 8,000 | 80,000 |
| F | 12,000 | 44,000 | 60,000 | 44,000 | 48,000 | 24,000 | 64,000 | 40,000 | 92,000 | 80,000 |
| G | 48,000 | 12,000 | 44,000 | 88,000 | 64,000 | 104,000 | 4,000 | 60,000 | 20,000 | 4,000 |
| H | 24,000 | 64,000 | 80,000 | 32,000 | 84,000 | 52,000 | 4,000 | 60,000 | 88,000 | 32,000 |

cess line total cleaning times are higher, since changeovers for batches of different products are more frequent. Finally, it is worth mentioning that in the feasible schedule reported by Bongers and Bakker (2006), the process line is utilized 90% (that is 9.11% higher than that of the optimal schedule of this work) of the available time, thus resulting to higher production costs (due to higher changeover costs) comparing it with the proposed optimal production schedule.

In general, it should be mentioned that solution strategies that do not optimally integrate the scheduling of all processing stages (i.e., process line, ageing vessels, and packing lines) face the risk of not generating optimal solutions. In this specific (high demand) case study, these solution strategies probably cannot give a feasible schedule. In other words, they may propose solutions where full demand satisfaction is not achieved inside the available production horizon. Manual intervention may still be necessary in order to obtain feasible (i.e., full demand satisfaction), and probably not optimal schedules (Bongers & Bakker, 2006).

**Problem Instances PI.02 to PI.10.** In problem instances PI.02 to PI.10, we consider no overlapping decisions from the previous week schedule. This fact allows us to predefine the relative sequence for products in each packing line, taking into account the sequence-dependent changeover times included into Table 5.1. It can be observed that the optimal relative sequence with respect to the minimization of changeover times in PACK1 is D → C → B → A, and in PACK2 is H → G → F → E. That means that $X_{iD} = X_{BC} = X_{AC} = X_{AB} = 0$ and $X_{iH} = X_{FG} = X_{EG} = X_{EF} = 0$ in PACK1 and PACK2, respectively.

Table 5.4 presents the optimal makespan and the computational characteristics for all problem instances. The proposed MIP formulation in tandem with the proposed solution methodology results in very low computational times for all cases. It is noted that seven of ten problem instances have been solved in less than 2 CPU s. These problem instances have been solved in the first step of the proposed solution method. It is worthwhile to note that zero nodes were explored for these problems. Also, notice that the remaining problem instances, which were infeasible in the first step of the proposed solution method, have been solved in less than a CPU minute in the second step of the solution method. Despite the complexity of the scheduling problems addressed in this work, all problem instances have been solved to optimality with low computational effort.

**Table 5.3:** Process line and packing lines utilization breakdown for PI.01.

| Processing unit | Unit operation | Time (h) | Operation utilization | Total unit utilization |
|---|---|---|---|---|
| PROC | processing | 76.48 | 63.73% | 80.89% |
| | cleaning | 20.58 | 17.15% | |
| | idle | 22.94 | 19.11% | |
| PACK1 | packing | 115.05 | 95.88% | 98.79% |
| | cleaning | 3.50 | 2.92% | |
| | idle | 1.45 | 1.21% | |
| PACK2 | packing | 106.00 | 88.33% | 92.08% |
| | cleaning | 4.50 | 3.75% | |
| | idle | 9.50 | 7.92% | |

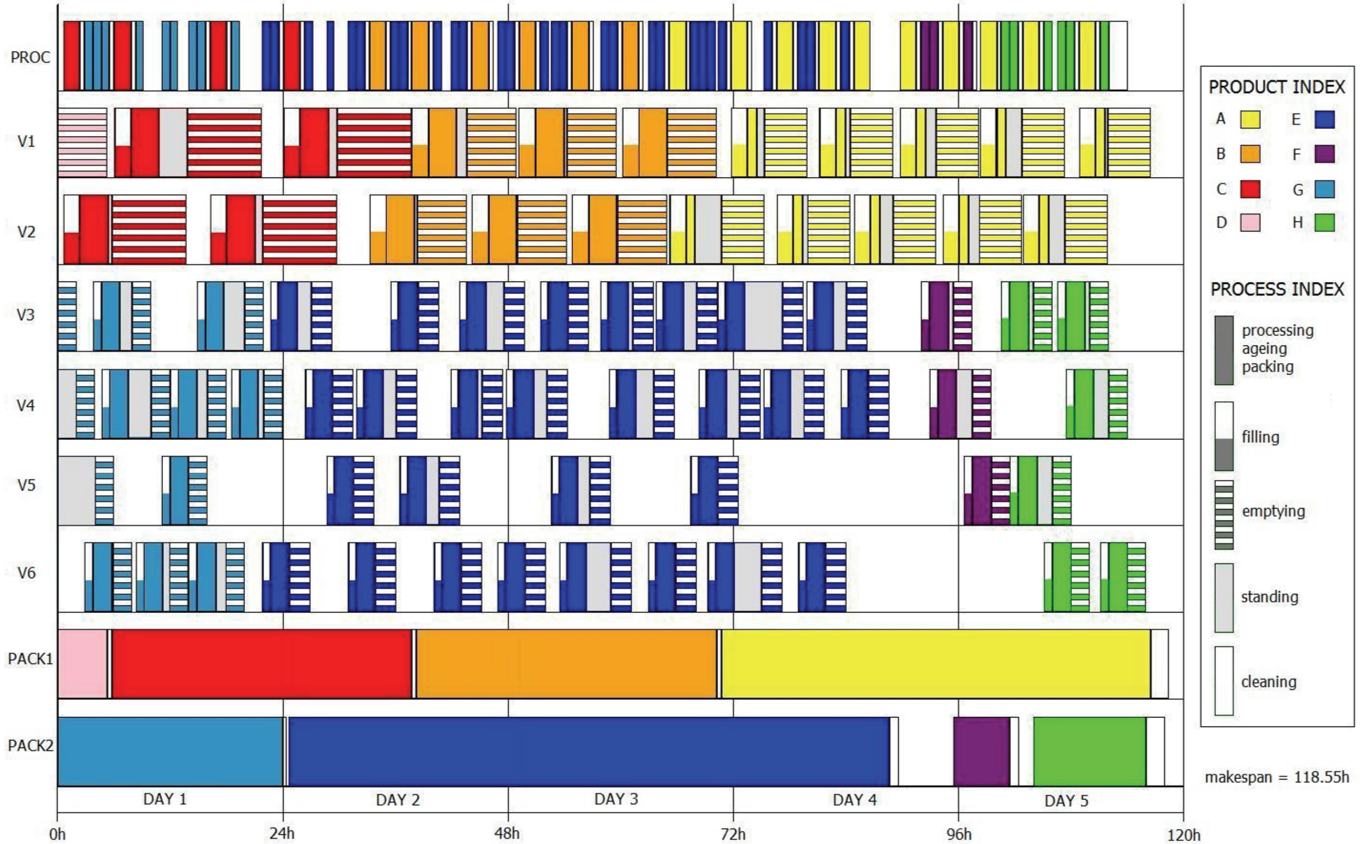



**Figure 5.5:** Optimal production schedule for PI.01 (minimization of makespan).

**Table 5.4:** Makespan and computational features for all problem instances.

| | PI.01 | PI.02 | PI.03 | PI.04 | PI.05 | PI.06 | PI.07 | PI.08 | PI.09 | PI.10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $C_{max}$ | 118.55 | 118.04 | 116.67 | 118.10 | 116.90 | 110.10 | 116.52 | 110.42 | 115.37 | 113.85 |
| CPU s | 1.83 | 0.88 | 23.20 | 51.41 | 1.22 | 15.70 | 0.39 | 0.58 | 0.54 | 0.75 |
| Nodes | 0 | 0 | 467 | 751 | 0 | 510 | 0 | 0 | 0 | 0 |

* makespan includes 2 h of cleaning before shutting down the packing lines.

## 5.8 Concluding Remarks

In this chapter, a novel mathematical programming framework and an efficient solution approach have been proposed for the production scheduling in food process industries similar to the ice-cream production facility studied in details. This model can easily be the core element of a computer-aided advanced scheduling and planning system in order to facilitate decision-making in relevant industrial environments. As the challenging case study reveals, the proposed approach features a salient computational performance due to the efficient modeling approach of the sequencing decisions, and the strong valid integer cuts introduced. However, it should be mentioned that in extremely large-scale scheduling problems, potentially involving hundreds of products, the proposed MIP model may result into huge model sizes difficult to be solved within a reasonable (acceptable) computational time. In that case the proposed mathematical formulation can be easily used as the core MIP model in the MIP-based decomposition strategy described in Chapter 8, in an attempt to make it attractive for the solution of complex large-scale industrial scheduling problems. Finally, it is worth noting that the proposed MIP model is well-suited to a real-life ice-cream production facility, however it could be also used, with minor modifications, in scheduling problems arising in other semicontinuous industries with similar processing features (e.g., yogurt production lines, milk processing plants, etc.).

## 5.9 Nomenclature

**Indices / Sets**

| | |
|---|---|
| $b, b' \in B$ | product batches (batches) |
| $i, i' \in I$ | product orders (products) |
| $j, j' \in J$ | processing units (units) |
| $s \in S$ | processing stages (stages) |

**Subsets**

| | |
|---|---|
| $I_j$ | products $i$ that can be processed in unit $j$ |
| $J_i$ | available units $j$ to process product $i$ |
| $J_s$ | available units $j$ to process stage $s$ |
| $J^{min}$ | packing line that appears the highest lower bound for unit makespan |

**Parameters**

| | |
|---|---|
| $\alpha_j^{min}$ | minimum number of products assigned to packing line $j \in J_{s_3}$ |
| $\beta_i^{min}$ | minimum number of batches for product $i$ |
| $\gamma_{ii'j}$ | sequence-dependent changeover time between orders $i$ and $i'$ in unit $j \in (J_i \cap J_{i'})$ |
| $\gamma_j^{min}$ | minimum sequence-dependent changeover time between two different products in packing line $j \in J_{s_3}$ |
| $\gamma_j^{total}$ | minimum total sequence-dependent changeover time in packing line $j \in J_{s_3}$ |
| $\varepsilon_i^{life}$ | shelf life for product $i$ in ageing vessels |
| $\zeta_i$ | demand for product $i$ |
| $\mu_j^{max}$ | maximum capacity of ageing vessel $j \in J_{s_2}$ |
| $\xi_j^{min}$ | lower bound for the makespan for packing line $j \in J_{s_3}$ |
| $\rho_{ij}$ | processing rate for every product $i$ in the process line $j \in (J_i \cap J_{s_1})$ and the packing lines $j \in (J_i \cap J_{s_3})$ |
| $\tau_i^{age}$ | minimum ageing time for product $i$ |
| $\tau_i^{empt}$ | emptying time of ageing vessel for product $i$ |
| $\tau_i^{fill}$ | filling time of ageing vessel for product $i$ |
| $\phi_j^{min}$ | minimum wait time to begin using packing line $j$ |
| $\omega$ | available scheduling horizon |

**Continuous Variables**

| | |
|---|---|
| $C_{ibs}$ | completion time for stage $s$ of batch $b$ of product $i$ |
| $C_{max}$ | makespan |
| $L_{ibs}$ | starting time for stage $s$ of batch $b$ of product $i$ |
| $W_{ibs}$ | standing (waiting) time for stage $s$ of batch $b$ of product $i$ |

**Binary Variables**

| | |
|---|---|
| $X_{ii'}$ | $= 1$ if product $i$ is processed before product $i'$ (for the ageing vessels and the packing lines) |
| $\bar{X}_{ibi'b'}$ | $= 1$ if batch $b$ of product $i$ is processed before batch $b'$ of product $i'$ (for the process line) |
| $Y_{ibsj}$ | $= 1$ if stage $s$ of batch $b$ of product $i$ is assigned to unit $j$ |

Chapter 6

# Resource-Constrained Production Scheduling in Semicontinuous Process Industries

## 6.1 Introduction

The production planning and scheduling problem in a single production site is usually concerned with meeting fairly specific production requirements. Customer orders, stock imperatives or higher-level supply chain or long-term planning would usually set these. Production planning and scheduling deals with the allocation over time of scarce resources between competing activities to meet these requirements in an efficient fashion. The key components of the resulting resource-constrained planning problem are resources, tasks and time. The resources need not be limited to processing equipment items, but may include material storage equipment, transportation equipment (intra- and inter-plant), operators, utilities (e.g., steam, electricity, cooling water), auxiliary devices and so on. The tasks typically comprise processing operations (e.g., reaction, separation, blending, packing) as well as other activities which change the nature of materials and other resources such as transportation, quality control, cleaning, changeovers, etc. There are both external and internal elements to the time component. The external element arises out of the need to co-ordinate manufacturing and inventory with expected product listings or demands, as well as scheduled raw material receipts and even service outages. The internal element relates to executing the tasks in an appropriate sequence and at right times, taking account of the external time events and resource availabilities. Overall, this arrangement of tasks over time and the assignment of appropriate resources to the tasks in a resource-constrained framework must be performed in an efficient fashion, which implies the optimization, as far as possible, of some objective. Typical objectives include the minimization of cost or the maximization of profit, the maximization of customer satisfaction, and the minimization of deviation from target performance (Shah, 1998).

In this chapter, a general mathematical programming approach is presented for the resource-constrained production problem in semicontinuous processes. The work has been motivated by a challenging problem in food processing industries related to yogurt production lines. In this problem, labor (i.e., the number of available workers) constitutes the limited resource constraint. The proposed mathematical approach for the resource-constrained production problem can also cope with unexpected events such as workers absence, and products orders modifications.

## 6.2 Resource-Constrained Planning and Scheduling Aspects in Dairy Processing Industries

In the food processing industries, quite often homogeneous products have to be packed (e.g., milk, yogurt). For an unpacked product a variety of packing materials (cups, bins, etc) and packing sizes are available. Generally the packing lines are used for various products in one type of packing material and various packing sizes. These possibilities make the scheduling of the packing lines rather complex (Van Dam, Gaalman, & Sierksma, 1993). In recent years, under market pressure the number of products in dairy plants has been increased, the order sizes have been reduced and the delivery times have been shortened. This has caused augmented scheduling tasks and usually the scheduling system supporting these tasks have not followed these changes sufficiently.

Of particular interest in the dairy industry is the production of yogurt. In order to make yogurt, milk has to be pasteurized and fermented to create batches of *white mass*, which are then placed in containers of various sizes, sometimes with fruit or other ingredients. A fresh dairy plant may make more than 10 types of white mass that differ, for example, in percentage of cream and type of yeast. A typical plant produces more than 100 final products on several packing lines corresponding to different flavors, sizes, percentage of fats (Nakhla, 1995). Between pasteurization, fermentation and storage, as many as 100 tanks can be used. And because the plant produces food, extra care must be taken to ensure high standards of sanitation, control of allergens, batch traceability and maximum product freshness. Figure 6.1 illustrates the main processing steps for producing stirred yogurt.

The plant must closely coordinate the two primary production steps: the transformation of raw materials – such as milk, milk powders and yeast – into white mass and the filling and packing of final products. The right amount of white mass has to be appropriately scheduled, and it has to be used as quickly as possible. Operational scheduling challenges include:

- Deciding which and how much white mass to produce in each tank given the available connections to the filling and packing lines.

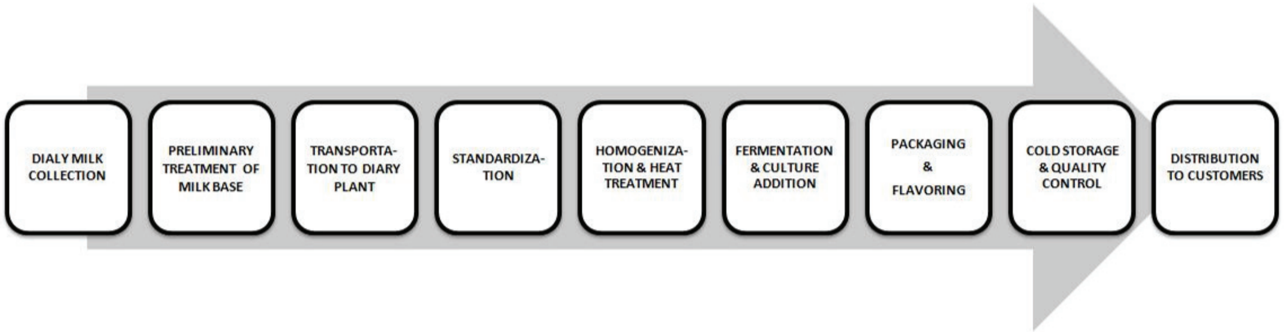



**Figure 6.1:** Yogurt production process (except from set yogurt).

- Finding the best time to clean the tanks and the filling lines given health and nutrition labeling requirements, and cleaning equipment availability.
- Synchronizing material consumption with white mass availability and freshness.
- Respecting batching policies for compliance with traceability regulations.
- Determining an optimal schedule for labor resources.
- Maintaining a steady supply of finished goods within a minimum and maximum inventory corridor.
- Optimal production rescheduling under unexpected events such as modifications in product orders.

To these production challenges must be added those for high demand variability. Fresh dairy products are consumer goods with significant promotional marketing and a steady introduction of new products. Demand is often uncertain. New products may steal sales from old products or simply contribute to market share, and marketing campaigns can result in sales that are higher or lower than forecasted. And in the fresh dairy industry, the challenges associated with demand variability are compounded by the short shelf life of the finished products and relatively long production lead times - three to four days from milk pasteurization to final product. Poor production plans lead to both product waste and stock shortages, making agility and regular rescheduling critical.

The extension of the range of products in a typical dairy processing industry makes scheduling particularly difficult in order to meet mix in demand. The existence of different production lines which can carry out the same operations but which are distinguished by different production rates and the cleaning-in-place operations required between different products being processed in the same equipment makes the problem additional complex. This is further complicated by the existence of limited and shared resources (e.g., labor, utilities, etc.). There are two key shared resources in the process: operators and utilities. Different categories of operators may be available in each shift to perform the various duties involved in plant operation. If an appropriate number of operators is not available in a particularly category needed to start a task associated with a product, the initiation of this task is delayed until the required number becomes available. Alternatively an external production service should be used if very tight product delivery times exist (Nakhla, 1995).

## 6.3 Problem Statement

In this chapter, the resource-constrained production planning problem of a multiproduct semicontinuous dairy plant is addressed. More specifically, the production line under consideration produces set, stirred or flavored yogurt. It is noted that flavored yogurt is stirred yogurt with additional fruit (or other type) flavor. Thus, flavored yogurt production should pass through fruit-mixer equipments in order

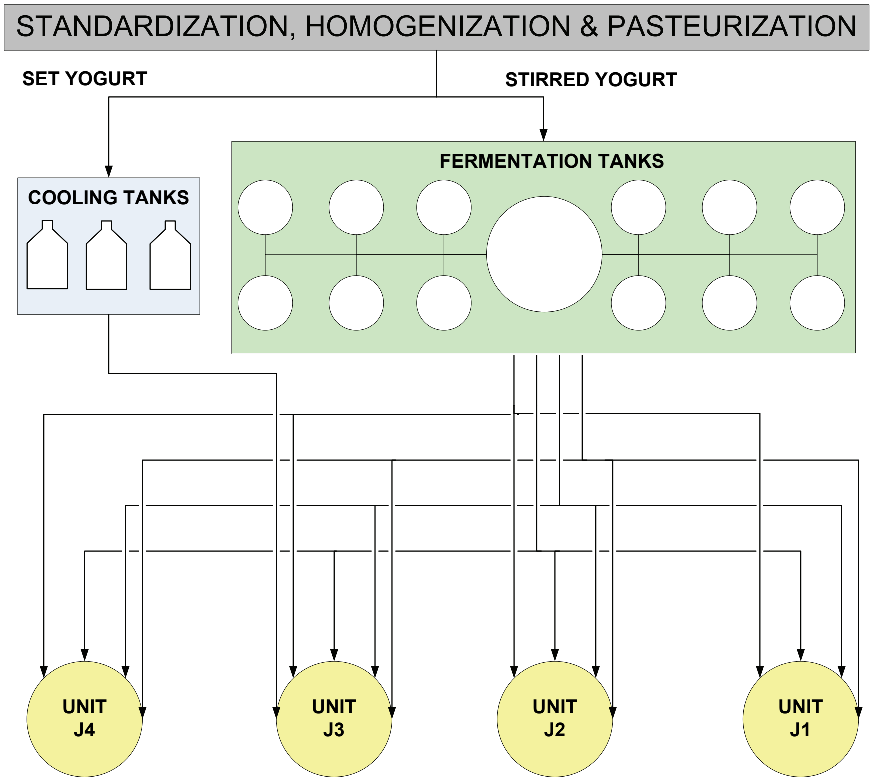



**Figure 6.2:** Yogurt production line layout.

to perform the addition and the mixing of fruit substances. Each packing unit has its own fruit-mixer. The yogurt production line consists of: a set of cooling tanks (set yogurt), a set of fermentation tanks (stirred and flavored yogurt), and 4 packing machines. The main yogurt production line layout is illustrated in Figure 6.2. It is clear that packing units operate in parallel. Moreover, they share common resources, such as manpower.

The problem addressed here is formally defined in terms of the following items:

(i) A known planning horizon divided into a set of periods $n \in N$.

(ii) A set of processing units $j \in J$ with available production time in period $n$ equal to $\omega_{jn}$.

(iii) A set of products $p \in P$ with specific production targets $\zeta_{pn}^{cup}$, inventory costs $\xi_{pn}$, production rates $\rho_{pj}$, minimum processing runs $\pi_{pjn}^{min}$, fixed $v_{jn}$ and variable operating costs $\theta_{pjn}$, cup weights $\eta_{p}^{cup}$, and external production cost

$\psi_{pn}$. $P_j$ is the subset of products that can be assigned to unit $j$, and $J_p$ is the subset of units that can produce product $p$.

(iv) A set of batch recipes $r \in R$ with minimum preparation time $\tau_r$, preparation cost $\chi_{rn}$, and minimum and maximum production capacity $\mu_{rn}^{min}$ and $\mu_{rn}^{max}$, respectively. The subset of products that come from recipe $r$ is denoted by $P_r$.

(v) A set of product families or simply *families* $f \in F$ wherein all products are grouped into; $F_j$ is the subset of families that can be assigned to unit $j$, and $J_f$ is the subset of units that can process family $f$, while the subset of products in family $f$ is denoted by $P_f$.

(vi) A set of renewable resources $k \in K$. Parameter $\varepsilon_{kfj}$ denotes the requirement of resource $k$ for processing family $f$ in unit $j$, and $E_{kn}^{max}$ is the maximum capacity for resource $k$ in period $n$.

(vii) A (sequence-dependent) changeover operation is required in each processing unit whenever the production is changed between two different families; the required changeover time is $\gamma_{ff'j}$, while the changeover cost is $\phi_{ff'jn}$.

(viii) A (sequence-independent) setup operation is required whenever a product $p$ is assigned to a processing unit $j$; the setup time is $\delta_{pj}$.

(ix) Forbidden processing sequences for families.

The key decision variables are:

(i) the allocation of products to processing units $Y_{pjn}^{P}$;

(ii) the sequencing of families in every processing line $X_{ff'jn}$ and $\bar{X}_{ff'jn}$;

(iii) the amount of product $p$ produced in unit $j$ ($Q_{pjn}$), and the inventory level of product $p$ at the end of planning period $n$ ($I_{pn}$);

(iv) the starting ($S_{fjn}$) and completion times ($C_{fjn}$) for every family.

So that an economic objective function typically representing total production costs is optimized.

## 6.4 Conceptual Model Design

Production planning in semicontinuous processing plants typically deals with a large number of products. Fortunately, many products appear similar characteristics. Therefore, products that share the same processing features could be treated as a product family group (family). Thus, the production planning problem under question could be *partially* focused on product families rather than on each product separately, following a similar modeling concept to Chapter 4. The definition of product families significantly reduces the size of the underlying mathematical model and, thus, the necessary computational effort without sacrificing any feasibility constraint. In the proposed approach products belong to the same family *if and only if*: (i) they come from the same batch recipe (e.g., fermentation recipe), (ii) they require the same labor resources, and (iii) there is no need for changeover operations among them.

When changing the production between two products that are not based on the same recipe, it is always necessary to perform changeover cleaning and/or sterilizing operations. In dairy plants, a "natural" sequence of products often exists (e.g., from the lower taste to the stronger or from the brighter color to the darker) thus the relative sequence of products within a family is usually known a priory. Therefore, when changing the production between two products of the same family, cleaning and sterilizing operations are not needed. Hence, not only the relative sequence of products, belonging to the same family, may be known but also the relative sequence of families in each processing line. In that case, different families are enumerated according to their relative position within the production day.

It should be noted that (sequence-independent) setup times, mainly depending on the cup size or product type changes, (among products of the same family) may exist and they can be treated appropriately by the proposed mathematical model. Finally, it is worth mentioning that a salient feature of the proposed modeling approach is that it allows products that belong to the same family to have different: (i) processing rates, (ii) setup times/costs, (iii) minimum and maximum production runs, (iv) operating costs, and (v) inventory costs.

## 6.5 Mathematical Formulation

In the proposed mathematical framework, constraints have been grouped according to the type of decision (assignment, timing, sequencing, etc.) upon which they are imposed on. It should be emphasized that the proposed model is a crossbreed between a continuous and a discrete time representation model. More specifically, the planning horizon of interest is discretized into a number of time periods each having the duration of one production day. Then, operations within the same day are modeled using a continuous time representation (see Figure 6.3). Mass balance is realized at the end of each production day. To facilitate the presentation of the MIP model, we use uppercase Latin letters for optimization variables and sets, and lowercase Greek letters for parameters.

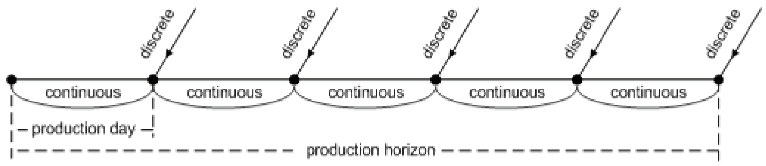



**Figure 6.3:** Time representation.

**Products Lot-Sizing and Allocation Constraints.** Lower and upper bounds on the produced amounts of product $p$ are imposed by:

$$\pi^{min}_{pjn} Y^P_{pjn} \leq Q_{pjn} \leq \pi^{max}_{pjn} Y^P_{pjn} \qquad \forall p, j \in J_p, n \tag{6.1}$$

Tighter maximum produced quantities can be estimated by:

$$\pi^{max}_{pjn} = \begin{cases} \sum\limits_{n' \geq n}^{N} \zeta_{pn'} & \text{if } \sum\limits_{n' \geq n}^{N} \zeta_{pn'} < (\Lambda_{jn} - \underset{r \in R_p}{\tau_r})\rho_{pj}, \\ (\Lambda_{jn} - \underset{r \in R_p}{\tau_r})\rho_{pj} & \text{if } \sum\limits_{n' \geq n}^{N} \zeta_{pn'} \geq (\Lambda_{jn} - \underset{r \in R_p}{\tau_r})\rho_{pj}. \end{cases}$$

where $\Lambda_{jn} = \omega_{jn} - \alpha_{jn} - \beta_{jn}$, and production targets $\zeta_{pn}$ for product $p$ are given by:

$$\zeta_{pn} = \zeta^{cup}_{pn} \eta^{cup}_{p} \qquad \forall p, n$$

Notice that $\zeta^{cup}_{pn}$ is provided by the logistics department of the company and usually reflects production targets which are based on actual products demands as well as on forecasts.

**Family Processing Time Definition.** Sequencing and timing decisions need to be taken regarding families $f$ rather than products $p$, because products that belong to the same family ($p \in P_f$) do not require changeovers among them. However, it should be noted that setup times, $\delta_{pj}$, may exist. In order to define sequencing and timing decisions for families, the definition of family processing time is introduced as follows:

$$T_{fjn} = \sum_{p \in P_f} \left(\frac{Q_{pjn}}{\rho_{pj}} + \delta_{pj} Y^P_{pjn}\right) \qquad \forall f, j \in J_f, n \tag{6.2}$$

In the proposed approach processing rates $\rho_{pj}$ are considered fixed as potential fluctuations may provoke quality problems (Soman, Donk, & Gaalmann, 2004).

**Families Allocation Constraints.** A family $f$ is assigned to a processing unit $j$ in a production day $n$ if at least one product $p \in P_f$, that belongs to this family, is processed on this unit during the same production day:

$$Y_{fjn} \geq Y^P_{pjn} \qquad \forall f, p \in P_f, j \in J_f, n \tag{6.3}$$

**Families Sequencing and Timing Constraints.** Constraint sets (6.4) and (6.5) state that if a family $f$ is allocated to processing unit $j$ in period $n$, (i.e., $Y_{fjn} = 1$) at most one family $f'$ is processed before and/or after it, respectively.

$$\sum_{f' \neq f, f' \in F_j} X_{f'fjn} \leq Y_{fjn} \qquad \forall f, j \in J_f, n \tag{6.4}$$

$$\sum_{f' \neq f, f' \in F_j} X_{ff'jn} \leq Y_{fjn} \qquad \forall f, j \in J_f, n \tag{6.5}$$

The total number of active sequencing binary variables $X_{ff'jn}$ plus the unit utilization binary variable $V_{jn}$ should be equal to the total number of active allocation binary variables $Y_{fjn}$ in a processing unit $j$ at period $n$, according to constraint set (6.6). For instance, if three families are assigned to a unit $j$ then two sequencing variables will be active.

$$\sum_{f\in F_j}\sum_{f'\neq f, f'\in F_j} X_{ff'jn} + V_{jn} = \sum_{f\in F_j} Y_{fjn} \qquad \forall j, n \tag{6.6}$$

Constraint set (6.7) ensures that the processing unit $j$ is used in period $n$, (i.e., $V_{jn}$=1) if at least one family $f$ is assigned to it in this period (i.e., $Y_{fjn} = 1$). Note that no lower bound on the binary variable $V_{jn}$ is necessary as far as a cost term (related to the unit utilization), is included into the objective function, thus enforcing $V_{jn}$ to zero.

$$V_{jn} \geq Y_{fjn} \qquad \forall f, j \in J_f, n \tag{6.7}$$

Constraint set (6.8) states that the starting time of a family $f'$, $S_{f'jn}$, that follows another family $f$ in a processing line $j$ in period $n$, (i.e., $X_{ff'jn} = 1$) is greater than the completion time of family $f$, $C_{fjn}$, plus the necessary changeover time $\gamma_{ff'j}$ between these families. Note that the big-M parameter $M_{jn}$ can be set equal to $(\omega_{jn} - \beta_{jn})$, where $\omega_{jn}$ is the available production time horizon and $\beta_{jn}$ corresponds to the daily plant shutdown time.

$$C_{fjn} + \gamma_{ff'j} \leq S_{f'jn} + M_{jn}(1 - X_{ff'jn}) \quad \forall f, f' \neq f, j \in (J_f \cap J_{f'}), n \tag{6.8}$$

Obviously,

$$S_{fjn} = C_{fjn} - T_{fjn} \qquad \forall f, j \in J_f, n \tag{6.9}$$

**Families Completion Times Lower and Upper Bounds.** Constraints (6.10) and (6.11) impose a lower and upper bound on each family completion time, $C_{fjn}$, respectively. More specifically, the completion time has to be greater than the daily plant setup time, $\alpha_{jn}$, plus the minimum time $\tau_r$ for preparing the batch recipe (e.g., fermentation recipe) $r$, plus the processing time, $T_{fjn}$, and the changeover time, $\gamma_{f'fj}$, for changing the production to family $f'$. An additional unit preparation time $o_{jn}$ is also taken into account. This time stands for the additional preparation time of a processing unit $j$ due to potential maintenance or other technical reasons. Additionally, the release batch recipe time $\sigma_{rn}$ is also considered. In order to commence the production of a batch recipe $r$ all recipes ingredients need to be present. Otherwise, the production of the batch recipe $r$ will be postponed until the arrival of its missing substances.

$$\begin{aligned} C_{fjn} \geq{} & (\alpha_{jn} + max[o_{jn}, \sigma_{rn}] + \tau_r) Y_{fjn} + T_{fjn} \\ & + \sum_{f'\neq f, f'\in F_j} \gamma_{f'fj} X_{f'fjn} \qquad \forall f, r \in R_f, j \in J_f, n \end{aligned} \tag{6.10}$$

Constraint set (6.11) ensures that the completion time of each family is smaller than the daily production time horizon $\omega_{jn}$ minus the daily plant shutdown time

$\beta_{jn}$. Production line shutdown is realized on a daily basis, as a typical production policy to guarantee high quality of final products and to comply with hygienic standards.

$$C_{fjn} \leq (\omega_{jn} - \beta_{jn}) Y_{fjn} \qquad \forall f, j \in J_f, n \tag{6.11}$$

**Batch Recipe Stage Constraints.** Batch recipe stage (e.g., fermentation and pasteurization) constraints must be included into the mathematical model in order to guarantee a feasible production plan in yogurt production lines. The cumulative produced quantity of products $p \in P_r$ should be greater than the minimum produced batch recipe amount (e.g., in the pasteurization and fermentation stages) $\mu_{rn}^{min}$ and lower than the maximum production capacity $\mu_{rn}^{max}$:

$$\mu_{rn}^{min} Y_{rn}^R \leq \sum_{p \in P_r} \sum_{j \in J_p} Q_{pjn} \leq \mu_{rn}^{max} Y_{rn}^R \qquad \forall r, n \tag{6.12}$$

Constraint set (6.13) ensures that a batch recipe $r$ is produced in period $n$, (i.e., $Y_{rn}^R = 1$), if at least one family $f \in F_r$ is processed on a processing unit $j$ in the same period $n$ (i.e., $Y_{fjn} = 1$) .

$$Y_{rn}^R \geq \sum_{j \in J_f} Y_{fjn} \qquad \forall r, f \in F_r, n \tag{6.13}$$

**Tightening Constraints.** In order to reduce the computational effort, constraints (6.14) can further tighten the linear relaxation of the proposed mathematical model by imposing an upper bound on the total processing time for every processing line $j$ in each period $n$.

$$\begin{aligned}\sum_{f \in F_j} T_{fjn} \leq & \,(\omega_{jn} - \alpha_{jn} - \beta_{jn} - \min_{r \in R_j}[\tau_r]) V_{jn} \\ & - \sum_{f \in F_j} \sum_{f' \neq f, f' \in F_j} \gamma_{ff'j} X_{ff'jn} \qquad \forall j, n\end{aligned} \tag{6.14}$$

Note that by incorporating constraint set (6.14) into the mathematical formulation, constraint set (6.7) can be omitted, thus further reducing the model size.

**Products Mass Balance Constraints.** The total quantity of product $p$ produced on the plant (internal production) in period $n$, $Q_{pn}^{int}$, is given by:

$$Q_{pn}^{int} = \sum_{j \in J_p} Q_{pjn} \qquad \forall p, n \tag{6.15}$$

At this point, it is worth pointing out that demand satisfaction is of great importance in the dairy industry. Inability to satisfy customer demand on time may result to losses of the market share, competitive advantage, increase customers disappointment, etc. Therefore, full demand satisfaction is desired. Constraints

(6.16) enforce full demand satisfaction. Figure 6.4 presents a network representation of the production planning problem under question. The inventory $I_{pn}$ of product $p$ is the summation of the previous period inventory, $I_{pn-1}$, plus the total internal, $Q_{pn}^{int}$, and external production, $Q_{pn}^{ext}$, minus the production target, $\zeta_{pn}$, in the current period $n$:

$$I_{pn} = I_{pn-1} + Q_{pn}^{int} + Q_{pn}^{ext} - \zeta_{pn} \qquad \forall p, n \tag{6.16}$$

External production usually expresses production targets that exceed the production capacity of the dairy plant. It can be realized to an affiliated production facility, if one exists, otherwise it represents the unsatisfied demand. A high penalty cost for external production will enforce the MIP model to generate solutions that comply with full demand satisfaction by internal production. Obviously, the external production of product $p$ in production day $n$ cannot be greater than the product demand at the same production day:

$$Q_{pn}^{ext} \leq \zeta_{pn} \qquad \forall p, n \tag{6.17}$$

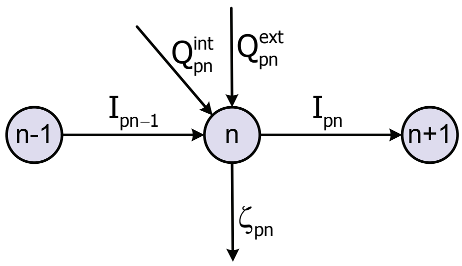



**Figure 6.4:** Network representation of the production planning problem.

Constraint set (6.18) is added to the proposed MIP model if product safety stocks are desired. If product-dependent storage limitations exist, constraint set (6.19) is used. Otherwise, constraint set (6.20) is included to account for the total plant storage capacity.

$$I_{pn} \geq \text{product safety stock} \qquad \forall p, n \tag{6.18}$$

$$I_{pn} \leq \text{product storage capacity} \qquad \forall p, n \tag{6.19}$$

$$\sum_{p} I_{pn} \leq \text{total plant storage capacity} \qquad \forall n \tag{6.20}$$

**Objective Function.** The objective function to be minimized is the total cost including several factors such as: (i) inventory costs, (ii) operating costs, (iii) batch recipes preparation costs, (iv) unit utilization costs, (v) families changeover costs, and (vi) external production costs, as follows:

$$\begin{aligned}\min \quad & \sum_{p}\sum_{n}\xi_{pn} I_{pn} + \sum_{p}\sum_{j\in J_p}\sum_{n}\frac{\theta_{pjn}}{\rho_{pj}} Q_{pjn} + \sum_{r}\sum_{n}\chi_{rn} Y^{R}_{rn} \\ & + \sum_{j}\sum_{n}\nu_{jn} V_{jn} + \sum_{f}\sum_{f'\neq f}\sum_{j\in(J_f\cap J_{f'})}\sum_{n}\phi_{ff'jn} X_{ff'jn} \\ & + \sum_{p}\sum_{n}\psi_{pn} Q^{ext}_{pn}\end{aligned} \tag{6.21}$$

In a dairy plant, final yogurt products are kept at low temperatures, thus resulting to a significant inventory cost (mainly due to high energy requirements), which should be considered in the optimization procedure. Moreover, inventory costs are include shelf life issues. Roughly speaking, the lower the shelf life of a product the higher its inventory cost. Operating costs mainly include labor and energy costs plus costs due to material losses. The fermentation recipe cost account for all costs associated with the preparation of each fermentation recipe. The unit utilization cost basically stands for the shutdown cleaning operation cost plus the initial unit setup cost. Changeover costs correspond to cleaning and/or sterilization operations. Finally, external production costs reflect the penalty cost of producing the requested production targets to an affiliated production facility. The nature of this cost is more qualitative than quantitative. A high external production cost will enforce demand satisfaction by internal production. In this case, external production will appear only if the production targets are higher than the production capacity of the plant. In other words, a full demand satisfaction by internal production is indirectly favored. It is worthy mentioning that since full demand satisfaction is imposed, the minimization of total costs is identical to the maximization of total profit.

### 6.5.1 Extension to renewable resources constraints

In most industrial environments resource limitations often constitute a crucial part of the production planning problem. By neglecting potential resource constraints in the optimization procedure, there is no guarantee that a feasible production plan will not be obtained.

Roughly speaking, resources could be mainly classified into *non-renewable* and *renewable*. Non-renewable resources do not recover their capacity after the completion of the tasks that consumed them. For instance, raw materials and intermediate products can be considered as non-renewable resources. On the other hand, renewable resources recover their capacity after the completion of the tasks that used them. Renewable resources like manpower are called discrete renewable resources, while resources such as utilities (e.g., electricity, vapor, cooling water, etc.) are usually referred as continuous renewable resources.

In the dairy processing industry, the available manpower usually constitutes the major resource limitation. This is the case in the plant under consideration, where a limited number of employees is available during each production day. The modeling approach of labor resources in the present work follows similar modeling concepts to the recent contribution of Marchetti and Cerdá (2009b).

**Basic Conditions for Modeling Resources Constraints.** By definition, a family $f'$ that is overlapping the starting time of family $f$ must satisfy the following conditions:

(A) It should demand some resource $k$ also required by family $f$.

(B) It is assigned to a processing unit different from the one that is allocated to family $f$.

(C) It starts before or exactly at the time that family $f$ starts being processed (i.e., $S_{f'j'n} \leq S_{fjn}$).

(D) It should end after the starting time of family $f$ (i.e., $C_{f'j'n} > S_{fjn}$).

An illustrative example for the basic overlapping conditions is shown in Figure 6.5. Note that family $f$: (i) is overlapped by family $f'$, and (ii) is overlapping family $f''$.

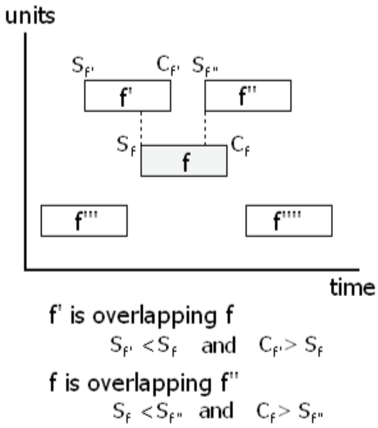



**Figure 6.5:** Illustrative example for overlapping conditions.

**Sequencing Constraints for Families Assigned to Different Units.** Constraints (6.22) to (6.24) are included into the MIP model to ensure the families relative sequencing related to their starting times; condition (C). Global sequencing binary variables $\bar{X}_{f'j'fjn}$ are introduced for each pair of families $f'$ and $f$ that are assigned to different units. When family $f'$, which is allocated to processing unit $j'$, starts before family $f$, which is allocated to processing unit $j \neq j'$, the binary variable, $\bar{X}_{f'j'fjn}$, is active (i.e., $\bar{X}_{f'j'fjn} = 1$). A very small number $\lambda$ is added in constraint set (6.24) to effectively cope with the case when two families $f$ and $f'$ start at the same time point. In this case, it is assumed that family $f' < f$ starts slightly before family $f$. In other words, the family with the lower index begins first. Note that, if both or one of the families $f$ and $f'$ is not assigned to unit $j$ or unit $j' \neq j$, respectively, (i.e., $Y_{fjn} = 0$ and/or $Y_{f'j'n} = 0$) constraints (6.22) to (6.24) become redundant.

$$S_{f'j'n} - S_{fjn} \leq M_{jn}(1 - \bar{X}_{f'j'fjn}) + M_{jn}(2 - Y_{f'j'n} - Y_{fjn}) \quad \forall f, f', j \in J_f, j' \in J_{f'}, n : j' \neq j \tag{6.22}$$

$$S_{fjn} - S_{f'j'n} \leq M_{jn}\bar{X}_{f'j'fjn} + M_{jn}(2 - Y_{f'j'n} - Y_{fjn}) \quad \forall f, f' \geq f, j \in J_f, j' \in J_{f'}, n : j' \neq j \tag{6.23}$$

$$S_{fjn} - S_{f'j'n} + \lambda \leq M_{jn}\bar{X}_{f'j'fjn} + M_{jn}(2 - Y_{f'j'n} - Y_{fjn}) \quad \forall f, f' < f, j \in J_f, j' \in J_{f'}, n : j' \neq j \tag{6.24}$$

**Families Overlapping Constraints.** In order to derive the mathematical expression for the overlapping condition (D), an auxiliary overlapping binary variable, $Z_{f'j'fjn}$, is defined. This variable is active (i.e., $Z_{f'j'fjn} = 1$) whenever family $f'$ is completed after the starting time of family $f$, as constraint set (6.25) states. If both or one of the families $f$ and $f'$ is not assigned to unit $j$ or unit $j' \neq j$, respectively, (i.e., $Y_{fjn} = 0$ and/or $Y_{f'j'n} = 0$) constraint set (6.25) becomes redundant.

$$C_{f'j'n} - S_{fjn} \leq M_{jn} Z_{f'j'fjn} + M_{jn}(2 - Y_{f'j'n} - Y_{fjn}) \quad \forall f, f', j \in J_f, j' \in J_{f'}, n : j' \neq j \tag{6.25}$$

Whenever the RHS of constraint set (6.25) is positive (i.e., $C_{f'j'n} - S_{fjn} > 0$), condition (D) is satisfied. Note that, in this case, the auxiliary overlapping binary variable $Z_{f'j'fjn}$ is enforced to take the value of 1.

It can be easily proven that if two families $f$ and $f'$ are running in parallel the Boolean condition $(C_{f'j'n} > S_{fjn}) \wedge (C_{fjn} > S_{f'j'n})$ is satisfied and, therefore, $Z_{fjf'j'n} + Z_{f'j'fjn} = 2$. However, it is important to keep in mind that not every family $f'$ satisfying the necessary condition $Z_{fjf'j'n} + Z_{f'j'fjn} = 2$ is an overlapping family, but only those families running at the starting time $S_{fjn}$. Given the condition $Z_{fjf'j'n} + Z_{f'j'fjn} = 2$, a global sequencing variable $\bar{X}_{f'j'fjn}$ is required to decide which family ($f$ or $f'$) overlaps the other one. If $Z_{fjf'j'n} + Z_{f'j'fjn} = 2$ and $\bar{X}_{f'j'fjn} = 1$, then family $f'$ is overlapping family $f$; according to constraints (6.26) and the overlapping binary variable $W_{f'j'fjn}$ takes the value of 1 in this case.

$$W_{f'j'fjn} \geq Z_{f'j'fjn} + \bar{X}_{f'j'fjn} - Y_{f'j'n} \quad \forall f, f', j \in J_f, j' \in J_{f'}, n : j' \neq j \tag{6.26}$$

**Families Resources Capacity.** Constraint set (6.27) does not allow family $f$ to start being processed if the maximum resource capacity $E_{kn}^{max}$ is reached. Thus, resource overloads, which result in infeasible solutions, are avoided.

$$\varepsilon_{kfj} Y_{fjn} + \sum_{f' \in F_k} \sum_{j' \neq j, j' \in J_{f'}} \varepsilon_{kf'j'} W_{f'j'fjn} \leq E_{kn}^{max} \qquad \forall k, f \in F_k, j \in J_f, n \tag{6.27}$$

The proposed modeling approach is able to tackle problems of multiple renewable resources constraints; either discrete (such as manpower) or continuous (such as utilities) types. Note that unit-dependent resource requirements can be also considered explicitly by the proposed set of resource constraints.

**Modified Objective Function.** To express the effect of labor resources, a non quantitative managerial term is added in the overall objective function. This term expresses the number of employees that are working simultaneously. It is preferable to keep this number as low as possible in order to use the remaining manpower in other tasks or to preserve them in case of the occurrence of an unexpected event. More importantly, the less the manpower used, the lower the possibilities for manpower errors and bad co-ordination and the higher the production flexibility. Finally, an auxiliary penalty term is also introduced to explicitly take account of potential resource constraints.

$$\begin{aligned} \min \quad & \sum_p \sum_n \xi_{pn} St_{pn} + \sum_p \sum_{j \in J_p} \sum_n \frac{\theta_{pjn}}{\rho_{pj}} Q_{pjn} + \sum_r \sum_n \chi_{rn} Y_{rn}^R \\ & + \sum_j \sum_n \nu_{jn} V_{jn} + \sum_f \sum_{f' \neq f} \sum_{j \in (J_f \cap J_{f'})} \sum_n \phi_{ff'jn} X_{ff'jn} \\ & + \sum_p \sum_n \psi_{pn} Q_{pn}^{ext} + \sum_f \sum_{f'} \sum_{j \in J_f} \sum_{j' \neq j, j' \in J_{f'}} \sum_n W_{f'j'fjn} \\ & + \sum_f \sum_{f'} \sum_{j \in J_f} \sum_{j' \neq j, j' \in J_{f'}} \sum_n Z_{f'j'fjn} \end{aligned} \tag{6.28}$$

## 6.6 Industrial Case Studies

In this section, a number of complex real-world production planning problems in the yogurt production line of the KRI-KRI diary production facility, located in Northern Greece, are considered. The facility under study consists of four packing units (J1 – J4), working in parallel and producing a total of 93 yogurt products which are grouped into 23 families (F01 – F23). Real data have been slightly modified due to confidentiality issues.

The production time horizon in the underlying yogurt production facilities is usually one week (Nakhla, 1995). The regular production is performed from Monday to Friday and overtime may be permitted on Sunday and/or on Saturday (see

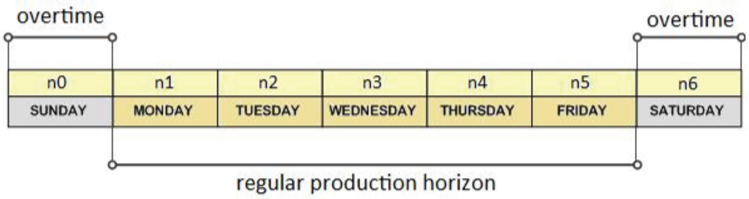



**Figure 6.6:** Production planning horizon.

Figure 6.6). The duration of each production period is 24 h. The daily scheduled cleaning operations of the plant $\beta_{jn}$ last 2 h. The total plant setup time $\alpha_{jn}$ is 3 h representing the necessary time for the completion of production stages before the fermentation stage (pasteurization, homogenization, etc.). Product demand data represent packing stage production targets have been provided from the logistics department of the plant. Demands quantities and due dates are based on product orders from Sunday to Tuesday of the following week as well as on forecasts. Demand due dates are given for packed final products (subtracting the necessary final cold storage and quality control time, which varies between 2 and 5 days).

The main processing data for final products and families (i.e., classification of products to families, cup weights, inventory costs, minimum production runs, packing rates, and families changeover times and costs) can be found in Appendix C. Table 6.1 provides: (i) the main data for each fermentation recipe including the minimum fermentation time (stirred yogurt) or the minimum cooling time (set yogurt) for preparing each fermentation recipe $r$, (ii) the recipe preparation cost and (iii) the set of product families $f \in F_r$ that share the same fermentation recipe. Table 6.2 illustrates the families relative sequence in a production day. The minimum produced quantity of any fermentation recipe $\mu_r^{min}$, due to pasteurization and fermentation stage operability issues, is 1,200 kg. The minimum packing time for any family is equal to 0.5 h. Manpower requirements for each family are shown in Table 6.3.

The variable operating cost, $\theta_{pjn}$, expresses mainly labor and utilities costs of the packing stage. This is equal to 1,000 €/h for any packing unit, during a regular

**Table 6.1:** Main data for recipes.

| Recipe | Process Type | Preparation Time (h) | Cost (€) | Families |
|---|---|---|---|---|
| R01 | fermentation | 4.75 | 545 | F04, F11 |
| R02 | fermentation | 4.50 | 540 | F05, F12 |
| R03 | fermentation | 8.25 | 565 | F13 |
| R04 | fermentation | 7.75 | 555 | F14, F15 |
| R05 | fermentation | 5.25 | 525 | F20 |
| R06 | fermentation | 7.25 | 565 | F19, F21, F22 |
| R07 | fermentation | 8.75 | 625 | F06, F07, F18 |
| R08 | cooling | 1.50 | 505 | F01 |
| R09 | cooling | 1.50 | 510 | F02 |
| R10 | cooling | 1.50 | 515 | F03 |
| R11 | fermentation | 8.75 | 625 | F08, F09, F10, F16, F17 |
| R12 | fermentation | 8.75 | 600 | F23 |

**Table 6.2:** Families relative sequences in a production day per packing line.

| Unit | Families Relative Sequence |
|---|---|
| J1 | F20 ➩ F21 ➩ F22 |
| J2 | F12 ➩ F11 ➩ F19 ➩ F18 ➩ F13 ➩ F14 ➩ F15 ➩ F16 ➩ F17 |
| J3 | F01 ➩ F02 ➩ F03 ➩ F05 ➩ F04 ➩ F08 ➩ F09 ➩ F10 ➩ F06 ➩ F07 |
| J4 | F08 ➩ F09 ➩ F10 ➩ F06 ➩ F23 |

production day (week days) and 10,000 €/h in overtime periods (Saturday and Sunday); this actually reflects the industrial policy to keep the production facility closed during weekend. Moreover, the cost for the production of a fermentation recipe in overtime periods (weekend) is taken twice the cost for producing it into a regular production period (week days). The fixed utilization and cleaning packing unit cost, $v^{unit}$, for each packing line is 1,000 € in a regular production day and 5,000 € in overtime periods. In order to avoid the undesirable case of external production, a high external production penalty cost $\psi_{pn}$ equal to 30 €/kg is imposed. The plant employs maximum 12 workers.

All case studies have been solved to optimality in an Intel Core 2 Quad 2.84 GHz with 3.5 GB RAM using CPLEX 11 under standard configurations via a GAMS 22.8 interface (Rosenthal, 2010). The detailed production plan for every case study is reported in Appendix D.

**Case Study I.** Production targets for this case study are provided in Appendix C. There is no minimum safety stock for any period. The mathematical model consists of 17,709 equations, 10,734 binary variables, and 2,664 continuous variables. The optimal solution was reached in just 142 CPU s corresponding to a total cost equal to 315,627 €.

Figure 6.7 presents the production plan for families as well as the manpower profile over the entire planning horizon of interest. The solution does not indicate production over the weekends, thus minimizing total costs. The total cost breakdown is shown in Figure 6.8. Note that the total inventory cost represents approximately 41% of the total costs. The total changeovers cost reflects the 21%

**Table 6.3:** Manpower requirements for families in every packing unit.

| | Units | | | | | Units | | | |
|---|---|---|---|---|---|---|---|---|---|
| Family | J1 | J2 | J3 | J4 | Family | J1 | J2 | J3 | J4 |
| F01 | | | 5 | | F13 | | 2 | | |
| F02 | | | 5 | | F14 | | 2 | | |
| F03 | | | 5 | | F15 | | 2 | | |
| F04 | | | 4 | | F16 | | 2 | | |
| F05 | | | 4 | | F17 | | 2 | | |
| F06 | | | 4 | 4 | F18 | | 2 | | |
| F07 | | | 4 | | F19 | | 2 | | |
| F08 | | | 4 | 4 | F20 | 2 | | | |
| F09 | | | 4 | 4 | F21 | 3 | | | |
| F10 | | | 4 | 4 | F22 | 2 | | | |
| F11 | | 2 | | | F23 | | | | 3 |
| F12 | | 2 | | | | | | | |

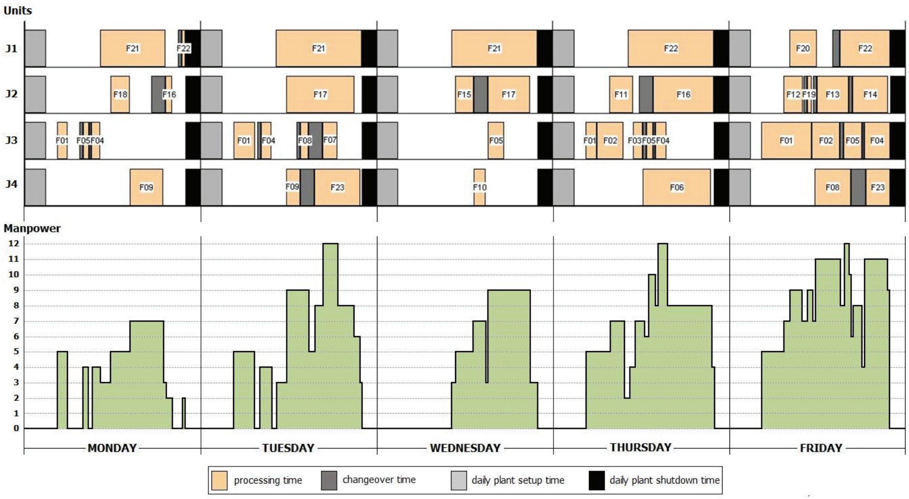



**Figure 6.7:** Case Study I: Production plan and manpower profile.

of the total costs. The inventory cost profile for each production day is illustrated in Figure 6.9. Thursday is the day with the higher inventory cost representing about 42.3% of the total inventory cost. On the other hand, Monday is the day with the lower inventory cost contribution representing 12.4% of the total inventory cost. It is worth noting that the proposed solution does not lead to any external production.

**Case Study II.** This case study, considers the unexpected case of the absence of an employee (illness, etc.) from Wednesday to Friday. That means that the maximum manpower capacity is reduced from 12 to 11 workers. Therefore, a new production plan should be generated from Wednesday to Friday since the previous one (Case Study I), as indicated in Figure 6.7, becomes infeasible (see also the manpower profile on Thursday and on Friday). The production re-planning prob-

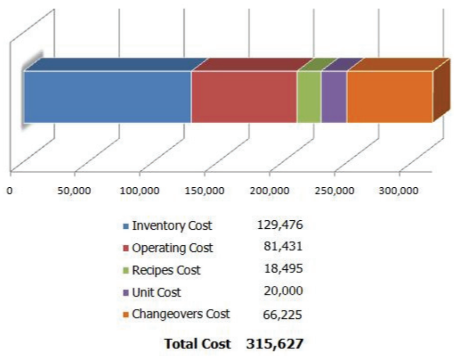



**Figure 6.8:** Case Study I: Breakdown of total cost (€).

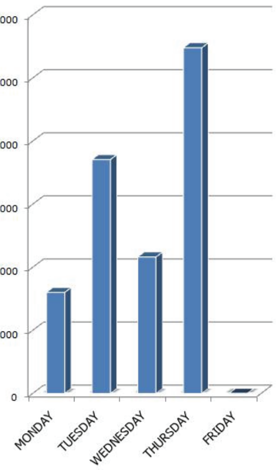



**Figure 6.9:** Case Study I: Total inventory cost per production period (€).

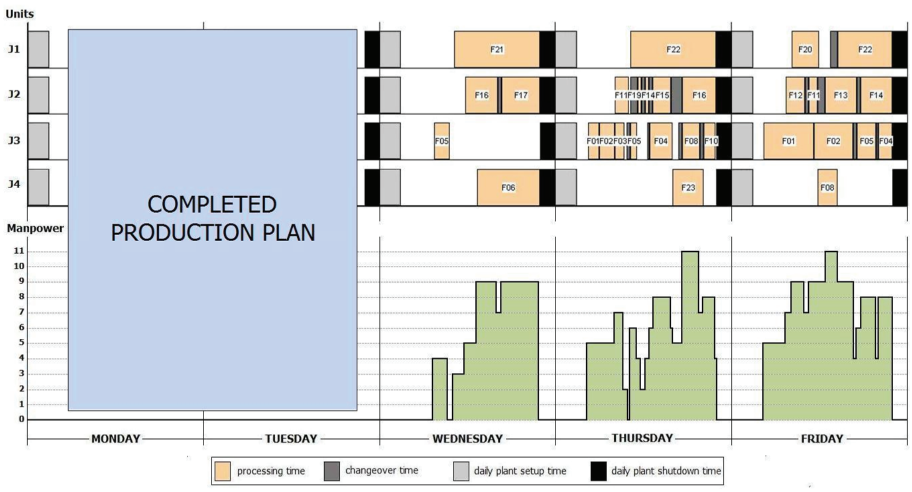



**Figure 6.10:** Case Study II: Production plan and manpower profile.

lem from Wednesday to Friday using the updated manpower capacity and actual production targets is therefore considered using the proposed model. The mathematical model consists of 8,901 equations, 5,367 binary variables, and 1,425 continuous variables. The optimal solution was reached in 516 CPU s leading to a total cost of 222,847 €.

The proposed production plan for all families and the manpower profile over the whole production planning horizon is depicted in Figure 6.10. Again there is no need for external production. Figure 6.11 illustrates the total cost breakdown. The total inventory cost and the total changeovers cost represent approximately 40% and 24% of the total cost, respectively. The inventory cost profile for each production day is illustrated in Figure 6.12. As expected the solution leads to an increase in the inventory cost on Wednesday and Thursday comparing to the initial production plan (Case Study I). This is due to the fact that the number of available employees is decreased thus resulting into a production capacity decrease. Note that the inventory cost generated on Thursday is approximately three times higher than the corresponding cost on Wednesday.

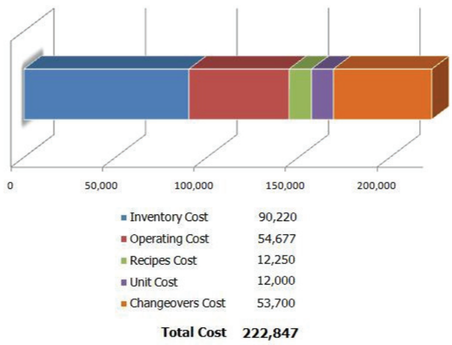



**Figure 6.11:** Case Study II: Breakdown of total cost (€).

**Case Study III.** A salient feature of the dairy industry is that customers usually confirm (i.e., change) their order quantities a few days prior to dispatch. This case study considers the case where production targets levels change due to potential orders cancellation, arrival of new orders and modification of old orders quanti-

**Table 6.4:** Case Study III. Updated production targets (cups).

| | old production targets | | new production targets | | Order |
|---|---|---|---|---|---|
| Product | Thursday | Friday | Thursday | Friday | Modification Type |
| P01 | | | 1,402 | | new arrival |
| P04 | | 14,001 | | 11,474 | quantity modified |
| P05 | | 5,480 | | 7,985 | quantity modified |
| P08 | | 4,000 | | | cancelled |
| P11 | | 715 | | 946 | quantity modified |
| P13 | | | 1,628 | | new arrival |
| P16 | | 3,715 | | 4,115 | quantity modified |
| P17 | | | 1,928 | | new arrival |
| P21 | | 1,620 | | 2,220 | quantity modified |
| P22 | | 1,380 | | 1,790 | quantity modified |
| P24 | 4,193 | | | | cancelled |
| P25 | 14,974 | | 13,792 | | quantity modified |
| P28 | | | | 2,146 | new arrival |
| P33 | | 4,057 | | 3,002 | quantity modified |
| P41 | | 1,172 | | 2,276 | quantity modified |
| P44 | | 2,019 | | 1,626 | quantity modified |
| P58 | 3,188 | | 1,985 | | quantity modified |
| P64 | | | | 1,856 | new arrival |
| P72 | 2,040 | | | | cancelled |
| P80 | | | | 120 | new arrival |
| P81 | | | | 240 | new arrival |

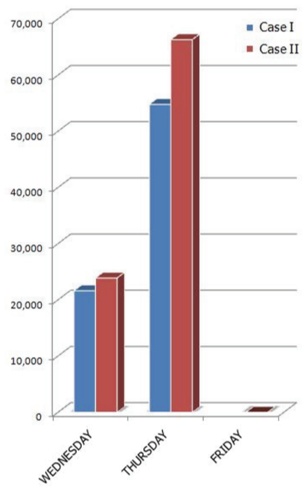



**Figure 6.12:** Case Study II: Total inventory cost per production period (€).

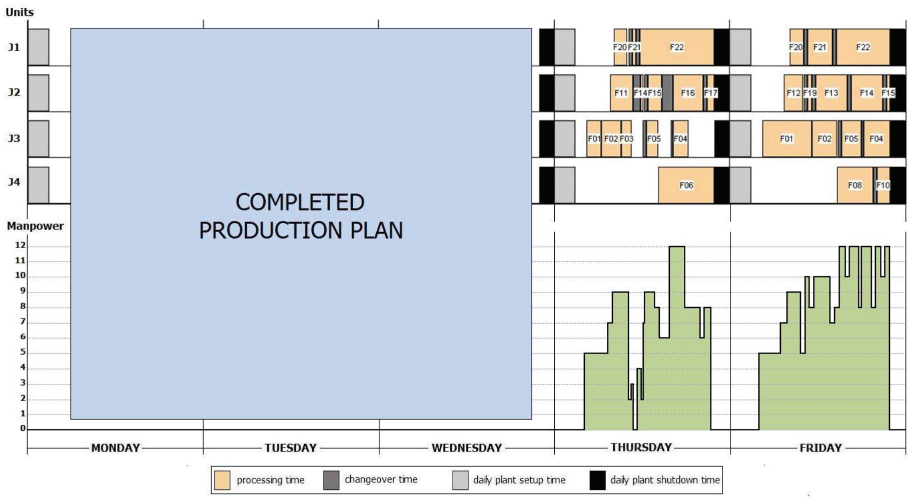



**Figure 6.13:** Case III: Production plan and manpower profile.

ties. New production targets from Tuesday to Friday arrived from the logistics department on Wednesday night (see Table 6.4). Therefore, a new production plan, considering the new production targets, should be generated from Tuesday to Friday. The initial production plan is that of Case Study I. The optimal production plan was obtained in 242 CPU s leading to a total cost of 179,793 €. An external production of 3,290 kg for product P87 is observed, since the production capacity of the dairy plant is not able to achieve full demand satisfaction. Figure 6.13 illustrates the production plan for families and the manpower profile over the entire production planning horizon. An inventory cost equal to 68,161 € is generated on Thursday while this figure for Friday is 1,530 €. The cancelled product order P72 reflects the inventory cost of Friday; since P72 had been already produced in advance on Monday (see Table D.1 in Appendix D). Figure 6.14 illustrates the total cost breakdown. The external production cost is not included in the objective function since it reflects a penalty cost and not a real cost term.

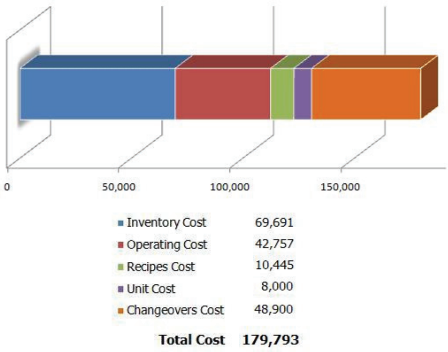



**Figure 6.14:** Case III: Breakdown of total cost (€).

## 6.7 Concluding Remarks

In this chapter, a novel MIP framework for the resource-constrained production planning problem in semicontinuous processing industries (e.g., dairy industries) has been developed. Quantitative as well as qualitative optimization goals are included in the proposed model. Renewable resource limitations are appropriately taken into account. Moreover, the MIP formulation has been extended to deal with unexpected events such as absence of an employee, product order cancellation or modification, etc. The properly treatment of uncertainty in semicontinuous industries is of great importance since unpredicted events take place very frequently. Food processing industries involve the production of perishable products therefore strategies of building up inventories are inappropriate because they compromise the freshness, the quality, and the selling price of the final products. Therefore, and as illustrated in Case Study II and III, production re-planning should be done *on-line* after the occurrence of an unexpected event.

The presented MIP model aims at being the core element of a computer-aided advanced planning system in order to facilitate decision making in related industrial environments. More specifically, the proposed approach can help users analyze plans and schedules, run what-if analysis, compare scenarios, balance the optimization of multiple goals, modify the recommended solution, and determine whether a modification violates any constraints. The results indicate the best possible production plans and schedules to maximize profitability and customer service, while taking into account the full set of operating costs and constraints, from inventory carrying and changeover costs to equipment management and labor resource availability. The proposed planning model delivers value beyond plan feasibility and schedule optimization. It may also serve as a tool for negotiations between the manufacturing and supply chain departments, allowing them to collaborate more easily to find the best balance between inventory levels and operational efficiency. Furthermore it can provide the basis to analyze the impact of new production plans on manufacturing efficiency, and scheduling decisions on inventory levels and demand satisfaction.

In lack of computer-aided production planning tools, empirical production plans are usually sent to the plant floor in dairy processing industries; thus thwarting the lucrative performance of the production facility. It should be pointed out that it may be difficult to directly quantify the benefits of the proposed MIP framework because the pre-computer situation is not usually known in detail, so there is no sufficient basis for comparison. However, this single fact is an excellent argument in favor of computer-aided production planning as discussed by Jakeman (1994): *If you do not know how well you are doing, how can you improve your performance?*.

## 6.8 Nomenclature

**Indices / Sets**

| | |
|---|---|
| $f, f' \in F$ | product families (families) |
| $j, j' \in J$ | processing units (units) |
| $k \in K$ | renewable resources |
| $n \in N$ | planning time periods |
| $p \in P$ | products |
| $r \in R$ | batch recipes (recipes) |

**Subsets**

| | |
|---|---|
| $F_j$ | families $f$ that can be processed in unit $j$ |
| $F_k$ | families $f$ that share the same renewable resource $k$ |
| $F_r$ | families $f$ that have the same recipe origin $r$ |
| $J_f$ | available units $j$ to process family $f$ |
| $J_p$ | units $j$ that can process product $p$ |
| $P_f$ | products $p$ that belong to the same family $f$ |
| $P_r$ | products $p$ that have the same recipe origin $r$ |
| $R_f$ | recipe origin $r$ for family $f$ |
| $R_j$ | recipes $r$ that can be processed in unit $j$ |
| $R_p$ | product $p$ that comes from recipe $r$ |

**Parameters**

| | |
|---|---|
| $\alpha_{jn}$ | daily opening setup time for every unit $j$ in period $n$ (e.g., accounts for the pasteurization and homogenization stages) |
| $\beta_{jn}$ | daily shutdown time for every unit $j$ in period $n$ (e.g., cleaning of yogurt production line for hygienic and quality reasons) |
| $\gamma_{ff'j}$ | changeover time between family $f$ and $f'$ in unit $j$ (e.g., accounts for cleaning and sterilizing operations) |

| | |
|---|---|
| $\delta_{pj}$ | setup time for product $p$ on unit $j$ |
| $\varepsilon_{kfj}$ | renewable resource $k$ requirements for family $f$ when processed in unit $j$; in the current study corresponds to the number of workers |
| $E_{kn}^{max}$ | maximum total capacity of renewable resource $k$ at period $n$ |
| $\zeta_{pn}$ | production target for product $p$ in period $n$ |
| $\zeta_{pn}^{cup}$ | production target for product $p$ in period $n$ (in cups) |
| $\eta_{p}^{cup}$ | cup weight for product $p$ |
| $\theta_{pjn}$ | variable operating cost for processing product $p$ in unit $j$ in period $n$ (e.g., includes labor and utilities costs) |
| $\lambda$ | a very small number (0.001) |
| $\Lambda_{jn}$ | $= \omega_{jn} - \alpha_{jn} - \beta_{jn}$ |
| $M_{jn}$ | $= \omega_{jn} - \beta_{jn}$ |
| $\mu_{rn}^{max}$ | maximum production capacity of recipe $r$ in period $n$ |
| $\mu_{rn}^{min}$ | minimum produced quantity of recipe $r$ in period $n$ (e.g., accounts for pasteurization and fermentation tanks capacity restrictions) |
| $\nu_{jn}$ | fixed cost for utilizing processing unit $j$ in period $n$ |
| $\xi_{pn}$ | inventory cost for product $p$ at time $n$ |
| $o_{jn}$ | additional unit preparation time for processing unit $j$ in period $n$ |
| $\pi_{pjn}^{max}$ | maximum production run for product $p$ in unit $j$ in period $n$ |
| $\pi_{pjn}^{min}$ | minimum production run for product $p$ in unit $j$ in period $n$ |
| $\rho_{pj}$ | processing rate for product $p$ in unit $j \in J_p$ |
| $\sigma_{rn}$ | release time for recipe $r$ in period $n$ |
| $\tau_{r}$ | minimum time for preparing recipe $r$; (e.g., for producing stirred yogurt products stands for the minimum fermentation time, while for set yogurt products reflects the minimum cooling time before the packing stage) |
| $\phi_{ff'jn}$ | changeover cost between family $f$ and $f'$ in unit $j$ in period $n$ (e.g., accounts for cleaning and sterilizing operations) |
| $\chi_{rn}$ | cost for producing recipe $r$ in period $n$ |
| $\psi_{pn}$ | external production penalty cost for product $p$ in period $n$ |
| $\omega_{jn}$ | physical available processing time in unit $j$ at period $n$ |

**Continuous Variables**

| | |
|---|---|
| $C_{fjn}$ | completion time for family $f$ in unit $j$ in period $n$ |
| $I_{pn}$ | inventory of product $p$ at time $n$ |
| $Q_{pjn}$ | produced amount of product $p$ in unit $j$ in period $n$ |
| $Q_{pn}^{ext}$ | external production of product $p$ in period $n$ |
| $Q_{pn}^{int}$ | total internal production of product $p$ in period $n$ |
| $S_{fjn}$ | starting time for family $f$ in unit $j$ in period $n$ |
| $T_{fjn}$ | processing time for family $f$ in unit $j$ in period $n$ |

**Binary Variables**

| | |
|---|---|
| $V_{jn}$ | $= 1$ if unit $j$ is used in period $n$ |
| $W_{f'j'fjn}$ | $= 1$ if family $f'$, assigned to unit $j'$ in period $n$, is overlapped by family $f$, assigned to unit $j \neq j'$ in the same period $n$ |
| $X_{ff'jn}$ | $= 1$ if family $f$ is processed exactly before family $f'$, when both are assigned to the same unit $j$ in the same period $n$ |
| $\bar{X}_{f'j'fjn}$ | $= 1$ if family $f'$, assigned to unit $j'$ in period $n$, starts processing before family $f$, assigned to unit $j \neq j'$ in the same period $n$ |

| | |
|---|---|
| $Y_{fjn}$ | $= 1$ if family $f$ is assigned to unit $j$ in period $n$ |
| $\bar{Y}_{pjn}$ | $= 1$ if product $p$ is assigned to unit $j$ in period $n$ |
| $Y^R_{rn}$ | $= 1$ if batch recipe $r$ is produced in period $n$ |
| $Z_{f'j'fjn}$ | $= 1$ if family $f'$, assigned to unit $j'$ in period $n$, is completed after starting family $f$, assigned to unit $j \neq j'$ in the same period $n$ |

Chapter 7

# Simultaneous Optimization of Production & Logistics Operations in Semicontinuous Process Industries

## 7.1 Introduction

In the semicontinuous process industry there is an ongoing trend towards an increased product variety and shorter replenishment cycle times. Hence, manufacturers seek a better coordination of production and distribution activities in order to avoid excessive inventories and improve customers service. While traditionally minimizing production costs has been considered as the major objective, attention has shifted towards faster replenishment and improved logistical performance. Thus, finished product inventories are merely regarded as buffers between the manufacturing and the distribution stage of the supply chain. As a result, distribution costs have to be included into the overall objective function (Bilgen & Günther, 2010).

In this chapter, the production and logistics operations planning in large-scale single- or multi-site semicontinuous process industries is addressed. A novel mixed discrete/continuous-time mixed integer programming model for the problem in question, based on the definition of families of products, is developed. A remarkable feature of the proposed approach is that in the production planning problem timing and sequencing decisions are taken for product families rather than for products. However, material balances are realized for every specific product, thus permitting the detailed optimization of production, inventory, and transportation costs. Sequence-dependent changeovers are also explicitly taken into account and optimized. Moreover, alternative transportation modes are considered for the delivery of final products from production sites to distribution centers. The efficiency and the applicability of the proposed approach is demonstrated by solving to optimality two industrial-size case studies, for an emerging real-life dairy industry which is considered as a representative semicontinuous process industry.

## 7.2 Problem Statement

In this chapter, the production and logistics operations planning problem in multisite multiproduct semicontinuous process industries is addressed. The basic features of the problem under consideration are summarized as follow:

(i) A known planning horizon divided into a set of periods $n \in N$.

(ii) A set of production sites $s \in S$, and a set of distribution centers $d \in D$.

(iii) A set of transportation trucks $l \in L$ which can transfer final products from production sites to distribution centers, $l \in L_{sd}$. Each transportation truck is characterized by a minimum and maximum capacity, $\varepsilon_l^{min}$ and $\varepsilon_l^{max}$, respectively.

(iv) A set of processing units $j \in J$ which are installed on production site $s$, $j \in J_s$; with available processing time in period $n$ equal to $\omega_{sjn}$.

(v) A set of products $p \in P$ with specific demand in period $n$, inventory costs $\xi_{spn}$, production rates $\rho_{psj}$, minimum processing runs $\pi_{psjn}^{min}$, processing costs $\theta_{psjn}$, and minimum storage time for processed products $\lambda_p$. $P_j$ is the subset of products that can be assigned to unit $j$, and $J_p$ is the subset of units that can produce product $p$.

(vi) A set of batch recipes $r \in R$ (e.g., fermentation recipes) with minimum preparation time $\tau_r$, preparation cost $\chi_{srn}$, and minimum and maximum production capacity $\mu_{srn}^{min}$ and $\mu_{srn}^{max}$, respectively. The subset of products that come from batch recipe $r$ is denoted by $P_r$.

(v) A set of product families or simply *families* $f \in F$ wherein all products are grouped into; $F_j$ is the subset of families that can be assigned to unit $j$, and $J_f$ is the subset of units that can process family $f$, while the subset of products in family $f$ is denoted by $P_f$.

(vi) A sequence-dependent changeover or simply *changeover* operation is required on each processing unit whenever the production is changed between two different families; the required changeover time is $\gamma_{ff'sj}$, while the changeover cost is $\phi_{ff'sjn}$.

(vii) A sequence-independent setup operation, henceforth referred to as *setup*, is required whenever product $p$ is assigned to a processing unit $j$; the setup time is $\delta_{psj}$.

We assume a non-preemptive operation mode, and no resource restrictions (e.g., manpower, steam, electricity, etc.).

The main key decision variables are:

(i) the optimal assignment of families and products to each processing unit in production period, $Y_{fsjn}$ and $\bar{Y}_{psjn}$, respectively;

(ii) the sequencing between families $f$ and $f'$ on each unit in every period, $X_{ff'sjn}$;

(iii) the assignment of transportation trucks to processing sites - distribution centers in each period $Z_{sdln}$ as well as the transportation load for each truck $\bar{U}_{sdln}$;

(iv) the produced quantity for each product in each processing site at period $Q_{psjn}$ and the total produced amount of product $p$ per period $\bar{Q}_{psn}$; and, final

(v) the inventory profiles for each product at period $n$, $I_{spn}$.

The objective is to fully satisfy customer demand at minimum total cost, including production, changeover, inventory and transportation costs.

## 7.3 Modeling Approach

As already discussed in Section 6.4, production planning in semicontinuous process industries typically deals with a large number of products with similar characteristics. This fact allow us to group products with similar characteristics into product families (families). In the proposed approach products belong to the same family *if and only if*: (i) they come from the same batch recipe, and (ii) there is no need for changeover operations among them. Therefore, the production planning problem under question could be *partially* focused on families rather than on each product separately. More specifically, sequencing and timing decisions are taken for families and not for each separate product, as Figure 7.1 illustrates. Obviously, the definition of families significantly reduces the size of the underlying mathematical model and, thus, the necessary computational burden without sacrificing any feasibility or optimality constraint.

A salient feature of the proposed mathematical formulation is the integration of three different modeling approaches (see Figure 7.1). More specifically, we use: (i) a discrete-time approach for the calculation of inventories and transported quantities for products at the end of each period $n$ in the production and logistics operations planning level, (ii) a continuous-time approach for the sequencing of families in the scheduling level for families, and (iii) lot-sizing type capacity constraints in the short-term scheduling level for products. Further, it should be emphasized that the proposed modeling approach allows products that belong to the same family to have different: (i) processing rates (e.g., packing rates), (ii) operating costs, (iii) setup times, (iv) inventory costs, (v) transportation costs, and (vi) customer type.

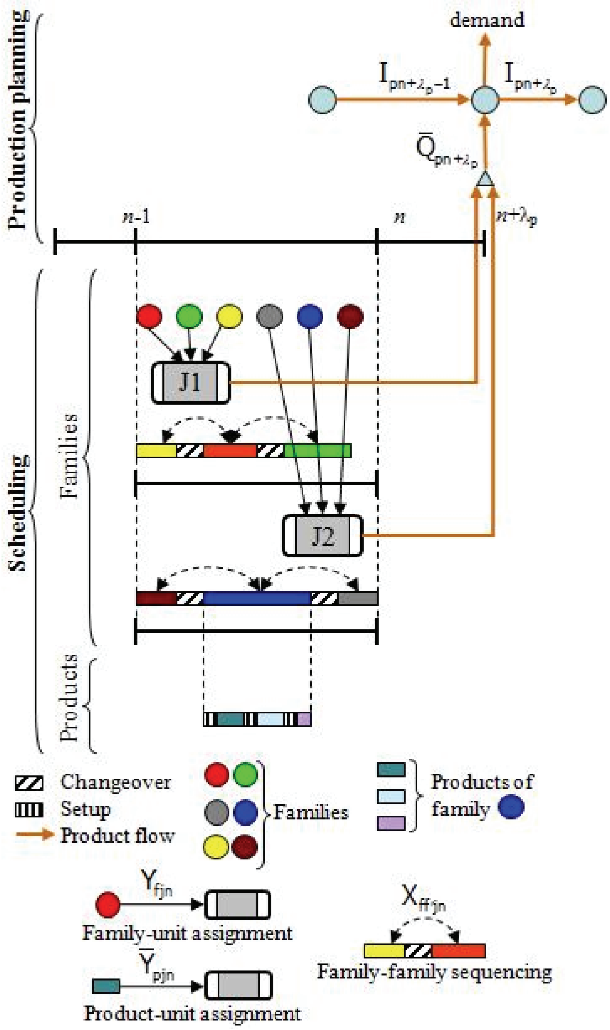



**Figure 7.1:** Modeling approach.

## 7.4 Mathematical Formulation

In the proposed mathematical framework, constraints have been grouped according to the type of decision (assignment, timing, sequencing, etc.) upon which they are imposed on. For the sake of clarity of the model presented, we use uppercase Latin letters for decision variables and sets, and lowercase Greek letters for parameters.

**Material Balance and Logistics Operations Constraints.** The transportation of final products to customers (or distribution centers) is assumed to be done by three potential *transportation modes*: (a) transportation trucks owned by the customers, (b) transportation trucks owned by the industry, and (c) contracted transportation trucks from third party logistics companies. Final products $p \in P^a$ whose final destination is the international market or big national supermarket customers are transported to their customers by transportation mode (a). The transportation of final products to the distribution centers owned by the enterprise can be performed by any of the other two transportation modes (b), and/or (c).

The total quantity of product $p$ produced in production plant $s$ in period $n-\lambda_p$, which is ready to ship to customers in period $n$, is given by:

$$\bar{Q}_{spn} = \sum_{j \in (J_s \cap J_p)} Q_{psjn-\lambda_p} \qquad \forall s, p, n > \lambda_p \tag{7.1}$$

where $\lambda_p$ denotes the days that processed product $p$ should be kept in storage (e.g., for cooling or refrigeration purpose). $Q_{psjn}$ corresponds to the quantity of product $p$ processed in unit $j$ of production site $s$ during period $n$. It should be noticed that $\bar{Q}_{spn} = 0 \quad \forall s, p, n \leq \lambda_p$.

Constraint set (7.2) expresses the material balance of products $p \in P^a$ whose destination is the international market or big national supermarket clients.

$$I_{spn} = I_{spn-1} + \bar{Q}_{spn} - U^a_{spn} \qquad \forall s, p \in P^a, n \tag{7.2}$$

where $U^a_{spn}$ denotes the quantity of product $p \in P^a$ transported from production site $s$ to the international market or big national supermarket clients by customer trucks, at period $n$, in order to fully meet the demand according to:

$$\sum_{s} U^a_{spn} = \zeta^a_{pn} \qquad \forall d, p \in P^a, n \tag{7.3}$$

$I_{spn}$ corresponds to the inventory level of product $p$ in production plant $s$ at time point $n$. Also, note that $I_{spn=0}$ reflects the initial inventory for product $p$ in production site $s$.

The multiperiod material balance constraints for products $p \notin P^a$ transported to company's distribution centers are given by:

$$I_{spn} = I_{spn-1} + \bar{Q}_{spn} - \sum_{d \in D_s} \sum_{l \in L_{sd}} U_{sdlpn} \qquad \forall s, p \notin P^a, n \tag{7.4}$$

where $U_{sdlpn}$ denotes the quantity of product $p$ transported from production site $s$ to distribution center $d$ by transportation truck $l$ at period $n$. Once final products reach distribution centers, they are stored for a day due to product quality purpose before sending them to final customers, as follows:

$$\sum_{s \in S_d} \sum_{l \in L_{sd}} U_{sdlpn-1} = \zeta_{dpn} \qquad \forall d, p \notin P^a, n > 1 \tag{7.5}$$

The total load for any transportation truck $l$ that transfers products from production facility $s$ to distribution center $d$ in period $n$ is calculated as follows:

$$\bar{U}_{sdln} = \sum_{p \notin P^a} U_{sdlpn} \qquad \forall s, d \in D_s, l \in L_{sd}, n \tag{7.6}$$

Hence, every truck $l$ has a specific minimum and maximum capacity ($\varepsilon_l^{min}$, and $\varepsilon_l^{max}$, respectively) as given by:

$$\varepsilon_l^{min} Z_{sdln} \leq \bar{U}_{sdln} \leq \varepsilon_l^{max} Z_{sdln} \qquad \forall s, d \in D_s, l \in L_{sd}, n \tag{7.7}$$

Binary variables $Z_{sdln}$ denote the use of truck $l$ for transporting products from production site $s$ to distribution center $d$ at period $n$. Any transportation truck $l$ can transfer products only between one production site $s$ and one distribution center $d$ during any period $n$, according to:

$$\sum_{s \in S_l} \sum_{d \in (D_s \cap D_l)} Z_{sdln} \leq 1 \qquad \forall l, n \tag{7.8}$$

**Product Lot-Sizing Constraints.** Lower and upper bounds on the produced amounts of product $p$ are imposed by:

$$\pi_{psjn}^{min} \bar{Y}_{psjn} \leq Q_{psjn} \leq \pi_{psjn}^{max} \bar{Y}_{psjn} \qquad \forall p, s, j \in (J_s \cap J_p), n \tag{7.9}$$

Tighter maximum produced quantities for $p \in P^a$ can be estimated by:

$$\pi_{psjn}^{max} = \begin{cases} 0 & \text{if } \sum\limits_{n' \geq n+\lambda_p}^{N} \zeta_{pn'}^{a} = 0 \\ (\omega_{jn} - \alpha_{jn} - \beta_{jn} - \underset{r \in R_p}{\tau_r})\rho_{psj} & \text{if } \sum\limits_{n' \geq n+\lambda_p}^{N} \zeta_{pn'}^{a} \geq 0 \end{cases}$$

It should be noted that demands $\zeta_{pn'}^{a}$ must be met (i.e., full demand satisfaction). Similar expressions can be written for products $p \notin P^a$.

**Family Processing Time Definition.** Because products that belong to the same family ($p \in P_f$) do not require changeover operations among them, sequencing and timing constraints should be solely imposed on families. However, it should be noted that setup times $\delta_{psj}$ may exist. In order to define sequencing and timing decisions for families, the definition of family processing time is introduced, as follows:

$$T_{fsjn} = \sum_{p \in P_f} (\frac{Q_{psjn}}{\rho_{psj}} + \delta_{psj} \bar{Y}_{psjn}) \qquad \forall f, s, j \in (J_f \cap J_s), n \tag{7.10}$$

Product processing rates, $\rho_{psj}$, are considered fixed as potential fluctuations may provoke quality problems (Soman *et al.*, 2004).

**Family Allocation Constraints.** A family $f$ is assigned to a processing unit $j$ of production site $s$ in period $n$ if at least one product $p \in P_f$, that belongs to this family, is processed in this unit during the same period:

$$Y_{fsjn} \geq \bar{Y}_{psjn} \qquad \forall f, p \in P_f, s, j \in (J_f \cap J_s), n \tag{7.11}$$

Hence, constraint set (7.12) enforces the binary variables $Y_{fsjn}$ to zero when no products $p \in P_f$ are processed in unit $j$ at production site $s$ during period $n$.

$$Y_{fsjn} \leq \sum_{p \in P_f} \bar{Y}_{psjn} \qquad \forall f, s, j \in (J_f \cap J_s), n \tag{7.12}$$

**Family Sequencing and Timing Constraints.** We introduce binary variables $X_{ff'sjn}$ to define the local precedence between two families $f$ and $f'$ in unit $j$, at production plant $s$ in period $n$. Constraints (7.13) and (7.14) state that, if a family $f$ is allocated to processing unit $j$ at production site $s$ in period $n$, (i.e., $Y_{fsjn} = 1$), then at most one family $f'$ is processed before and after it, respectively.

$$\sum_{f' \neq f, f' \in F_j} X_{f'fsjn} \leq Y_{fsjn} \qquad \forall f, s, j \in (J_f \cap J_s), n \tag{7.13}$$

$$\sum_{f' \neq f, f' \in F_j} X_{ff'sjn} \leq Y_{fsjn} \qquad \forall f, s, j \in (J_f \cap J_s), n \tag{7.14}$$

Obviously, the total number of active sequencing binary variables $X_{ff'sjn}$ plus the unit utilization binary variable $V_{sjn}$ should be equal to the total number of active allocation binary variables $Y_{fsjn}$ in a processing unit $j$ at production facility $s$ in period $n$, according to:

$$\sum_{f \in F_j} \sum_{f' \neq f, f' \in F_j} X_{ff'sjn} + V_{sjn} = \sum_{f \in F_j} Y_{fsjn} \qquad \forall s, j \in J_s, n \tag{7.15}$$

Constraint set (7.16) ensures that the processing unit $j$ in production site $s$ is used at period $n$, (i.e., $V_{sjn}$=1) if at least one family $f$ is assigned to it over this period (i.e., $Y_{fsjn} = 1$). Note that no lower bound on the binary variable $V_{sjn}$ is necessary as far as a cost term (related to the unit utilization), is included into the objective function, thus enforcing $V_{sjn}$ to zero.

$$V_{sjn} \geq Y_{fsjn} \qquad \forall f, s, j \in (J_f \cap J_s), n \tag{7.16}$$

The starting time of family $f'$, that directly follows another family $f$ on a processing line $j$ in production plant $s$ at period $n$, (i.e., $X_{ff'sjn} = 1$) should be greater than the completion time of family $f$, $C_{fsjn}$, plus the necessary changeover time $\gamma_{ff'sj}$ between those families:

$$\begin{aligned} C_{fsjn} + \gamma_{ff'sj} \leq C_{f'sjn} - T_{f'sjn} + M_{sjn}(1 - X_{ff'sjn}) \\ \forall f, f' \neq f, s, j \in (J_f \cap J_{f'} \cap J_s), n \end{aligned} \tag{7.17}$$

Note that the big-M parameter $M_{sjn}$ can be set equal to $\omega_{sjn} - \beta_{sjn}$, where $\omega_{sjn}$ is the available production time horizon and $\beta_{sjn}$ corresponds to the daily plant shutdown time.

**Family Starting and Completion Time Bounds.** Constraints (7.18) and (7.19) impose bounds on the starting and completion time of each family. More specifically, the starting time (i.e., $C_{fsjn} - T_{fsjn}$) has to be greater than the daily plant setup time, $\alpha_{sjn}$, plus the minimum batch time (fermentation process in the case of yogurt production) $\tau_r$ for preparing the recipe $r$, plus the changeover time $\gamma_{f'fsj}$ for changing the production to family $f'$. An additional unit preparation time $o_{sjn}$ is also taken into account. This time stands for the additional preparation time of a processing unit $j$ due to potential maintenance or other technical reasons. Additionally, the release batch recipe time $\sigma_{srn}$ is also considered. In order to commence the production of a batch recipe $r$ all recipe ingredients need to be present. Otherwise, the production of the batch recipe $r$ will be postponed until the arrival of its missing substances.

$$\begin{aligned} C_{fsjn} - T_{fsjn} \geq (\alpha_{sjn} + max[o_{sjn}, \sigma_{srn}] + \tau_r) Y_{fsjn} \\ + \sum_{f' \neq f, f' \in F_j} \gamma_{f'fsj} X_{f'fsjn} \qquad \forall s, f, r \in R_f, j \in (J_f \cap J_s), n \end{aligned} \tag{7.18}$$

Hence, the completion time of each family should be smaller than the daily production time horizon $\omega_{sjn}$ minus the daily plant shutdown time $\beta_{sjn}$, as follows:

$$C_{fsjn} \leq (\omega_{sjn} - \beta_{sjn}) Y_{fsjn} \qquad \forall f, s, j \in (J_f \cap J_s), n \tag{7.19}$$

Production line shutdown is realized on a daily basis, as a typical production policy to guarantee high quality of final products and to comply with hygienic standards.

**Batch Recipe Stage Constraints.** Batch recipe stage (e.g., fermentation and pasteurization) constraints must be included into the mathematical model in order to ensure a feasible production plan. The cumulative produced quantity of products $p \in P_r$ should be greater than the minimum produced recipe amount in the batch recipe stages $\mu_{srn}^{min}$ and lower than the maximum production capacity $\mu_{srn}^{max}$:

$$\mu_{srn}^{min} W_{srn} \leq \sum_{p \in P_r} \sum_{j \in (J_p \cap J_s)} Q_{psjn} \leq \mu_{srn}^{max} W_{srn} \qquad \forall s, r, n \tag{7.20}$$

Constraint set (7.21) states that a batch recipe $r$ is produced in production facility $s$ at period $n$, (i.e., $W_{srn} = 1$), if at least one family $f \in F_r$ is processed on a processing unit $j$ in production site $s$ at the same period $n$ (i.e., $Y_{fsjn} = 1$) .

$$W_{srn} \geq \sum_{j \in (J_f \cap J_s)} Y_{fsjn} \qquad \forall s, r, f \in F_r, n \tag{7.21}$$

**Tightening Constraints.** In order to reduce the computational effort, constraint set (7.22) can further tighten the linear relaxation of the proposed mathematical model by imposing an upper bound on the total processing time for each processing line $j$ at each period $n$.

$$\begin{aligned}&\sum_{f\in F_j} T_{fsjn} + \sum_{f\in F_j}\sum_{f'\neq f, f'\in F_j} \gamma_{ff'sj} X_{ff'sjn}\\ &\leq (\omega_{sjn} - \alpha_{sjn} - \beta_{sjn} - \min_{r\in R_j}[\tau_r]) V_{sjn} \qquad \forall s, j\in J_s, n\end{aligned} \tag{7.22}$$

It should be noted that by incorporating constraint set (7.22) into the mathematical formulation, constraint set (7.16) can be omitted, thus further reducing the model size. Similarly, an upper bound on the family processing time can be defined as follows:

$$\begin{aligned}&T_{fsjn} + \sum_{f'\neq f, f'\in F_j} \gamma_{ff'sj} X_{ff'sjn} \leq (\omega_{sjn} - \alpha_{sjn} - \beta_{sjn} - \tau_r) Y_{fsjn}\\ &\qquad \forall f, r\in R_f, s, j\in (J_f\cap J_s), n\end{aligned} \tag{7.23}$$

**Objective Function.** The objective function to be minimized is the total cost including several factors such as: (i) inventory costs, (ii) operating costs, (iii) batch recipes preparation costs, (iv) unit utilization costs, (v) families changeover costs, and (vi) transportation costs, as follows:

$$\begin{aligned}\min &\sum_s\sum_p\sum_n \xi_{spn} I_{spn} + \sum_s\sum_p\sum_{j\in(J_s\cap J_p)}\sum_n \frac{\theta_{psjn}}{\rho_{psj}} Q_{psjn}\\ &+ \sum_s\sum_r\sum_n \chi_{srn} W_{srn} + \sum_s\sum_{j\in J_s}\sum_n \nu_{sjn} V_{sjn}\\ &+ \sum_s\sum_f\sum_{f'\neq f}\sum_{j\in(J_s\cap J_f\cap J_{f'})}\sum_n \phi_{ff'sjn} X_{ff'sjn}\\ &+ \sum_s\sum_{d\in D_s}\sum_{l\in L_{sd}}\sum_n (\psi_{sl} Z_{sdln} + \upsilon_{sdl} \bar{U}_{sdln})\end{aligned} \tag{7.24}$$

In a dairy plant, final yogurt products are kept at low temperatures, thus resulting to significant inventory cost (mainly due to high energy requirements), which should be considered in the optimization procedure. It should be also noted that the short shelf lives of yogurt products are indirectly taken into account through the inventory costs. Operating costs mainly include labor and energy costs plus costs due to material losses. The batch recipe (e.g., fermentation) cost account for all costs associated with the preparation of each batch recipe. The unit utilization cost basically stands for the shutdown cleaning operation cost, and the initial unit setup cost. Changeover costs correspond to cleaning and/or sterilization operations for switchover operations between families. Transportation costs include a fixed costs term for contracting the transportation vehicles and a variable costs term for the quantities transfered from production sites to distribution

**Table 7.1:** Families relative sequences in a planning period per unit.

| Units | Families Relative Sequence |
|---|---|
| J1 | F20 ➩ F21 ➩ F22 |
| J2 | F12 ➩ F11 ➩ F19 ➩ F18 ➩ F13 ➩ F14 ➩ F15 ➩ F16 ➩ F17 |
| J3 | F01 ➩ F02 ➩ F03 ➩ F05 ➩ F04 ➩ F08 ➩ F09 ➩ F10 ➩ F06 ➩ F07 |
| J4 | F08 ➩ F09 ➩ F10 ➩ F06 ➩ F23 |

centers. Note that products $p \in P^a$ whose final destination is the international market or big national supermarket customers are transported by using their own trucks, therefore there is no transportation cost for the industry. Finally, it should be mentioned that since full demand satisfaction is required, the minimization of total costs is identical to the maximization of total profit. The overall MIP model optimizes objective function (7.24) subject to constraints (7.1) – (7.23) .

## 7.5 Case Studies

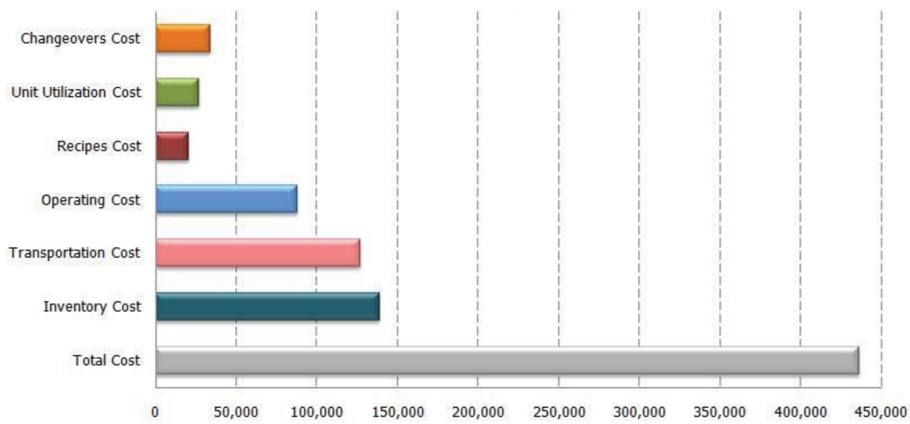
In this section, two industrial-size case studies are considered using the proposed MIP model. The first case (Case Study I) concerns the single-site production (already described in Section 6.6) and distribution planning of an emerging Greek dairy industry. Real data have been slightly modified due to confidentiality issues. The second case (Case Study II) considers the multi-site production and distribution planning problem, and is inspired by Case Study I.

At this point, it should be emphasized that in semicontinuous process plants, as well as in many other food processing industries, a natural sequence of products


**Figure 7.2:** Case Study I: Production site and distribution centers locations.

often exists (e.g., from the lower taste to the stronger or from the brighter color to the darker) thus the relative sequence of products within a family is known a priory. Therefore, when changing the production between two products of the same family, cleaning and sterilizing can be neglected. Hence, in such production plants not only the relative sequence of products belonging to the same product family may be known but also the relative sequence of families in each unit. In this case, different families are enumerated according to their relative position within the day. Table 7.1 illustrates the families relative sequence inside a planning period for the case studies under consideration.

There are no initial inventories and setup times for products in both cases. Finally, all case studies have been solved to global optimality in a Dell Inspiron 1520 2.0 GHz with 2 GB RAM using CPLEX 11 under standard configurations via a GAMS 22.8 interface (Rosenthal, 2010).




| Total Cost | Inventory Cost | Transportation Cost | Operating Cost | Recipes Cost | Unit Utilization Cost | Changeovers Cost |
|---|---|---|---|---|---|---|
| 436,170 | 139,210 | 126,798 | 88,262 | 20,750 | 27,000 | 34,150 |

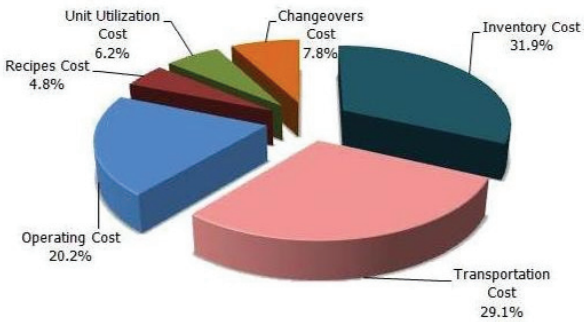



**Figure 7.3:** Case Study I: Total cost breakdown (€) and cost terms contribution.

**Table 7.2:** Case Study I: Transportation plan (kg).

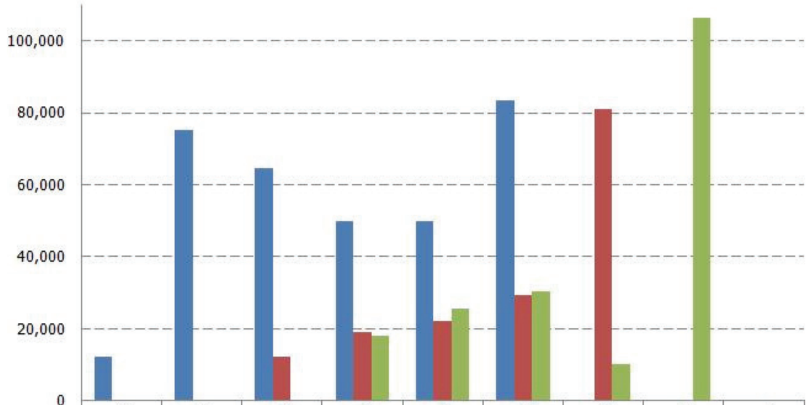
| Truck | Distr. Center | n0 | n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 |
|---|---|---|---|---|---|---|---|---|---|---|
| OWN-1 | Thessaloniki | 0 | 0 | 0 | 6,000 | 6,000 | 0 | 5,250 | 6,000 | 0 |
| | Xanthi | 0 | 0 | 0 | 0 | 0 | 4,621 | 0 | 0 | 0 |
| OWN-2 | Thessaloniki | 0 | 0 | 0 | 4,506 | 0 | 0 | 0 | 0 | 0 |
| | Xanthi | 0 | 0 | 0 | 0 | 4,151 | 4,621 | 0 | 5,924 | 0 |
| 3PLT-1 | Athens | 0 | 0 | 0 | 7,465 | 8,918 | 0 | 5,000 | 10,795 | 0 |
| | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 9,776 | 0 | 0 | 0 |
| 3PLT-2 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Thessaloniki | 0 | 0 | 0 | 0 | 6,450 | 5,699 | 0 | 10,795 | 0 |
| 3PLT-3 | Athens | 0 | 0 | 0 | 0 | 0 | 5,699 | 0 | 10,795 | 0 |
| | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3PLT-4 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10,795 | 0 |
| 3PLT-5 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10,795 | 0 |
| 3PLT-6 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10,795 | 0 |
| 3PLT-7 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9,901 | 0 |
| | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3PLT-8 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9,901 | 0 |
| | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3PLT-9 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9,901 | 0 |

### 7.5.1 Case Study I

The production and distribution network of the dairy industry under study consists of one production site and three distribution centers, as shown in Figure 7.2. The production facility, situated in the city of Serres, has to fully satisfy the demand for: (i) products $p \notin P_a$ for the distribution centers, and (ii) products $p \in P_a$ for international customers and big local supermarket clients. The plant operates with 4 packing lines (J1 – J4). The 93 final products (P01 – P93) are grouped into 23 families (F01 – F23). The dairy industry owns a pair of transportation trucks (OWN-1, and OWN-2) with $\psi_{sl} = 50$ € and minimum and maximum load capacity, $\varepsilon_l^{min} = 1{,}000$ kg and $\varepsilon_l^{max} = 6{,}000$ kg, respectively. These trucks can supply the distribution centers situated in Thessaloniki and Xanthi. Ten third party logistics trucks (3PLT-1 to 3PLT-10) with $\psi_{sl} = 700$ € and $\varepsilon_l^{min} = 1{,}000$ kg and $\varepsilon_l^{max} = 12{,}000$ kg are also available. The distribution centers in Thessaloniki and Athens can be supplied by 3PLT trucks. The remaining data are not provided due to confidentiality issues.

The resulting mathematical model consists of 9,639 constraints, 2,160 binary variables, and 15,462 continuous variables. The optimal solution, corresponding to a total cost of 436,167 €, was reached in just 7.7 CPU s after exploring 610 nodes in the branch-and-bound tree.

Figure 7.3 presents the total cost breakdown as well as the contribution of each cost term in the total cost. Inventory and transportation costs stands for the 61.0% of the total cost while production costs (i.e., operating, recipe, unit utilization, and changeovers costs) represent the 39.0% of the total cost. The profiles of total produced quantities, inventories, and transported quantities for each planning period are shown in Figure 7.4. The production site operates from n0 to n5 period. Also,

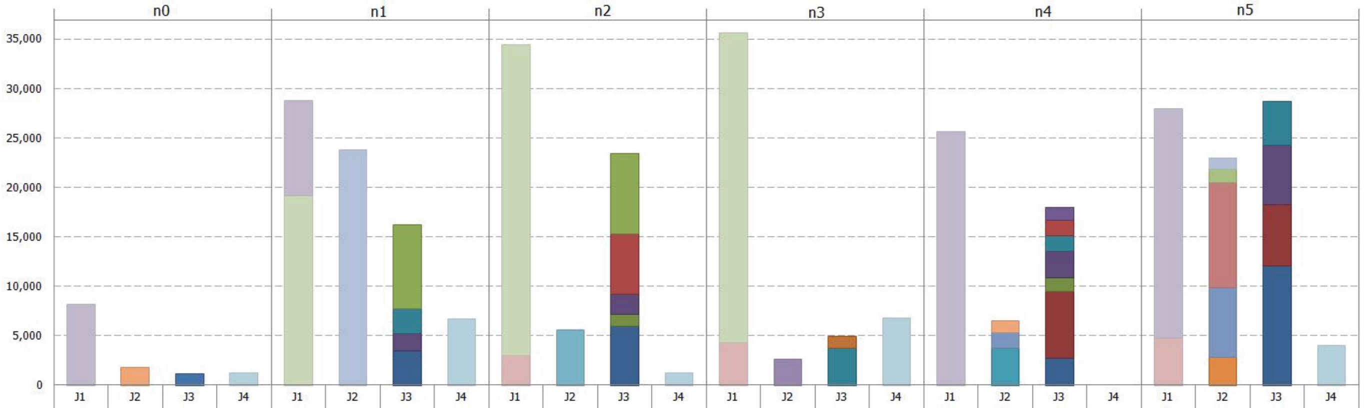



| | n0 | n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 |
|---|---|---|---|---|---|---|---|---|---|
| Production | 12,227 | 75,285 | 64,448 | 49,814 | 50,033 | 83,463 | 0 | 0 | 0 |
| Inventory | 0 | 0 | 12,227 | 18,949 | 22,092 | 29,228 | 81,165 | 0 | 0 |
| Transportation | 0 | 0 | 0 | 17,971 | 25,520 | 30,415 | 10,250 | 106,400 | 0 |

**Figure 7.4:** Case Study I: Production, inventory and transportation profiles per period (kg).

note that due to high demand requirements the production facility operates in period n0, which is an overtime period (i.e., higher operating costs). The highest total production is observed in period n5, with 83,463 kg of production. In period n6, a very high inventory level of 81,165 kg is detected. The transportation schedule is realized from n3 to n7 period. The peak of transportation quantity is observed in period n7 where a total of 106,400 kg is transferred from the production site to the distribution centers.

Figure 7.5 illustrates the detailed production plan for families. The sequences between families can be found in Table 7.1. The optimal transportation plan is given in Table 7.2. A total number of 11 trucks are occupied in period n7, wherein the peak of logistics operations is observed, as Figure 7.4 illustrates. Moreover, the proposed MIP formulation provide us with the detailed transportation plan for each product (i.e., assignment of product to truck, assignment of truck to distribution center, and quantity of product transported by each truck). An example is presented in Table 7.3 where the detailed product transportation plan in periods n3 and n4 is shown.




| | n0 | | | | n1 | | | | n2 | | | | n3 | | | | n4 | | | | n5 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | J1 | J2 | J3 | J4 | J1 | J2 | J3 | J4 | J1 | J2 | J3 | J4 | J1 | J2 | J3 | J4 | J1 | J2 | J3 | J4 | J1 | J2 | J3 | J4 |
| F23 | 0 | 0 | 0 | 1,200 | 0 | 0 | 0 | 6,616 | 0 | 0 | 0 | 1,200 | 0 | 0 | 0 | 6,770 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3,927 |
| F22 | 8,107 | 0 | 0 | 0 | 9,577 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 25,633 | 0 | 0 | 0 | 23,195 | 0 | 0 | 0 |
| F21 | 0 | 0 | 0 | 0 | 19,125 | 0 | 0 | 0 | 31,388 | 0 | 0 | 0 | 31,388 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,953 | 0 | 0 | 0 | 4,200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4,709 | 0 | 0 | 0 |
| F19 | 0 | 0 | 0 | 0 | 0 | 23,764 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,188 | 0 | 0 |
| F18 | 0 | 1,782 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,240 | 0 | 0 | 0 | 0 | 0 | 0 |
| F17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5,526 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,553 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,351 | 0 | 0 |
| F14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10,652 | 0 | 0 |
| F13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,489 | 0 | 0 | 0 | 6,970 | 0 | 0 |
| F12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,792 | 0 | 0 |
| F11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3,726 | 0 | 0 | 0 | 0 | 0 | 0 |
| F10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,361 | 0 | 0 | 0 | 0 | 0 |
| F09 | 0 | 0 | 0 | 0 | 0 | 0 | 8,560 | 0 | 0 | 0 | 8,151 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F08 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6,099 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,541 | 0 | 0 | 0 | 0 | 0 |
| F07 | 0 | 0 | 1,138 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F06 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,221 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F05 | 0 | 0 | 0 | 0 | 0 | 0 | 2,513 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3,683 | 0 | 0 | 0 | 1,555 | 0 | 0 | 0 | 4,504 | 0 |
| F04 | 0 | 0 | 0 | 0 | 0 | 0 | 1,692 | 0 | 0 | 0 | 2,033 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,650 | 0 | 0 | 0 | 5,974 | 0 |
| F03 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,397 | 0 | 0 | 0 | 0 | 0 |
| F02 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6,792 | 0 | 0 | 0 | 6,221 | 0 |
| F01 | 0 | 0 | 0 | 0 | 0 | 0 | 3,438 | 0 | 0 | 0 | 5,898 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,650 | 0 | 0 | 0 | 11,980 | 0 |

**Figure 7.5:** Case Study I: Production plan for families (kg).

**Table 7.3:** Case Study I: Detailed transportation plan for period n3 and n4 (kg).

| Distr. Center | Truck | Product | n3 | Distr. Center | Truck | Product | n4 |
|---|---|---|---|---|---|---|---|
| Athens | 3PLT-1 | P01 | 2,538 | Athens | 3PLT-1 | P28 | 1,850 |
| | | P12 | 1,152 | | | P29 | 1,850 |
| | | P17 | 1,703 | | | P34 | 1,325 |
| | | P90 | 438 | | | P35 | 1,325 |
| | | P91 | 864 | | | P02 | 1,308 |
| | | P92 | 578 | | | P13 | 1,098 |
| | | P93 | 192 | | | P64 | 164 |
| | | total | **7,465** | | | total | **8,918** |
| Thessaloniki | OWN-1 | P84 | 1,500 | Thessaloniki | 3PLT-2 | P34 | 3,435 |
| | | P90 | 909 | | | P28 | 1,200 |
| | | P91 | 1,836 | | | P29 | 1,200 |
| | | P92 | 923 | | | P13 | 616 |
| | | P93 | 832 | | | total | **6,450** |
| | | total | **6,000** | | | | |
| | | | | Thessaloniki | OWN-1 | P35 | 4,050 |
| Thessaloniki | OWN-2 | P01 | 900 | | | P02 | 1,177 |
| | | P12 | 540 | | | P34 | 615 |
| | | P17 | 810 | | | P64 | 158 |
| | | P88 | 1,013 | | | total | **6,000** |
| | | P90 | 1,244 | | | | |
| | | total | **4,506** | Xanthi | OWN-2 | P02 | 1,548 |
| | | | | | | P04 | 864 |
| | | | | | | P01 | 852 |
| | | | | | | P10 | 374 |
| | | | | | | P12 | 175 |
| | | | | | | P05 | 149 |
| | | | | | | P13 | 144 |
| | | | | | | P64 | 45 |
| | | | | | | total | **4,151** |

### 7.5.2 Case Study II

This case is concerned with the multi-site production and logistics operations planning problem. The production and distribution network under consideration consists of two production sites (situated in Serres, and Karditsa) and five distribution centers, as shown in Figure 7.6. Processing units J1 to J4 are installed in the production plant situated in Serres while processing units J1 to J3 are installed on the production site of Karditsa.

The production plants have to fully meet the demand for all products. The production site in Serres owns a pair of transportation trucks (OWN-1, and OWN-2) with $\psi_{sl} = 50$ € and minimum and maximum load capacity, $\varepsilon_l^{min} = 1{,}000$ kg and $\varepsilon_l^{max} = 6{,}000$ kg, respectively. These trucks can supply distribution centers located to Thessaloniki, and Xanthi. Ten third party logistics trucks (3PLT-1 to 3PLT-10) with $\psi_{sl} = 700$ € and $\varepsilon_l^{min} = 1{,}000$ kg and $\varepsilon_l^{max} = 12{,}000$ kg are also available in the production plant of Serres. The distribution centers of Thessaloniki, Athens, and Ioannina can be supplied from the production site in Serres by 3PLT trucks. The production facility located to Karditsa owns a pair of transportation trucks (OWN-3, and OWN-4) with $\psi_{sl} = 50$ € and minimum and maximum load capacity, $\varepsilon_l^{min} = 1{,}000$ kg and $\varepsilon_l^{max} = 6{,}000$ kg, respectively. These trucks can supply distribution centers situated in Ioannina, and Patras. In addition, six third party logistics trucks (3PLT-11 to 3PLT-16) with $\psi_{sl} = 600$ € and $\varepsilon_l^{min} = 1{,}000$


**Figure 7.6:** Case Study II: Production sites and distribution centers locations.

kg and $\varepsilon_l^{max}$ = 12,000 kg are available in the production facility of Karditsa. The distribution centers of Thessaloniki, Athens, and Ioannina can be supplied from production site in Karditsa by 3PLT trucks. Notice that the production facility in Serres cannot supply the distribution center of Patras, and the production plant in Karditsa cannot supply the distribution center of Xanthi. The remaining data are not provided due to confidentiality issues.

The resulting mathematical model consists of 19,371 constraints, 4,070 binary variables, and 35,614 continuous variables. The optimal solution corresponds to 520,047 € of total cost, and it was obtained in 379.2 CPU s after exploring 1,837 nodes in the branch-and-bound tree. The profile of total produced quantities, inventories, and transported quantities for each planning period are illustrated in Figure 7.7. The production plants work from n0 to n5 period, where period n0 is an overtime period (i.e., higher operating costs). The highest production is observed in period n5, with 129,723 kg of total production. Generally speaking, inventory levels are maintained low throughout the planning horizon, with an exception in period n6 where a relatively high inventory level of 64,702 kg is detected. Transportation operations from production facilities to distribution centers are realized from n3 to n7 period. In period n7, a total of 124,171 kg of products is transferred from the production sites to the distribution centers. Figure 7.8 illustrates the total, and per production facility, cost breakdown as well as the contribution of each cost term on the total cost. Inventories cost represent 14.5% of the total cost while transportation costs stands for the 35.7% of the total cost. Production costs (i.e., operating, recipe, unit utilization, and changeovers costs) represent 49.8% of the total cost. It should be also noted that 62.5% of the total inventory cost, and the 57.5% of the total transportation cost is occurred in production site of Serres.

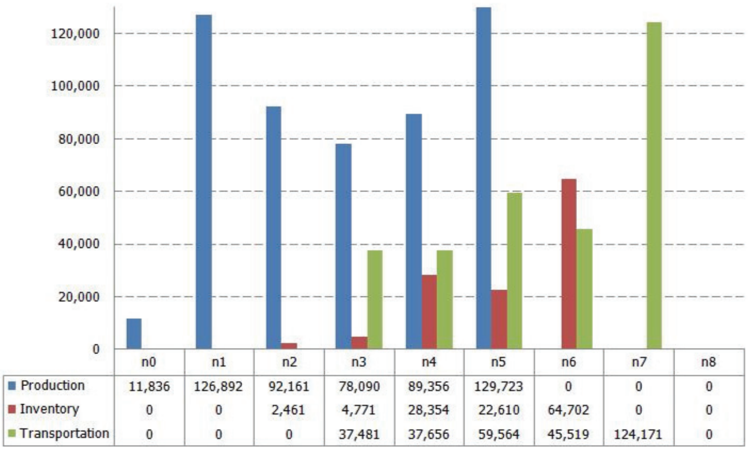



**Figure 7.7:** Case Study II: Total production, inventory and transportation profiles per period (kg).

**Table 7.4:** Case Study II: Transportation plan (kg).

| Site | Truck | Distr. Center | n0 | n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SERRES | OWN-1 | Thessaloniki | 0 | 0 | 0 | 6,000 | 0 | 0 | 6,000 | 0 | 0 |
| | | Xanthi | 0 | 0 | 0 | 0 | 4,151 | 3,241 | 0 | 5,924 | 0 |
| | OWN-2 | Thessaloniki | 0 | 0 | 0 | 2,256 | 6,000 | 0 | 6,000 | 6,000 | 0 |
| | | Xanthi | 0 | 0 | 0 | 0 | 0 | 6,000 | 0 | 0 | 0 |
| | 3PLT-1 | Athens | 0 | 0 | 0 | 2,590 | 11,148 | 0 | 0 | 0 | 0 |
| | | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 3,475 | 12,000 | 11,886 | 0 |
| | | Ioannina | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 3PLT-2 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Thessaloniki | 0 | 0 | 0 | 0 | 6,450 | 12,000 | 0 | 11,932 | 0 |
| | | Ioannina | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 3PLT-3 | Athens | 0 | 0 | 0 | 0 | 0 | 7,124 | 0 | 0 | 0 |
| | | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Ioannina | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6,549 | 0 |
| | 3PLT-4 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10,697 | 0 |
| | | Ioannina | 0 | 0 | 0 | 0 | 0 | 4,830 | 0 | 0 | 0 |
| | 3PLT-5 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10,582 | 0 |
| | | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Ioannina | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 3PLT-6 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6,812 | 0 |
| | | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Ioannina | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| KARDITSA | OWN-3 | Ioannina | 0 | 0 | 0 | 0 | 0 | 6,000 | 0 | 5,652 | 0 |
| | | Patras | 0 | 0 | 0 | 5,189 | 3,906 | 0 | 5,769 | 0 | 0 |
| | OWN-4 | Ioannina | 0 | 0 | 0 | 6,000 | 0 | 6,000 | 0 | 5,569 | 0 |
| | | Patras | 0 | 0 | 0 | 0 | 6,000 | 0 | 1,000 | 0 | 0 |
| | 3PLT-11 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 11,500 | 12,000 | 0 |
| | | Thessaloniki | 0 | 0 | 0 | 2,250 | 0 | 0 | 0 | 0 | 0 |
| | | Ioannina | 0 | 0 | 0 | 0 | 0 | 10,894 | 0 | 0 | 0 |
| | 3PLT-12 | Athens | 0 | 0 | 0 | 6,741 | 0 | 0 | 0 | 0 | 0 |
| | | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 2,250 | 12,000 | 0 |
| | | Ioannina | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 3PLT-13 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 1,000 | 0 | 0 |
| | | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6,567 | 0 |
| | | Ioannina | 0 | 0 | 0 | 6,454 | 0 | 0 | 0 | 0 | 0 |
| | 3PLT-14 | Athens | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12,000 | 0 |
| | | Thessaloniki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Ioannina | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

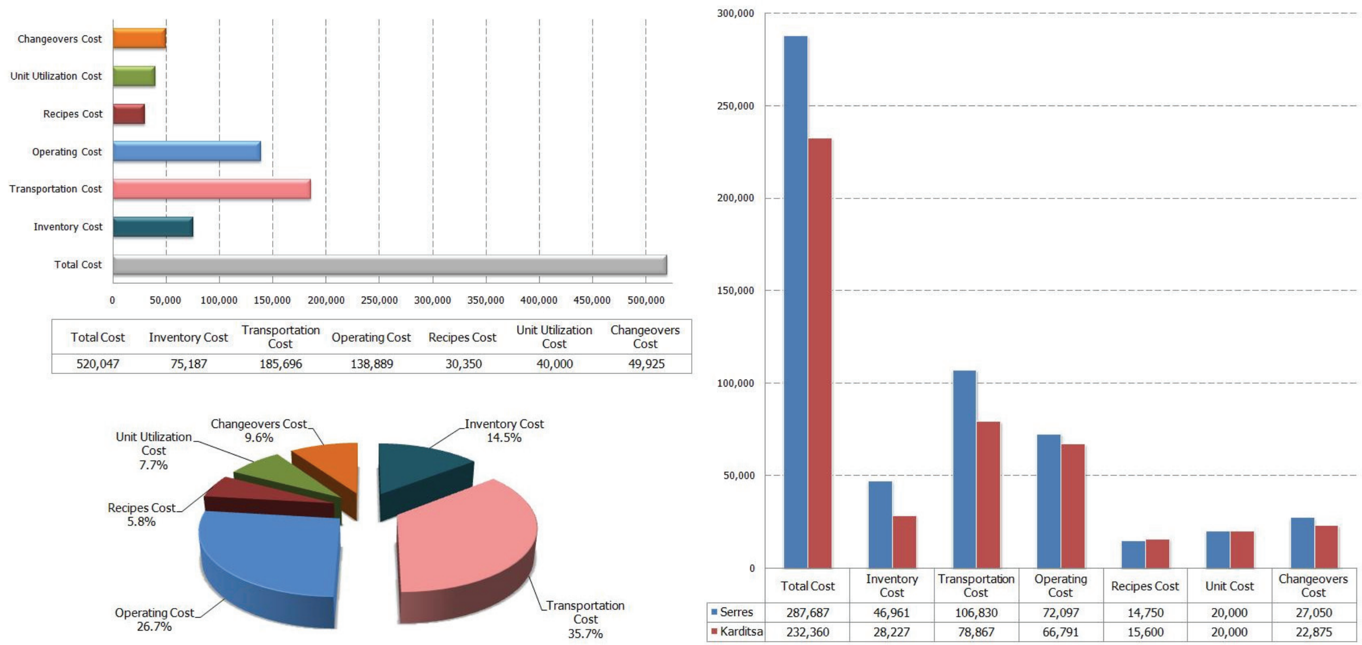
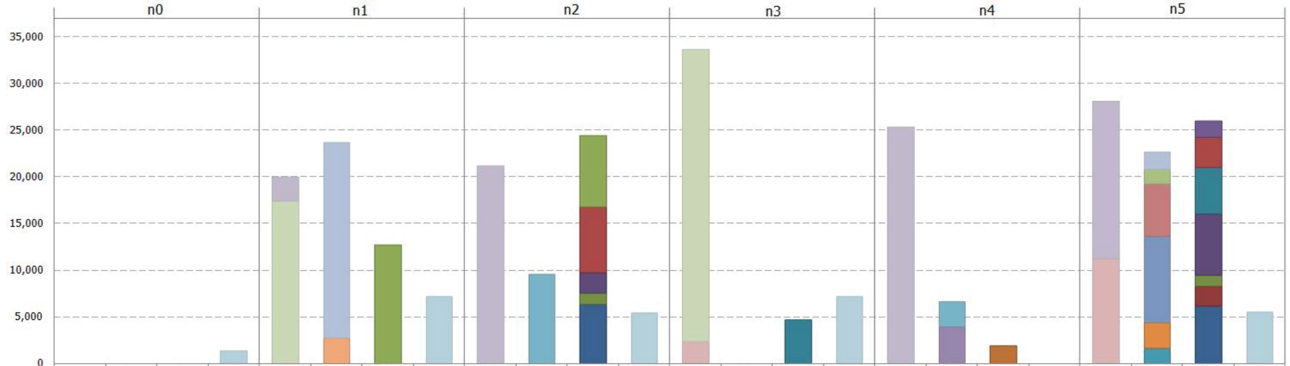
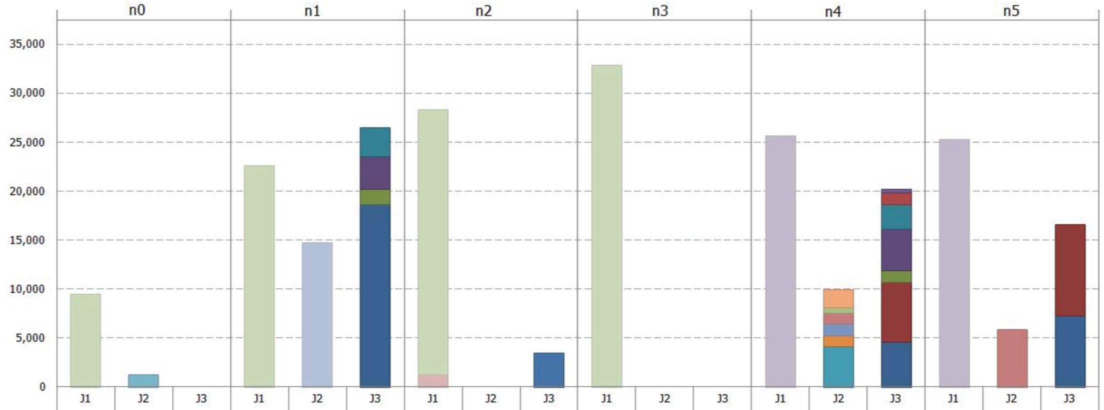



**Figure 7.8:** Case Study II: Total cost breakdown (€) and cost terms contribution.




| | n0 | | | | n1 | | | | n2 | | | | n3 | | | | n4 | | | | n5 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | J1 | J2 | J3 | J4 | J1 | J2 | J3 | J4 | J1 | J2 | J3 | J4 | J1 | J2 | J3 | J4 | J1 | J2 | J3 | J4 | J1 | J2 | J3 | J4 |
| F23 | 0 | 0 | 0 | 1,261 | 0 | 0 | 0 | 7,073 | 0 | 0 | 0 | 5,350 | 0 | 0 | 0 | 7,073 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5,453 |
| F22 | 0 | 0 | 0 | 0 | 2,513 | 0 | 0 | 0 | 21,184 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 25,356 | 0 | 0 | 0 | 16,904 | 0 | 0 | 0 |
| F21 | 0 | 0 | 0 | 0 | 17,400 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 31,388 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,244 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11,160 | 0 | 0 | 0 |
| F19 | 0 | 0 | 0 | 0 | 0 | 20,952 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,874 | 0 | 0 |
| F18 | 0 | 0 | 0 | 0 | 0 | 2,673 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9,467 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,705 | 0 | 0 | 0 | 0 | 0 | 0 |
| F16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3,830 | 0 | 0 | 0 | 0 | 0 | 0 |
| F15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,536 | 0 | 0 |
| F14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5,637 | 0 | 0 |
| F13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9,269 | 0 | 0 |
| F12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,787 | 0 | 0 |
| F11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,539 | 0 | 0 |
| F10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,763 | 0 |
| F09 | 0 | 0 | 0 | 0 | 0 | 0 | 12,672 | 0 | 0 | 0 | 7,683 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F08 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7,024 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3,183 | 0 |
| F07 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F06 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,831 | 0 | 0 | 0 | 0 | 0 |
| F05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4,604 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4,996 | 0 |
| F04 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,307 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6,579 | 0 |
| F03 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,227 | 0 |
| F02 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,089 | 0 |
| F01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6,225 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6,121 | 0 |

**Figure 7.9:** Case Study II: Production plan for families in production facility in Serres (kg).




| | n0 | | | n1 | | | n2 | | | n3 | | | n4 | | | n5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | J1 | J2 | J3 | J1 | J2 | J3 | J1 | J2 | J3 | J1 | J2 | J3 | J1 | J2 | J3 | J1 | J2 | J3 |
| F22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 25,597 | 0 | 0 | 25,271 | 0 | 0 |
| F21 | 9,375 | 0 | 0 | 22,500 | 0 | 0 | 27,105 | 0 | 0 | 32,783 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F20 | 0 | 0 | 0 | 0 | 0 | 0 | 1,200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F19 | 0 | 0 | 0 | 0 | 14,667 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,860 | 0 | 0 | 0 | 0 |
| F17 | 0 | 1,200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 575 | 0 | 0 | 0 | 0 |
| F14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,100 | 0 | 0 | 5,827 | 0 |
| F13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,200 | 0 | 0 | 0 | 0 |
| F12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,100 | 0 | 0 | 0 | 0 |
| F11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4,050 | 0 | 0 | 0 | 0 |
| F10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 425 | 0 | 0 | 0 |
| F09 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F08 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,125 | 0 | 0 | 0 |
| F07 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3,415 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F06 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F05 | 0 | 0 | 0 | 0 | 0 | 2,939 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,541 | 0 | 0 | 0 |
| F04 | 0 | 0 | 0 | 0 | 0 | 3,337 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4,231 | 0 | 0 | 0 |
| F03 | 0 | 0 | 0 | 0 | 0 | 1,591 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,200 | 0 | 0 | 0 |
| F02 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6,098 | 0 | 0 | 9,309 |
| F01 | 0 | 0 | 0 | 0 | 0 | 18,577 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4,532 | 0 | 0 | 7,199 |

**Figure 7.10:** Case Study II: Production plan for families in production facility in Karditsa (kg).

Figure 7.9 presents the detailed production plan for the production plant in Serres, and Figure 7.10 shows the detailed production plan for the production facility of Karditsa. The sequences between families are predetermined (see Table 7.1). The total amount of products transported from production facilities to distribution centers by transportation trucks is given in Table 7.4. A total of 14 trucks are needed in period n7, wherein the peak of logistics operations is observed, as illustrated in Figure 7.7. Finally, the proposed MIP model also generates the detailed transportation plan for each product (i.e., assignment of product to truck, assignment of truck to distribution center, and quantity of product transported by each truck).

## 7.6 Concluding Remarks

In this chapter, we have developed a novel MIP formulation, based on the definition of families of products, for the simultaneous optimization of single- or multi-site production and logistics operations in semicontinuous process industries. Two industrial-size case studies for a real-life dairy industry have been solved to optimality in order to shed light on the special features of the suggested MIP model. It should be emphasized that while production timing and sequencing decisions are taken for families (rather than for products), material balances are realized for each specific product, thus permitting the detailed optimization of production, inventory, and transportation costs. Additionally, alternative transportation modes are considered for the delivery of final products from production sites to distribution centers, a reality that most of the current approaches totally neglect. Despite the complexity of the problems addressed, the proposed approach appears a remarkable computational performance. Finally, it is worth mentioning that the proposed MIP model aims at being the core element of a computer-aided advanced planning system in order to facilitate decision making in relevant industrial environments by better coordination of production and distribution activities.

## 7.7 Nomenclature

**Indices / Sets**

| | |
|---|---|
| $d \in D$ | distribution centers |
| $f, f' \in F$ | product families (families) |
| $j, j' \in J$ | processing unit types (units) |
| $l \in L$ | transportation trucks |
| $n, n' \in N$ | planning time periods |
| $p \in P$ | products |
| $r \in R$ | batch recipe types (recipes) |
| $s \in S$ | production sites |

**Subsets**

| | |
|---|---|
| $D_l$ | distribution centers $d$ that can be supplied by truck $l$ |

| | |
|---|---|
| $D_s$ | distribution centers $d$ that can be supplied by production site $s$ |
| $F_j$ | families $f$ that can be processed on unit $j$ |
| $F_r$ | families $f$ that have the same batch recipe type origin $r$ |
| $J_f$ | available units $j$ to process family $f$ |
| $J_p$ | units $j$ that can process product $p$ |
| $J_s$ | processing units $j$ that are installed on production site $s$ |
| $L_{sd}$ | transportation trucks $l$ that can transfer products from production site $s$ to distribution center $d$ |
| $P^a$ | products $p$ that are destined for international customers or big national supermarket clients, which have their own trucks |
| $P_f$ | products $p$ that belong to the same family $f$ |
| $P_r$ | products $p$ that have the same batch recipe $r$ origin |
| $R_f$ | batch recipe origin $r$ for family $f$ |
| $R_j$ | batch recipes $r$ that can be processed on unit $j$ |
| $R_p$ | products $p$ that come from batch recipe $r$ |
| $S_d$ | production sites $s$ that can supply distribution center $d$ |
| $S_l$ | production sites $s$ that can use transportation truck $l$ |

**Parameters**

| | |
|---|---|
| $\alpha_{sjn}$ | daily opening setup time for every unit $j$ of production site $s$ in period $n$; accounts for the pasteurization and homogenization stages |
| $\beta_{jn}$ | daily shutdown time for every unit $j$ of production site $s$ in period $n$; cleaning of production line for hygienic and quality reasons |
| $\gamma_{ff'sj}$ | changeover time between family $f$ and family $f'$ on unit $j$ of production site $s$; accounts for cleaning and sterilizing operations |
| $\delta_{spj}$ | setup time for product $p$ in unit $j$ of production site $s$ |
| $\varepsilon_l^{max}$ | maximum capacity of transportation truck $l$ |
| $\varepsilon_l^{min}$ | minimum capacity of transportation truck $l$ |
| $\zeta_{dpn}$ | demand for product $p \notin P^a$ of customers supplied by distribution center $d$ at time $n$ |
| $\zeta_{pn}^a$ | demand for product $p \in P^a$ at time $n$ |
| $\theta_{psjn}$ | variable operating cost for product $p$ on processing unit $j$ of production site $s$ in period $n$; includes labor and utilities costs |
| $\lambda_p$ | minimum cooling storage time for processed products (in periods n) |
| $M_{sjn}$ | a big number |
| $\mu_{srn}^{max}$ | maximum production capacity of batch recipe $r$ in production site $s$ in period $n$ |
| $\mu_{rn}^{min}$ | minimum produced quantity of batch recipe $r$ in production site $s$ in period $n$; accounts for pasteurization and fermentation tanks capacity restrictions |
| $\nu_{sjn}$ | fixed cost for utilizing unit $j$ of production site $s$ in period $n$ |
| $\xi_{spn}$ | inventory cost for product $p$ in production site $s$ in period $n$ |
| $o_{sjn}$ | additional unit preparation time for processing unit $j$ of production site $s$ in periods $n$ |
| $\pi_{psjn}^{max}$ | maximum production run for product $p$ on unit $j$ of production site $s$ in period $n$ |
| $\pi_{psjn}^{min}$ | minimum production run for product $p$ on unit $j$ of production site $s$ in period $n$ |
| $\rho_{psj}$ | processing rate for product $p$ on unit $j \in J_p$ of production site $s$ |

| | |
|---|---|
| $\sigma_{srn}$ | release time for batch recipe $r$ in production site $s$ in period $n$ |
| $\tau_r$ | minimum time for preparing batch recipe $r$; for producing stirred yogurt products stands for the minimum fermentation time, while for set yogurt products reflects the minimum cooling time before the packing stage |
| $\upsilon_{sdl}$ | variable cost for transferring products from production site $s$ to distribution center $d$ by truck $l$ |
| $\phi_{ff'sjn}$ | changeover cost between family $f$ and family $f'$ in unit $j$ of production site $s$ in period $n$; accounts for cleaning and sterilizing operations |
| $\chi_{srn}$ | cost for producing batch recipe $r$ in production site $s$ in period $n$ |
| $\psi_{sl}$ | fix cost for contracting transportation truck $l$ to carry products from production site $s$ |
| $\omega_{sjn}$ | physical available processing time in period $n$ |

**Continuous Variables**

| | |
|---|---|
| $C_{fsjn}$ | completion time for family $f$ in unit $j$ of production site $s$ in period $n$ |
| $I_{spn}$ | inventory of product $p$ in production site $s$ at time $n$ |
| $Q_{psjn}$ | produced amount of product $p$ in unit $j$ of production site $s$ in period $n$ |
| $\bar{Q}_{psn}$ | total produced amount of product $p$ in production site $s$ in period $n$ |
| $T_{fsjn}$ | processing time for family $f$ in unit $j$ of production site $s$ in period $n$ |
| $U_{sdlpn}$ | quantity of product $p \notin P^a$ transported from production site $s$ to distribution center $d$ by truck $l$ in period $n$ |
| $\bar{U}_{sdln}$ | total transported quantity from production site $s$ to distribution center $d$ by truck $l$ in period $n$ |
| $U^a_{spn}$ | quantity of product $p \in P^a$ transported from production site $s$ to international market or big national supermarket clients by customer trucks in period $n$ |

**Binary Variables**

| | |
|---|---|
| $V_{sjn}$ | $= 1$, if unit $j$ of production site $s$ is used in period $n$ |
| $W_{srn}$ | $= 1$, if batch recipe $r$ is produced in production site $s$ in period $n$ |
| $X_{ff'sjn}$ | $= 1$, if family $f'$ is processed exactly after family $f$, when both are assigned to the same unit $j$ of production site $s$ in the period $n$ |
| $Y_{fsjn}$ | $= 1$, if family $f$ is assigned to unit $j$ of production site $s$ in period $n$ |
| $\bar{Y}_{psjn}$ | $= 1$, if product $p$ is assigned to unit $j$ of production site $s$ in period $n$ |
| $Z_{sdln}$ | $= 1$, if transportation truck $l$ transfers material from production facility $s$ to distribution center $d$ in period $n$ |

Part IV

# Batch Processes

# Chapter 8

# Production Scheduling in Large-scale Multistage Batch Process Industries

## 8.1 Introduction

Nowadays, it is widely recognized that the current gap between practice and theory in the area of short-term scheduling needs to be bridged, as clearly remarked in Méndez *et al.* (2006) and Ruiz, Serifoglu, and Urlings (2008). New academic developments are mostly tested on relatively small problems whereas current real-world industrial applications consist of hundreds of batches, numerous multiple units available for each task and long sequence of processing stages. Additionally, there exist a wide range of operational constraints which should be taken into account in order to guarantee the feasibility of the proposed schedule. Most industrial problems are very hard-constrained, thus optimization solvers have to find optimal or near-optimal solutions in a huge search space with a relatively small feasible region. This fact may result in huge computational requirements which often do not allow finding even good feasible solutions, which is definitely not suitable for industrial environments.

Since most industrial scheduling problems are highly combinatorial and complex decision-making processes, they rarely can be solved to optimality within a reasonable computational time. In addition, the computational effort to find a good solution tends to be as important as the scheduling problem itself; since industry demands solutions that are both optimal, or at least close-optimal, and quick to be reached.

In this chapter, an efficient systematic iterative MIP-based solution strategy for solving real-world scheduling problems in multiproduct multistage batch plants is presented. A novel precedence-based concept has been also developed here. The proposed solution strategy consists of a constructive step, wherein a feasible and initial solution is rapidly generated by following an iterative insertion procedure, and an improvement step, wherein the initial solution is systematically enhanced by implementing iteratively several rescheduling techniques; based on the mathematical model. A salient feature of our approach is that the scheduler can maintain the number of decisions at a reasonable level thus reducing appropriately the search space. This usually results in manageable model sizes that often favors a more stable and predictable optimization model behavior.

## 8.2 Problem Statement

The problem under consideration is concerned with industrial-scale multiproduct multistage batch processes with the following features:

(i) A set of product orders $i \in I$ should be processed by following a predefined sequence of processing stages $s \in S$ with, in general, unrelated processing units $j \in J$ working in parallel.

(ii) Each product order $i$ comprises a single batch that must follow a set of processing stages $s \in S_i$.

(iii) Some products $i$ may skip certain processing stages $s \notin S_i$, since different production recipes are considered.

(iv) A product order $i$ can be processed in a specific subset of units $j \in J_i$. Similarly, a processing stage $s$ can be processed in a specific subset of units $j \in J_s$.

(v) Transition times between consecutive product orders involve two terms. The first depends on both the unit and the order being processed ($\pi_{ij}$) while the second also varies with the order previously manufactured in that unit ($\gamma_{ii'j}$). Transition times must be explicitly taken into account in the schedule generation process since they are usually of the same order of magnitude or even larger than the processing times. Consequently, they become a very critical feature when scheduling real-world batch processes such as pharmaceuticals, chemicals, food, etc.

(vi) Model parameters like order due dates ($\delta_i$), processing times ($\tau_{isj}$), unit-dependent setup times ($\pi_{ij}$), sequence-dependent setup (or simply changeover) times ($\gamma_{ii'j}$) and costs ($\xi_{ii'j}$), order release times ($o_i$), unit available times ($\varepsilon_j$), and operating cost ($\psi$) are all deterministic.

(vii) Once the processing of an order in a given stage is started, it should be carried out until completion without interruption (non-preemptive mode).

(viii) Mixing or splitting of product orders is not allowed.

The key decision variables are:

(i) the allocation of products $i$ to units $j \in J_i$ per stage, $Y_{isj}$;

(ii) the relative sequence for any pair of products $i$, $i'$ in unit $j \in (J_i \cap J_{i'})$, $X_{ii'j}$;

(iii) the completion time of products $i$ in processing stage $s \in S_i$, $C_{is}$.

Alternative objective functions can be considered, such as the minimization of makespan, total weighted lateness or total operating and changeovers cost.

## 8.3 Mathematical Formulations

In this section, two batch-oriented mathematical models are presented for solving scheduling problems in multiproduct multistage batch plants. Both models are based on a continuous-time domain and utilize sequencing variables. The first model is based on the general (global) precedence sequencing concept, and the latter one is based on the unit-specific general precedence sequencing concept which has developed as a part of this thesis (see Appendix B).

Global precedence formulations result in models with small model size and they are computationally faster on average. However, a drawback of these models is that they cannot optimize objectives containing changeover issues (e.g., minimization of changeover costs). For this reason, a unit-specific general precedence model, for scheduling multiproduct multistage batch plants, able to cope with a wide variety of objective functions, is also presented in the context of a more general mathematical formulation.

It is worth noticing that the MIP models, presented in this work, are not claimed to be either the fastest or the tightest. However, for the sake of clarity of the presentation of the proposed MIP-based solution strategy, the MIP models adopted were entirely developed along this thesis rather than using readily available models from the literature. Otherwise, other mathematical formulations found in the literature could be used as core MIP models in the proposed solution strategy. The description of the mathematical frameworks used in this work follows.

### 8.3.1 A general precedence multistage scheduling framework

The problem under study can be formulated by the following sets of constraints using the general precedence notion:

$$\sum_{j \in (J_i \cap J_s)} Y_{isj} = 1 \qquad \forall i \in I^{in}, s \in S_i \tag{8.1}$$

$$C_{is} \geq \sum_{j \in (J_i \cap J_s)} (max[\varepsilon_j, o_i] + \pi_{ij} + \tau_{isj}) Y_{isj} \qquad \forall i \in I^{in}, s \in S_i : s = 1 \tag{8.2}$$

$$C_{is} - \sum_{j \in (J_i \cap J_s)} (\pi_{ij} + \tau_{isj}) Y_{isj} = C_{is-1} + W_{is-1} + \mu_{s-1s} \qquad \forall i \in I^{in}, s \in S_i : s > 1 \tag{8.3}$$

$$C_{is} + \gamma_{ii'j} \leq C_{i's} - \pi_{i'j} - \tau_{i'sj} + M(1 - X_{ii'j}) + M(2 - Y_{isj} - Y_{i'sj}) \quad \forall i \in I^{in}, i' \in I^{in}, s \in S_i, j \in (J_s \cap J_i \cap J_{i'}) : i' > i \tag{8.4}$$

$$C_{i's} + \gamma_{i'ij} \leq C_{is} - \pi_{ij} - \tau_{isj} + M X_{ii'j} + M(2 - Y_{isj} - Y_{i'sj}) \quad \forall i \in I^{in}, i' \in I^{in}, s \in S_i, j \in (J_s \cap J_i \cap J_{i'}) : i' > i \tag{8.5}$$

$$\begin{aligned} Y_{isj} \in \{0,1\} \quad & \forall i \in I^{in}, s \in S_i, j \in (J_s \cap J_i) \\ X_{ii'j} \in \{0,1\} \quad & \forall i \in I^{in}, i' \in I^{in}, j \in (J_i \cap J_i') : i' \neq i \\ W_{is} \geq 0 \quad & \forall i \in I^{in}, s \in S_i : s < S \\ C_{is} \geq 0 \quad & \forall i \in I^{in}, s \in S_i \end{aligned} \tag{8.6}$$

Constraint set (8.1) ensures that every product order goes through one unit $j \in (J_s \cap J_i)$ at each stage $s \in S_i$. Constraint set (8.2) defines the completion time of the first stage for every product. Notice that this set of constraints takes into account possible release order $o_i$ and available unit $\varepsilon_j$ times. Constraint set (8.3) provides the timing for every product order between consecutive stages. This set of constraints allows for the consideration of possible transferring times between two sequential stages. The positive variable $W_{is-1}$ reflects the wait time of each batch product before proceeding to the following processing stage. Note that in a Zero Wait (ZW) storage policy $W_{is-1}$ is set to zero. In Unlimited Intermediate Storage (UIS) policy, $W_{is-1}$ is left free or, alternatively, it can be eliminated and the equality can be substituted by a greater-or-equal inequality. In order to model storage policies like Non Intermediate Storage (NIS) and Finite Intermediate Storage (FIS), appropriate sets of constrains found in the literature can be easily added to the current model. Constraint sets (8.4) and (8.5) define the relative sequencing of product batches at each processing unit. These sets of big-M constraints force the starting time of a product $i'$ to be greater than the completion time of whichever product $i$ processed beforehand. Note that $X_{ii'j}$ corresponds to the global sequencing binary variable. Have in mind that $X_{ii'j}$ is active (i.e., $X_{ii'j} = 1$) for all product batches $i'$ that are processed after product batch $i$. Finally, the decision variables are defined by (8.6). Henceforth, we will refer to the MIP model that constitutes by constraint sets (8.1) to (8.6) as GP.

### 8.3.2 A unit-specific general precedence multistage scheduling framework

The following sets of constraints are proposed for scheduling problems where the changeover issues should be optimally integrated into the optimization framework. The constraints are:

$$\sum_{j \in (J_i \cap J_s)} Y_{isj} = 1 \qquad \forall i \in I^{in}, s \in S_i \tag{8.7}$$

$$C_{is} \geq \sum_{j \in (J_i \cap J_s)} (max[\varepsilon_j, o_i] + \pi_{ij} + \tau_{isj}) Y_{isj} + \sum_{i' \neq i} \sum_{j \in (J_s \cap J_i \cap J_{i'})} \gamma_{i'ij} \bar{X}_{i'ij} \qquad \forall i \in I^{in}, s \in S_i : s = 1 \tag{8.8}$$

$$C_{is} - \sum_{j \in (J_i \cap J_s)} (\pi_{ij} + \tau_{isj}) Y_{isj} = C_{is-1} + W_{is-1} + \mu_{s-1s} \qquad \forall i \in I^{in}, s \in S_i : s > 1 \tag{8.9}$$

$$C_{is} + \gamma_{ii'j} \bar{X}_{ii'j} \leq C_{i's} - \pi_{i'j} - \tau_{i'sj} + M(1 - X_{ii'j}) \qquad \forall i \in I^{in}, i' \in I^{in}, s \in S_i, j \in (J_s \cap J_i \cap J_{i'}) : i' \neq i \tag{8.10}$$

$$Y_{isj} + Y_{i'sj} \leq 1 + X_{ii'j} + X_{i'ij} \qquad \forall i \in I^{in}, i' \in I^{in}, s \in S_i, j \in (J_s \cap J_i \cap J_{i'}) : i' > i \tag{8.11}$$

$$2(X_{ii'j} + X_{i'ij}) \leq Y_{isj} + Y_{i'sj} \qquad \forall i \in I^{in}, i' \in I^{in}, s \in S_i, j \in (J_s \cap J_i \cap J_{i'}) : i' > i \tag{8.12}$$

$$Z_{ii'j} = \sum_{i'' \in I^{in} : i'' \neq [i,i']} (X_{ii''j} - X_{i'i''j}) + M(1 - X_{ii'j}) \qquad \forall i \in I^{in}, i' \in I^{in}, j \in (J_i \cap J_{i'}) : i' \neq i \tag{8.13}$$

$$Z_{ii'j} + \bar{X}_{ii'j} \geq 1 \qquad \forall i \in I^{in}, i' \in I^{in}, j \in (J_i \cap J_{i'}) : i' \neq i \tag{8.14}$$

$$\begin{aligned} Y_{isj} \in \{0,1\} \quad & \forall i \in I^{in}, s \in S_i, j \in (J_s \cap J_i) \\ X_{ii'j} \in \{0,1\} \ \& \ \bar{X}_{ii'j} \in \{0,1\} \quad & \forall i \in I^{in}, i' \in I^{in}, j \in (J_i \cap J_i') : i' \neq i \\ Z_{ii'j} \in \Re \quad & \forall i \in I^{in}, i' \in I^{in}, j \in (J_i \cap J_i') : i' \neq i \\ W_{is} \geq 0 \quad & \forall i \in I^{in}, s \in S_i : s < S \\ C_{is} \geq 0 \quad & \forall i \in I^{in}, s \in S_i \end{aligned} \tag{8.15}$$

Constraint set (8.7) forces that every product order goes through one unit $j \in (J_s \cap J_i)$ at each stage $s \in S_i$. Constraint set (8.8) determines the completion time of the first stage for every product. Notice that $\bar{X}_{ii'j}$ is the unit-specific immediate precedence binary variable. Constraint set (8.9) defines the timing for every product order between to consecutive stages and is similar to constraint set (8.3) of the GP model. Constraint sets (8.10) to (8.12) define the relative sequencing of product batches at each processing unit. Big-M constraint set (8.10) forces the starting time of a product batch $i'$ to be greater than the completion time of whichever product batch $i$ processed beforehand at the same unit. Constraint sets (8.11) and (8.12) state that when two product batches are allocated to the same unit (i.e., $Y_{isj} = Y_{i'sj} = 1$), one of the two global sequencing binary variables $X_{ii'j}$ and $X_{i'ij}$ should be active. If the two product batches are not allocated to the same unit then $X_{ii'j} = X_{i'ij} = 0$. It is clear that two orders $i$ and $i'$ are consecutive only in

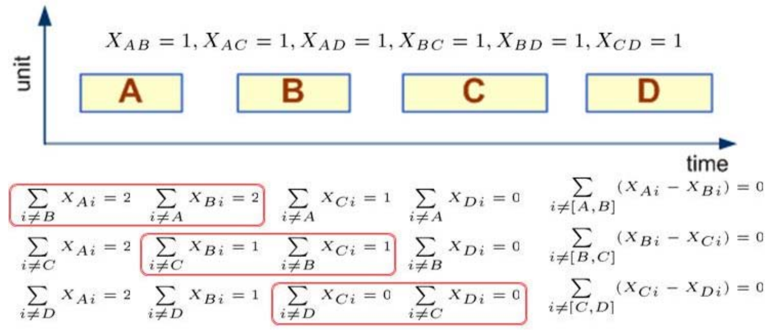



**Figure 8.1:** The unit-specific general precedence concept.

the case that $X_{ii'j} = 1$ and, moreover, when there is no other order $i''$ between them. In other words, two product batches $i$ and $i'$ are consecutive *if and only if* the total number of batches that are processed after batch $i$, if batch $i'$ is excluded, is equal to the total number of batches that are processed after batch $i'$, when batch $i$ is excluded; see Figure 8.1. Constraint sets (8.13) and (8.14) formulate this concept. Note that the auxiliary variable $Z_{ii'j}$ is zero whenever two products $i$ and $i'$ are sequentially processed in the same unit. The RHS of constraints (8.14) can be substituted by $X_{ii'j}$; in some instances this reduces the computational time. For a more detailed description the unit-specific general precedence concept refer to Appendix B. Finally, the decision variables are defined by (8.15). Henceforth, we will refer to the MIP model that constitutes by constraint sets (8.7) to (8.15) as USGP.

### 8.3.3 Objective functions

In this subsection, different optimization goals for solving the short-term scheduling problem under consideration, are reviewed.

**Makespan.** The time point at which all product orders are accomplished corresponds to the makespan, which is calculated by equation (8.16). The makespan objective is closely related to the throughput objective. For instance, minimizing the makespan in a parallel-machine environment with changeover times forces the scheduler to balance the load over the various machines and to minimize the sum of all the setup times in the critical bottleneck path (Pinedo & Chao, 1999).

$$\min \quad C_{max} \geq C_{is} \quad \forall i \in I^{in}, s \in S_i^{last} \tag{8.16}$$

**Total weighted lateness.** The minimization of a combined function of earliness and tardiness, as given in equation (8.17), is one of the most widely used objective functions in the scheduling literature. It is also known as weighted lateness. The weighing coefficients $\alpha_i$ and $\beta_i$ are used to specify the significance of every product order earliness or tardiness, respectively.

$$\min \quad \sum_{i \in I^{in}} (\alpha_i E_i + \beta_i T_i) \tag{8.17}$$

Earliness and tardiness for every product order $i$ are estimated by constraint set (8.18) and (8.19), respectively.

$$E_i \geq \delta_i - C_{is} \quad \forall i \in I^{in}, s \in S_i^{last} \tag{8.18}$$

$$T_i \geq C_{is} - \delta_i \quad \forall i \in I^{in}, s \in S_i^{last} \tag{8.19}$$

This objective, in a sense, accounts for minimizing storage and handling costs while maximizing service and customer satisfaction level.

**Operating and changeovers costs.** The minimization of operating and changeovers costs constitutes a reasonable goal in production environments where changeover costs are significant. The operating cost is denoted by $\psi$ and is defined as the cost for operating the production facility per time unit. Obviously, makespan corresponds to the total operating time. Parameter $\xi_{ii'j}$ stands for the changeover cost from product order $i$ to $i'$ in processing unit $j$.

$$\min \quad (\psi \, C_{max} + \sum_{i \in I^{in}} \sum_{i' \in I^{in}, i' \neq i} \sum_{j \in (J_i \cap J_{i'})} \xi_{ii'j} \bar{X}_{ii'j}) \tag{8.20}$$

## 8.4 The MIP-based Solution Strategy

Although the above mathematical formulations are able to describe a large number of scheduling problems, in practice, they can only solve problems of modest size. Given that the combinatorial complexity strongly increases with the number of product orders considered the solution of real-life industrial scheduling problems by exact methods is impossible. According to Herrmann (2006), algorithms that can find optimal solutions to these hard problems in a reasonable amount of time are unlikely to exist.

In a nutshell, the proposed MIP-based solution strategy has as a core a MIP scheduling framework and consists of two major procedure steps: (i) the constructive step, and (ii) the improvement step. The objective in the constructive step is the generation of a feasible schedule in a short amount of time. Afterwards, this schedule is gradually improved by implementing some elaborate rescheduling techniques, in the improvement step. As a sequence, the generation of feasible and fairly good schedules in reasonable computational times is favored. A description of the proposed solution strategy steps follows (see Figure 8.2).

### 8.4.1 Constructive step

In the constructive step, the large-scale scheduling problem is decomposed, in an iterative mode, into a subset of the involved product orders. This way the MIP

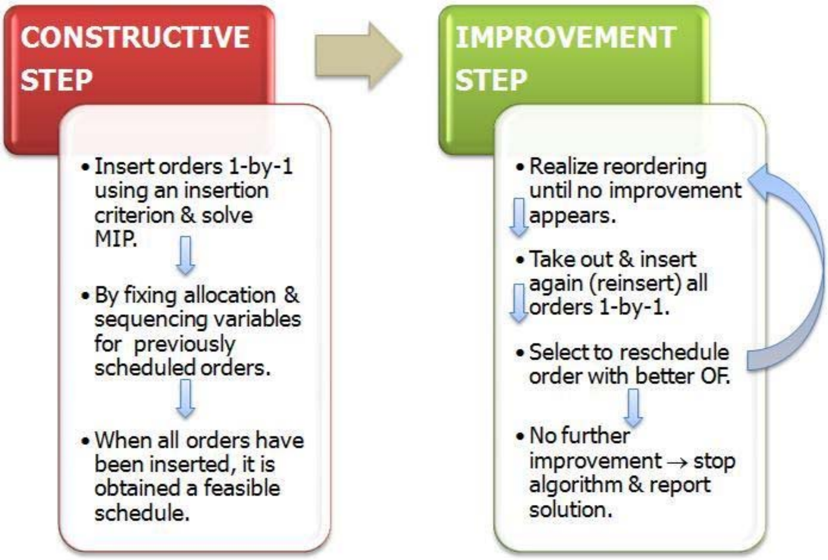



**Figure 8.2:** Representative scheme of the proposed MIP-based solution strategy.

solver search space is reduced and the resolution of the problem is favored. More specifically, a predefined number of product orders ($i \in I^{in}$) are scheduled (by solving the MIP model) at each iteration, until all product orders are finally scheduled. The user defines the number of product order for each iteration. It should be noted that the number of orders inserted into each iteration should be small enough to ensure a quick MIP model resolution for each iteration, and thus generating a feasible schedule in short time. In this study, it is proposed to insert (schedule) product orders one-by-one, since it has been observed, after a series of experiments, that insertion of a higher number of products per iteration: (i) does not guarantee a better constructive step solution, and (ii) is more computationally expensive.

The user should also specify the order that products are inserted into the constructive step procedure. An insertion criterion could be adopted in order to

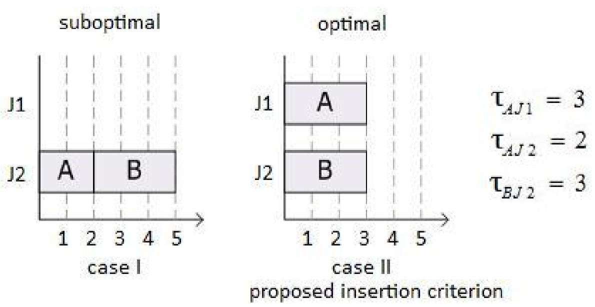



**Figure 8.3:** Illustrative example for insertion criterion.

ALLOWED SEQUENCES

| | | | |
|---|---|---|---|
| A | B | C | D |
| A | B | D | C |
| A | D | B | C |
| D | A | B | C |

inserted product order

previously scheduled product order

FORBIDDEN SEQUENCES

| | | | |
|---|---|---|---|
| A | C | B | D |
| A | C | D | B |
| A | D | B | C |
| B | A | C | D |
| B | A | D | C |
| B | C | A | D |
| B | C | D | A |
| B | D | A | C |
| B | D | C | A |
| C | A | B | D |
| C | A | D | B |
| C | B | A | D |
| C | B | D | A |
| C | D | A | B |
| C | D | B | A |
| D | A | C | B |
| D | B | A | C |
| D | B | C | A |
| D | C | A | B |
| D | C | B | A |

**Figure 8.4:** Illustrative example for allowed sequences in constructive step.

decrease the possibility of obtaining a bad constructive step solution. Here, it is proposed to insert as a priority products with less unit-stage allocation flexibility. In other words, products with less alternative units should be scheduled first. By doing so, unit allocation decisions are first taken for the less unit-stage-flexible products. Consider a single-stage two-product (A and B) batch plant with two parallel processing units (J1 and J2). The processing time of product A in unit J1 is 3 h and in unit J2 is 2 h. Product B can be only processed in unit J2 in 3 h. The minimization of makespan is the optimization goal. Consider the following insertion sequences: case I according to which product A is first inserted, which opposes our proposed insertion criterion, and case II that product B is first, which is in accordance with our proposed insertion criterion. As one can observe, the first insertion strategy (case I) results in a makespan of 5 h. Note that both products are allocated to unit J2. Following our insertion criterion (case II) a makespan of 3 h is obtained. Figure 8.3 illustrates the schedules for both cases.

After the resolution of the MIP model at each iteration, allocation and global sequencing binary variables for the already scheduled product orders are fixed. In other words, unit allocation decisions and relative sequencing relations between the already scheduled products cannot be modified in the following iterations. However, timing decisions may change thus permitting the insertion of new product orders among the previously scheduled product orders. Figure 8.4 delineates an illustrative example (single-stage products and single-unit) of the allowed sequences when a product D is inserted to a current schedule containing products A, B, and C. Note that just 4 sequences are permitted, instead of the 24 possible sequences, thus reducing significantly the computational effort. When all product orders have been inserted, a feasible schedule can be finally obtained in relatively short time.

Similar insertion methods have also been implemented to other types of scheduling problems by Nawaz, Enscore, and Ham (1983), Werner and Winkler (1995), Röslof *et al.* (2001), and Röslof, Harjunkoski, Westerlund, and Isaksson (2002). It is pointed out that the insertion order of product orders influences the quality of the solution. Therefore, a more detailed study and the development of other insertion criteria seems a promising future research direction for enhancing the proposed approach.

### 8.4.2 Improvement step

The initial feasible schedule provided by the constructive step can be systematically improved through reordering and/or reassignment MIP-based operations; in accordance with the main rescheduling concepts introduced by Röslof *et al.* (2001) and Méndez and Cerdá (2003a). The improvement step is a two-stage closed loop procedure that consists of the reordering and the reinsertion stage, which are performed sequentially until no improvement is observed. A description of the improvement step follows.

**Reordering stage** At this stage, unit allocation decisions, are fixed. Reordering actions are iteratively applied on the initial schedule, by solving a MIP model, until no further improvement is observed. A full unit reordering tactic results impractical due to the large number of batches and processing units in real-world industrial scheduling problems. Instead, the alternative of limited reordering operations may usually improve the current schedule with relatively low computational effort. It is common sense that there exists a strong trade-off between the degrees of freedom and the solution time. In an industrial environment, the scheduler should appropriately define the reordering tactic/limitations, followed in this step, depending on the complexity of the scheduling problem. A local reordering tactic is adopted in this study. Thus, in an attempt to maintain manageable model sizes, reordering of batches with their direct predecessor or successor is only allowed. An illustrative example is used here to highlight the local reordering computational benefits. Consider the reordering scheduling problem of 4 single-stage products (A, B, C, and D) on a single-unit. As Figure 8.5 shows, a local reordering policy will only examine 4 potential sequences instead of the 23 total possible sequences. On the one hand, the solution quality is probably decreased since one of the 19 unexplored sequences may yield a better solution. On the other hand, the optimization search space is significantly reduced. Keep in mind that considering the whole set of possible sequences impacts drastically the computational performance of the reordering step. Other less-limited reordering tactics could be also easily applied. More details are provided in the work of Méndez and Cerdá (2003a).

**Reinsertion stage** The schedule of the reordering step constitutes the initial schedule in the reinsertion stage. Here, unit allocation and relative sequencing decisions for a small number of product orders are left free by the scheduler. Let us refer to these product orders as reinserted orders. Allocation and relative sequencing decisions, among the non-reinserted orders, are fixed. In other words,

| A | B | C | D |
|---|---|---|---|
| A | B | D | C |
| A | C | B | D |
| A | C | D | B |
| A | D | B | C |
| A | D | C | B |
| B | A | C | D |
| B | A | D | C |
| B | C | A | D |
| B | C | D | A |
| B | D | A | C |
| B | D | C | A |

| | | | |
|---|---|---|---|
| C | A | B | D |
| C | A | D | B |
| C | B | A | D |
| C | B | D | A |
| C | D | A | B |
| C | D | B | A |
| D | A | B | C |
| D | A | C | B |
| D | B | A | C |
| D | B | C | A |
| D | C | A | B |
| D | C | B | A |

local reordering possible sequence

current schedule sequence

**Figure 8.5:** Illustrative example for local reordering.

some products orders are extracted from the current schedule, and they are reinserted aiming at improving the actual schedule. Note that the reinsertion stage is quite similar to the last iteration of the constructive step (see Figure 8.4). Since our scope is to propose a general standard algorithm for large-scale industrial scheduling problems, we adopt the lowest number of reinsertion orders (i.e., one at a time) in order to favor low solution times. However, the scheduler could set the number of reinserted orders depending on the specific scheduling problem. In the standard reinsertion stage, the number of iterations (reinsertions) equals the number of product orders. The solutions of all reinserted orders (iterations) are compared, and the best one is finally chosen as the solution of the reinsertion stage. Note that if the number of product orders is too high, someone could have preferred to end the reinsertion stage once a better solution (comparing it with the previous stage) is reached. This way computational savings are achieved. If the best solution at this stage is better than the solution of the reordering stage, the algorithm goes to the reordering stage again. Otherwise, the solution algorithm terminates and reports the best solution found.

In Appendix E, some illustrative pseudocodes can be found for the constructive and the improvement stage of our MIP-based solution strategy.

## 8.5 Pharmaceutical Production Process

A real-world multiproduct multistage pharmaceutical batch plant is studied in the current work. Recently, Castro *et al.* (2009) have also studied this pharmaceutical facility. More specifically, they solved two problem instances (for 30 and 50 product orders) minimizing the makespan under UIS policy. In this work, we use partially different sets of data (e.g., we introduce due dates, changeover costs) and we deal with more objective functions.

In this study, the short-term scheduling problem of a considerably high number of multistage product orders (30 and 60) using 17 processing units in the production plant is addressed. The production process has 6 processing stages, as Figure 8.6 depicts. Some products bypass the third processing stage S3. Changeover times are also explicitly considered thus increasing the complexity of the prob-

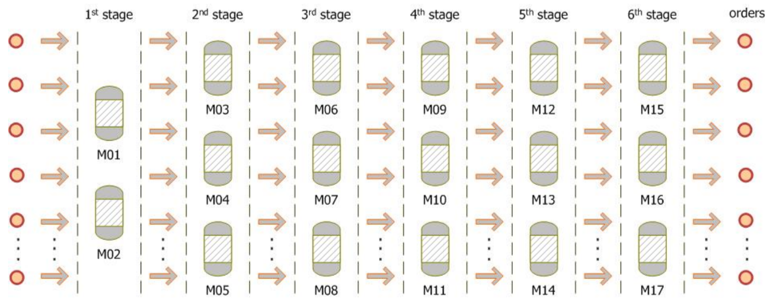



**Figure 8.6:** Pharmaceutical multistage process.

lem. An interesting feature of the production process is that in some processing stages changeover times are higher than the processing times. Changeover times are zero in the first stage S1 and 0.45 h in the second stage S2 among all products. Changeover times for the remaining stages (S3 – S6) and processing times can be found in Appendix F. Finally, changeover costs are defined as the multiplication of the impact factors given in Table 8.1 and the corresponding changeover time.

# 8.6 Experimental Studies

In this section, the problem instances details are firstly introduced and the results of these experimental studies are presented and discussed afterwards.

## 8.6.1 Details of problem instances

Twelve different problem instances have been solved. These case studies differ in: (i) the optimization goal (makespan, weighted lateness, and operating and changeover costs), (ii) the number of product orders (30 products, and 60 products), and (iii) the storage policy type (ZW, and UIS).

Notice that two batches for every product are considered in order to address the 60-product cases. Therefore, for instance, the product order P31 has the same processing characteristics with product order P01, where product order P32 has the same processing data with product order P02, and so on. Moreover, notice that the changeover times/costs between P01 and P31 are equal to the changeover times/costs between P01 and P01, as they are given in the data tables, and so on. For the problem instances where the optimization goal is the minimization of weighted lateness (i.e., PI.05 – PI.08), due dates for every product order are considered, according to Table 8.2. It should be noted that due dates for two

**Table 8.1:** Changeover costs' impact factors per time unit ($10^3$\$/h).

| Products | P01 - P10 | P11 - P20 | P21 - P30 |
|---|---|---|---|
| P01 - P10 | 0.36 | 0.27 | 0.27 |
| P11 - P20 | 0.27 | 0.45 | 0.27 |
| P21 - P30 | 0.27 | 0.27 | 0.54 |

**Table 8.2:** Due dates for product orders (h).

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| P01 | 30.6 | P16 | 20.7 | P31 | 34.2 | P46 | 54.0 |
| P02 | 16.2 | P17 | 14.4 | P32 | 37.8 | P47 | 49.5 |
| P03 | 23.4 | P18 | 23.4 | P33 | 37.8 | P48 | 46.8 |
| P04 | 18.0 | P19 | 20.7 | P34 | 48.6 | P49 | 52.2 |
| P05 | 27.0 | P20 | 27.0 | P35 | 40.5 | P50 | 54.0 |
| P06 | 16.2 | P21 | 30.6 | P36 | 34.2 | P51 | 48.6 |
| P07 | 28.8 | P22 | 9.0 | P37 | 46.8 | P52 | 52.2 |
| P08 | 20.7 | P23 | 18.0 | P38 | 54.0 | P53 | 27.0 |
| P09 | 18.0 | P24 | 23.4 | P39 | 46.8 | P54 | 52.2 |
| P10 | 10.8 | P25 | 23.4 | P40 | 49.5 | P55 | 41.4 |
| P11 | 30.6 | P26 | 18.0 | P41 | 52.2 | P56 | 49.5 |
| P12 | 14.4 | P27 | 14.4 | P42 | 40.5 | P57 | 54.0 |
| P13 | 30.6 | P28 | 9.0 | P43 | 54.0 | P58 | 52.2 |
| P14 | 14.4 | P29 | 18.0 | P44 | 36.0 | P59 | 40.5 |
| P15 | 27.0 | P30 | 10.8 | P45 | 52.2 | P60 | 36.0 |

batches of the same product (e.g., P01 and P31) may be different. Additionally, the weighing coefficient for earliness, $\alpha_i$, equals to 0.9 and the weighing coefficient for tardiness, $\beta_i$, is set to 4.5 for all products. Regarding the problem instances with objective the simultaneous minimization of operating and changeovers costs (i.e., PI.09 – PI.12), the operating cost per time unit, $\psi$, is equal to 0.9 $10^3$\$/h.

At this point, it is worth mentioning that the GP model has been used to solve the problem instances that involves the minimization of the makespan or the weighted lateness (i.e., PI.01 – PI.08), and USGP model has been employed to cope with the operating and changeovers costs objective (PI.e., PI.09 – PI.12). Moreover, it is emphasized that the 30-product problem instances (PI.01, PI.02, PI.05, PI.06, PI.09 and PI.10) deal with the complex scheduling of 168 product batches, and the 60-product problem instances (PI.03, PI.04, PI.07, PI.08, PI.11 and PI.12) tackle the intricate scheduling problem of 336 product batches.

### 8.6.2 Results and discussion

The proposed solution strategy has been tested on a total number of twelve complex problem instances in order to validate its performance. A time limit of 1 CPU h has been imposed on the solution of every problem instance. All problem instances have been solved in a Dell Inspiron 1520 2.0 GHz with 2 GB RAM using CPLEX 11 via a GAMS 22.8 interface (Rosenthal, 2010).

Table 8.3 presents the constructive step's solution (initial solution) and the best solution found for each problem instance. The computational time for the constructive step ($1^{st}$-stage) as well as the total computation time are also included in the same table. Note that feasible schedules are obtained in a short computational times. Problem instance PI.11 is the most time-demanding since almost half a CPU h was needed in order to obtain a feasible solution. The remaining problem instances reached a feasible solution in relatively low computational times ranging from some CPU s to no more than 7 CPU min.

The MIP-based solution strategy is able to quickly generate feasible solutions and then gradually improve the quality these solutions. It was observed that the necessary computational time to improve a given solution mainly depends on: (i) the total number of batches to be scheduled, (ii) the objective function, (iii) the storage policy, and (iv) the core mathematical model. Obviously, the lower the total number of bathes the faster the problem is solved. It has also been observed that the case studies considering ZW storage policy are solved faster comparing them with the cases under UIS policy. The MIP model used depends on the optimization goal. Generally, the more complex the objective function the bigger the size of the model; such is the case of minimizing operating and changeovers costs.

All problem instances were solved by using the original undecomposed mathematical formulations in order to underline the high complexity of the problems addressed and to highlight the practical benefits of the proposed solution approach. Table 8.4 contains the computational features of the original mathematical models and the best solution found, for all problem instances, within the predefined time limit of 1 CPU h. It is worth mentioning that the number of the sequencing binary variables is strongly augmented by increasing the number of product orders. Note that the least complex problem instances (PI.01 and PI.02) result into a MIP model of 10,230 equations, 5,326 binary variables, and 295 continuous variables, while, the most complex problem instances (PI.11 and PI.12) result into a huge MIP model of 161,828 constraints, 41,056 binary variables, and 81,626 continuous variables. It should be noted that a feasible solution was not found by the original mathematical models in 8 of the 12 problem instances. In the remaining problem instances, feasible but very bad solutions (PI.e., with big integrality gap) were obtained. For example, the original GP model in problem instance PI.01 reported a makespan equal to 34.810 h, with an integrality gap of 58%, after 1 CPU h while our solution approach gave a makespan of 26.559 h in just 542 CPU s. The solution found by the original GP model is 31.07% worse than that of our approach. It is worth mentioning that all problem instances were also solved by the original MIP models without setting a time limit. However, in all cases the MIP solver terminated because memory capacity was exceeded.

According to Table 8.4, it is evident that the proposed MIP-based solution strategy overwhelms the original MIP models. By using our approach, highly complicated scheduling problems in multiproduct multistage batch plants can be solved,

**Table 8.3:** Problem Instances: Best schedules found within the maximum predefined time limit (3,600 CPU s).

| problem instance | objective function | products (batches) | storage policy | $1^{st}$-stage solution | $1^{st}$-stage CPU s | best solution | total CPU s | improvement |
|---|---|---|---|---|---|---|---|---|
| PI.01 | $C_{max}$ | 30 (168) | UIS | 28.507 | 38 | 26.559 | 542 | 6.83% |
| PI.02 | $C_{max}$ | 30 (168) | ZW | 31.520 | 7 | 30.532 | 187 | 3.14% |
| PI.03 | $C_{max}$ | 60 (336) | UIS | 49.161 | 155 | 48.548 | 1,502 | 1.25% |
| PI.04 | $C_{max}$ | 60 (336) | ZW | 58.104 | 106 | 56.061 | 1,718 | 3.52% |
| PI.05 | W.L. | 30 (168) | UIS | 48.613 | 22 | 19.085 | 720 | 60.74% |
| PI.06 | W.L. | 30 (168) | ZW | 115.016 | 15 | 84.438 | 262 | 26.59% |
| PI.07 | W.L. | 60 (336) | UIS | 118.683 | 403 | 87.943 | 3,600 | 25.90% |
| PI.08 | W.L. | 60 (336) | ZW | 629.672 | 356 | 515.876 | 1,478 | 18.07% |
| PI.09 | O.C.C | 30 (168) | UIS | 66.158 | 94 | 62.910 | 3,600 | 4.91% |
| PI.10 | O.C.C | 30 (168) | ZW | 72.318 | 58 | 70.209 | 3,600 | 2.92% |
| PI.11 | O.C.C | 60 (336) | UIS | 119.759 | 1,780 | 117.909 | 3,600 | 1.54% |
| PI.12 | O.C.C | 60 (336) | ZW | 139.104 | 880 | 134.624 | 3,600 | 3.22% |

† W.L. = Weighted Lateness, and O.C.C = Operating & Changeovers Costs

**Table 8.4:** Comparison between the original MIP model & the proposed MIP-based strategy best solutions found within the maximum predefined time limit (3,600 CPU s).

| | ORIGINAL MATHEMATICAL MODEL | | | | | | OUR STRATEGY | |
|---|---|---|---|---|---|---|---|---|
| problem instance | cons-trains | binary variables | continuous variables | gap | best solution | total CPU s | best solution | total CPU s |
| PI.01 | 10,230 | 5,326 | 295 | 58% | 34.810 | 3,600 | 26.559 | 542 |
| PI.02 | 10,230 | 5,326 | 295 | - | - | 3,600 | 30.532 | 187 |
| PI.03 | 40,988 | 20,916 | 589 | 90% | 109.960 | 3,600 | 48.548 | 1,502 |
| PI.04 | 40,988 | 20,916 | 589 | - | - | 3,600 | 56.061 | 1,718 |
| PI.05 | 10,261 | 5,326 | 355 | 100% | 428.146 | 3,600 | 19.085 | 720 |
| PI.06 | 10,261 | 5,326 | 355 | - | - | 3,600 | 84.438 | 262 |
| PI.07 | 41,049 | 20,916 | 709 | 100% | 23,453.744 | 3,600 | 87.943 | 3,600 |
| PI.08 | 41,049 | 20,916 | 709 | - | - | 3,600 | 515.876 | 1478 |
| PI.09 | 39,858 | 10,264 | 20,286 | - | - | 3,600 | 62.910 | 3,600 |
| PI.10 | 39,858 | 10,264 | 20,286 | - | - | 3,600 | 70.209 | 3,600 |
| PI.11 | 161,828 | 41,056 | 81,626 | - | - | 3,600 | 117.909 | 3,600 |
| PI.12 | 161,828 | 41,056 | 81,626 | - | - | 3,600 | 134.624 | 3,600 |

as the current experimental study reveals. Although optimality cannot be guaranteed in general, feasible solutions can be obtained in relatively short computational time. Bear in mind that feasibility is the principal goal in practical scheduling problems. To the best of our knowledge, neither other standard solution methods nor heuristics exist for tackling the studied scheduling problem efficiently.

Some Gantt charts of the best schedules for some representative problem instances are provided in order to provide the reader with a visual demonstration of the complexity of the addressed problems. More specifically, Figure 8.7 presents the best solution found for solving the 30-product case by minimizing $C_{max}$ under UIS policy (problem instance PI.01). Figure 8.8 illustrates the best schedule reported for solving the 60-product case by minimizing total weighted lateness under UIS policy (problem instance PI.07). Finally, Figure 8.9 graphically depicts the best schedule found for solving the 60-product case by minimizing total operating and changeovers costs under ZW storage policy (problem instance PI.12).

## 8.7 Concluding Remarks

A novel iterative two-step MIP-based solution strategy has been presented for the solution of large-scale scheduling problems in multiproduct multistage batch plants. A benchmark scheduling problem in a multiproduct multistage pharmaceutical batch plant has been introduced and solved in this study. The proposed solution technique is able to generate good feasible solutions in relatively short times, as the several problem instances of the pharmaceutical scheduling problem reveal. It is worthwhile to note that the user can appropriately define the degrees of freedom of the decision variables by balancing the trade-off between computational time and solution quality. The proposed solution strategy can be also applied to other types of scheduling problems by adopting a different MIP core model that describes the particular scheduling problem. Moreover, this work aims to be a step towards reducing the gap between scheduling theory and practice, since it has clearly demonstrated that real-world industrial scheduling problems can be solved by using effective MIP-based optimization solution strategies.

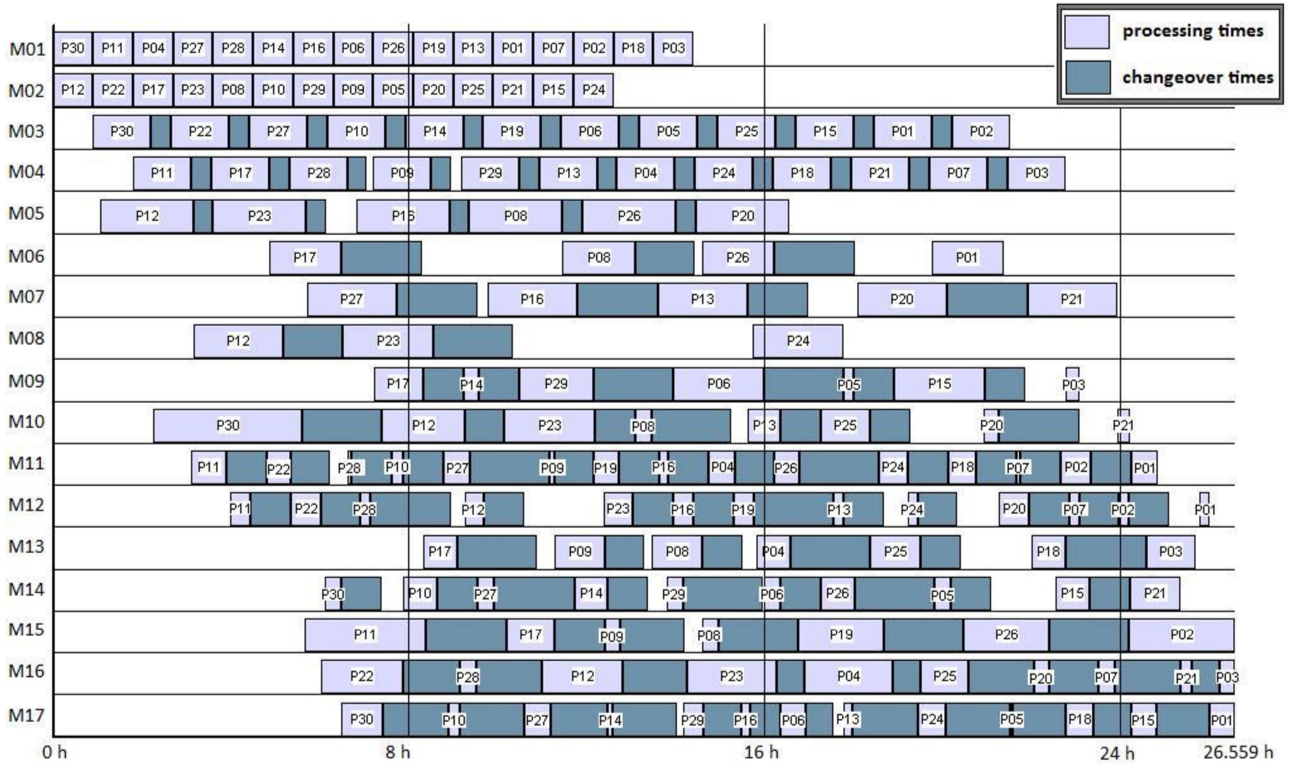



**Figure 8.7:** Best schedule for PI.01 (30-product case: minimization of makespan under UIS policy).

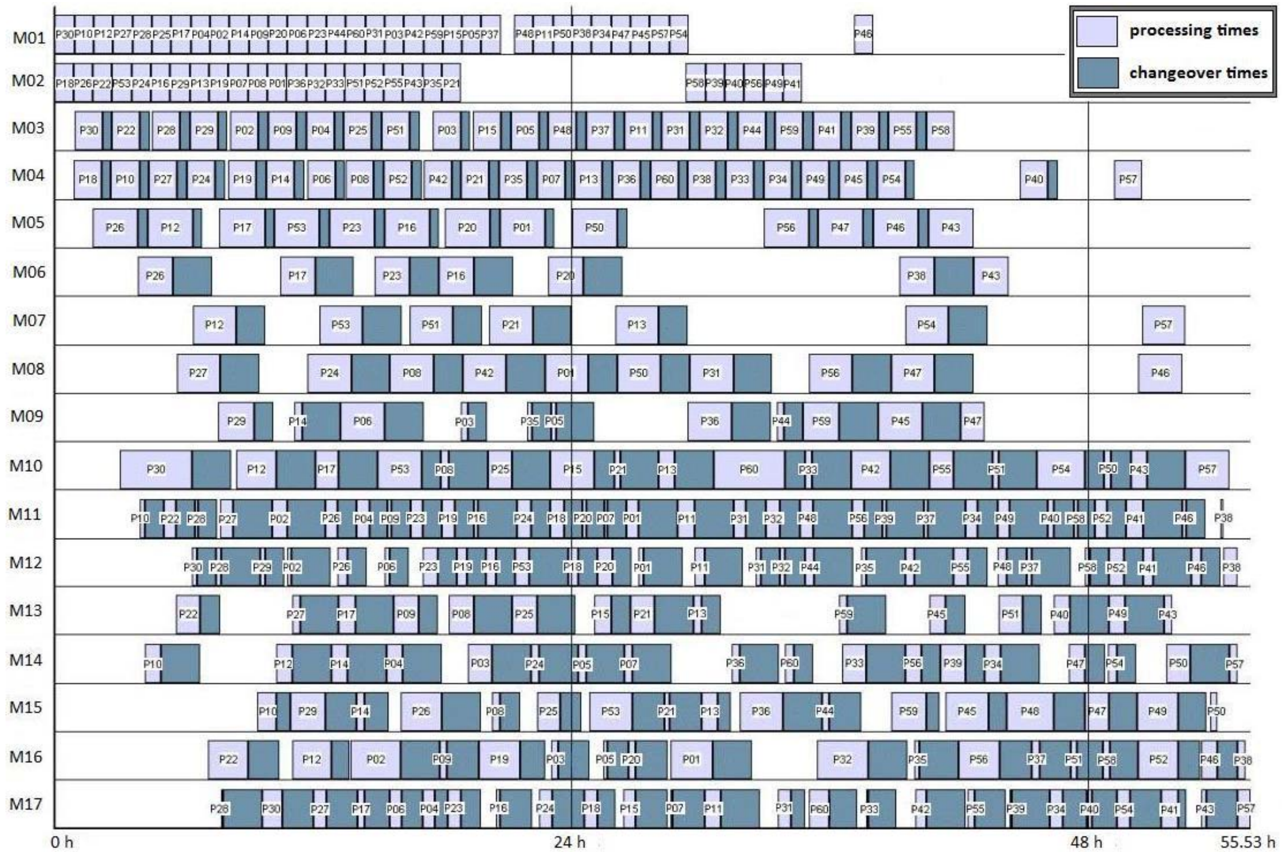



**Figure 8.8:** Best schedule for PI.07 (60-product case: minimization of total weighted lateness under UIS policy).

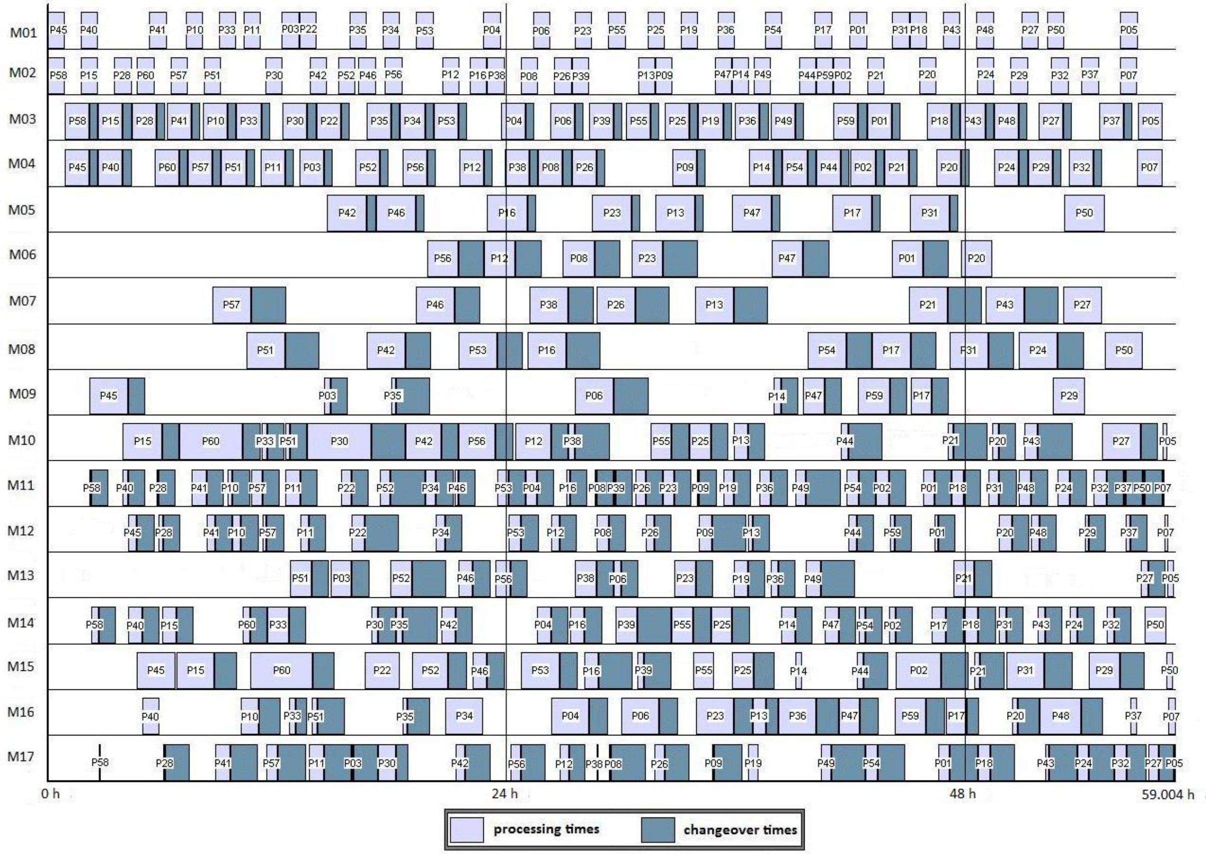



**Figure 8.9:** Best schedule for PI.12 (60-product case: minimization of total operating and changeovers costs under ZW policy).

## 8.8 Nomenclature

**Indices / Sets**

| | |
|---|---|
| $i, i', i'' \in I$ | product orders (products) |
| $j \in J$ | processing units (units) |
| $s \in S$ | processing stages (stages) |

**Subsets**

| | |
|---|---|
| $I^{in}$ | set of products $i$ that are included into the optimization |
| $J_i$ | available units $j$ to process product $i$ |
| $J_s$ | available units $j$ to process stage $s$ |
| $S_i$ | set of stages $s$ for each product order $i$ |
| $S_i^{last}$ | last processing stage for product order $i$ |

**Parameters**

| | |
|---|---|
| $\alpha_i$ | weighing coefficient for earliness for product $i$ |
| $\beta_i$ | weighing coefficient for tardiness for product $i$ |
| $\gamma_{ii'j}$ | sequence-dependent setup (changeover) time between products $i$ and $i'$ in unit $j$ |
| $\delta_i$ | due date for product $i$ |
| $\varepsilon_j$ | time point that unit $j$ is available to start processing |
| $M$ | a big number |
| $\mu_{s-1s}$ | batch transfer time between two consecutive stages $s-1$ and $s$ |
| $\xi_{ii'j}$ | sequence-dependent setup (changeover) cost between products $i$ and $i'$ in unit $j$ |
| $o_i$ | release time for product $i$ |
| $\pi_{ij}$ | sequence-independent setup time of product $i$ in unit $j$ |
| $\tau_{isj}$ | processing time for stage $s$ of product $i$ in unit $j$ |
| $\psi$ | operating cost of production facility per time unit |

**Continuous Variables**

| | |
|---|---|
| $C_{is}$ | completion time of stage $s$ of product $i$ |
| $C_{max}$ | makespan |
| $E_i$ | earliness for product $i$ |
| $T_i$ | tardiness for product $i$ |
| $W_{is}$ | the time that stage $s$ of a product $i$ is stored (waits) before proceeding to the following processing stage $s+1$ |
| $Z_{ii'j}$ | allocation position difference between products $i$ and $i'$ in unit $j$ |

**Binary Variables**

| | |
|---|---|
| $X_{ii'j}$ | $= 1$ for every product $i$ that is processed before product $i'$ in unit $j$ |
| $\bar{X}_{ii'j}$ | $= 1$ if product $i$ is processed exactly before product $i'$ in unit $j$ |
| $Y_{isj}$ | $= 1$ if stage $s$ of product $i$ is assigned to unit $j$ |

Chapter 9

# Production Scheduling in Large-scale Multipurpose Batch Process Industries

## 9.1 Introduction

The batch production scheduling problem, consists in determining the optimal allocation of a number of resources (e.g., processing units, raw materials, utilities, manpower, etc.) over time to a number of processing tasks transforming raw materials into a number of desired final products. The extent of each processing task (total amount of materials consumed/produced by the task) is constrained by minimum and maximum batch sizes. Interactions between resources and processing tasks are discrete, occurring at fixed time intervals relative to the tasks starting times. The problem is NP-hard in the strong sense, and even simplified versions of it are NP-hard too (Burkard *et al.*, 1998). Accordingly, a number of approaches have been developed in order to maintain computational tractability in systems of industrial relevance; including rigorous and heuristic time-based decomposition of detailed scheduling models, mixed-integer linear programming, heuristics, local search, methods relating to project scheduling, and combined batching/scheduling techniques. The later have been shown to yield good solutions to realistic production scheduling problems with reduced computational effort.

In this chapter, a new two-layered decomposition methodology to the batch process scheduling problem in multipurpose production plants is proposed. In the first level, an approximate scheduling model derived from the detailed STN-based time-indexed scheduling formulation is solved; the model partially relaxes the allocation of task instances to processing units details of the full scheduling formulation. In the second level, the output of the approximate scheduling problem is used to provide batching targets for the detailed scheduling model within an iterative decomposition scheme. The procedure is tested on published instances of the Westenberger-Kallrath (W-K) benchmark scheduling problem.

## 9.2 Problem Statement

In this chapter, the problem under consideration is concerned with industrial-scale multiproduct multipurpose batch processes. The scheduling problem in question is formally defined in terms of the following items:

(i) A number of undefined time points $t$ inside the available production scheduling horizon $T$.

(ii) A set of multipurpose batch processing units $u \in U$.

(iii) A set of batch processing tasks $i \in I$ (i.e., raw materials, intermediates, final products) with minimum and maximum allowable batch sizes; $\beta_i^{min}$ and $\beta_i^{max}$, respectively. Batch processing times ($\tau_{iu}$) and setup times ($\gamma_{iu}$) for tasks are unit dependent.

(iv) A set of material states $s \in S$ with initial inventory $R_{st=0}$, maximum inventory level $\theta_s^{max}$, and external demand $\zeta_s$.

(v) For each material state $s$, there is a set of consumption tasks $I_s^c$ with consumption coefficients $\alpha_{is} \leq 0$, and a set of production tasks $I_s^p$ with production coefficients $\alpha_{is} \geq 0$. These coefficients are expressed as fractions of the total batch size of task $i$. $I_u$ is the subset of products that can be processed in unit $u$ while subset $U_i$ denotes the subset of units where product $i$ can be processed.

(vi) A subset of processing tasks with variable production coefficients $I^f$. For each state $s$ of these tasks, there exist minimum and maximum production coefficients; $\tilde{\alpha}_{is}^{min}$ and $\tilde{\alpha}_{is}^{max}$, respectively.

The key decision variables are:

(i) if task $i$ starts processing in unit $u$ at time $t$, $X_{iut}$;

(ii) the batch size of task $i$ starting in unit $u$ at time $t$, $\bar{B}_{iut}$;

(iii) the amount of state $s$ processed by task $i$ in unit $u$ at time $t$, $B_{isut}$; and

(iv) the amount of state $s$ at time $t$, $R_{st}$.

For a given set of final product demands, the scheduling objective is the minimization of makespan subject to a number of operational constraints.

## 9.3 The W-K Process

The Westenberger-Kallrath (W-K) benchmark batch scheduling problem represents a typical multipurpose multiproduct batch chemical process (Kallrath, 2002b), which consists of 17 distinct processing tasks (T1 – T17) that can take place in 9 multipurpose units, and 19 material states (S1 – S19) with 5 final products (S15 – S19). The process features convergent, divergent, and cyclic production flows, constituting a complex batch scheduling problem. Figure 9.1 illustrates the State-Task-Network (STN) representation for the W-K process flow.

The main W-K process specifications are shown in Tables 9.1 and 9.2. For each processing task, Table 9.1 summarizes input/output states, min/max allowable batch sizes, a suitable subset of processing units, and the corresponding unit processing times. Note that processing times are independent of batch size. For each non-perishable material state, Table 9.2 shows initial ($R_{st=0}$) and maximum ($\theta_s^{max}$) inventory levels for material states. The overall process involves a number of perishable products (S6, S10, S11, and S13), for which both initial and maximum inventory levels are zero.

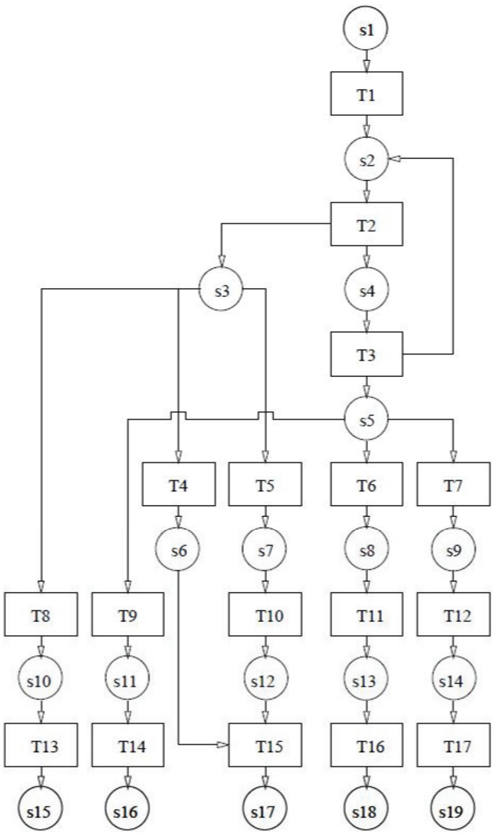



**Figure 9.1:** STN representation for the W-K process.

**Table 9.1:** W-K process: Tasks specifications.

| Task | Input states | Output states | Batch size bounds (kg) | Alternative units | Processing times (h) |
|---|---|---|---|---|---|
| T1 | S1 | S2 | 3 – 10 | U1 | 2 |
| T2 | S2 | S3, S4 | 5 – 20 | U2 | 4 |
| T3 | S4 | S2, S5 | 4 – 10 | U3 | 2 |
| T4 | S3 | S6 | 4 – 10 | U4 | 4 |
| T5 | S3 | S7 | 4 – 10 | U4 | 4 |
| T6 | S5 | S8 | 4 – 10 | U4 | 4 |
| T7 | S5 | S9 | 4 – 10 | U4 | 4 |
| T8 | S3 | S10 | 4 – 10 | U5 | 6 |
| T9 | S5 | S11 | 4 – 10 | U5 | 6 |
| T10 | S7 | S12 | 3 – 7 | U6, U7 | 4, 5 |
| T11 | S8 | S13 | 3 – 7 | U6, U7 | 5, 6 |
| T12 | S9 | S14 | 3 – 7 | U6, U7 | 6, 6 |
| T13 | S10 | S15 | 4 – 12 | U8, U9 | 4, 6 |
| T14 | S11 | S16 | 4 – 12 | U8, U9 | 4, 6 |
| T15 | S6, S12 | S17 | 4 – 12 | U8 | 4 |
| T16 | S13 | S18 | 4 – 12 | U8, U9 | 6, 6 |
| T17 | S14 | S19 | 4 – 12 | U8, U9 | 6, 6 |

**Table 9.2:** W-K process: States specifications.

| States | S1 | S2 | S3 | S4 | S5 | S7 | S8 | S9 | S12 | S14 | S15 | S16 | S17 | S18 | S19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $R_{st=0}$ | ∞ | 20 | 20 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\theta_s^{max}$ | ∞ | 30 | 30 | 15 | 30 | 10 | 10 | 10 | 10 | 10 | ∞ | ∞ | ∞ | ∞ | ∞ |

It is worth mentioning that two main issues complicating scheduling considerations in the W-K process are: (i) the existence of one processing task (T2) with flexible output proportions, and (ii) the need for sequence-independent batch setup times. For task T2, the first output state (S3) can be produced in amounts ranging from 20% to 70% of the task batch size. Consequently, the second output state (S4) can be produced in amounts ranging from 80% to 30% of the task batch size. These and all other non-trivial stoichiometric coefficients are summarized in Table 9.3. For each processing unit, cleaning must be performed whenever production switches between two different task types.

**Table 9.3:** W-K process: Consumption/production coefficients.

| Task | State | Coefficient | State | Coefficient |
|---|---|---|---|---|
| T2 | S3 | 0.2 – 0.7 | S4 | 0.3 – 0.8 |
| T3 | S2 | 0.31 | S5 | 0.69 |
| T15 | S6 | -0.5 | S12 | -0.5 |

## 9.4 Bi-Level Decomposition Approach

In this section, a novel bi-level decomposition approach for complex batch process scheduling problems in multiproduct multipurpose plants is presented. First, a detailed STN-based MIP formulation for the scheduling problem in question is developed, and afterwards an approximate STN-based MIP model is derived by partially relaxing the allocation details of tasks to units. Finally, the output of the approximate scheduling problem is used to provide batching targets for the detailed scheduling model within an iterative decomposition scheme.

### 9.4.1 Detailed scheduling formulation

In this subsection, the detailed scheduling formulation is presented. The detailed MIP model follows well-known STN-based models (Shah *et al.*, 1993a), with the addition of: (i) constraints permitting flexible production proportions for some or all tasks in the process, and (ii) constraints for enforcing sequence-independent batch setup (cleaning) times. The scheduling problem under consideration can be formulated by the following sets of constraints:

$$\beta_i^{min} X_{iut} \leq \bar{B}_{iut} \leq \beta_i^{max} X_{iut} \qquad \forall u \in U, i \in I_u, t \in T \tag{9.1}$$

$$\tilde{\alpha}_{is}^{min} \bar{B}_{iut} \leq B_{isut} \leq \tilde{\alpha}_{is}^{max} \bar{B}_{iut} \quad \forall u \in U, s \in S, i \in (I_u \cup I_s^p \cup I^f), t \in T \tag{9.2}$$

$$\sum_{s \in S_i^p} B_{isut} = \bar{B}_{iut} \qquad \forall u \in U, i \in I_u, t \in T \tag{9.3}$$

$$\begin{aligned} R_{st} = R_{st-1} &+ \sum_{u \in U} \sum_{i \in (I_u \cup I_s^c)} \alpha_{is} \bar{B}_{iut} + \sum_{u \in U} \sum_{i \in (I_u \cup I_s^p \cap I^f)} \alpha_{is} \bar{B}_{iut-\tau_{iu}} \\ &+ \sum_{u \in U} \sum_{i \in (I_u \cup I_s^p \cup I^f)} B_{isut-\tau_{iu}} - \zeta_s \qquad \forall s \in S, t \in T \end{aligned} \tag{9.4}$$

$$R_{st} \leq \theta_s^{max} \qquad \forall s \in S, t \in T \tag{9.5}$$

$$\sum_{i \in I_u} \sum_{t'=t-\tau_{iu}+1}^{t'=t} X_{iut} \leq 1 \qquad \forall u \in U, t \in T \tag{9.6}$$

$$\begin{aligned} \sum_{i' \in I_u : i' \neq i} \sum_{t'=t+\tau_{iu}}^{t'=t+\tau_{iu}+\gamma_{iu}-1} X_{iut} \leq \frac{\tau_{iu} + \gamma_{iu}}{min[\tau_{i'u}]} (1 - X_{iut}) \\ \forall u \in U, i \in I_u, t \in T \end{aligned} \tag{9.7}$$

$$C_{max} \geq t \sum_{i \in I_u} (\tau_{iu} + \gamma_{iu}) X_{iut} \qquad \forall u \in U \tag{9.8}$$

Constraints (9.1) to (9.3) express batch size limitations. The basic STN formulation has been extended to account for flexible production recipes by including constraint set (9.2) into the MIP formulation. Constraints (9.4) and (9.5) define the material balance for each state $s$ at time $t$. In addition, constraint set (9.6) forbid any two tasks to be performed in the same unit concurrently while constraints set (9.7) ensures unit unavailability during production switchover from any task $i$ to a different task $i'$. Finally, constraint set (9.8) is only related to makespan.

Having as optimization goal the minimization of the makespan, the detailed scheduling problem takes the form:

$$\min C_{max} \qquad \text{s.t. } (9.1) - (9.8) \qquad \text{(DSP)}$$

where DSP represents a MIP formulation with binary decision variables $X_{iut}$ and non-negative continuous decision variables $B_{isut}$ and $\bar{B}_{iut}$.

### 9.4.2 Approximate scheduling formulation

An approximate scheduling model can be derived from the detailed scheduling formulation DSP with the following modifications. First, alternative processing units for the same production task are aggregated based on a mean processing time ($\bar{\tau}_i$). Second, the computationally expensive unit allocation constraints are partially relaxed. The functional form of all other constraints is retained. In effect, the suggested approximate model concentrates on ordering the production steps within the process relative to each other, without explicitly considering scheduling details at the unit level.

Some additional notation has been introduced in order to formulate the approximate scheduling model. More specifically, we introduce integer variables $Z_{it}$ that denote the number of batches of task $i$ starting at time $t$, continuous variables $\bar{Q}_{it}$ which stand for the cumulative batch size of task $i$ starting at time $t$, and continuous variables $Q_{ist}$ that define the cumulative amount of state $s$ being processed by task $i$ at time $t$. The constraints for the approximate scheduling problem are summarized below.

$$\beta_i^{min} Z_{it} \le \bar{Q}_{it} \le \beta_i^{max} Z_{it} \qquad \forall i \in I, t \in T \tag{9.9}$$

$$\tilde{\alpha}_{is}^{min} \bar{Q}_{it} \le Q_{ist} \le \tilde{\alpha}_{is}^{max} \bar{Q}_{it} \quad \forall s \in S, i \in (I \cup I_s^p \cup I^f), t \in T \tag{9.10}$$

$$\sum_{s \in S_i^p} Q_{ist} = \bar{Q}_{it} \qquad \forall i \in I, t \in T \tag{9.11}$$

$$\begin{aligned} R_{st} = R_{st-1} + \sum_{i \in I_s^c} \alpha_{is} \bar{Q}_{it} + \sum_{i \in (I_s^p \cap I^f)} \alpha_{is} \bar{Q}_{it-\bar{\tau}_i} \\ + \sum_{i \in (I_s^p \cup I^f)} Q_{ist-\bar{\tau}_i} - \zeta_s \qquad \forall s \in S, t \in T \end{aligned} \tag{9.12}$$

$$R_{st} \le \theta_s^{max} \qquad \forall s \in S, t \in T \tag{9.13}$$

$$Z_{it} \le \psi_i \qquad \forall i \in I, t \in T \tag{9.14}$$

Constraints (9.9) – (9.12) are the aggregated form of constraints (9.1) – (9.4) in the DSP formulation, while constraint set (9.14) is the approximate form of constraint set (9.6) in DSP model. Notice that parameter $\psi_i$, in constraint set (9.14), denotes the total number of alternative units for processing task $i$. It is worth mentioning that changeovers are ignored in the approximate model, thus makespan cannot be meaningfully related to the approximate model either.

Having as objective the minimization of the total number of batches, the approximate scheduling problem takes the form:

$$\min \sum_{i \in I} \sum_{t \in T} Z_{it} \qquad \text{s.t. (9.9) – (9.14)} \qquad \text{(ASP)}$$

where ASP is also a MIP formulation with integer decision variables $Z_{it}$ and non-negative continuous decision variables $Q_{ist}$ and $\bar{Q}_{it}$. Obviously, the absence of unit-level scheduling considerations in model ASP may render its output infeasible with respect to DSP, since the timings of batch instances will likely be underestimated. Nevertheless, as it will be demonstrated in the following subsection, ASP can provide batching targets to an heuristic iterative procedure for solving problem instances of practical size. Note that other regular objectives can be specified in the ASP problem, such as the minimization of the total flow time.

### 9.4.3 Decomposition scheme

The total number of batches required to satisfy demand is known by solving ASP model. This information can be utilized to split the detailed scheduling problem into a number of tractable subproblems. Hence, if $\mu_k$ is the (constant) number of batches to be scheduled at each iteration $k$, then the total number of subproblems to be solved will be at most $\sum_i \sum_t Z_{it}/\mu_k$. In practice, an adaptive adjustment of $\mu_k$ can be employed based on the makespan achieved at the previous subproblem.

The iterative procedure examines the solution of ASP problem sequentially starting from the first time period. When the desired number of batches $\mu_k$ has been reached (let $\hat{t}$ denote the corresponding time period in ASP), model DSP is solved by adding the following cuts:

$$\sum_{u \in U_i} \sum_{t \in T} X_{iut} \geq \sum_{t \leq \hat{t}} Z_{it'} \qquad \forall i \in I \tag{9.15}$$

$$\sum_{u \in U_i} \sum_{t \in T} \bar{B}_{iut} \geq \sum_{t \leq \hat{t}} \bar{Q}_{it'} \qquad \forall i \in I \tag{9.16}$$

Henceforth, this subproblem will be refereed to as DSP-k. The solution to DSP-k will be used in the next iteration in two ways: (i) the makespan achieved in DSP-k subproblem will provide a lower bound for the makespan of DSP-(k+1) subproblem, and (ii) the timings of scheduled batches from DSP-k subproblem will be fixed and provided as input to DSP-(k+1). In practice, the timings obtained by DSP-k subproblem can be fixed only partially (with the exception of the last batch of each task $i$ scheduled in DSP-k) so that feasibility of next subproblem DSP-(k+1) is facilitated. For each subproblem DSP-k, the choice of an appropriate upper bound on makespan is critical in the computational performance of the algorithm. In this study, a greedy heuristic has been employed, which schedules all batches in $\mu_k$ with two simple dispatching rules: (i) batches starting earlier in ASP are scheduled first, and (ii) each batch is scheduled on the available unit with the earliest completion time. The later is obviously only applicable to batch types (tasks) that can be performed in more than one unit.

## 9.5 Experimental Studies

Two sets of 22 problem instances each of the computational experiments of the W-K benchmark scheduling problem are considered; one ignoring and the other considering cleaning times. All cleaning times are set equal to 50% of the corresponding task processing times. All problem instances have been solved in CPLEX

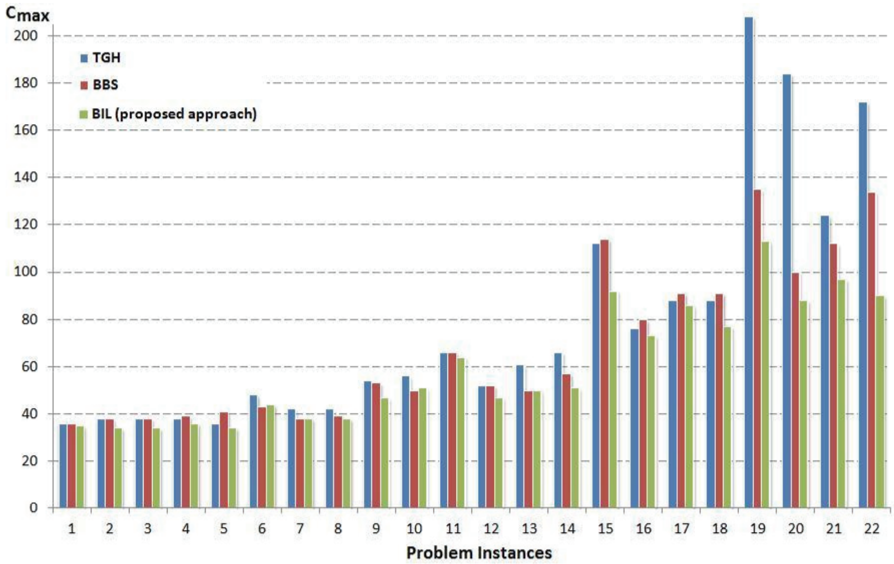



**Figure 9.2:** Experimental results for the case without cleaning times ($C_{max}$ in h).

6.6 solver via a GAMS (Rosenthal, 2010) interface on a single 400 MHz Unix processor. The number of DSP sub-problems has been determined dynamically for each problem instance, so that at least 20 batches were scheduled in every DSP iteration. For completeness, it is noted that TGH results were obtained on a single 266 MHz PC and B+BS results were obtained on an 800 MHz PC.

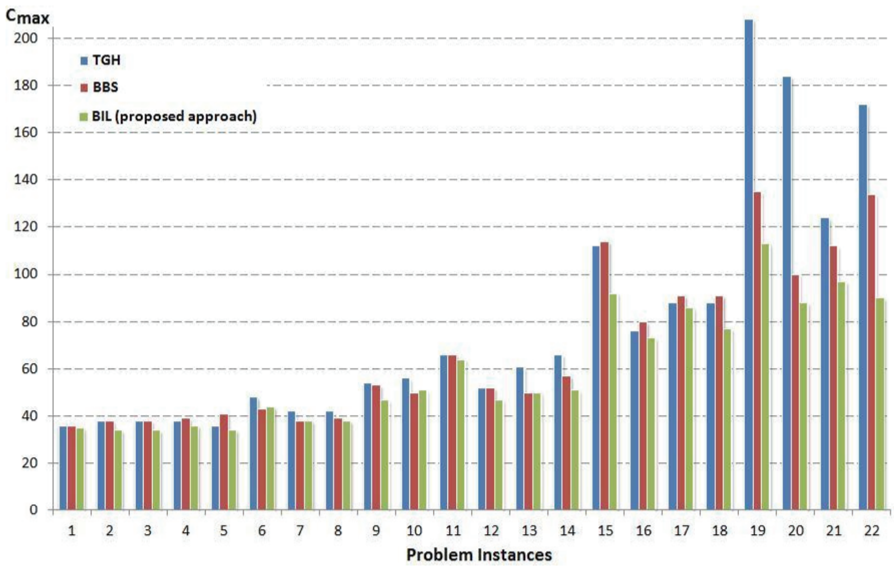



**Figure 9.3:** Experimental results for the case with cleaning times ($C_{max}$ in h).

**Table 9.4:** CPU s times for the cases: (a) with, and (b) without cleaning times.

| | Problem Instances | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| TGH | 1,100 | 2,247 | 2,487 | 1,550 | 1,778 | 3,600 | 2,587 | 3,123 | 3,600 | 3,600 | 3,600 |
| BBS | 3 | 13 | 17 | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 |
| BIL,a | 24 | 41 | 29 | 27 | 31 | 44 | 38 | 65 | 73 | 64 | 191 |
| BIL,b | 44 | 66 | 111 | 103 | 116 | 133 | 108 | 50 | 59 | 97 | 64 |
| | Problem Instances | | | | | | | | | | |
| Method | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| TGH | 3,600 | 3,600 | 3,600 | 3,600 | 3,600 | 3,600 | 3,600 | 5,152 | 3,600 | 3,600 | 3,600 |
| BBS | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 |
| BIL,a | 182 | 167 | 197 | 234 | 247 | 271 | 232 | 265 | 228 | 263 | 211 |
| BIL,b | 97 | 89 | 179 | 107 | 110 | 126 | 219 | 291 | 279 | 284 | 261 |

Consequently, the solutions of all problem instances of the W-K benchmarck scheduling problem have been compared with publicly available results in order to shed light on the advantages of the proposed approach. Comparisons are attempted with the Time-Grid Heuristic (TGH) methodology (Blömer & Günther, 1998, 2000b) and the Batching/Batch Scheduling (B+BS) approach (Neumann, Schwindt, & Trautmann, 2002). TGH employs relax-and-fix as well as dive-and-fix heuristics for the resolution of the detailed scheduling model (Wolsey, 1998). B+BS methodology employs a simple batching model in combination with resource-constrained project scheduling algorithms. BIL is used here as an acronym for the proposed combined bi-level approach.

For the problem instances without cleaning times (see Figure 9.2), equal or better solutions than TGH are obtained in 21 out of 22 cases, with an average makespan improvement of 8%. Compared to B+BS, BIL finds equal or better makespans in all 22 cases, with an average improvement of 8.16%. Figure 9.3 presents the computational experiments with cleaning times. Notice that BIL manages to find equal or better solutions than TGH in all cases, with an average improvement of 15.5%. Compared to B+BS, equal or better solutions are obtained in 20 cases with an average improvement of 9.3%. In total, the results indicate that BIL manages to find good quality solutions in reasonable computational time, as Table 9.4 shows.

## 9.6 Concluding Remarks

A decomposition approach has been developed for the solution of large-size batch process scheduling problems. The decomposition is principally based on the solution of an approximate scheduling problem, derived from a detailed time-indexed scheduling model by unit aggregation and partial relaxation of unit allocation constraints. The output of the approximate model forms the basis of an order-based decomposition scheme, within which scheduling subproblems are solved with the aid of the detailed scheduling model. The method has been tested on several published instances of a benchmark multipurpose batch process scheduling problem with good results; particularly, for the larger problem instances.

## 9.7 Nomenclature

**Indices / Sets**

| | |
|---|---|
| $i, i', \in I$ | batch processing tasks (tasks) |
| $u \in U$ | batch processing units (units) |
| $s \in S$ | material states (states) |
| $t \in T$ | time points |

**Subsets**

| | |
|---|---|
| $I^f$ | set of tasks with variable production coefficients |
| $I_s^c$ | set of consumption tasks for state $s$ |
| $I_s^p$ | set of production tasks for state $s$ |
| $I_u$ | subset of tasks that can be processed in unit $u$ |
| $S_i^p$ | subset of states that are produced by task $i$ |
| $U_i$ | subset of units that can process task $i$ |

**Parameters**

| | |
|---|---|
| $\alpha_{is}$ | consumption/production coefficient of state $s$ from task $i$; they are expressed as fractions of the total batch size of task $i$ |
| $\tilde{\alpha}_{is}^{min}$ | minimum production coefficient for flexible tasks $i \in I^f$ |
| $\tilde{\alpha}_{is}^{max}$ | maximum production coefficient for flexible tasks $i \in I^f$ |
| $\beta_i^{min}$ | minimum batch size for task $i$ |
| $\beta_i^{max}$ | maximum batch size for task $i$ |
| $\gamma_{iu}$ | setup (cleaning) time of task $i$ in unit $u$ |
| $\zeta_s$ | external demand for state $s$ |
| $\theta_s^{max}$ | maximum inventory level for state $s$ |
| $\tau_{iu}$ | batch processing time of task $i$ in unit $u$ |
| $\bar{\tau}_i$ | mean processing time for task $i$ considering its alternative units |
| $\psi_i$ | total number of alternative units for processing task $i$ |

**Continuous Variables**

| | |
|---|---|
| $B_{isut}$ | the amount of state $s$ processed by task $i$ in unit $u$ at time $t$ |
| $\bar{B}_{iut}$ | the batch size of task $i$ starting in unit $u$ at time $t$ |
| $C_{max}$ | makespan |
| $Q_{ist}$ | the cumulative amount of state $s$ being processed by task $i$ at time point $t$ |
| $\bar{Q}_{it}$ | the cumulative batch size of task $i$ starting at time $t$ |
| $R_{st}$ | the amount of state $s$ at time $t$ |

**Integer Variables**

| | |
|---|---|
| $Z_{it}$ | the number of batches for task $i$ starting at time $t$ |

**Binary Variables**

| | |
|---|---|
| $X_{iut}$ | $= 1$ if task $i$ starts processing in unit $u$ at time $t$ |

Part V

# Conclusions and Outlook

# Chapter 10

# Conclusions and Future Work

## 10.1 Conclusions

The objective of this thesis has been to establish mathematical programming techniques and solution approaches for the efficient solution of complex process scheduling and planning problems. For this reason, a number of real-life case studies, contemplating representative sectors and significant process industries (chemical, pharmaceuticals, food and beverage industries), have been addressed and solved by new mathematical programming frameworks devised in this thesis.

Part I identifies the major challenges to be addressed through an extensive state-of-the-art review (Chapter 2). Although production planning and scheduling has become the subject of intensive research, an attentive review reveals the areas where new contributions are needed for a major impact in real-world applications. In Chapter 3, the fundamental theory and concepts behind the methods and tools used throughout the thesis is briefly described.

Part II deals with continuous processes. More specifically, in Chapter 4, an industrial case study considering the simultaneous planning and scheduling problem in the bottling stage of a real-life beer industry, producing hundreds of final products, has been addressed. A special feature of the problem in question is that final products can be classified into product families. The grouping into families is based on various criteria, including product similarities, processing similarities, and/or changeover considerations. A hybrid discrete/continuous-time mathematical approach to the simultaneous production planning and scheduling of continuous parallel units producing a large number of final products that can be classified into product families has been developed. In contrast with previous research works, a more general case has been considered based on: (i) product families, (ii) short planning periods that may lead to idle units for entire periods, (iii) changeovers spanning multiple periods, and (iv) maintenance activities. The proposed approach also addresses appropriately aspects such as changeover carryover and crossover, thereby leading to solutions with higher utilization of resources. Very good solutions to problems with hundreds of products can be obtained within 5 CPU min, while optimal solutions can also be found in reasonable time. Furthermore, the proposed formulation yields solutions which are substantially better than the ones obtained using commercial tools, suggesting that MIP methods can be used to address large-scale problems of practical interest.

Part III deals with food process industries that combine batch and continuous operation modes in their overall production process. In Chapter 5, a multiproduct multistage semicontinuous ice-cream production facility is considered. A new MIP framework and a solution strategy have been presented for the optimal production scheduling of this production facility. Although, the proposed mathematical formulation is well-suited to the ice-cream production facility considered, it could be also used, with minor modifications, in scheduling problems arising in other semicontinuous industries with similar processing features. The overall mathematical framework relies on an efficient modeling approach of the sequencing decisions, the integrated modeling of all production stages and the inclusion of strong valid integer cuts in the MIP formulation. The simultaneous optimization of all processing stages increases the plant production capacity, reduces the production cost for final products, and facilitates the interaction among the different departments of the production facility. The proposed MIP formulation and the proposed solution methodology results in very low computational times for the several problem instances solved.

In Chapter 6, a multiproduct semicontinuous yogurt production facility, where labor (i.e., the number of available workers) is a limited resource, is studied. Production planning in semicontinuous processing plants typically deals with a large number of products, however many products appear with similar characteristics, and therefore final products can be grouped into product families. Thus, the production planning problem under question could be partially focused on product families rather than on each product separately, following a similar modeling concept to Chapter 4. A general MIP approach has been presented for the resulting resource-constrained production. Quantitative as well as qualitative optimization goals are included in the proposed model. The definition of product families significantly reduces the size of the underlying mathematical model and, thus, the necessary computational effort without sacrificing any feasibility constraints. A number of cases studies, also considering unexpected event scenarios (i.e., workers absence, and products orders modifications), have been solved in reasonable computational time.

Chapter 7 addresses the production and logistics operations planning in large-scale single- or multi-site semicontinuous process industries. A novel mixed discrete continuous-time mixed integer programming model, based on the concept of families of products, for the problem in question has been developed. A remarkable feature of the proposed approach is that in the production planning problem timing and sequencing decisions are taken for product families rather than for products. However, material balances are realized for every specific product, thus permitting the detailed optimization of production, inventory, and transportation costs. Additionally, alternative transportation modes are considered for the delivery of final products from production sites to distribution centers, a reality that most of the current approaches totally neglect. The efficiency and the applicability of the proposed approach is demonstrated by solving to optimality two industrial-size case studies, for an emerging real-life dairy industry. It is worth noting that despite the complexity of the problems addressed, the proposed approach appears a remarkable computational performance.

Part IV deals with scheduling in batch processes. In Chapter 8, a real-life multiproduct multistage pharmaceuticals scheduling problem is considered. A systematic two-stage iterative solution strategy, based on mathematical programming, has been developed. More specifically, the proposed solution strategy consists of a constructive step, wherein a feasible and initial solution is rapidly generated by following an iterative insertion procedure, and an improvement step, wherein the initial solution is systematically enhanced by implementing iteratively several rescheduling techniques; based on the mathematical model. A salient feature of the proposed approach is that the scheduler can maintain the number of decisions at a reasonable level thus reducing appropriately the search space. This usually results in manageable model sizes that often guarantees a more stable and predictable optimization model behavior. Several challenging large-scale scheduling problem instances, considering alternative optimization goals, of a pharmaceuticals production facility have been solved. Also, it is worth mentioning that a new precedence concept (i.e., the unit-specific general precedence that is included in Appendix B) has been developed in order to cope with objectives containing changeover issues.

Finally, in Chapter 9, a two-layered decomposition methodology to the batch process scheduling problem in multipurpose production plants has been developed. In the first level, an approximate scheduling model derived from the detailed STN-based time-indexed scheduling formulation is solved; the model partially relaxes the allocation of task instances to processing units details of the full scheduling formulation. In the second level, the output of the approximate scheduling problem is used to provide batching targets for the detailed scheduling model within an iterative decomposition scheme. The procedure is tested on published problem instances of the Westenberger-Kallrath benchmark scheduling problem considering a polymers production plant. Despite the promising results, future work is needed in order to further improve the performance of proposed decomposition strategy.

## 10.2 Future work

This thesis has been focused on the development of smart modeling concepts and mathematical programming approaches to efficiently tackle real-life industrial planning and/or scheduling problems in the process industries, in an attempt to make more attractive the mathematical approaches and bridge the gap between planning and scheduling theory and practice. A range of issues requiring further investigation has been revealed in the course of this work. In particular:

- Further study and improvement of the mathematical-based solution techniques. For instance, the MIP-based solution strategy, presented in Chapter 8, could be combined with metaheuristics in an attempt to reduce the computational burden.
- Since process industries are dynamic in nature, the consideration of the uncertainty also arises as a challenging future research task. As briefly described in Section 2.4, proactive (e.g., stochastic programming or parametric optimization) or reactive (e.g., full or partial re-planning) approaches could be used (Acevedo & Pistikopoulos, 1997b,a; Pistikopoulos *et al.*, 2002; Balasubramanian & Grossmann, 2004; Dua, Kouramas, Dua, & Pistikopoulos, 2008).
- Continual improvement in mathematical problem formulation and preprocessing to improve relaxation characteristics, and tailor-made solution procedures for problems with relatively large integrality gaps.
- More attention should be given in the multisite problem. A major task should be the simultaneous optimization of production and logistics operations across multiple production facilities and distribution centers, in order to enhance the overall performance, responsiveness, and profitability of the enterprise.
- Regarding multisite problems, when an extremely large number of final products, production sites and distribution centers are present, appropriate modeling frameworks and solution strategies should be devised in order to tackle efficiently these highly complicated optimization problems.
- Implementation of the proposed models, after further improvements,in a computer-aided advanced scheduling and planning system.
- Further improvement of the bi-level decomposition strategy presented in Chapter 9.
- There are process industries that have received little attention; regarding scheduling and/or planning research, such as the food industries considered in this thesis. For instance, scheduling and planning approaches in the ceramics and tiles process industry is rather poor, despite the fact that there are many optimization challenges.
- New efficient approaches to integrate scheduling decisions into SC design may be further explored. One interesting approach to be examined is the use of attainable regions for modeling feasible aggregated production rates (Sung & Maravelias, 2007).

- More rigorous treatment of the financial aspects as scheduling and planning become integrated (Badell, Romero, Huertas, & Puigjaner, 2004; Guillén, Badell, & Puigjaner, 2007; Laínez *et al.*, 2007). And, further extension of the proposed mathematical approaches to address environmental and sustainability considerations (Stefanis, Livingston, & Pistikopoulos, 1997), thus necessitating the development of multiobjective optimization frameworks (Papageorgiou, 2009).

- Solution techniques and concepts developed for the planning and scheduling of the process industries can be implemented to other classes of planning or scheduling problems. For instance, process scheduling concepts could be applied into project scheduling problems.

# Appendixes

Appendix A

# Publications

This is a list of the works carried out so far within the scope of interest of this thesis, in reversed chronological order. The list has been divided in manuscripts to international refereed journals and conference proceedings.

## Scientific Journals

### Manuscripts published

[1] Kopanos, G. M.; Puigjaner, L.; Maravelias, C. T. Production Planning and Scheduling of Parallel Continuous Processes with Product Families. *Industrial and Engineering Chemistry Research*, ISSN: 0888-5885, 50 (3): 1369 – 1378 (**2011**).

[2] Kopanos, G. M.; Méndez, C. A.;Puigjaner, L. MIP-based Decomposition Strategies for Large-scale Scheduling Problems in Multiproduct Multistage Batch Plants: A Benchmark Scheduling Problem of the Pharmaceutical Industry. *European Journal of Operational Research*, ISSN: 0377-2217, 207 (2): 644 – 655 (**2010**).

[3] Kopanos, G. M.; Puigjaner, L.; Georgiadis, M. C. Optimal Production Scheduling and Lot-Sizing in Dairy Plants: The Yogurt Production Line. *Industrial and Engineering Chemistry Research*, ISSN: 0888-5885, 49 (2): 701 – 718 (**2010**).

[4] Kopanos, G. M.; Laínez, J. M.; Puigjaner, L. An Efficient Mixed-Integer Linear Programming Scheduling Framework for Addressing Sequence-Dependent

Setup Issues in Batch Plants. *Industrial and Engineering Chemistry Research*, ISSN: 0888-5885, 48 (13): 6346 – 6357 (**2009**).

[5] Laínez, J. M.; Kopanos, G.; Espuña, A.; Puigjaner, L. Flexible Design-Planning of Supply Chain Networks. *AIChE Journal*, ISSN: 0001-1541, 55 (7): 1736 – 1753 (**2009**).

[6] 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 and Engineering Chemistry Research*, ISSN: 0888-5885, 47 (22): 8785 – 8795 (**2008**).

### Manuscripts submitted

[1] Kopanos, G. M.; Puigjaner, L.; Georgiadis, M. C. Resource-Constrained Production Planning in Semicontinuous Food Industries. *Computers and Chemical Engineering* (**2011**).

[2] Kyriakidis, T. S.; Kopanos, G. M.; Georgiadis, M. C. MILP Formulations for Single- and Multi-Mode Resource-Constrained Project Scheduling Problems. *Computers and Chemical Engineering* (**2011**).

[3] Kopanos, G. M.; Puigjaner, L.; Georgiadis, M. C. Production Scheduling in Multiproduct Multistage Semicontinuous Food Processes. *Industrial and Engineering Chemistry Research* (**2011**).

[4] Kopanos, G. M.; Puigjaner, L.; Georgiadis, M. C. Production and Logistics Operations Planning in Semicontinuous Food Processing Industries. *European Journal of Operational Research* (**2010**).

## Conference Proceeding Articles

The work realized in this thesis has been also presented and published to different international specialized conferences. A list of publications in conferences proceedings follows.

[1] Kopanos, G. M.; Puigjaner, L.; Georgiadis, M. C.; Bongers, P. M. M. An efficient mathematical framework for detailed production scheduling in food industries: The ice-cream production line. *21$^{st}$ European Symposium on Computer Aided Process Engineering, Chalkidiki, Greece*, (**2011**).

[2] Kyriakidis, T. S.; Kopanos, G. M.; Georgiadis, M. C. MILP formulations for resource-constrained project scheduling problems. *21$^{st}$ European Symposium on Computer Aided Process Engineering, Chalkidiki, Greece*, (**2011**).

[3] Kopanos, G. M.; Puigjaner, L.; Georgiadis, M. C. Optimal production scheduling and lot-sizing in yoghurt production lines. *20$^{th}$ European Symposium on Computer Aided Process Engineering, Ischia - Naples, Italy* (Eds. S. Pierucci and G. Buzzi Ferraris: Elsevier, Amsterdam) *Computer Aided Chemical Engineering*, ISBN 13: 978-0-444-53569-6, 28: 1153 – 1158 (**2010**).

[4] Kopanos, G. M.; Puigjaner, L.; Méndez, C. A. A decomposition strategy for large-scale scheduling problems in multi-stage multi-product batch plants. *International Conference on Industrial Logistics, Rio de Janeiro, Brazil* (Eds. M. C. F. de Sinay, M. I. Faé and A. G. Canen): 125 – 132 (**2010**).

[5] Kopanos, G. M.; Puigjaner, L. Simultaneous batching and scheduling in multi-product multi-stage batch plants through mixed-integer linear programming. *13$^{th}$ Conference on Process Integration, Modeling and Optimization for Energy Saving and Pollution Reduction, Prague, Czech Republic* (Eds. J. J. Klemeš, H. L. Lam and P. S. Varbanov) *Chemical Engineering Transactions*, ISBN 978-88-95608-05-1, 21: 505 – 510 (**2010**).

[6] Kopanos, G. M.; Méndez, C. A.; Puigjaner, L. Solving scheduling problems in multi-stage multi-product batch in pharmaceuticals industry. *13$^{th}$ Conference on Process Integration, Modeling and Optimization for Energy Saving and Pollution Reduction, Prague, Czech Republic* (Eds. J. J. Klemeš, H. L. Lam and P. S. Varbanov) *Chemical Engineering Transactions*, ISBN 978-88-95608-05-1, 21: 511 – 516 (**2010**).

[7] Kopanos, G. M.; Puigjaner, L. Multi-site scheduling/batching and production planning for batch process industries. *10$^{th}$ International Symposium on Process Systems Engineering, Salvador-Bahía, Brazil* (Eds. R. de Brito Alves, C. O. de Nascimento and E. Biscaia: Elsevier, Amsterdam) *Computer Aided Chemical Engineering*, ISBN 13: 978-0-444-53472-9, 27: 2109 – 2114 (**2009**).

[8] Kopanos, G. M.; Puigjaner, L.; Georgiadis, M. C. A bi-level decomposition methodology for scheduling batch chemical production facilities. *10$^{th}$ International Symposium on Process Systems Engineering, Salvador-Bahía, Brazil* (Eds. R. de Brito Alves, C. O. de Nascimento and E. Biscaia: Elsevier, Amsterdam) *Computer Aided Chemical Engineering*, ISBN 13: 978-0-444-53472-9, 27: 681 – 686 (**2009**).

[9] Kopanos, G. M.; Puigjaner, L. A MILP scheduling model for multi-stage batch plants. *19$^{th}$ European Symposium on Computer Aided Process Engineering, Cracow, Poland* (Eds. J. Jezowski and J. Thullie: Elsevier, Amsterdam) *Computer Aided Chemical Engineering*, ISBN 13: 978-0-444-53433-0, 26: 369 – 374 (**2009**).

[10] Muñoz, E.; Kopanos, G. M.; Espuña, A.; Puigjaner, L. Towards an ontological informatics infrastructure in chemical batch process. *19$^{th}$ European Symposium on Computer Aided Process Engineering, Cracow, Poland* (Eds. J.

Jezowski and J. Thullie: Elsevier, Amsterdam) *Computer Aided Chemical Engineering*, ISBN 13: 978-0-444-53433-0, 26: 883 – 888 (**2009**).

[11] Laínez, J. M.; Kopanos, G. M.; Badell, M.; Espuña, A.; Puigjaner, L. Integrating strategic, tactical and operational supply chain decision levels in a model predictive control framework. *18$^{th}$ European Symposium on Computer Aided Process Engineering, Lyon, France* (Eds. B. Braunschweig and X. Julia: Elsevier, Amsterdam) *Computer Aided Chemical Engineering*, ISBN 13: 978-0-444-53227-5, 25: 477 – 482 (**2008**).

[12] Capón-García, E.; Kopanos, G. M; Espuña, A.; Puigjaner, L. A novel proactive-reactive scheduling approach in chemical multiproduct batch plants. *18$^{th}$ European Symposium on Computer Aided Process Engineering, Lyon, France* (Eds. B. Braunschweig and X. Julia: Elsevier, Amsterdam) *Computer Aided Chemical Engineering*, ISBN 13: 978-0-444-53227-5, 25: 435 – 440 (**2008**).

## Other Conferences and Congresses

Kopanos, G. M.; Puigjaner, L.; Georgiadis, M. C. Scheduling and lot-sizing in the dairy industry: The yoghurt production case. *AIChE Annual Meeting, Nashville, USA* (Eds. W. S. Winston Ho and T. B. Halloway: OMNIPRESS), ISBN: 978-0-8169-1058-6: 472d, (2009).

Kopanos, G. M.; Laínez, J. M.; Puigjaner, L. Short-term scheduling in multi-stage batch plants through lagrangean decomposition. *AIChE Annual Meeting, Nashville, USA* (Eds. W. S. Winston Ho and T. B. Halloway: OMNIPRESS), ISBN: 978-0-8169-1058-6: 423d, (2009).

Laínez, J. M.; Kopanos, G. M.; A. Espuña; Puigjaner, L. Supply chain design considering operational level details. *AIChE Annual Meeting, Nashville, USA* (Eds. W. S. Winston Ho and T. B. Halloway: OMNIPRESS), ISBN: 978-0-8169-1058-6: 423d, (2009).

Kopanos, G. M.; Puigjaner, L. The unit-specific general precedence scheduling model: Addressing sequence-dependent setup times and costs issues. *AIChE Annual Meeting, Philadelphia, USA*, (OMNIPRESS), ISBN: 978-0-8169-1050-2: 554d, (2008).

Kopanos, G. M.; Laínez, J. M.; Espuña, A.; Puigjaner, L. Enhancing supply chain network design by considering financial analysis issues. *AIChE Annual Meeting, Philadelphia, USA*, (OMNIPRESS), ISBN: 978-0-8169-1050-2: 577i, (2008).

Kopanos, G. M.; Laínez, J. M.; Puigjaner, L. Supply chain management considering international logistics - production issues. *11$^{th}$ Mediterranean Congress of Chemical Engineering*, Barcelona, Spain, 2008.

Laínez, J. M.; Kopanos, G. M.; Espuña, A.; Puigjaner, L. A process operations conscious design of supply chains. *$11^{th}$ Mediterranean Congress of Chemical Engineering*, Barcelona, Spain, 2008.

Kopanos, G. M.; Puigjaner, L. Scheduling of multi-product single-stage chemical batch plants considering changeovers issues. *$11^{th}$ Mediterranean Congress of Chemical Engineering*, Barcelona, Spain, 2008.

Kopanos, G. M.; Laínez, J. M.; Pérez-Fortes, M.; Puigjaner, L. Biomass for energy production supply chain network design. *$11^{th}$ Mediterranean Congress of Chemical Engineering*, Barcelona, Spain, 2008.

Kopanos, G. M.; Espuña, A.; Puigjaner, L. Rescheduling actions by preserving schedule stability in multiproduct batch plants. *Computer Aided Process Engineering Forum* (CAPE FORUM), Thessaloniki, Greece, 2008.

Kopanos, G. M; Capón-García, E.; Bonfill, A.; Espuña, A.; Puigjaner, L. Towards a novel proactive-reactive scheduling approach in chemical multiproduct batch plants. *AIChE Annual Meeting*, Salt Lake City, USA, 2007.

Kopanos, G. M.; Laínez, J. M.; Badell, M.; Espuña, A.; Puigjaner, L. Operations and logistic planning considering vendor uncertainties to enhance the flexibility and reduce the risk of chemical supply chain networks. *$6^{th}$ European Congress of Chemical Engineering, Copenhagen, Denmark* (Eds. R. Gani and K. Dam-Johansen: Norhaven Books) ISBN: 978-87-91435-56-0: 483 – 484, (2007).

Badell, M.; Kopanos, G. M.; Laínez, J. M.; Espuña, A.; Puigjaner, L. Making value with order management for agent based systems. *$6^{th}$ European Congress of Chemical Engineering, Copenhagen, Denmark* (Eds. R. Gani and K. Dam-Johansen: Norhaven Books) ISBN: 978-87-91435-56-0: 657 – 658, (2007).

## Participation in Research Projects

EHMAN: *Expanding the Horizons of Manufacturing: Solving the Integration Paradox*, supported by the "Ministerio de Ciencia e Innovacin (MICINN)" and the "European Regional Development Fund (ERDF, European Community)" (DPI2009-09386), 2010 – 2013.

ToleranT: *Sistema de Soporte Avanzado para Procesos de Fabricación Flexible en la Industria Química y Petroquímica*, supported by the "Ministerio de Ciencia e Innovación (MICINN)" and the "European Regional Development Fund (ERDF, European Community)" (DPI2006-05673), 2006 – 2009.

XaRTAP: *Xarxa de Referència en Tècniques Avançades de la Producció. Estació de Muntatge Universal*, supported by the "Direcció General de Recerca de la Generalitat de Catalunya" (I0303), 2006 – 2008.

# Appendix B

## The Unit-Specific General Precedence Concept

This appendix contains a detailed description of the unit-specific general precedence concept used in the MIP-based solution strategy presented previously in Chapter 8.

### Introduction

Continuous time representation strategies based on the precedence relationships between batches to be processed have been developed to deal with the process scheduling problem. In these mathematical formulations, model variables and constraints enforcing the sequential use of shared resources are explicitly employed, and therefore sequence-dependent setups (changeovers) can be treated straight-forwardly Méndez *et al.* (2006). The three different precedence-based approaches that can be found in the literature are: (i) the immediate precedence, (ii) the unit-specific immediate precedence, and (iii) the general precedence. Immediate (or local) Precedence (IP) explores the relation between each pair of consecutive orders in the production schedule time horizon without taking into account whether the orders are assigned to the same unit. Unit-Specific Immediate Precedence (USIP) is based on the immediate precedence concept. The difference is that it only takes into account the immediate precedence of the orders that are assigned to the same processing unit. General (or global) Precedence (GP) generalizes the precedence concept by exploring the precedence relations of each batch, taking into account all the remaining batches and not only the immediate predecessor Méndez and Cerdá (2003a). The last approach results in a lower number of binary variables and, compared with the other two approaches, it significantly reduces the computational effort on average. However, it cannot cope with changeovers issues explicitly (especially if there are changeover times greater than a batch processing time), as it is clearly demonstrated in the illustrative example. Moreover, scheduling models based on the GP notion cannot be used to address problems with sequence-dependent changeover costs because the global-sequencing variables are active for all the batch pairs assigned to the same unit. In order to address this limitation, a new precedence-based scheduling formulation, the Unit-Specific General Precedence (USGP), is developed here. Figure B.1 shows the precedence-based frameworks for the scheduling in batch processes.

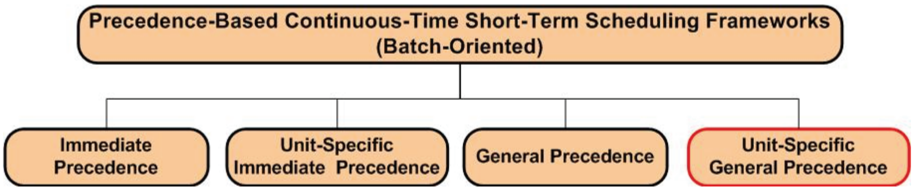



**Figure B.1:** Current and proposed precedence-based scheduling frameworks.

## Problem Statement

The scheduling problem in single-stage multiproduct batch plants with different processing units working in parallel is addressed here. Batch to unit assignment and batch sequencing in order to meet a production goal constitutes the under study scheduling problem. Changeover times, which greatly increase the complexity of the problem, are explicitly considered. The main assumptions of the proposed model include:

(i) Only single-stage product orders are considered.

(ii) An equipment unit cannot process more than one batch at a time.

(iii) To begin another task in a processing unit, the current task must have been completed (i.e., non-preemptive operation mode).

(iv) Processing times, unit setup times and changeover times and/or costs and due dates are deterministic.

(v) Unforeseen events, such as unit breakdowns, that may disrupt the normal plant operation do not appear during the scheduling time horizon.

(vi) No resource constraints except for equipment availability are taken into account.

(vii) Product batch sizing is carried out beforehand, and thus batch sizes are known a priori.

In industrial batch plants, assumption (iii) is frequently satisfied. This is not the case, however, for assumption (i), since there are many industrial applications that are multistage. The proposed model can be appropriately modified to deal with multistage plants (see Chapter 8). By adding a set of resource constraints similar to the ones that were reported in the work by Marchetti and Cerdá (2009b), assumption (vi) can be satisfied. If assumptions (iv) and/or (v) are relaxed, then uncertainty should be included in the optimization procedure. There are generally two ways of treating unexpected events: proactively (off-line) or reactively (on-line). The proposed scheduling framework can be easily adapted to both cases. Finally, assumption (vii) is in accordance with the sequential modeling strategy in scheduling problems, in which first the lot-sizing problem is solved and then, once the number and sizes of batches are known, the pure scheduling problem is solved. This approach probably results in less optimal solutions than the monolithic approach, in which lot-sizing and scheduling are simultaneously optimized. However, the sequential scheduling approach is less computationally expensive, and in some cases it can be viewed as a good approximation to the industrial reality.

## The Unit-Specific General Precedence Scheduling Model

In this section, the proposed MIP scheduling model is described in detail. The concept of the USGP is also introduced and explained. In the proposed mathematical formulation, constraints have been grouped according to the type of decision (e.g., assignment, timing and sequencing) they are imposed on.

**Allocation Constraints.** Constraint set (B.1) presents the unit allocation constraints for every order $i$. As this expression states, each order $i$ can be assigned to only one processing unit $j$ or to none (permitting unsatisfied demand). $Y_{ij}$ represents the binary decision of whether to assign a product order $i$ to a processing unit $j$ or not. $Y_{ij}$ is active, i.e. $Y_{ij} = 1$, whenever product $i$ is allocated to unit $j$, otherwise it is set to zero. Let $J_i$ denote the set of units $j$ that can process product $i$. By changing the inequality to an equality total demand satisfaction is imposed.

$$\sum_{j \in JI_i} Y_{ij} \leq 1 \qquad \forall i \tag{B.1}$$

**Timing Constraints.** The completion time $C_i$ for batch $i$, when it is assigned to unit $j$, should be greater than the summation of its corresponding processing time $\tau_{ij}$ and setup time $\pi_{ij}$ in this unit $j$. The maximum of the ready unit time $\varepsilon_j$ and the release order time $o_i$ are also added to this summation as the following equation states.

$$C_i \geq \sum_{j \in J_i} (max[\varepsilon_j, o_i] + \tau_{ij} + \pi_{ij}) Y_{ij} \qquad \forall i \tag{B.2}$$

**Sequencing-Timing Constraints.** Constraint set (B.3) expresses the order sequencing constraints between two orders, $i$ and $i'$. This equation is formulated as a big-M constraint, and it is activated for every task $i'$ that is processed after task $i$, when both are assigned to the same unit $j$ (i.e., $X_{ii'j} = 1$).

$$C_i + \gamma_{ii'j}\bar{X}_{ii'j} \leq C_{i'} - \tau_{i'j} - \pi_{i'j} + M\,(1 - X_{ii'j}) \qquad \forall i, i' \neq i, j \in (J_i \cap J_{i'}) \quad \text{(B.3)}$$

Binary variable $\bar{X}_{ii'j}$ defines the immediate precedence of two tasks $i$ and $i'$, when both are assigned to the same unit $j$. If two orders $i$ and $i'$ are allocated to the same processing unit $j$ and order $i'$ is processed directly after order $i$, then $\bar{X}_{ii'j} = 1$. Hence, this formulation allows us to consider explicitly and efficiently the changeover times and/or costs.

**Sequencing-Allocation Constraints.** In order to assess binary variable $\bar{X}_{ii'j}$ with the sequencing-assignment binary variable $X_{ii'j}$, a set of additional constraints, presented below, is needed. As mentioned above, the binary variable $X_{ii'j}$ only stands for two products, $i$ and $i'$, that are assigned to the same equipment unit $j$. In disjunctive programming this statement can be expressed as follows:

$$X_{ii'j} \Rightarrow \left[Y_{ij} \wedge Y_{i'j}\right] \qquad \forall i, i' \neq i, j \in (I_i \cap J_{i'})$$

Later, the aforementioned disjunctive programming expression is decomposed into constraints (B.4) and (B.5). It can be clearly seen that $X_{ii'j}$ may take the value of 1 only if both orders $i$ and $i'$ are into the same unit $j$; otherwise, it is set to zero without exploring the sequencing of the orders further.

$$Y_{ij} + Y_{i'j} \leq 1 + X_{ii'j} + X_{i'ij} \qquad \forall i, i', j \in (J_i \cap J_{i'}) : i' > i \qquad \text{(B.4)}$$

$$2\,(X_{ii'j} + X_{i'ij}) \leq Y_{ij} + Y_{i'j} \qquad \forall i, i', j \in (J_i \cap J_{i'}) : i' > i \qquad \text{(B.5)}$$

In order to explicitly tackle scheduling problems with changeover times and/or costs, the immediate precedence of every pair of orders must be assessed. We now describe our approach. Obviously, two orders $i$ and $i'$ may be consecutive only in the case that the sequencing-allocation binary variable is $X_{ii'j} = 1$ and when there is no other order $i''$ between orders $i$ and $i'$, and vice versa. In disjunctive programming this expression can be stated as follows:

$$X_{ii'j} \wedge \neg \left[ \bigvee_{i'' \neq [i,i']} (X_{ii''j} - X_{i'i''j}) \right] \Leftrightarrow \bar{X}_{ii'j} \qquad \forall i, i' \neq i, j \in (J_i \cap J_{i'})$$

Figure 8.1, found in page 164, shows an illustrative example of the concept of this expression. It can be seen that if the total number of batches that follow batch $i$, excluding batch $i'$ is equal to the total number of batches that follow batch $i'$, excluding batch $i$, then batches $i$ and $i'$ are consecutive. For the sake of simplicity, only one processing unit is considered. As a unique machine case is studied, the unit index $j$ is omitted in this example. Constraints (B.6) and (B.7) correspond to the mathematical formulation of the aforementioned disjunctive programming expression. Constraint set (B.6) states that the auxiliary variable $Z_{ii'j}$ will be set to zero only if batch $i$ is processed before batch $i'$ and they are allocated to the same equipment unit $j$ (i.e., $X_{ii'j} = 1$) and simultaneously $\sum_{i'' \neq [i,i']} X_{ii''j} - X_{i'i''j} = 0$ (see Figure 8.1 in page 164).

$$Z_{ii'j} = \sum_{i'' \neq [i,i']} (X_{ii''j} - X_{i'i''j}) + M\,(1 - X_{ii'j}) \qquad \forall i, i' \neq i, j \in (J_i \cap J_{i'}) \tag{B.6}$$

If the position difference variable $Z_{ii'j}$ is equal to zero, i.e, when order $i$ has been processed exactly before order $i'$, constraint set (B.7) activates the binary variable $Seq_{ii'j}$, i.e., $\bar{X}_{ii'j}$=1. Therefore, the consecutiveness of the orders is assessed and sequence-dependent setup time and/or cost issues can be effectively treated.

$$Z_{ii'j} + \bar{X}_{ii'j} \geq 1 \qquad \forall i, i' \neq i, j \in (J_i \cap J_{i'}) \tag{B.7}$$

**Objective Function.** Different objective functions can be optimize by using the proposed scheduling framework. For instance, the earliness and the tardiness for every product order $i$ are given by:

$$E_i \geq \delta_i - C_i \quad \forall i \tag{B.8}$$

$$T_i \geq C_i - \delta_i \quad \forall i \tag{B.9}$$

The minimization of a combined function of earliness and tardiness, as given in equation (B.10), is one of the most widely used objective functions in the scheduling literature. It is also known as weighted lateness. The weighting coefficients $\alpha_i$ and $\beta_i$ are used to specify the significance of order earliness or tardiness respectively.

$$min \sum_i (\alpha_i\, E_i + \beta_i\, T_i) \tag{B.10}$$

If tardiness is not permitted ($T_i = 0$) for any order $i$, the aforementioned objective function can be substituted by the maximization of the order completion time, $C_i$:

$$max \sum_i C_i \tag{B.11}$$

This objective function is identical to the minimization of earliness. Note that if changeover costs are proportional to the changeover times, then the minimization of earliness will correspond to minimization of changeover costs. Alternative objective functions can be also used (e.g., minimization of makespan, total costs minimization, changeover costs minimization, maximization of profit).

At this point it should be noted that general precedence based formulations cannot optimize changeover costs, because batch consecutiveness is not assessed explicitly. In the process industries, changeover considerations and optimization are of great importance. Moreover, since scheduling constitutes a part of the supply chain network optimization problem which ought to meet financial goals, it must also be examined and optimized considering financial and economic issues. In view of this industrial strategy in the contemporary highly competitive market environment, the significant advantages of adopting the proposed scheduling framework (to explicitly deal with changeover issues) are clear. What is more, general precedence models may generate suboptimal solution when changeover times are greater than processing times, as will be demonstrated in the illustrative example presented.

## A General Precedence Scheduling Model

In this section, the representative precedence-based mathematical formulation of Méndez and Cerdá (2003a) is presented.

$$\sum_{j \in J_i} Y_{ij} = 1 \qquad \forall i \tag{B.12}$$

$$C_i \geq \sum_{j \in J_i} (max[\varepsilon_j, o_i] + \tau_{ij} + \pi_{ij}) Y_{ij} \qquad \forall i \tag{B.13}$$

$$\begin{aligned} C_i + \gamma_{ii'j} \leq C_{i'} - \tau_{i'j} - \pi_{i'j} + M\,(1 - X^{GP}_{ii'}) + M\,(2 - Y_{ij} - Y_{i'j}) \\ \forall i, i', j \in (J_i \cap J_{i'}) : i' > i \end{aligned} \tag{B.14}$$

$$\begin{aligned} C_{i'} + \gamma_{i'ij} \leq C_i - \tau_{ij} - \pi_{ij} + M\,X^{GP}_{ii'} + M\,(2 - Y_{ij} - Y_{i'j}) \\ \forall i, i', j \in (J_i \cap J_{i'}) : i' > i \end{aligned} \tag{B.15}$$

Constraint set (B.12) corresponds to the unit allocation constraints, and constraint set (B.13) defines the completion time for every product. Constraint sets (B.14) and (B.15) define the relative sequencing of product batches at each processing unit. These sets of big-M constraints enforce the starting time of a product $i'$ to be greater than the completion time of whichever product $i$ processed beforehand. Note that $X^{GP}_{ii'}$ corresponds to the global sequencing binary variable, which is active (i.e., $X^{GP}_{ii'} = 1$) for all products $i'$ that are processed after product $i$.

## Illustrative Example

A modified version of a plastic compounding plant, first introduced by Pinto and Grossmann (1995), is used as a simple illustrative example. Table B.1 shows the data for this scheduling problem. The optimization goal is to minimize earliness and tardiness (where $\alpha_i = 1$ and $\beta_i = 5$).

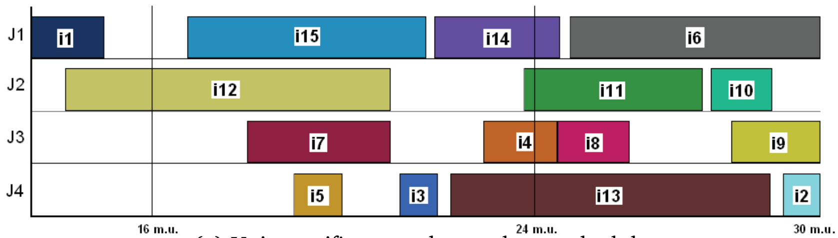
First, the addressed scheduling problem is solved without changeover times considerations (i.e $\gamma_{ii'j} = 0$). The objective function value is equal to 6.681 m.u.; the schedule obtained is shown in Figure B.2a. Afterwards, the scheduling problem is solved by considering changeover times only between order $i_{12}$ and order $i_{10}$

**Table B.1:** Data for the motivating example.

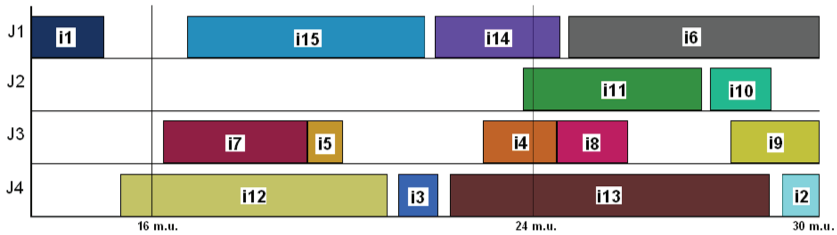
| order | due date (day) | processing times (days) unit 1 | unit 2 | unit 3 | unit 4 |
|---|---|---|---|---|---|
| 1 | 15 | 1.538 | | | 1.194 |
| 2 | 30 | 1.500 | | | 0.789 |
| 3 | 22 | 1.607 | | | 0.818 |
| 4 | 25 | | | 1.564 | 2.143 |
| 5 | 20 | | | 0.736 | 1.017 |
| 6 | 30 | 5.263 | | | 3.200 |
| 7 | 21 | 4.865 | | 3.025 | 3.214 |
| 8 | 26 | | | 1.500 | 1.440 |
| 9 | 30 | | | 1.869 | 2.459 |
| 10 | 29 | | 1.282 | | |
| 11 | 30 | | 3.750 | | 3.000 |
| 12 | 21 | | 6.796 | 7.000 | 5.600 |
| 13 | 30 | 11.250 | | | 6.716 |
| 14 | 25 | 2.632 | | | 1.527 |
| 15 | 24 | 5.000 | | | 2.985 |
| setup times (days) | | 0.180 | 0.175 | - | 0.237 |

($\gamma_{i_{12}i_{10}j} = 9$). As the schedule in Figure B.2a shows, order $i_{12}$ and order $i_{10}$ are not consecutive; thus, someone will expect to obtain the same schedule even if a changeover time is assigned between these orders. However, this is not the case when the general precedence model is applied. Figure B.2b illustrates the schedule obtained, which is, perhaps unsurprisingly, different from the previous one. Its objective function value is equal to 8.472 m.u. (26% worse).

The cause of this fault is that the general precedence sequencing constraints take into account all the sequence-dependent changeover times (and not only between consecutive orders) of the orders assigned to the same unit $j$. This point can be clearly seen by observing constraints (B.14) and (B.15). It can be clearly seen that changeover times, $\gamma_{ii'j}$, are taken into account whenever the general precedence sequencing variable $X_{ii'}^{GP}$ is active (i.e., $X_{ii'}^{GP} = 1$).




(a) Unit-specific general precedence schedule.




(b) General precedence schedule.

**Figure B.2:** Motivating example schedules: General precedence fault demonstration.

Therefore, coming back to the illustrative example, because $X_{i_{12}i_{10}} = 1$ (in the schedule in Figure B.2a order $i_{12}$ is processed before order $i_{10}$, although they are not consecutive orders) the sequence-dependent changeover time $\gamma_{i_{12}i_{10}j} = 9$ (of two no-consecutive orders) is incorrectly taken into account. In other words, the general precedence model will try to make an order completion time plus the changeover time less than or equal to the starting time of any other following order, and consecutiveness is not explicitly considered.

All in all, in cases that there exist some sequence-dependent changeover times higher than some batch processing times, as in this illustrative example, general precedence may result to a suboptimal solution. If all changeover times are lower than all the orders processing times then general precedence is valid and can be implemented. Nevertheless, note that sequence-dependent changeover cost issues still cannot be addressed explicitly by general precedence models.

# Nomenclature

**Indices / Sets**

| | |
|---|---|
| $i, i', i'' \in I$ | product orders (products) |
| $j \in J$ | processing units (units) |

**Subsets**

| | |
|---|---|
| $J_i$ | available units $j$ to process product $i$ |

**Parameters**

| | |
|---|---|
| $\alpha_i$ | weighing coefficient for earliness for product $i$ |
| $\beta_i$ | weighing coefficient for tardiness for product $i$ |
| $\gamma_{ii'j}$ | sequence-dependent setup (changeover) time between products $i$ and $i'$ in unit $j$ |
| $\delta_i$ | due date for product $i$ |
| $\varepsilon_j$ | time point that unit $j$ is available to start processing |
| $M$ | a big number |
| $o_i$ | release time for product $i$ |
| $\pi_{ij}$ | sequence-independent setup time of product $i$ in unit $j$ |
| $\tau_{ij}$ | processing time for product $i$ in unit $j$ |

**Continuous Variables**

| | |
|---|---|
| $C_i$ | completion time of product $i$ |
| $E_i$ | earliness for product $i$ |
| $T_i$ | tardiness for product $i$ |
| $Z_{ii'j}$ | allocation position difference between products $i$ and $i'$ in unit $j$ |

**Binary Variables**

| | |
|---|---|
| $X_{ii'j}$ | $= 1$ for every product $i$ that is processed before product $i'$ in unit $j$ |
| $\bar{X}_{ii'j}$ | $= 1$ if product $i$ is processed exactly before product $i'$ in unit $j$ |
| $X^{GP}_{ii'j}$ | $= 1$ for every product $i$ that is processed before product $i'$ |
| $Y_{ij}$ | $= 1$ if product $i$ is assigned to unit $j$ |

Appendix B

# Appendix C

## Data for the Resource-Constrained Yoghurt Production Process

This appendix contains the main processing data for the yogurt production process described and studied in Chapter 6. More specifically, Table C.1 provides the main data for all final yogurt products including: (i) the assigned products to families set $P_f$, (ii) the product cup weight, (iii) the product inventory cost $\xi_{pn}$, and (iv) the minimum production amount for any product. Packing rates for every product can be found in Table C.2. Changeover times among families are given in Table C.3. Changeover costs are related to changeover times, according to Table C.4, but setups are irrelevant. Finally, Table C.5 provides the production targets (for Case Study I and II).

**Table C.1:** Main data for final products.

| Product | Family | weight (kg) | inv. cost (€) | min. run (kg) |
|---|---|---|---|---|
| P01 | F01 | 0.600 | 9.00 | 82.80 |
| P02 | F01 | 0.600 | 7.50 | 82.80 |
| P03 | F01 | 0.600 | 6.75 | 82.80 |
| P04 | F01 | 0.600 | 6.00 | 82.80 |
| P05 | F01 | 0.600 | 6.00 | 82.80 |
| P06 | F02 | 0.600 | 6.00 | 82.80 |
| P07 | F02 | 0.600 | 5.25 | 82.80 |
| P08 | F02 | 0.600 | 5.25 | 82.80 |
| P09 | F03 | 0.600 | 6.00 | 82.80 |
| P10 | F03 | 0.600 | 5.25 | 82.80 |
| P11 | F03 | 0.600 | 5.25 | 82.80 |
| P12 | F04 | 0.600 | 8.25 | 82.80 |
| P13 | F04 | 0.600 | 6.75 | 82.80 |
| P14 | F04 | 0.600 | 6.00 | 82.80 |
| P15 | F04 | 0.600 | 5.25 | 82.80 |
| P16 | F04 | 0.600 | 5.25 | 82.80 |
| P17 | F05 | 0.600 | 9.00 | 82.80 |
| P18 | F05 | 0.600 | 6.75 | 82.80 |
| P19 | F05 | 0.600 | 7.50 | 82.80 |
| P20 | F05 | 0.600 | 6.00 | 82.80 |
| P21 | F05 | 0.200 | 5.70 | 27.60 |
| P22 | F05 | 0.200 | 5.70 | 27.60 |
| P23 | F06 | 0.400 | 0.75 | 55.20 |
| P24 | F06 | 0.200 | 0.75 | 27.60 |
| P25 | F06 | 0.200 | 0.75 | 27.60 |
| P26 | F07 | 0.380 | 0.75 | 19.76 |
| P27 | F07 | 0.380 | 0.75 | 19.76 |
| P28 | F08 | 0.400 | 1.35 | 55.20 |
| P29 | F08 | 0.400 | 1.35 | 55.20 |
| P30 | F08 | 0.600 | 1.65 | 82.80 |
| P31 | F08 | 0.600 | 1.65 | 82.80 |
| P32 | F08 | 0.600 | 1.65 | 82.80 |
| P33 | F08 | 0.600 | 1.65 | 82.80 |
| P34 | F09 | 0.450 | 0.45 | 62.10 |
| P35 | F09 | 0.450 | 0.45 | 62.10 |
| P36 | F09 | 0.150 | 1.20 | 20.70 |
| P37 | F09 | 0.150 | 1.20 | 20.70 |
| P38 | F09 | 0.150 | 1.20 | 20.70 |
| P39 | F10 | 0.125 | 1.80 | 11.25 |
| P40 | F10 | 0.125 | 1.80 | 11.25 |
| P41 | F10 | 0.125 | 1.80 | 11.25 |
| P42 | F11 | 1.000 | 3.30 | 33.00 |
| P43 | F11 | 1.000 | 4.20 | 33.00 |
| P44 | F11 | 1.000 | 4.20 | 33.00 |
| P45 | F12 | 1.000 | 4.20 | 33.00 |
| P46 | F12 | 1.000 | 3.00 | 33.00 |
| P47 | F12 | 1.000 | 3.00 | 33.00 |
| P48 | F13 | 1.000 | 2.55 | 33.00 |
| P49 | F13 | 1.000 | 2.10 | 33.00 |
| P50 | F13 | 1.000 | 2.10 | 33.00 |
| P51 | F14 | 1.000 | 3.60 | 33.00 |
| P52 | F14 | 1.000 | 3.60 | 33.00 |
| P53 | F15 | 0.500 | 3.60 | 28.00 |
| P54 | F15 | 0.500 | 3.60 | 28.00 |
| P55 | F16 | 0.500 | 1.80 | 16.50 |
| P56 | F16 | 0.500 | 1.80 | 16.50 |
| P57 | F16 | 0.500 | 1.80 | 16.50 |
| P58 | F16 | 0.500 | 1.80 | 16.50 |
| P59 | F16 | 0.500 | 1.80 | 16.50 |
| P60 | F16 | 0.500 | 1.80 | 16.50 |
| P61 | F17 | 0.750 | 0.90 | 27.00 |
| P62 | F17 | 0.750 | 0.90 | 27.00 |
| P63 | F17 | 0.750 | 0.90 | 27.00 |
| P64 | F17 | 0.750 | 0.60 | 27.00 |
| P65 | F17 | 0.750 | 0.90 | 27.00 |
| P66 | F17 | 0.750 | 0.60 | 27.00 |
| P67 | F17 | 0.750 | 0.90 | 27.00 |
| P68 | F17 | 0.750 | 0.60 | 27.00 |
| P69 | F17 | 0.750 | 0.90 | 27.00 |
| P70 | F17 | 0.750 | 0.60 | 27.00 |
| P71 | F17 | 0.750 | 0.60 | 27.00 |
| P72 | F18 | 1.000 | 0.75 | 33.00 |
| P73 | F18 | 1.000 | 0.75 | 33.00 |
| P74 | F19 | 1.000 | 0.45 | 33.00 |
| P75 | F19 | 1.000 | 0.75 | 33.00 |
| P76 | F19 | 1.000 | 0.75 | 33.00 |
| P77 | F20 | 5.000 | 3.30 | 80.00 |
| P78 | F20 | 5.000 | 3.60 | 80.00 |
| P79 | F20 | 5.000 | 3.60 | 80.00 |
| P80 | F21 | 30.000 | 0.45 | 90.00 |
| P81 | F21 | 30.000 | 0.75 | 90.00 |
| P82 | F21 | 30.000 | 0.75 | 90.00 |
| P83 | F22 | 10.000 | 0.75 | 100.00 |
| P84 | F22 | 10.000 | 0.45 | 100.00 |
| P85 | F22 | 5.000 | 0.30 | 80.00 |
| P86 | F22 | 5.000 | 0.45 | 80.00 |
| P87 | F22 | 5.000 | 0.75 | 80.00 |
| P88 | F22 | 5.000 | 0.45 | 80.00 |
| P89 | F22 | 5.000 | 0.45 | 80.00 |
| P90 | F23 | 0.150 | 1.80 | 32.30 |
| P91 | F23 | 0.150 | 1.80 | 32.30 |
| P92 | F23 | 0.150 | 1.80 | 32.30 |
| P93 | F23 | 0.150 | 1.80 | 32.30 |

**Table C.2:** Products packing rates $\rho_{pj}$ (kg/h).

| Product | J1 | J2 | J3 | J4 |
|---|---|---|---|---|
| P01 | | | 1,710 | |
| P02 | | | 1,710 | |
| P03 | | | 1,710 | |
| P04 | | | 1,710 | |
| P05 | | | 1,710 | |
| P06 | | | 1,710 | |
| P07 | | | 1,710 | |
| P08 | | | 1,710 | |
| P09 | | | 1,710 | |
| P10 | | | 1,710 | |
| P11 | | | 1,710 | |
| P12 | | | 1,710 | |
| P13 | | | 1,710 | |
| P14 | | | 1,710 | |
| P15 | | | 1,710 | |
| P16 | | | 1,710 | |
| P17 | | | 1,710 | |
| P18 | | | 1,710 | |
| P19 | | | 1,710 | |
| P20 | | | 1,710 | |
| P21 | | | 1,820 | |
| P22 | | | 1,820 | |
| P23 | | | 1,320 | 1,150 |
| P24 | | | 1,390 | 1,204 |
| P25 | | | 1,390 | 1,204 |
| P26 | | | 1,140 | |
| P27 | | | 1,140 | |
| P28 | | | 2,400 | 2,700 |
| P29 | | | 2,400 | 2,700 |
| P30 | | | 2,250 | 2,520 |
| P31 | | | 2,250 | 2,520 |

| Product | J1 | J2 | J3 | J4 |
|---|---|---|---|---|
| P32 | | | 2,250 | 2,520 |
| P33 | | | 2,250 | 2,520 |
| P34 | | | 1,215 | 1,215 |
| P35 | | | 1,215 | 1,215 |
| P36 | | | 1,350 | 1,350 |
| P37 | | | 1,350 | 1,350 |
| P38 | | | 1,350 | 1,350 |
| P39 | | | 850 | 850 |
| P40 | | | 850 | 850 |
| P41 | | | 850 | 850 |
| P42 | | 2,200 | | |
| P43 | | 2,200 | | |
| P44 | | 2,200 | | |
| P45 | | 2,200 | | |
| P46 | | 2,200 | | |
| P47 | | 2,200 | | |
| P48 | | 2,200 | | |
| P49 | | 2,200 | | |
| P50 | | 2,200 | | |
| P51 | | 2,200 | | |
| P52 | | 2,200 | | |
| P53 | | 1,150 | | |
| P54 | | 1,150 | | |
| P55 | | 1,150 | | |
| P56 | | 1,150 | | |
| P57 | | 1,150 | | |
| P58 | | 1,150 | | |
| P59 | | 1,150 | | |
| P60 | | 1,150 | | |
| P61 | | 1,320 | | |
| P62 | | 1,320 | | |

| Product | J1 | J2 | J3 | J4 |
|---|---|---|---|---|
| P63 | | 1,320 | | |
| P64 | | 1,320 | | |
| P65 | | 1,320 | | |
| P66 | | 1,320 | | |
| P67 | | 1,320 | | |
| P68 | | 1,320 | | |
| P69 | | 1,320 | | |
| P70 | | 1,320 | | |
| P71 | | 1,320 | | |
| P72 | | 2,100 | | |
| P73 | | 2,100 | | |
| P74 | | 2,100 | | |
| P75 | | 2,100 | | |
| P76 | | 2,100 | | |
| P77 | 2,250 | | | |
| P78 | 2,100 | | | |
| P79 | 2,100 | | | |
| P80 | 2,790 | | | |
| P81 | 2,790 | | | |
| P82 | 2,790 | | | |
| P83 | 2,350 | | | |
| P84 | 2,350 | | | |
| P85 | 2,250 | | | |
| P86 | 2,250 | | | |
| P87 | 2,150 | | | |
| P88 | 2,150 | | | |
| P89 | 2,150 | | | |
| P90 | | | | 690 |
| P91 | | | | 690 |
| P92 | | | | 690 |
| P93 | | | | 690 |

**Table C.3:** Changeover times among families (h).

| | | Family | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Unit | Family | F01 | F02 | F03 | F04 | F05 | F06 | F07 | F08 | F09 | F10 | F11 | F12 | F13 | F14 | F15 | F16 | F17 | F18 | F19 | F20 | F21 | F22 | F23 |
| J3 | F01 | - | 0.00 | 0.00 | 0.50 | 0.50 | 0.50 | 0.50 | 1.00 | 1.50 | 1.50 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| J3 | F02 | FS | - | 0.00 | 0.50 | 0.50 | 0.50 | 0.50 | 1.00 | 1.50 | 1.50 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| J3 | F03 | FS | FS | - | 0.50 | 0.50 | 0.50 | 0.50 | 1.00 | 1.50 | 1.50 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| J3 | F04 | FS | FS | FS | - | FS | 1.00 | 1.50 | 0.50 | 0.75 | 0.75 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| J3 | F05 | FS | FS | FS | 0.25 | - | 0.50 | 0.75 | 1.00 | 1.50 | 1.50 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| J3 | F06 | FS | FS | FS | FS | FS | - | 0.50 | FS | FS | FS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| J4 | F06 | - | - | - | - | - | - | - | FS | FS | FS | - | - | - | - | - | - | - | - | - | - | - | - | 2.00 |
| J3 | F07 | FS | FS | FS | FS | FS | FS | - | FS | FS | FS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| J3 | F08 | FS | FS | FS | FS | FS | 2.00 | 2.00 | - | 0.50 | 0.50 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| J4 | F08 | - | - | - | - | - | 2.00 | - | - | 0.50 | 0.50 | - | - | - | - | - | - | - | - | - | - | - | - | 2.00 |
| J3 | F09 | FS | FS | FS | FS | FS | 2.00 | 2.00 | FS | - | 0.50 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| J4 | F09 | - | - | - | - | - | 2.00 | - | FS | - | 0.50 | - | - | - | - | - | - | - | - | - | - | - | - | 2.00 |
| J3 | F10 | FS | FS | FS | FS | FS | 2.00 | 2.00 | FS | FS | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| J4 | F10 | - | - | - | - | - | 2.00 | - | FS | FS | - | - | - | - | - | - | - | - | - | - | - | - | - | 2.00 |
| J2 | F11 | - | - | - | - | - | - | - | - | - | - | - | FS | 1.00 | 1.00 | 1.50 | 2.00 | 2.00 | 1.00 | 1.00 | - | - | - | - |
| J2 | F12 | - | - | - | - | - | - | - | - | - | - | 0.50 | - | 0.50 | 0.50 | 1.00 | 2.00 | 2.00 | 0.50 | 0.50 | - | - | - | - |
| J2 | F13 | - | - | - | - | - | - | - | - | - | - | FS | FS | - | 0.50 | 1.00 | 2.00 | 2.00 | FS | FS | - | - | - | - |
| J2 | F14 | - | - | - | - | - | - | - | - | - | - | FS | FS | FS | - | 0.50 | 2.00 | 2.00 | FS | FS | - | - | - | - |
| J2 | F15 | - | - | - | - | - | - | - | - | - | - | FS | FS | FS | FS | - | 1.50 | 2.00 | FS | FS | - | - | - | - |
| J2 | F16 | - | - | - | - | - | - | - | - | - | - | FS | FS | FS | FS | FS | - | 0.50 | FS | FS | - | - | - | - |
| J2 | F17 | - | - | - | - | - | - | - | - | - | - | FS | FS | FS | FS | FS | FS | - | FS | FS | - | - | - | - |
| J2 | F18 | - | - | - | - | - | - | - | - | - | - | FS | FS | 0.50 | 0.50 | 1.00 | 2.00 | 2.00 | - | FS | - | - | - | - |
| J2 | F19 | - | - | - | - | - | - | - | - | - | - | FS | FS | 0.50 | 0.50 | 1.00 | 2.00 | 2.00 | 0.50 | - | - | - | - | - |
| J1 | F20 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 0.50 | 1.00 | - |
| J1 | F21 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | FS | - | 0.50 | - |
| J1 | F22 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | FS | FS | - | - |
| J4 | F23 | - | - | - | - | - | FS | - | FS | FS | FS | - | - | - | - | - | - | - | - | - | - | - | - | - |

-: Impossible processing sequence. FS: Forbidden processing sequence.

**Table C.4:** Correlation between changeover times and changeover costs.

| changeover time (h) | changeover cost (€) |
|---|---|
| 0.00 | 750 |
| 0.25 | 1,125 |
| 0.50 | 1,800 |
| 0.75 | 2,250 |
| 1.00 | 6,000 |
| 1.50 | 15,000 |
| 2.00 | 22,500 |

**Table C.5:** Production targets $\zeta_{pn}^{cup}$ (cups).

| Product | n1 | n2 | n3 | n4 | n5 |
|---|---|---|---|---|---|
| P01 | 3,915 | | | | |
| P02 | | 2,190 | | | |
| P03 | | | | 4,416 | |
| P04 | | 6,130 | | | 14,001 |
| P05 | | | | | 5,480 |
| P06 | | | | 5,888 | |
| P07 | | | | | 11,241 |
| P08 | | | | | 4,000 |
| P09 | | | | 1,888 | |
| P10 | | | | | 1,229 |
| P11 | | | | | 715 |
| P12 | 3,560 | | | | |
| P13 | | 4,215 | | | |
| P14 | | | | 4,416 | |
| P15 | | | | | 6,341 |
| P16 | | | | | 3,715 |
| P17 | 2,319 | | | | |
| P18 | | | | 2,592 | |
| P19 | | | 6,138 | | |
| P20 | | | | | 6,480 |
| P21 | | | | | 1,620 |
| P22 | | | | | 1,380 |
| P23 | | | | 17,318 | |
| P24 | | | | 4,193 | |
| P25 | | | | 14,974 | |
| P26 | | 4,671 | | | |
| P27 | | 1,325 | | | |
| P28 | | 3,312 | | | |
| P29 | | 3,312 | | | |
| P30 | | | | | 3,682 |
| P31 | | | | | 4,801 |

| Product | n1 | n2 | n3 | n4 | n5 |
|---|---|---|---|---|---|
| P32 | | | | | 8,249 |
| P33 | | | | | 4,057 |
| P34 | | 2,472 | | | |
| P35 | | 2,472 | | | |
| P36 | 26,496 | | | | |
| P37 | 9,531 | | | | |
| P38 | 4,717 | | | | |
| P39 | | | | | 4,093 |
| P40 | | | | | 5,743 |
| P41 | | | | | 1,172 |
| P42 | | | | 3,300 | |
| P43 | | | | | 1,807 |
| P44 | | | | | 2,019 |
| P45 | | | | | 1,578 |
| P46 | | | | | 2,518 |
| P47 | | | | | 1,690 |
| P48 | | | | | 2,132 |
| P49 | | | | | 5,495 |
| P50 | | | | | 1,830 |
| P51 | | | | | 9,380 |
| P52 | | | | | 1,272 |
| P53 | | | | | 4,386 |
| P54 | | | | | 1,315 |
| P55 | | | | 4,782 | |
| P56 | | | | 4,316 | |
| P57 | | | | 3,162 | |
| P58 | | | | 3,188 | |
| P59 | | | | 3,188 | |
| P60 | | | | 2,316 | |
| P61 | | | | | 1,408 |
| P62 | | | | | 1,262 |

| Product | n1 | n2 | n3 | n4 | n5 |
|---|---|---|---|---|---|
| P63 | | 5,753 | | | |
| P64 | | 1,919 | | | |
| P65 | | 3,648 | | | |
| P66 | | | | | 1,962 |
| P67 | | 2,648 | | | |
| P68 | | | | | 4,683 |
| P69 | | 2,296 | | | |
| P70 | | | | | 544 |
| P71 | | | | | 219 |
| P72 | 2,199 | | | 2,040 | |
| P73 | 1,283 | | | | |
| P74 | | | | | 40 |
| P75 | | | | | 1,071 |
| P76 | | | | | 117 |
| P77 | | | | | 57 |
| P78 | | | | | 1,348 |
| P79 | | | | | 195 |
| P80 | | | 960 | | |
| P81 | | | 1,160 | | |
| P82 | | | 900 | | |
| P83 | | | | | 710 |
| P84 | | | | | 290 |
| P85 | | | | | 200 |
| P86 | | | | | 518 |
| P87 | | | | | 1,130 |
| P88 | | | | | 1,442 |
| P89 | | | | 3,150 | |
| P90 | | 3,140 | | | 10,140 |
| P91 | | 3,120 | | | 5,410 |
| P92 | | 9,890 | | | 2,900 |
| P93 | | 8,220 | | | 1,200 |

# Appendix D

## Detailed Production Plans for the Resource-Constrained Yoghurt Production Process Case Studies

This appendix contains the detailed production plans for the yogurt production process case studies solved. More specifically, the detailed production plan for each product for Case Study I is reported in Table D.1. Table D.2 presents the detailed production plan for every product for Case Study II, and Table D.3 contains the detailed production plan for Case Study III.

**Table D.1:** Case Study I. Detailed production plan (kg).

| Product | Unit | Monday | Tuesday | Wednesday | Thursday | Friday |
|---|---|---|---|---|---|---|
| P01 | J3 | 2,349 | 0 | 0 | 0 | 0 |
| P02 | J3 | 0 | 1,314 | 0 | 0 | 0 |
| P03 | J3 | 0 | 0 | 0 | 2,650 | 0 |
| P04 | J3 | 0 | 3,678 | 0 | 0 | 8,401 |
| P05 | J3 | 0 | 0 | 0 | 0 | 3,288 |
| P06 | J3 | 0 | 0 | 0 | 3,533 | 0 |
| P07 | J3 | 0 | 0 | 0 | 2,995 | 3,749 |
| P08 | J3 | 0 | 0 | 0 | 0 | 2,400 |
| P09 | J3 | 0 | 0 | 0 | 1,133 | 0 |
| P10 | J3 | 0 | 0 | 0 | 737 | 0 |
| P11 | J3 | 0 | 0 | 0 | 429 | 0 |
| P12 | J3 | 2,136 | 0 | 0 | 0 | 0 |
| P13 | J3 | 0 | 2,529 | 0 | 0 | 0 |
| P14 | J3 | 0 | 0 | 0 | 2,650 | 0 |
| P15 | J3 | 0 | 0 | 0 | 0 | 3,805 |
| P16 | J3 | 0 | 0 | 0 | 0 | 2,229 |
| P17 | J3 | 1,391 | 0 | 0 | 0 | 0 |
| P18 | J3 | 0 | 0 | 0 | 1,555 | 0 |
| P19 | J3 | 0 | 0 | 3,683 | 0 | 0 |
| P20 | J3 | 0 | 0 | 0 | 0 | 3,888 |
| P21 | J3 | 0 | 0 | 0 | 0 | 324 |
| P22 | J3 | 0 | 0 | 0 | 0 | 276 |
| P23 | J3 | 660 | 0 | 0 | 0 | 0 |
| P23 | J4 | 0 | 0 | 0 | 6,267 | 0 |
| P24 | J3 | 0 | 0 | 0 | 0 | 0 |
| P24 | J4 | 0 | 0 | 0 | 839 | 0 |
| P25 | J3 | 0 | 0 | 0 | 0 | 0 |
| P25 | J4 | 0 | 0 | 0 | 2,995 | 0 |
| P26 | J3 | 1,775 | 0 | 0 | 0 | 0 |
| P27 | J3 | 504 | 0 | 0 | 0 | 0 |
| P28 | J3 | 0 | 1,325 | 0 | 0 | 0 |
| P28 | J4 | 0 | 0 | 0 | 0 | 0 |
| P29 | J3 | 0 | 1,325 | 0 | 0 | 0 |
| P29 | J4 | 0 | 0 | 0 | 0 | 0 |
| P30 | J3 | 0 | 0 | 0 | 0 | 0 |
| P30 | J4 | 0 | 0 | 0 | 0 | 2,209 |
| P31 | J3 | 0 | 0 | 0 | 0 | 0 |
| P31 | J4 | 0 | 0 | 0 | 0 | 2,881 |
| P32 | J3 | 0 | 0 | 0 | 0 | 0 |
| P32 | J4 | 0 | 0 | 0 | 0 | 4,949 |
| P33 | J3 | 0 | 0 | 0 | 0 | 0 |
| P33 | J4 | 0 | 0 | 0 | 0 | 2,434 |
| P34 | J3 | 0 | 0 | 0 | 0 | 0 |
| P34 | J4 | 1,112 | 0 | 0 | 0 | 0 |
| P35 | J3 | 0 | 0 | 0 | 0 | 0 |
| P35 | J4 | 1,112 | 0 | 0 | 0 | 0 |
| P36 | J3 | 0 | 0 | 0 | 0 | 0 |
| P36 | J4 | 3,974 | 0 | 0 | 0 | 0 |
| P37 | J3 | 0 | 0 | 0 | 0 | 0 |
| P37 | J4 | 1,430 | 0 | 0 | 0 | 0 |
| P38 | J3 | 0 | 0 | 0 | 0 | 0 |
| P38 | J4 | 708 | 0 | 0 | 0 | 0 |
| P39 | J3 | 0 | 0 | 0 | 0 | 0 |
| P39 | J4 | 0 | 0 | 0 | 0 | 512 |
| P40 | J3 | 0 | 0 | 0 | 0 | 0 |
| P40 | J4 | 0 | 0 | 0 | 0 | 718 |
| P41 | J3 | 0 | 0 | 0 | 0 | 0 |
| P41 | J4 | 0 | 0 | 0 | 0 | 147 |
| P42 | J2 | 0 | 0 | 0 | 3,300 | 0 |
| P43 | J2 | 0 | 0 | 0 | 0 | 1,807 |

**Table D.1:** Case Study I. Detailed Production plan (kg).

| Product | Unit | Monday | Tuesday | Wednesday | Thursday | Friday |
|---|---|---|---|---|---|---|
| P44 | J2 | 0 | 0 | 0 | 1,121 | 898 |
| P45 | J2 | 0 | 0 | 0 | 0 | 1,578 |
| P46 | J2 | 0 | 0 | 0 | 0 | 2,518 |
| P47 | J2 | 0 | 0 | 0 | 0 | 1,690 |
| P48 | J2 | 0 | 0 | 0 | 0 | 2,132 |
| P49 | J2 | 0 | 0 | 0 | 0 | 5,495 |
| P50 | J2 | 0 | 0 | 0 | 0 | 1,830 |
| P51 | J2 | 0 | 0 | 0 | 1,100 | 8,280 |
| P52 | J2 | 0 | 0 | 0 | 0 | 1,272 |
| P53 | J2 | 0 | 0 | 0 | 2,193 | 0 |
| P54 | J2 | 0 | 0 | 0 | 658 | 0 |
| P55 | J2 | 0 | 0 | 2,351 | 40 | 0 |
| P56 | J2 | 0 | 0 | 0 | 2,158 | 0 |
| P57 | J2 | 0 | 0 | 0 | 1,581 | 0 |
| P58 | J2 | 0 | 0 | 0 | 1,594 | 0 |
| P59 | J2 | 0 | 0 | 1,578 | 17 | 0 |
| P60 | J2 | 0 | 0 | 1,158 | 0 | 0 |
| P61 | J2 | 0 | 0 | 1,056 | 0 | 0 |
| P62 | J2 | 0 | 0 | 947 | 0 | 0 |
| P63 | J2 | 0 | 4,315 | 0 | 0 | 0 |
| P64 | J2 | 0 | 1,439 | 0 | 0 | 0 |
| P65 | J2 | 0 | 2,736 | 0 | 0 | 0 |
| P66 | J2 | 0 | 27 | 1,445 | 0 | 0 |
| P67 | J2 | 0 | 1,986 | 0 | 0 | 0 |
| P68 | J2 | 0 | 27 | 3,485 | 0 | 0 |
| P69 | J2 | 0 | 1,722 | 0 | 0 | 0 |
| P70 | J2 | 0 | 336 | 72 | 0 | 0 |
| P71 | J2 | 0 | 137 | 27 | 0 | 0 |
| P72 | J2 | 4,239 | 0 | 0 | 0 | 0 |
| P73 | J2 | 1,283 | 0 | 0 | 0 | 0 |
| P74 | J2 | 0 | 0 | 0 | 40 | 0 |
| P75 | J2 | 0 | 0 | 0 | 1,071 | 0 |
| P76 | J2 | 0 | 0 | 0 | 117 | 0 |
| P77 | J1 | 0 | 0 | 0 | 0 | 285 |
| P78 | J1 | 0 | 0 | 0 | 0 | 6,740 |
| P79 | J1 | 0 | 0 | 0 | 0 | 975 |
| P80 | J1 | 25,035 | 3,765 | 0 | 0 | 0 |
| P81 | J1 | 0 | 29,018 | 5,783 | 0 | 0 |
| P82 | J1 | 0 | 0 | 27,000 | 0 | 0 |
| P83 | J1 | 0 | 0 | 0 | 0 | 7,100 |
| P84 | J1 | 0 | 0 | 0 | 0 | 2,900 |
| P85 | J1 | 0 | 0 | 0 | 1,000 | 0 |
| P86 | J1 | 0 | 0 | 0 | 0 | 2,590 |
| P87 | J1 | 0 | 0 | 0 | 0 | 5,650 |
| P88 | J1 | 0 | 0 | 0 | 5,242 | 1,968 |
| P89 | J1 | 0 | 0 | 0 | 15,750 | 0 |
| P90 | J4 | 0 | 471 | 1,521 | 0 | 0 |
| P91 | J4 | 0 | 468 | 812 | 0 | 0 |
| P92 | J4 | 0 | 1,484 | 435 | 0 | 0 |
| P93 | J4 | 0 | 1,233 | 180 | 0 | 0 |
| Total daily production | | 47,708 | 59,333 | 51,531 | 62,764 | 97,916 |

**Table D.2:** Case Study II. Detailed production plan (kg).

| Product | Unit | Wednesday | Thursday | Friday |
|---|---|---|---|---|
| P01 | J3 | 0 | 0 | 0 |
| P02 | J3 | 0 | 0 | 0 |
| P03 | J3 | 0 | 2,650 | 0 |
| P04 | J3 | 0 | 0 | 8,401 |
| P05 | J3 | 0 | 0 | 3,288 |
| P06 | J3 | 0 | 3,533 | 0 |
| P07 | J3 | 0 | 0 | 6,745 |
| P08 | J3 | 0 | 0 | 2,400 |
| P09 | J3 | 0 | 1,133 | 0 |
| P10 | J3 | 0 | 737 | 0 |
| P11 | J3 | 0 | 429 | 0 |
| P12 | J3 | 0 | 0 | 0 |
| P13 | J3 | 0 | 0 | 0 |
| P14 | J3 | 0 | 2,650 | 0 |
| P15 | J3 | 0 | 2,676 | 1,129 |
| P16 | J3 | 0 | 0 | 2,229 |
| P17 | J3 | 0 | 0 | 0 |
| P18 | J3 | 0 | 1,555 | 0 |
| P19 | J3 | 3,683 | 0 | 0 |
| P20 | J3 | 0 | 0 | 3,888 |
| P21 | J3 | 0 | 0 | 324 |
| P22 | J3 | 0 | 0 | 276 |
| P23 | J3 | 0 | 0 | 0 |
| P23 | J4 | 6,267 | 0 | 0 |
| P24 | J3 | 0 | 0 | 0 |
| P24 | J4 | 839 | 0 | 0 |
| P25 | J3 | 0 | 0 | 0 |
| P25 | J4 | 2,995 | 0 | 0 |

| Product | Unit | Wednesday | Thursday | Friday |
|---|---|---|---|---|
| P26 | J3 | 0 | 0 | 0 |
| P27 | J3 | 0 | 0 | 0 |
| P28 | J3 | 0 | 0 | 0 |
| P28 | J4 | 0 | 0 | 0 |
| P29 | J3 | 0 | 0 | 0 |
| P29 | J4 | 0 | 0 | 0 |
| P30 | J3 | 0 | 83 | 0 |
| P30 | J4 | 0 | 0 | 2,126 |
| P31 | J3 | 0 | 0 | 0 |
| P31 | J4 | 0 | 0 | 2,881 |
| P32 | J3 | 0 | 4,949 | 0 |
| P32 | J4 | 0 | 0 | 0 |
| P33 | J3 | 0 | 555 | 0 |
| P33 | J4 | 0 | 0 | 1,879 |
| P34 | J3 | 0 | 0 | 0 |
| P34 | J4 | 0 | 0 | 0 |
| P35 | J3 | 0 | 0 | 0 |
| P35 | J4 | 0 | 0 | 0 |
| P36 | J3 | 0 | 0 | 0 |
| P36 | J4 | 0 | 0 | 0 |
| P37 | J3 | 0 | 0 | 0 |
| P37 | J4 | 0 | 0 | 0 |
| P38 | J3 | 0 | 0 | 0 |
| P38 | J4 | 0 | 0 | 0 |
| P39 | J3 | 0 | 512 | 0 |
| P39 | J4 | 0 | 0 | 0 |
| P40 | J3 | 0 | 718 | 0 |
| P40 | J4 | 0 | 0 | 0 |

**Table D.2:** Case Study II. Detailed production plan (kg).

| Product | Unit | Wednesday | Thursday | Friday |
|---|---|---|---|---|
| P41 | J3 | 0 | 147 | 0 |
| P41 | J4 | 0 | 0 | 0 |
| P42 | J2 | 0 | 3,300 | 0 |
| P43 | J2 | 0 | 33 | 1,774 |
| P44 | J2 | 0 | 1,088 | 931 |
| P45 | J2 | 0 | 0 | 1,578 |
| P46 | J2 | 0 | 0 | 2,518 |
| P47 | J2 | 0 | 0 | 1,690 |
| P48 | J2 | 0 | 0 | 2,132 |
| P49 | J2 | 0 | 0 | 5,495 |
| P50 | J2 | 0 | 0 | 1,830 |
| P51 | J2 | 0 | 1,100 | 8,280 |
| P52 | J2 | 0 | 0 | 1,272 |
| P53 | J2 | 0 | 2,193 | 0 |
| P54 | J2 | 0 | 658 | 0 |
| P55 | J2 | 2,391 | 0 | 0 |
| P56 | J2 | 17 | 2,142 | 0 |
| P57 | J2 | 17 | 1,565 | 0 |
| P58 | J2 | 17 | 1,578 | 0 |
| P59 | J2 | 1,504 | 90 | 0 |
| P60 | J2 | 1,142 | 17 | 0 |
| P61 | J2 | 1,056 | 0 | 0 |
| P62 | J2 | 947 | 0 | 0 |
| P63 | J2 | 0 | 0 | 0 |
| P64 | J2 | 0 | 0 | 0 |
| P65 | J2 | 0 | 0 | 0 |
| P66 | J2 | 1,445 | 0 | 0 |
| P67 | J2 | 0 | 0 | 0 |
| P68 | J2 | 3,485 | 0 | 0 |
| P69 | J2 | 0 | 0 | 0 |
| P70 | J2 | 72 | 0 | 0 |
| P71 | J2 | 27 | 0 | 0 |
| P72 | J2 | 0 | 0 | 0 |
| P73 | J2 | 0 | 0 | 0 |
| P74 | J2 | 0 | 40 | 0 |
| P75 | J2 | 0 | 1,071 | 0 |
| P76 | J2 | 0 | 117 | 0 |
| P77 | J1 | 0 | 0 | 285 |
| P78 | J1 | 0 | 0 | 6,740 |
| P79 | J1 | 0 | 0 | 975 |
| P80 | J1 | 0 | 0 | 0 |
| P81 | J1 | 5,783 | 0 | 0 |
| P82 | J1 | 27,000 | 0 | 0 |
| P83 | J1 | 0 | 0 | 7,100 |
| P84 | J1 | 0 | 0 | 2,900 |
| P85 | J1 | 0 | 1,000 | 0 |
| P86 | J1 | 0 | 1,410 | 1,180 |
| P87 | J1 | 0 | 0 | 5,650 |
| P88 | J1 | 0 | 7,210 | 0 |
| P89 | J1 | 0 | 15,750 | 0 |
| P90 | J4 | 0 | 1,521 | 0 |
| P91 | J4 | 0 | 812 | 0 |
| P92 | J4 | 0 | 435 | 0 |
| P93 | J4 | 0 | 180 | 0 |
| Total daily production | | 58,684 | 65,632 | 87,895 |

**Table D.3:** Case Study III. Detailed production plan (kg).

| Product | Unit | Thursday | Friday | Product | Unit | Thursday | Friday | Product | Unit | Thursday | Friday | Product | Unit | Thursday | Friday |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P01 | J3 | 841 | 0 | P26 | J3 | 0 | 0 | P41 | J3 | 0 | 0 | P68 | J2 | 0 | 0 |
| P02 | J3 | 0 | 0 | P27 | J3 | 0 | 0 | P41 | J4 | 0 | 285 | P69 | J2 | 0 | 0 |
| P03 | J3 | 2,650 | 0 | P28 | J3 | 0 | 0 | P42 | J2 | 3,300 | 0 | P70 | J2 | 0 | 0 |
| P04 | J3 | 0 | 6,884 | P28 | J4 | 0 | 858 | P43 | J2 | 1,807 | 0 | P71 | J2 | 0 | 0 |
| P05 | J3 | 0 | 4,791 | P29 | J3 | 0 | 0 | P44 | J2 | 1,626 | 0 | P72 | J2 | 0 | 0 |
| P06 | J3 | 3,533 | 0 | P29 | J4 | 0 | 0 | P45 | J2 | 0 | 1,578 | P73 | J2 | 0 | 0 |
| P07 | J3 | 974 | 5,771 | P30 | J3 | 0 | 0 | P46 | J2 | 0 | 2,518 | P74 | J2 | 0 | 40 |
| P08 | J3 | 0 | 0 | P30 | J4 | 0 | 2,209 | P47 | J2 | 0 | 1,690 | P75 | J2 | 0 | 1,071 |
| P09 | J3 | 1,133 | 0 | P31 | J3 | 0 | 0 | P48 | J2 | 0 | 2,132 | P76 | J2 | 0 | 117 |
| P10 | J3 | 737 | 0 | P31 | J4 | 0 | 2,881 | P49 | J2 | 0 | 5,495 | P77 | J1 | 285 | 0 |
| P11 | J3 | 568 | 0 | P32 | J3 | 0 | 0 | P50 | J2 | 0 | 1,830 | P78 | J1 | 3,515 | 3,225 |
| P12 | J3 | 0 | 0 | P32 | J4 | 0 | 4,949 | P51 | J2 | 33 | 9,347 | P79 | J1 | 0 | 975 |
| P13 | J3 | 977 | 0 | P33 | J3 | 0 | 0 | P52 | J2 | 1,067 | 205 | P80 | J1 | 1,395 | 2,205 |
| P14 | J3 | 2,650 | 0 | P33 | J4 | 0 | 1,801 | P53 | J2 | 2,193 | 0 | P81 | J1 | 0 | 7,200 |
| P15 | J3 | 0 | 3,805 | P34 | J3 | 0 | 0 | P54 | J2 | 54 | 604 | P82 | J1 | 0 | 0 |
| P16 | J3 | 0 | 2,469 | P34 | J4 | 0 | 0 | P55 | J2 | 40 | 0 | P83 | J1 | 0 | 7,100 |
| P17 | J3 | 1,157 | 0 | P35 | J3 | 0 | 0 | P56 | J2 | 2,158 | 0 | P84 | J1 | 0 | 2,900 |
| P18 | J3 | 1,555 | 0 | P35 | J4 | 0 | 0 | P57 | J2 | 1,581 | 0 | P85 | J1 | 1,000 | 0 |
| P19 | J3 | 0 | 0 | P36 | J3 | 0 | 0 | P58 | J2 | 993 | 0 | P86 | J1 | 0 | 2,590 |
| P20 | J3 | 0 | 3,888 | P36 | J4 | 0 | 0 | P59 | J2 | 17 | 0 | P87 | J1 | 0 | 2,363 |
| P21 | J3 | 0 | 444 | P37 | J3 | 0 | 0 | P60 | J2 | 0 | 0 | P88 | J1 | 5,332 | 1,878 |
| P22 | J3 | 0 | 358 | P37 | J4 | 0 | 0 | P61 | J2 | 0 | 0 | P89 | J1 | 15,750 | 0 |
| P23 | J3 | 0 | 0 | P38 | J3 | 0 | 0 | P62 | J2 | 0 | 0 | P90 | J4 | 0 | 0 |
| P23 | J4 | 6,267 | 0 | P38 | J4 | 0 | 0 | P63 | J2 | 0 | 0 | P91 | J4 | 0 | 0 |
| P24 | J3 | 0 | 0 | P39 | J3 | 0 | 0 | P64 | J2 | 1,392 | 0 | P92 | J4 | 0 | 0 |
| P24 | J4 | 0 | 0 | P39 | J4 | 0 | 512 | P65 | J2 | 0 | 0 | P93 | J4 | 0 | 0 |
| P25 | J3 | 0 | 0 | P40 | J3 | 0 | 0 | P66 | J2 | 0 | 0 | | | | |
| P25 | J4 | 2,758 | 0 | P40 | J4 | 0 | 718 | P67 | J2 | 0 | 0 | Total daily production | | 69,336 | 99,685 |

Appendix E

# MIP-based Solution Strategy Pseudocodes

This appendix presents some representative pseudocodes for the two-stage MIP-based solution strategy described in Chapter 8.

**Algorithm E.1**: Pseudocode for iterative procedure in Constructive Step

Set step = 1, initial = 1 & pos(i) parameter. Also, set $I^{in} = \emptyset$

FOR $z$ = initial to card(i) by step
    LOOP i
        IF pos(i) $\leq$ z
            $I^{in}$=yes
        END IF
    END LOOP
    SOLVE MIP model
    fix $Y_{isj}$ & $X_{ii'j}$ binary variables $\forall i \in I^{in}$
END FOR

**Algorithm E.2**: : Pseudocode for iterative procedure in Improvement Step

Refer to Méndez and Cerdá (2003a) for an explanation of parameter $n$,
subsets $IS_i$, $ISS_{ii'}$, and Reordering-MIP model.

Set $iter^{max}$, $n = 1$, order(i), reins=1 parameters & $IS_i$ subset
$fixY_{isj} = Y_{isj}$, $fixX_{ii'j} = X_{ii'j}$ (solution of constructive step)
iteration = 1

WHILE ($OF^{reins}$ is better than $OF^{reord}$ or iteration = 1)
  → *Reordering Stage*
  $Y_{isj} = fixY_{isj}$
  $iter = 1$
  WHILE ($OF_{iter}$ better than $OF_{iter-1}$ and $iter \leq iter^{max}$)
    CLEAR $ISS_{ii'}$ subset
    assess $ISS_{ii'}$ subset
    CLEAR all variables apart from $Y_{isj}$
    SOLVE Reordering-MIP model
    $iter = iter + 1$
  END WHILE
  save best solution of reordering stage:
  CLEAR $fixX_{ii'j}$, and set $fixX_{ii'j} = X_{ii'j}$ & $OF^{reord} = OF_{iter-1}$

  → *Reinsertion Stage*
  iter = 1
  FOR $z$ = reins to card(i) by reins
    LOOP i
      IF order(i) $\leq z$ - reins + 1
        CLEAR all variables related to i (e.g., $Y_{isj}$, $X_{ii'j}$, $C_{is}$, etc.)
      ELSE
        $Y_{isj} = fixY_{isj}$ & $X_{ii'j} = fixX_{ii'j}$
      END IF
    END LOOP
    SOLVE MIP model
    IF $OF_{iter}$ is better than $OF^{reord}$
      Save Solution(iter) (e.g., save $OF$, $Y_{isj}$, $X_{ii'j}$, $C_{is}$, etc.)
    END IF
    $iter = iter + 1$
  END FOR
  save best solution of reinsertion stage:
  CLEAR $fixY_{isj}$, $fixX_{ii'j}$ & $OF^{reins}$
  $OF^{reins}$ is equal to the best $OF_{iter}$
  set $fixY_{isj} = Y_{isj}$, $fixX_{ii'j} = X_{ii'j}$ only for $OF_{iter} = OF^{reins}$
  iteration = iteration + 1
END WHILE

# Appendix F

## Data for the Pharmaceutical Multistage Batch Process

This appendix contains the data for the changeover times for stages S3 – S6 and the processing times for the pharmaceutical process described and studied in Chapter 8.

**Table F.1:** Changeover times in stage S3 (h).

| | P01 | P08 | P12 | P13 | P16 | P17 | P20 | P21 | P23 | P24 | P26 | P27 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P01 | 0.00 | 1.80 | 1.80 | 1.35 | 1.80 | 1.35 | 1.35 | 1.80 | 1.80 | 1.35 | 1.80 | 1.80 |
| P08 | 1.80 | 0.00 | 1.35 | 1.80 | 1.35 | 1.80 | 1.80 | 1.80 | 1.35 | 1.80 | 1.35 | 1.80 |
| P12 | 1.80 | 1.35 | 0.00 | 1.80 | 1.35 | 1.80 | 1.80 | 1.80 | 1.35 | 1.80 | 1.35 | 1.80 |
| P13 | 1.35 | 1.80 | 1.80 | 0.00 | 1.80 | 1.35 | 1.35 | 1.80 | 1.80 | 1.35 | 1.80 | 1.80 |
| P16 | 1.80 | 1.35 | 1.35 | 1.80 | 0.00 | 1.80 | 1.80 | 1.80 | 1.35 | 1.80 | 1.35 | 1.80 |
| P17 | 1.35 | 1.80 | 1.80 | 1.35 | 1.80 | 0.00 | 1.35 | 1.80 | 1.80 | 1.35 | 1.80 | 1.80 |
| P20 | 1.35 | 1.80 | 1.80 | 1.35 | 1.80 | 1.35 | 0.00 | 1.80 | 1.80 | 1.35 | 1.80 | 1.80 |
| P21 | 1.80 | 1.80 | 1.80 | 1.80 | 1.80 | 1.80 | 1.80 | 0.00 | 1.80 | 1.80 | 1.80 | 1.35 |
| P23 | 1.80 | 1.35 | 1.35 | 1.80 | 1.35 | 1.80 | 1.80 | 1.80 | 0.00 | 1.80 | 1.35 | 1.80 |
| P24 | 1.35 | 1.80 | 1.80 | 1.35 | 1.80 | 1.35 | 1.35 | 1.80 | 1.80 | 0.00 | 1.80 | 1.80 |
| P26 | 1.80 | 1.35 | 1.35 | 1.80 | 1.35 | 1.80 | 1.80 | 1.80 | 1.35 | 1.80 | 0.00 | 1.80 |
| P27 | 1.80 | 1.80 | 1.80 | 1.80 | 1.80 | 1.80 | 1.80 | 1.35 | 1.80 | 1.80 | 1.80 | 0.00 |

**Table F.2:** Changeover times in stages S4 and S5 (h).

| | P01 | P02 | P03 | P04 | P05 | P06 | P07 | P08 | P09 | P10 | P11 | P12 | P13 | P14 | P15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P01 | 0 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 |
| P02 | 0.9 | 0 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 |
| P03 | 1.8 | 1.8 | 0 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 |
| P04 | 1.8 | 1.8 | 1.8 | 0 | 1.8 | 0.9 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 |
| P05 | 1.8 | 1.8 | 0.9 | 1.8 | 0 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 |
| P06 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 0 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 |
| P07 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 |
| P08 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 0 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 |
| P09 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 0.9 | 0 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 |
| P10 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 |
| P11 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0 | 1.8 | 1.8 | 1.8 | 0.9 |
| P12 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 0 | 1.8 | 1.8 | 1.8 |
| P13 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0 | 0.9 | 1.8 |
| P14 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0 | 1.8 |
| P15 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0 |
| P16 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 |
| P17 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 |
| P18 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 |
| P19 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 |
| P20 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 |
| P21 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 |
| P22 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 |
| P23 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 |
| P24 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 |
| P25 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 |
| P26 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 |
| P27 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 |
| P28 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 |
| P29 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 |
| P30 | 1.8 | 1.8 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 |

**Table F.2:** Changeover times stages S4 and S5 (h).

| | P16 | P17 | P18 | P19 | P20 | P21 | P22 | P23 | P24 | P25 | P26 | P27 | P28 | P29 | P30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P01 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 |
| P02 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 |
| P03 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 |
| P04 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 |
| P05 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 |
| P06 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 |
| P07 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 |
| P08 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 |
| P09 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 |
| P10 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 |
| P11 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 |
| P12 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 |
| P13 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 |
| P14 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 |
| P15 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 |
| P16 | 0 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 |
| P17 | 1.8 | 0 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 |
| P18 | 1.8 | 0.9 | 0 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 |
| P19 | 0.9 | 1.8 | 1.8 | 0 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 |
| P20 | 1.8 | 0.9 | 0.9 | 1.8 | 0 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 |
| P21 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 |
| P22 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 |
| P23 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 |
| P24 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 |
| P25 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 0 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 |
| P26 | 0.9 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 1.8 | 1.8 | 0 | 1.8 | 1.8 | 1.8 | 1.8 |
| P27 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0 | 0.9 | 1.8 | 0.9 |
| P28 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0 | 1.8 | 0.9 |
| P29 | 1.8 | 0.9 | 0.9 | 1.8 | 0.9 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 0 | 1.8 |
| P30 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 1.8 | 1.8 | 1.8 | 0.9 | 0.9 | 1.8 | 0 |

**Table F.3:** Changeover times in stage S6 (h).

| | P01 | P02 | P03 | P04 | P05 | P06 | P07 | P08 | P09 | P10 | P11 | P12 | P13 | P14 | P15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P01 | 0.0000 | 1.8000 | 1.2933 | 1.5183 | 1.5183 | 1.8000 | 1.8000 | 1.1250 | 0.9558 | 1.4625 | 1.8000 | 0.9558 | 1.8000 | 1.4625 | 1.1808 |
| P02 | 1.8000 | 0.0000 | 1.4625 | 1.4625 | 1.8000 | 1.8000 | 1.2933 | 1.1808 | 1.8000 | 1.8000 | 1.8000 | 0.8433 | 1.8000 | 1.2933 | 1.8000 |
| P03 | 1.2933 | 1.4625 | 0.0000 | 0.7875 | 1.4625 | 1.4625 | 1.4625 | 1.8000 | 1.2933 | 1.1250 | 1.4625 | 1.2933 | 1.4625 | 1.4625 | 1.8000 |
| P04 | 1.5183 | 1.4625 | 0.7875 | 0.0000 | 1.1808 | 0.9558 | 1.4625 | 1.2933 | 1.8000 | 1.1250 | 0.9558 | 1.8000 | 0.9558 | 1.4625 | 1.5183 |
| P05 | 1.5183 | 1.8000 | 1.4625 | 1.1808 | 0.0000 | 1.1250 | 1.1250 | 1.8000 | 1.4625 | 0.9558 | 1.4625 | 1.8000 | 1.4625 | 1.8000 | 1.0125 |
| P06 | 1.8000 | 1.8000 | 1.4625 | 0.9558 | 1.1250 | 0.0000 | 1.1250 | 1.2933 | 1.4625 | 1.4625 | 0.6183 | 1.8000 | 0.6183 | 1.8000 | 1.8000 |
| P07 | 1.8000 | 1.2933 | 1.4625 | 1.4625 | 1.1250 | 1.1250 | 0.0000 | 1.8000 | 1.4625 | 1.4625 | 1.4625 | 1.8000 | 1.4625 | 1.2933 | 1.4625 |
| P08 | 1.1250 | 1.1808 | 1.8000 | 1.2933 | 1.8000 | 1.2933 | 1.8000 | 0.0000 | 1.4625 | 1.4625 | 1.2933 | 0.8433 | 1.2933 | 1.4625 | 1.4625 |
| P09 | 0.9558 | 1.8000 | 1.2933 | 1.8000 | 1.4625 | 1.4625 | 1.4625 | 1.4625 | 0.0000 | 1.8000 | 1.5183 | 1.2933 | 1.8000 | 1.1808 | 1.8000 |
| P10 | 1.4625 | 1.8000 | 1.1250 | 1.1250 | 0.9558 | 1.4625 | 1.4625 | 1.4625 | 1.8000 | 0.0000 | 1.4625 | 1.4625 | 1.4625 | 1.4625 | 0.9558 |
| P11 | 1.8000 | 1.8000 | 1.4625 | 0.9558 | 1.4625 | 0.6183 | 1.4625 | 1.2933 | 1.5183 | 1.4625 | 0.0000 | 1.8000 | 0.3753 | 1.1808 | 1.8000 |
| P12 | 0.9558 | 0.8433 | 1.2933 | 1.8000 | 1.8000 | 1.8000 | 1.8000 | 0.8433 | 1.2933 | 1.4625 | 1.8000 | 0.0000 | 1.8000 | 1.8000 | 1.4625 |
| P13 | 1.8000 | 1.8000 | 1.4625 | 0.9558 | 1.4625 | 0.6183 | 1.4625 | 1.2933 | 1.8000 | 1.4625 | 0.3753 | 1.8000 | 0.0000 | 1.4625 | 1.8000 |
| P14 | 1.4625 | 1.2933 | 1.4625 | 1.4625 | 1.8000 | 1.8000 | 1.2933 | 1.4625 | 1.1808 | 1.4625 | 1.1808 | 1.8000 | 1.4625 | 0.0000 | 1.8000 |
| P15 | 1.1808 | 1.8000 | 1.8000 | 1.5183 | 1.0125 | 1.8000 | 1.4625 | 1.4625 | 1.8000 | 0.9558 | 1.8000 | 1.4625 | 1.8000 | 1.8000 | 0.0000 |
| P16 | 1.4625 | 1.8000 | 1.8000 | 1.2933 | 1.8000 | 0.6750 | 1.8000 | 0.9558 | 1.8000 | 1.4625 | 0.9558 | 1.4625 | 0.9558 | 1.8000 | 1.1250 |
| P17 | 1.1250 | 1.8000 | 1.8000 | 1.8000 | 0.9558 | 1.1808 | 1.4625 | 1.4625 | 1.1250 | 1.2933 | 1.8000 | 1.8000 | 1.8000 | 1.4625 | 1.2933 |
| P18 | 1.5183 | 1.8000 | 1.8000 | 1.0125 | 1.1808 | 0.9558 | 1.1250 | 1.2933 | 1.4625 | 1.8000 | 1.2933 | 1.8000 | 1.2933 | 1.8000 | 0.8433 |
| P19 | 1.8000 | 1.1250 | 1.1808 | 1.4625 | 1.2933 | 1.8000 | 1.8000 | 1.8000 | 1.4625 | 1.2933 | 1.8000 | 1.4625 | 1.8000 | 1.8000 | 1.2933 |
| P20 | 1.4625 | 1.8000 | 1.8000 | 1.8000 | 0.9558 | 1.4625 | 1.1250 | 1.4625 | 1.1250 | 1.2933 | 1.8000 | 1.8000 | 1.5183 | 1.4625 | 0.9558 |
| P21 | 1.2933 | 1.4625 | 0.6183 | 1.1250 | 1.4625 | 1.4625 | 1.4625 | 1.8000 | 0.9558 | 1.4625 | 1.4625 | 1.2933 | 1.4625 | 1.8000 | 1.8000 |
| P22 | 1.8000 | 1.4625 | 1.8000 | 1.2933 | 1.4625 | 1.0125 | 1.8000 | 1.2933 | 1.8000 | 1.8000 | 0.9558 | 1.4625 | 0.9558 | 1.4625 | 1.8000 |
| P23 | 1.4625 | 1.8000 | 1.1250 | 0.6183 | 1.4625 | 0.9558 | 1.4625 | 0.9558 | 1.5183 | 0.7875 | 0.6750 | 1.4625 | 0.9558 | 1.1808 | 1.4625 |
| P24 | 1.4625 | 1.2933 | 1.4625 | 1.4625 | 1.4625 | 1.4625 | 0.4500 | 1.4625 | 1.8000 | 1.1250 | 1.4625 | 1.4625 | 1.4625 | 1.2933 | 1.1250 |
| P25 | 1.4625 | 1.4625 | 1.1250 | 0.6183 | 1.8000 | 1.0125 | 1.8000 | 0.9558 | 1.4625 | 1.4625 | 1.2933 | 1.8000 | 1.2933 | 1.1250 | 1.8000 |
| P26 | 1.2933 | 1.8000 | 0.9558 | 1.4625 | 1.8000 | 1.8000 | 1.5183 | 1.8000 | 1.2933 | 1.4625 | 1.4625 | 1.2933 | 1.4625 | 1.1250 | 1.4625 |
| P27 | 1.8000 | 1.0125 | 1.4625 | 1.4625 | 0.7875 | 1.1250 | 0.6183 | 1.5183 | 1.4625 | 1.4625 | 1.4625 | 1.5183 | 1.4625 | 1.2933 | 1.8000 |
| P28 | 1.4625 | 1.1250 | 1.4625 | 0.9558 | 1.8000 | 1.2933 | 1.8000 | 0.6183 | 1.4625 | 1.8000 | 1.2933 | 1.4625 | 1.2933 | 1.4625 | 1.8000 |
| P29 | 1.4625 | 1.8000 | 1.4625 | 1.4625 | 1.2933 | 1.5183 | 1.8000 | 1.4625 | 1.8000 | 0.6183 | 1.8000 | 1.4625 | 1.8000 | 1.4625 | 0.6183 |
| P30 | 0.6750 | 1.4625 | 1.2933 | 1.5183 | 1.1808 | 1.8000 | 1.8000 | 1.4625 | 1.2933 | 1.4625 | 1.8000 | 0.6183 | 1.8000 | 1.8000 | 1.1808 |

**Table F.3:** Changeover times in stage S6 (h).

| | P16 | P17 | P18 | P19 | P20 | P21 | P22 | P23 | P24 | P25 | P26 | P27 | P28 | P29 | P30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P01 | 1.4625 | 1.1250 | 1.5183 | 1.8000 | 1.4625 | 1.2933 | 1.8000 | 1.4625 | 1.4625 | 1.4625 | 1.2933 | 1.8000 | 1.4625 | 1.4625 | 0.6750 |
| P02 | 1.8000 | 1.8000 | 1.8000 | 1.1250 | 1.8000 | 1.4625 | 1.4625 | 1.8000 | 1.2933 | 1.4625 | 1.8000 | 1.0125 | 1.1250 | 1.8000 | 1.4625 |
| P03 | 1.8000 | 1.8000 | 1.8000 | 1.1808 | 1.8000 | 0.6183 | 1.8000 | 1.1250 | 1.4625 | 1.1250 | 0.9558 | 1.4625 | 1.4625 | 1.4625 | 1.2933 |
| P04 | 1.2933 | 1.8000 | 1.0125 | 1.4625 | 1.8000 | 1.1250 | 1.2933 | 0.6183 | 1.4625 | 0.6183 | 1.4625 | 1.4625 | 0.9558 | 1.4625 | 1.5183 |
| P05 | 1.8000 | 0.9558 | 1.1808 | 1.2933 | 0.9558 | 1.4625 | 1.4625 | 1.4625 | 1.4625 | 1.8000 | 1.8000 | 0.7875 | 1.8000 | 1.2933 | 1.1808 |
| P06 | 0.6750 | 1.1808 | 0.9558 | 1.8000 | 1.4625 | 1.4625 | 1.0125 | 0.9558 | 1.4625 | 1.0125 | 1.8000 | 1.1250 | 1.2933 | 1.5183 | 1.8000 |
| P07 | 1.8000 | 1.4625 | 1.1250 | 1.8000 | 1.1250 | 1.4625 | 1.8000 | 1.4625 | 0.4500 | 1.8000 | 1.5183 | 0.6183 | 1.8000 | 1.8000 | 1.8000 |
| P08 | 0.9558 | 1.4625 | 1.2933 | 1.8000 | 1.4625 | 1.8000 | 1.2933 | 0.9558 | 1.4625 | 0.9558 | 1.8000 | 1.5183 | 0.6183 | 1.4625 | 1.4625 |
| P09 | 1.8000 | 1.1250 | 1.4625 | 1.4625 | 1.1250 | 0.9558 | 1.8000 | 1.5183 | 1.8000 | 1.4625 | 1.2933 | 1.4625 | 1.4625 | 1.8000 | 1.2933 |
| P10 | 1.4625 | 1.2933 | 1.8000 | 1.2933 | 1.2933 | 1.4625 | 1.8000 | 0.7875 | 1.1250 | 1.4625 | 1.4625 | 1.4625 | 1.8000 | 0.6183 | 1.4625 |
| P11 | 0.9558 | 1.8000 | 1.2933 | 1.8000 | 1.8000 | 1.4625 | 0.9558 | 0.6750 | 1.4625 | 1.2933 | 1.4625 | 1.4625 | 1.2933 | 1.8000 | 1.8000 |
| P12 | 1.4625 | 1.8000 | 1.8000 | 1.4625 | 1.8000 | 1.2933 | 1.4625 | 1.4625 | 1.4625 | 1.8000 | 1.2933 | 1.5183 | 1.4625 | 1.4625 | 0.6183 |
| P13 | 0.9558 | 1.8000 | 1.2933 | 1.8000 | 1.5183 | 1.4625 | 0.9558 | 0.9558 | 1.4625 | 1.2933 | 1.4625 | 1.4625 | 1.2933 | 1.8000 | 1.8000 |
| P14 | 1.8000 | 1.4625 | 1.8000 | 1.8000 | 1.4625 | 1.8000 | 1.4625 | 1.1808 | 1.2933 | 1.1250 | 1.1250 | 1.2933 | 1.4625 | 1.4625 | 1.8000 |
| P15 | 1.1250 | 1.2933 | 0.8433 | 1.2933 | 0.9558 | 1.8000 | 1.8000 | 1.4625 | 1.1250 | 1.8000 | 1.4625 | 1.8000 | 1.8000 | 0.6183 | 1.1808 |
| P16 | 0.0000 | 1.5183 | 0.9558 | 1.8000 | 1.8000 | 1.8000 | 1.0125 | 0.9558 | 1.4625 | 1.0125 | 1.4625 | 1.8000 | 1.2933 | 0.8433 | 1.4625 |
| P17 | 1.5183 | 0.0000 | 1.4625 | 1.2933 | 0.6183 | 1.8000 | 1.5183 | 1.8000 | 1.8000 | 1.1808 | 1.8000 | 1.4625 | 1.4625 | 1.0125 | 1.8000 |
| P18 | 0.9558 | 1.4625 | 0.0000 | 1.8000 | 1.1250 | 1.8000 | 1.2933 | 1.2933 | 1.4625 | 1.2933 | 1.4625 | 1.4625 | 1.2933 | 1.4625 | 1.5183 |
| P19 | 1.8000 | 1.2933 | 1.8000 | 0.0000 | 1.2933 | 1.1250 | 1.4625 | 1.8000 | 1.8000 | 1.4625 | 1.8000 | 1.8000 | 1.4625 | 1.2933 | 1.4625 |
| P20 | 1.8000 | 0.6183 | 1.1250 | 1.2933 | 0.0000 | 1.8000 | 1.8000 | 1.8000 | 1.4625 | 1.4625 | 1.8000 | 1.4625 | 1.4625 | 1.2933 | 1.8000 |
| P21 | 1.8000 | 1.8000 | 1.8000 | 1.1250 | 1.8000 | 0.0000 | 1.8000 | 1.4625 | 1.4625 | 1.4625 | 1.2933 | 1.4625 | 1.1808 | 1.8000 | 1.2933 |
| P22 | 1.0125 | 1.5183 | 1.2933 | 1.4625 | 1.8000 | 1.8000 | 0.0000 | 1.2933 | 1.8000 | 1.0125 | 1.4625 | 1.4625 | 1.2933 | 1.5183 | 1.1250 |
| P23 | 0.9558 | 1.8000 | 1.2933 | 1.8000 | 1.8000 | 1.4625 | 1.2933 | 0.0000 | 1.1250 | 0.9558 | 1.4625 | 1.4625 | 1.2933 | 1.1250 | 1.4625 |
| P24 | 1.4625 | 1.8000 | 1.4625 | 1.8000 | 1.4625 | 1.4625 | 1.8000 | 1.1250 | 0.0000 | 1.8000 | 1.5183 | 0.9558 | 1.8000 | 1.4625 | 1.4625 |
| P25 | 1.0125 | 1.1808 | 1.2933 | 1.4625 | 1.4625 | 1.4625 | 1.0125 | 0.9558 | 1.8000 | 0.0000 | 1.4625 | 1.8000 | 0.6183 | 1.1808 | 1.8000 |
| P26 | 1.4625 | 1.8000 | 1.4625 | 1.8000 | 1.8000 | 1.2933 | 1.4625 | 1.4625 | 1.5183 | 1.4625 | 0.0000 | 1.8000 | 1.8000 | 1.1250 | 1.2933 |
| P27 | 1.8000 | 1.4625 | 1.4625 | 1.8000 | 1.4625 | 1.4625 | 1.4625 | 1.4625 | 0.9558 | 1.8000 | 1.8000 | 0.0000 | 1.8000 | 1.8000 | 1.4625 |
| P28 | 1.2933 | 1.4625 | 1.2933 | 1.4625 | 1.4625 | 1.1808 | 1.2933 | 1.2933 | 1.8000 | 0.6183 | 1.8000 | 1.8000 | 0.0000 | 1.8000 | 1.8000 |
| P29 | 0.8433 | 1.0125 | 1.4625 | 1.2933 | 1.2933 | 1.8000 | 1.5183 | 1.1250 | 1.4625 | 1.1808 | 1.1250 | 1.8000 | 1.8000 | 0.0000 | 1.4625 |
| P30 | 1.4625 | 1.8000 | 1.5183 | 1.4625 | 1.8000 | 1.2933 | 1.1250 | 1.4625 | 1.4625 | 1.8000 | 1.2933 | 1.4625 | 1.8000 | 1.4625 | 0.0000 |

**Table F.4:** Processing times (h).

| Product | M01 | M02 | M03 | M04 | M05 | M06 | M07 | M08 | M09 | M10 | M11 | M12 | M13 | M14 | M15 | M16 | M17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P01 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | 2.0979 | 1.6335 | 2.0205 | 2.0205 | - | - | 0.5778 | 0.2250 | 0.3780 | 0.3780 | 1.9818 | 1.9818 | 0.5661 |
| P02 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | - | 0.6894 | 0.2250 | 0.3780 | 0.3780 | 2.3634 | 2.3634 | 0.6750 |
| P03 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | 0.3339 | 0.3339 | - | 0.6750 | 1.1340 | 1.1340 | 0.3276 | 0.3276 | 0.0936 |
| P04 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | - | 0.5832 | 0.4500 | 0.7560 | 0.7560 | 1.9998 | 1.9998 | 0.5715 |
| P05 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | 0.2223 | 0.2223 | - | 0.2250 | 0.3780 | 0.3780 | 0.2178 | 0.2178 | 0.0621 |
| P06 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | 2.0403 | 2.0403 | 0.5832 | 0.2250 | 0.3780 | 0.3780 | 1.9998 | 1.9998 | 0.5715 |
| P07 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | - | 0.1062 | 0.2250 | 0.3780 | 0.3780 | 0.3636 | 0.3636 | 0.1035 |
| P08 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | 2.0979 | 1.6335 | 2.0205 | 2.0205 | - | 0.3708 | 0.1062 | 0.6750 | 1.1340 | 1.1340 | 0.3636 | 0.3636 | 0.1035 |
| P09 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | - | 0.1008 | 0.6750 | 1.1340 | 1.1340 | 0.3456 | 0.3456 | 0.0990 |
| P10 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | - | 0.2646 | 0.4500 | 0.7560 | 0.7560 | 0.9090 | 0.9090 | 0.2601 |
| P11 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | - | 0.7947 | 0.4500 | 0.7560 | 0.7560 | 2.7270 | 2.7270 | 0.7794 |
| P12 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | 2.0979 | 1.6335 | 2.0205 | 2.0205 | - | 1.8549 | - | 0.4500 | 0.7560 | 0.7560 | 1.8180 | 1.8180 | 0.5193 |
| P13 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | 2.0979 | 1.6335 | 2.0205 | 2.0205 | - | 0.7416 | - | 0.2250 | 0.3780 | 0.3780 | 0.7272 | 0.7272 | 0.2079 |
| P14 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | 0.3528 | 0.3528 | 0.1008 | 0.4500 | 0.7560 | 0.7560 | 0.3456 | 0.3456 | 0.0990 |
| P15 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | 2.0403 | 2.0403 | 0.5832 | 0.4500 | 0.7560 | 0.7560 | 1.9998 | 1.9998 | 0.5715 |
| P16 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | 2.0979 | 1.6335 | 2.0205 | 2.0205 | - | - | 0.2124 | 0.4500 | 0.7560 | 0.7560 | 0.7272 | 0.7272 | 0.2079 |
| P17 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | 2.0979 | 1.6335 | 2.0205 | 2.0205 | 1.1133 | 1.1133 | - | 0.4500 | 0.7560 | 0.7560 | 1.0908 | 1.0908 | 0.3114 |
| P18 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | - | 0.6363 | 0.4500 | 0.7560 | 0.7560 | 2.1816 | 2.1816 | 0.6237 |
| P19 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | - | 0.5616 | 0.4500 | 0.7560 | 0.7560 | 1.9269 | 1.9269 | 0.5508 |
| P20 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | 2.0979 | 1.6335 | 2.0205 | 2.0205 | - | 0.3339 | 0.0954 | 0.6750 | 1.1340 | 1.1340 | 0.3276 | 0.3276 | 0.0936 |
| P21 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | 2.0205 | 2.0205 | - | 0.2781 | - | 0.6750 | 1.1340 | 1.1340 | 0.2727 | 0.2727 | 0.0783 |
| P22 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | - | 0.5409 | 0.6750 | 1.1340 | 1.1340 | 1.8549 | 1.8549 | 0.5301 |
| P23 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | 2.0979 | 1.6335 | 2.0205 | 2.0205 | - | 2.0403 | 0.5832 | 0.6750 | 1.1340 | 1.1340 | 1.9998 | 1.9998 | 0.5715 |
| P24 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | 2.0205 | 2.0205 | - | 2.2257 | 0.6363 | 0.2250 | 0.3780 | 0.3780 | 2.1816 | 2.1816 | 0.6237 |
| P25 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | 1.1133 | - | 0.6750 | 1.1340 | 1.1340 | 1.0908 | 1.0908 | 0.3114 |
| P26 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | 2.0979 | 1.6335 | 2.0205 | 2.0205 | - | 1.9476 | 0.5562 | 0.4500 | 0.7560 | 0.7560 | 1.9089 | 1.9089 | 0.5454 |
| P27 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | 2.0205 | 2.0205 | - | 2.0403 | 0.5832 | 0.2250 | 0.3780 | 0.3780 | 1.9998 | 1.9998 | 0.5715 |
| P28 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | - | 0.1062 | 0.2250 | 0.3780 | 0.3780 | 0.3636 | 0.3636 | 0.1035 |
| P29 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | 1.6695 | 1.6695 | - | 0.2250 | 0.3780 | 0.3780 | 1.6362 | 1.6362 | 0.4671 |
| P30 | 0.9000 | 0.9000 | 1.3050 | 1.3050 | - | - | - | - | - | 3.3390 | - | 0.2250 | 0.3780 | 0.3780 | 3.2724 | 3.2724 | 0.9351 |

# References

# Bibliography


Abdeddaïm, Y., E. Asarin, & O. Maler. "Scheduling with timed automata." *Theoretical Computer Science* 354: (2006) 272–300.

Acevedo, J., & E. N. Pistikopoulos. "A hybrid parametric/stochastic programming approach for mixed-integer linear problems under uncertainty." *Industrial and Engineering Chemistry Research* 36, no. 6: (1997a) 2262–2270.

———. "A multiparametric programming approach for linear process engineering problems under uncertainty." *Industrial and Engineering Chemistry Research* 36, no. 3: (1997b) 717–728.

Ahumada, O., & J. R. Villalobos. "Application of planning models in the agri-food supply chain: A review." *European Journal of Operational Research* 195: (2009) 1–20.

Akkerman, R., P. Farahani, & M. Grunow. "Quality, safety and sustainability in food distribution: a review of quantitative operations management approaches and challenges." *OR Spectrum* 32: (2010) 863–904.

Aytug, H., M. Lawley, K. McKay, S. Mohan, & R. Uzsoy. "Executing production schedules in the face of uncertainties: A review and some future directions." *European Journal of Operational Research* 161, no. 1: (2005) 86–110.

Badell, M., J. Romero, R. Huertas, & L. Puigjaner. "Planning, scheduling and budgeting value-added chains." *Computers and Chemical Engineering* 28, no. 1-2: (2004) 45–61.

Balasubramanian, J., & I. E. Grossmann. "Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty." *Industrial and Engineering Chemistry Research* 43, no. 14: (2004) 3695–3713.

Baptiste, P., C. Le Pape, & W. Nuijten. *Constraint-based scheduling. Applying constraint programming to scheduling problems*. Dordrecht, Holland: Kluwer Academics, 2001.

Bassett, M. H., J. F. Pekny, & G. V. Reklaitis. "Using detailed scheduling to obtain realistic operating policies for a batch processing facility." *Industrial and Engineering Chemistry Research* 36: (1997) 1717–1726.

Berning, G., M. Brandenburg, K. Gürsoy, V. Mehta, & F.-J. Tölle. "An integrated system for supply chain optimisation in the chemical process industry." *OR Spectrum* 24, no. 4: (2002) 371–401.

Bilgen, B., & H.-O. Günther. "Integrated production and distribution planning in the fast moving consumer goods industry: a block planning application.” *OR Spectrum* 32: (2010) 927–955.

Birewar, D., & I. E. Grossmann. “Efficient optimization algorithms for zero-wait scheduling of multiproduct batch plants.” *Industrial and Engineering Chemistry Research* 28, no. 9: (1989) 1333–1345.

———. “Simultaneous planning and scheduling of multiproduct batch plants.” *Industrial and Engineering Chemistry Research* 29: (1990) 570–580.

Blackstone, J. H., D. T. Phillips, & G. L. Hogg. “A state-of-the-art survey of dispatching rules for manufacturing job shop operations.” *International Journal of Production Research* 20: (1982) 27–45.

Blömer, F., & H.-O. Günther. “Scheduling of a multi-product batch process in the chemical industry.” *Computers in Industry* 36: (1998) 245–259.

———. “LP-based heuristics for scheduling chemical batch processes.” *International Journal of Production Research* 38: (2000a) 1029–1051.

———. “LP-based heuristics for scheduling chemical batch processes.” *International Journal of Production Research* 35: (2000b) 1029–1052.

Bojarski, A. D., J. M. Laínez, A. Espuña, & L. Puigjaner. “Incorporating environmental impacts and regulations in a holistic supply chains modeling: An LCA approach.” *Computers and Chemical Engineering* 33, no. 10: (2009) 1747–1759.

Bonfill, A. *Proactive management of uncertainty to improve scheduling robustness in process industries*. Ph.D. thesis, Universitat Politecnica de Catalunya, 2006.

Bongers, P. M. M., & B. H. Bakker. “Application of multi-stage scheduling.” In *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*, edited by W. Marquardt, & C. Pantelides, Elsevier, 2006, Vol. 21 of *Computer Aided Chemical Engineering*, 1917–1922.

Brown, G., J. Keegan, B. Vigus, & K. Wood. “The Kellogg company optimizes production, inventory, and distribution.” *Interfaces* 31: (2001) 1–15.

Burkard, R. E., M. Hujter, B. Klinz, R. Rudolf, & M. T. Wennink. “A process scheduling problem arising from chemical production planning.” *Optimization Methods and Software* 10: (1998) 175–196.

Castillo, E., A. J. Conejo, P. Pedregal, R. García, & N. Alguaci. *Building and Solving Mathematical Programming Models in Engineering and Science*. New York, USA: John Wiley & Sons, 2001.

Castro, P. M., A. P. F. D. Barbosa-Póvoa, & H. A. Matos. “An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants.” *Industrial and Engineering Chemistry Research* 40, no. 9: (2001) 2059–2068.

Castro, P. M., A. P. F. D. Barbosa-Póvoa, H. A. Matos, & A. Q. Novais. “Simple continuous-time formulation for short-term scheduling of batch and continuous processes.” *Industrial and Engineering Chemistry Research* 43, no. 1: (2004) 105–118.

Castro, P. M., & I. E. Grossmann. “An efficient MILP model for the short-term scheduling of single stage batch plants.” *Computers and Chemical Engineering* 30, no. 6-7: (2006) 1003–1018.

Castro, P. M., I. E. Grossmann, & A. Q. Novais. “Two new continuous-time models for the scheduling of multistage batch plants with sequence dependent changeovers.” *Industrial and Engineering Chemistry Research* 45, no. 18: (2006) 6210–6226.

Castro, P. M., I. Harjunkoski, & I. E. Grossmann. “Optimal short-term scheduling of large-scale multistage batch plants." *Industrial and Engineering Chemistry Research* 48: (2009) 11,002–11,016.

Cavin, L., U. Fischer, F. Glover, & K. Hungerbhüler. "Multi-objective process design in multi-purpose batch plants using a tabu search optimization algorithm." *Computers and Chemical Engineering* 28: (2004) 459–478.

CEFIC. "Facts and Figures: The European chemical industry in a worldwide perspective." Technical report, Conseil Européen des Fédérations de l'Industrie Chimique, 2010.

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 and Engineering Chemistry Research* 36, no. 5: (1997) 1695–1707.

Chen, C.-L., C.-L. Liu, X.-D. Feng, & H.-H. Shao. "Optimal short-term scheduling of multi-product single-stage batch plants with parallel lines." *Industrial and Engineering Chemistry Research* 41, no. 5: (2002) 1249–1260.

Chen, P., L. G. Papageorgiou, & J. M. Pinto. "Medium-term planning of single-stage single-unit multiproduct plants using a hybrid discrete/continuous-time MILP model." *Industrial and Chemical Engineering Research* 47: (2008) 1925–1934.

Conejo, A. J., F. J. Nogales, & F.J. Prieto. "A decomposition procedure based on approximate Newton directions." *Mathematical Programming* 93, no. 3: (2002) 495–515.

CSCMP. "http://cscmp.org." Technical report, Council of Supply Chain Management Professionals, 2007.

Dannenbring, D. G. "Evaluation of flow shop sequencing heuristics." *Management Science* 23: (1977) 1174–1182.

Dantzig, G. B., & M. N. Thapa. *Linear Programming 1: Introduction*. Series in Operations Research. NY 10010, USA: Springer-Verlag New York, Inc., 1997.

———. *Linear Programming 2: Theory and Extensions*. Series in Operations Research. NY 10010, USA: Springer-Verlag New York, Inc, 2003.

Doganis, P., & H. Sarimveis. "Optimal production scheduling for the dairy industry." *Annals of Operations Research* 159: (2008) 315–331.

Dua, P., K. Kouramas, V. Dua, & E. N. Pistikopoulos. "MPC on a chip - Recent advances on the application of multi-parametric model-based control." *Computers and Chemical Engineering* 32, no. 4-5: (2008) 754–765.

Duran, M., & I. E. Grossmann. "An outer-approximation algorithm for a class of mixed-integer nonlinear programs." *Mathematical Programming* 36: (1986) 307–339.

Eksioglu, S. D., & M. Jin. "Cross-facility production and transportation planning problem with perishable inventory." In *In: Computational Science and Its Applications - ICCSA 2006*. Heidelberg: Springer Berlin, 2006, 708–717.

Elkamel, A. *Scheduling of process operations using mathematical programming techniques*. Ph.D. thesis, Purdue University, 1993.

Entrup, M. L., H.-O. Günther, P. Van Beek, M. Grunow, & T. Seiler. "Mixed-integer linear programming approaches to shelf-life-integrated planning and scheduling in yoghurt production." *International Journal of Production Research* 43: (2005) 5071–5100.

Erdirik-Dogan, M., & I. E. Grossmann. "A decomposition method for the simultaneous planning and scheduling of single-stage continuous multiproduct plants." *Industrial Engineering and Chemical Research* 45: (2006) 299–315.

———. "Simultaneous planning and scheduling of single-stage multi-product continuous plants with parallel lines." *Computers and Chemical Engineering* 32: (2008) 2664–2683.

Espuña., A., & L. Puigjaner. "Solving the production planning problem for parallel multi-product plants." *Institute of Chemical Engineering Symposium Series* 114: (1989) 15–25.

Ferrer-Nadal, S., E. Capón-García, C. A. Méndez, & L. Puigjaner. "Material transfer operations in batch scheduling. A critical modeling issue." *Industrial and Engineering Chemistry Research* 47, no. 20: (2008) 7721–7732.

Ferris, M. "MATLAB and GAMS: Interfacing optimization and visualization software." Technical report, University of Wisconsin, 1999.

Franca, P. M., M. Gendreau, G. Laporte, & F. M. Muller. "A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times." *International Journal of Production Economics* 43: (1996) 78–89.

Gass, S. I. *Linear Programming: Methods and Applications*. Mineola, New York, USA: Dover Publications, Inc., 2003.

Giannelos, N. F., & M. C. Georgiadis. "A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes." *Industrial and Engineering Chemistry Research* 41, no. 9: (2002) 2178–2184.

Gimenez, D. M., G. P. Henning, & C. T. Maravelias. "A novel network-based continuous-time representation for process scheduling: Part I. Main concepts and mathematical formulation." *Computers and Chemical Engineering* 33, no. 9: (2009a) 1511–1528.

———. "A novel network-based continuous-time representation for process scheduling: Part II. General framework." *Computers and Chemical Engineering* 33, no. 10: (2009b) 1644–1660.

Glover, F. "Tabu search: A tutorial." *Interfaces* 20: (1990) 74.

Günther, H.-O., M. Grunow, & U. Neuhaus. "Realizing block planning concepts in make-and-pack production using MILP modelling and SAP APO." *International Journal of Production Research* 44, no. 18-19: (2006) 3711–3726.

Gooding, W. B. *Specially structured formulations and solution methods for optimisation problems important to process scheduling*. Ph.D. thesis, Purdue University, 1994.

Gooding, W. B., J. F. Pekny, & P. S. McCroskey. "Enumerative approaches to parallel flowshop scheduling via problem transformation." *Computers and Chemical Engineering* 18: (1994) 909–927.

Graells, M., J. Cantón, B. Peschaud, & L. Puigjaner. "General approach and tool for the scheduling of complex production systems." *Computers and Chemical Engineering* 22, no. S1: (1998) S395–S402.

Graells, M., A. Espuña, & L. Puigjaner. "Sequencing intermediate products: A practical solution for multipurpose production scheduling." *Computers and Chemical Engineering* S20: (1996) S1137–S1142.

Graves, S. C. "Using Lagrangean techniques to solve hierarchical production planning problems." *Management Science* 28: (1982) 260–275.

Grossmann, I. E., & A. W. Westerberg. "Research challenges in process systems engineering." *AIChE Journal* 46, no. 9: (2000) 1700–1703.

Grunow, M., H.-O. Günther, & M. Lehmann. "Campaign planning for multi-stage batch processes in the chemical industry." *OR Spectrum* 24: (2002) 281–314.

Grunow, M., & J. van der Vorst. "Food production and supply chain management." *OR Spectrum* 32: (2010) 861–862.

Guillén, G., M. Badell, & L. Puigjaner. "A holistic framework for short-term supply chain management integrating production and corporate financial planning." *Computers and*

*Chemical Engineering* 106, no. 1: (2007) 288–306.

Gupta, A., & C. D. Maranas. "A hierarchical Lagrangean relaxation procedure for solving midterm planning problems." *Industrial and Engineering Chemistry Research* 38: (1999) 1937–1947.

Harjunkoski, I., & I. E. Grossmann. "A decomposition approach for the scheduling of a steel plant production." *Computers and Chemical Engineering* 55, no. 11: (2001) 1647–1660.

Henning, G. P., & J. Cerdá. "Knowledge-based predictive and reactive scheduling in industrial environments." *Computers and Chemical Engineering* 24, no. 9: (2000) 2315–2338.

Herrmann, J. W. *Handbook of Production Scheduling*. New York, NY 10013, USA: Springer Science + Business Media, Inc., 2006.

Hess, E. "Make advanced planning & scheduling work for your company." *APS* .

Higgins, A., G. Beashel, & A. Harrison. "Scheduling of brand production and shipping within a sugar supply chain." *Journal of the Operational Research Society* 57: (2006) 490–498.

Honkomp, S. J., S. Lombardo, O. Rosen, & J. F. Pekny. "The curse of reality - why process scheduling optimization problems are difficult in practice." *Computers and Chemical Engineering* 24, no. 2-7: (2000) 323–328.

Hui, C.-W., & A. Gupta. "A bi-index continuous-time mixed-integer linear programming model for single-stage batch scheduling with parallel units." *Industrial and Engineering Chemistry Research* 40, no. 25: (2001) 5960–5967.

Ierapetritou, M. G., & C. A. Floudas. "Effective continuous-time formulation for short-term scheduling. 1.Multipurpose batch processes." *Industrial and Engineering Chemistry Research* 37, no. 11: (1998) 4341–4359.

Inman, R. R., & P. C. Jones. "Decomposition for scheduling flexible manufacturing systems." *Operations Research* 41, no. 3: (1993) 608–617.

Jakeman, C. M. "Scheduling needs of the food processing industry." *Food Research International* 27: (1994) 117–120.

Kallrath, J. "Combined strategic and operational planning - an MILP success story in chemical industry." *OR Spectrum* 24, no. 3: (2002a) 315–341.

———. "Planning and scheduling in the process industry." *OR Spectrum* 24: (2002b) 219–250.

Karimi, I. A., & C. M. McDonald. "Planning and scheduling of parallel semicontinuous processes. 2. Short-term scheduling." *Industrial Engineering and Chemical Research* 36, no. 7: (1997) 2701–2714.

Karmarkar, U. S., & L. Schrage. "The deterministic dynamic product cycling problem." *Operations Research* 33, no. 2: (1985) 326–345.

Keskinocak, P., F. Wu, R. Goodwin, S. Murthy, R. Akkiraju, S. Kumaran, & A. Derebail. "Scheduling solutions for the paper industry." *Operations Research* 50, no. 2: (2002) 249–259.

Kim, M., J. H. Jung, & I.-B. Lee. "Optimal scheduling of Multiproduct batch processes for various intermediate storage policies." *Industrial and Engineering Chemistry Research* 35, no. 11: (1996) 4048–4066.

Kirkpatrick, S., C. D. Gelatt, & M. P. Vechi. "Optimization by simulated annealing." *Science* 220: (1983) 671–680.

Kondili, E., C. C. Pantelides, & R. W. H. Sargent. "A general algorithm for scheduling of batch operations." In *Proceedings of Third International Symposium on Process Systems Engineering*. Sydney, Australia, 1988, 62–75.

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

Kopanos, G. M., L. Puigjaner, & M. C. Georgiadis. "Optimal production scheduling and lot-sizing in dairy plants: The yogurt production line." *Industrial Engineering and Chemical Research* 49, no. 2: (2010) 701–718.

Kreipl, S., & M. Pinedo. "Planning and scheduling in supply chains: An overview of issues in practice." *Production and Operations Management* 213, no. 1: (2004) 77–92.

Ku, H., & I.A. Karimi. "An evaluation of simulated annealing for batch process scheduling." *Industrial and Chemical Engineering Research* 30: (1991) 163–169.

Kudva, G., A. Elkamel, J. F. Pekny, & G. V. Reklaitis. "Heuristic algorithm for scheduling batch and semicontinuous plants with production deadlines, intermediate storage limitations and equipment changeover costs." *Computers and Chemical Engineering* 18: (1994) 859–875.

Kuriyan, K., & G. V. Reklaitis. "Scheduling network flowshops so as to minimise makespan." *Computers and Chemical Engineering* 13: (1989) 187–200.

Laborie, P. "Algorithms for propagating resource constraints in AI planning and scheduling: Existing approaches and new results." *Artificial Intelligence* 143, no. 2: (2003) 151–188.

Lamba, N., & I. A. Karimi. "Scheduling parallel production lines with resource constraints. 1. Model formulation." *Industrial and Engineering Chemistry Research* 41, no. 4: (2002) 779–789.

Land, A. H., & A. G. Doig. "An automatic method for solving discrete programming problems." *Econometrica* 28: (1960) 497–520.

Laínez, J. M., G. Guillén-Gonsálbez, M. Badell, A. Espuña, & L. Puigjaner. "Enhancing corporate value in the optimal design of chemical supply chains." *Industrial and Engineering Chemistry Research* 46, no. 23: (2007) 7739–7757.

Laínez, J. M., M. Hegyháti, F. Friedler, & L. Puigjaner. "Using S-graph to Address uncertainty in batch plants." *Clean Technology and Environmental Policy* 12, no. 2: (2010a) 105–115.

Laínez, J. M., G. V. Reklaitis, & L. Puigjaner. "Linking marketing and supply chain models for improved business strategic decision support." *Computers and Chemical Engineering* 34, no. 12: (2010b) 2107–2117.

Le Pape, C. "Implementation of resource constraints in ILOG schedule: A library for the development of constrained-based scheduling systems." *Intelligent Systems Engineering* 3, no. 2: (1998) 55–66.

Lee, K.-H., S.-K. Heo, H.-K. Lee, & I.-B. Lee. "Scheduling of single-stage and continuous processes on parallel lines with intermediate due dates." *Industrial and Engineering Chemistry Research* 41, no. 1: (2002) 58–66.

Lee, S., & I. E Grossmann. "New algorithms for nonlinear disjunctive programming." *Computers & Chemical Engineering* 24: (2000) 2125–2141.

Li, Z., & M. G. Ierapetritou. "Process scheduling under uncertainty: Review and challenges." *Computers and Chemical Engineering* 32, no. 4-5: (2008) 715–727.

———. "Production planning and scheduling integration through augmented Lagrangian optimization." *Computers and Chemical Engineering* 34, no. 6: (2010) 996–1006.

Lim, M., & I. A. Karimi. "Resource-constrained scheduling of parallel production lines using asynchronous slots." *Industrial and Engineering Chemistry Research* 42, no. 26: (2003) 6832–6842.

Liu, S., J. M. Pinto, & L. G. Papageorgiou. "A TSP-based MILP model for medium-term planning of single-stage continuous multiproduct plants." *Industrial and Chemical Engineering Research* 47: (2008) 7733–7743.

———. "MILP-based approaches for medium-term planning of single-stage continuous multiproduct plants with parallel units." *Computers and Management Science* 7: (2009) 407–435.

Lázaro, M., & L. Puigjaner. "Simulation and optimization of multi-product plants for batch and semi-batch operations." *Institute of Chemical Engineering Symposium Series* 92: (1985) 209–222.

———. "Solution of integer optimization problems subjected to non-linear restrictions: An improved algorithm." *Computers and Chemical Engineering* 12, no. 5: (1988) 443–448.

Majozi, T., & F. Friedler. "Maximization of throughput in a multipurpose batch plant under a fixed time horizon: S-graph approach." *Industrial and Engineering Chemistry Research* 45, no. 20: (2006) 6713–6720.

Manzini, R., & F. Bindi. "Strategic design and operational management optimization of a multi stage physical distribution system." *Transportation Research Part E: Logistics and Transportation Review* 45, no. 6: (2009) 915–936.

Maravelias, C. T. "Mixed-time representation for state-task network models." *Industrial and Engineering Chemistry Research* 44, no. 24: (2005) 9129–9145.

Maravelias, C. T., & I. E. Grossmann. "New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants." *Industrial and Engineering Chemistry Research* 42, no. 13: (2003) 3056–3074.

Marchetti, P. A., & J. Cerdá. "A continuous-time tightened formulation for single-stage batch scheduling with sequence-dependent changeovers." *Industrial and Engineering Chemistry Research* 48, no. 1: (2009a) 483–498.

———. "A general resource-constrained scheduling framework for multistage batch facilities with sequence-dependent changeovers." *Computers and Chemical Engineering* 33, no. 4: (2009b) 871–886.

Marchetti, P. A., C. A. Méndez, & J. Cerdá. "Mixed-integer linear programming monolithic formulations for lot-sizing and scheduling of single-stage batch facilities." *Industrial and Engineering Chemistry Research* 49: (2010) 6482–6498.

Marinelli, F., M. E. Nenni, & A. Sforza. "Capacitated lot sizing and scheduling with parallel machines and shared buffers: A case study in a packaging company." *Annals of Operations Research* 150: (2007) 177–192.

Marquardt, W., L .V. Wedel, & B. Bayer. "Perspectives on lifecycle process modeling." *Proceedings of the AIChE Symposium Series* 96, no. 323: (2000) 192–214.

Melo, M. T., S. Nickel, & F. Saldanha-da-Gama. "Facility location and supply chain management - A review." *European Journal of Operational Research* 196: (2009) 401–412.

Min, H., & G. Zhou. "Supply chain modeling: past, present and future." *Computers and Industrial Engineering* 43, no. 1-2: (2002) 231–249.

Méndez, C. A., & J. Cerdá. "An MILP framework for short-term scheduling of single-stage batch plants with limited discrete resources." *Computer-Aided Chemical Engineering* 10: (2002a) 721–726.

———. "Short-term scheduling of multistage batch processes subject to limited finite resources." *Computer-Aided Chemical Engineering* 15B: (2002b) 984–989.

———. "Dynamic scheduling in multiproduct batch plants." *Computers and Chemical Engineering* 27: (2003a) 1247–1259.

———. "An MILP continuous-time framework for short-term scheduling of multipurpose batch processes under different operation strategies." *Optimization and Engineering* 4, no. 1-2: (2003b) 7–22.

Méndez, C. A., J. Cerdá, I. E. Grossmann, I. Harjunkoski, & M. Fahl. "Review: State-of-the-art of optimization methods for short-term scheduling of batch processes." *Computers and Chemical Engineering* 30, no. 6-7: (2006) 913–946.

Méndez, C. A., G. P. Henning, & J. Cerdá. "Optimal scheduling of batch plants satisfying multiple product orders with different due-dates." *Computers and Chemical Engineering* 24, no. 9-10: (2000) 2223–2245.

———. "An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities." *Computers and Chemical Engineering* 25, no. 4-6: (2001) 701–711.

Mockus, L., & G. V. Reklaitis. "Continuous time representation approach to batch and continuous process scheduling. 1. MINLP formulation." *Industrial and Engineering Chemistry Research* 38, no. 1: (1999a) 197–203.

———. "Continuous time representation approach to batch and continuous process scheduling. 2. Computational issues." *Industrial and Engineering Chemistry Research* 38, no. 1: (1999b) 204–210.

Moon, S., S. Park, & W. K. Lee. "New MILP models for scheduling of multiproduct batch plants under zero-wait policy." *Industrial and Engineering Chemistry Research* 35: (1996) 3458–3469.

Moula, J., D. Peidro, M. Díaz-Madroñero, & E. Vicens. "Mathematical programming models for supply chain production and transport planning." *European Journal of Operational Research* 204: (2010) 377–390.

Mulvey, J. M., R. J. Vanderbei, & S. A. Zenios. "Robust optimization of large-scale systems." *Operations Research* 43, no. 2: (1995) 264–281.

Murakami, Y., H. Uchiyama, S. Hasebe, & I. Hashimoto. "Application of repetitive SA method to scheduling problems of chemical processes." *Computers and Chemical Engineering* S21: (1997) S1087–S1092.

Nakhla, M. "Production control in the food processing industry." *International Journal of Operations & Production Management* 15: (1995) 73–88.

Nawaz, M., E. E. Enscore, & I. Ham. "A heuristic algorithm for the m-machine n-job flowshop sequencing problem." *OMEGA, The International Journal of Management Science* 11: (1983) 91–95.

Nemhauser, G., & L. A. Wolsey. *Integer and Combinatorial Optimization*. Series in Discrete Mathematics and Optimization. New York, USA: Wiley-Interscience, 1999.

Neumann, K., C. Schwindt, & N. Trautmann. "Advanced production scheduling for batch plants in process industries." *OR Spectrum* 24: (2002) 251–279.

Pacciarelli, D. "The alternative graph formulation for solving complex factory scheduling problems." *International Journal of Production Research* 40: (2002) 3641–3653.

Pantelides, C. C. "Unified frameworks for optimal process planning and scheduling." In *Proceedings of Second Conference on Foundations of Computer Aided Process Operations*. CACHE, 1994, 253–274.

Pantelides, C. C., M. J. Realff, & N. Shah. "Short-term scheduling of pipeless batch plants." *Chemical Engineering Research and Design* 73, no. A4: (1995) 431–444.

Panwalkar, S. S., & W. A. Iskander. "Survey of scheduling rules." *Operations Research* 25:

(1977) 45–61.

Papageorgiou, L. G. "Supply chain optimisation for the process industries: Advances and opportunities." *Computers and Chemical Engineering* 33, no. 12: (2009) 1931–1938.

Papageorgiou, L. G., & C. C. Pantelides. "Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 1. Mathematical formulation." *Industrial and Chemical Engineering Research* 35, no. 2: (1996a) 488–509.

———. "Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 2. A mathematical decomposition approach." *Industrial and Chemical Engineering Research* 35, no. 2: (1996b) 510–529.

Pekny, J. F., D. L. Miller, & G. J. McCrae. "Application of a parallel travelling salesman problem to no-wait flowshop scheduling." In *AIChE Annual Meeting 1988*. Washington D.C., USA, 1988.

———. "An exact parallel algorithm for scheduling when production costs depend on consecutive system states." *Computers and Chemical Engineering* 14: (1990) 1009–1023.

Pinedo, M. *Scheduling, Theory, Algorithms and Systems*. New York, USA: Prentice-Hall, 1995.

Pinedo, M., & X. Chao. *Operations scheduling with applications in manufacturing and services*. New York, USA: McGraw-Hill International Editions, 1999.

Pinto, J. M, & I. E. Grossmann. "A continuous time mixed integer linear programming model for short-term scheduling of multistage batch plants." *Industrial and Engineering Chemistry Research* 34, no. 9: (1995) 3037–3051.

———. "A logic-based approach to scheduling problems with resource constraints." *Computers and Chemical Engineering* 21, no. 8: (1997) 801–818.

Pistikopoulos, E. N., V. Dua, N. A. Bozinis, A. Bemporad, & M. Morari. "On-line optimization via off-line parametric optimization tools." *Computers and Chemical Engineering* 26, no. 2: (2002) 175–185.

Porter, M. E. *Competitive Advantage: Creating and Sustaining Superior Performance*. New York, USA: The Free Press, 1985.

Prasad, P., & C. T. Maravelias. "Batch selection, assignment and sequencing in multi-stage multi-product processes." *Computers and Chemical Engineering* 32, no. 6: (2008) 2008.

Puigjaner, L. "Handling the increasing complexity of detailed batch process simulation and optimisation." *Computers and Chemical Engineering* 23, no. S1: (1999) S929–S943.

Puigjaner, L., & G. Guillém-Gosálbez. "Towards an integrated framework for supply chain management in the batch chemical process industry." *Computers and Chemical Engineering* 32, no. 4-5: (2008) 650–670.

Puigjaner, L., J. M. Laínez, & C. R. Álvarez. "Tracking the dynamics of the supply chain for enhanced production sustainability." *Industrial and Engineering Chemistry Research* 48, no. 21: (2009) 9556–9570.

Raaymakers, W. H. M., & J. A. Hoogeveen. "Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing." *European Journal of Operational Research* 126: (2000) 131–151.

Raman, R., & I. E. Grossmann. "Modeling and computational techniques for logic based integer programming." *Computers & Chemical Engineering* 18: (1994) 563–578.

Reklaitis, G. V. "Reviewing of scheduling of process operations." In *Chemical Engineering Progress Symposium Series: Selected Topics in Computer Aided Process Design and Analysis*. AIChE, New York, USA, 1982.

Rippin, D. W. T. "Batch process systems engineering: A retrospective and prospective review." *Computers and Chemical Engineering* S17: (1993) S1–S13.

Rodrigues, M. T. M., L. G. Latre, & L. C. A. Rodrigues. "Short-term planning and scheduling in multipurpose batch chemical plants: A multi-level approach." *Computers and Chemical Engineering* 24, no. 9-10: (2000) 2247–2258.

Rong, A., R. Akkerman, & M. Grunow. "An optimization approach for managing fresh food quality throughout the supply chain." *International Journal of Production Economics (in press)* doi:10.1016/j.ijpe.2009.11.026.

Rosenthal, R. E. *GAMS - A User's Guide*. Washington, DC, USA: GAMS Development Corporation, 2010.

Röslof, J., I. Harjunkoski, J. Björkqvist, S. Karlsson, & T. Westerlund. "An MILP-based reordering algorithm for complex industrial scheduling and rescheduling." *Computers and Chemical Engineering* 25: (2001) 821–828.

Röslof, J., I. Harjunkoski, T. Westerlund, & J. Isaksson. "Solving a large-scale industrial scheduling problem using MILP combined with a heuristic procedure." *European Journal of Operational Research* 138: (2002) 29–42.

Ruiz, R., & C. Maroto. "A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility." *European Journal of Operational Research* 169: (2006) 781–800.

Ruiz, R., F. S. Serifoglu, & T. Urlings. "Modeling realistic hybrid flexible flowshop scheduling problems." *Computers and Operations Research* 35: (2008) 1151–1175.

Ruiz, R., & T. Stutzle. "An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives." *European Journal of Operational Research* 187: (2008) 1143–1159.

Sahinidis, N. V. "Optimization under uncertainty: State-of-the-art and opportunities." *Computers and Chemical Engineering* 28, no. 6-7: (2004) 971–983.

Sahinidis, N. V., & I. E. Grossmann. "MINLP model for cyclic multiproduct scheduling on continuous parallel lines." *Computers and Chemical Engineering* 15: (1991) 85–103.

Sanmartí, E., F. Friedler, & L. Puigjaner. "Combinatorial technique for short term scheduling of multipurpose batch plants based on schedule-graph representation." *Computers and Chemical Engineering* 22, no. S1: (1998) S847–S850.

Sanmartí, E., T. Holczinger, F. Friedler, & L. Puigjaner. "Combinatorial framework for effective scheduling of multipurpose batch plants." *AIChE Journal* 48, no. 11: (2002) 2557–2570.

Sargent, R. W. H. "What is chemical engineering?" *CAST Newsletter* 14, no. 1: (1991) 9–11.

Sauer, J., & R. Bruns. "Knowledge-based scheduling in industry and medicine." *IEEE Expert* 12: (1997) 24–31.

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

Schnelle, K. D. "Preliminary design and scheduling of a batch agrochemical plant." *Computers and Chemical Engineering* 24, no. 2-7: (2000) 1535–1541.

Shah, N. "Single- and multi-site planning and scheduling: current status and future challenge." In *Proceedings of the Third International Conference of the Foundations of Computer-Aided Process Operations*, edited by J. F. Pekny, & G. E. Blau. AIChE Symposium Series 94, AIChE: New York, 1998, 75–90.

Shah, N., C. C. Pantelides, & R. W. H. Sargent. "A general algorithm for short-term scheduling of batch operations - II. Computational issues." *Computers and Chemical Engineering* 17: (1993a) 229–244.

———. "Optimal periodic scheduling of multipurpose batch plants." *Annals of Operations Research* 42: (1993b) 193–228.

Shapiro, J. F. *Modeling The Supply Chain*. New York, USA: International Student Edition, Duxbury, 2007.

Shobrys, D. E. *The history of APS*. Report of the Supply Chain Consultants available at http://www.thesupplychain.com/, 2001.

Simchi-Levi, D., P. Kaminsky, & E. Simchi-Levi. *Managing the supply chain: The definitive guide for the business professional*. New York, USA: McGraw-Hill, 2004.

Soman, C. A., D. P. Van Donk, & G. J. C. Gaalmann. "A basic period approach to the economic lot scheduling problem with shelf life considerations." *International Journal of Production Research* 42: (2004) 1677–1689.

Stefanis, S. K., A. G. Livingston, & E. N. Pistikopoulos. "Environmental impact considerations in the optimal design and scheduling of batch processes." *Computers and Chemical Engineering* 21, no. 10: (1997) 1073–1094.

Subbiah, S., & S. Engell. "Short-term scheduling of multi-product batch plants with sequence-dependent changeovers using timed automata models." In *20th European Symposium on Computer Aided Process Engineering*, edited by S. Pierucci, & G. Buzzi Ferraris, Elsevier, 2010, Vol. 28 of *Computer Aided Chemical Engineering*, 1201–1206.

Sundaramoorthy, A., & I. A. Karimi. "A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants." *Chemical Engineering Science* 60, no. 10: (2005) 2679–2702.

Sundaramoorthy, A., & C. T. Maravelias. "Modeling of storage in batching and scheduling of multistage processes." *Industrial and Engineering Chemistry Research* 47, no. 17: (2008a) 6648–6660.

———. "Simultaneous batching and scheduling in multistage multiproduct processes." *Industrial and Engineering Chemistry Research* 47, no. 5: (2008b) 1546–1555.

Sundaramoorthy, A., C. T. Maravelias, & P. Prasad. "Scheduling of multistage batch processes under utility constraints." *Industrial and Engineering Chemistry Research* 48, no. 13: (2009) 6050–6058.

Sung, C., & C. T. Maravelias. "An attainable region approach for production planning of multi-product processes." *AIChE Journal* 53, no. 5: (2007) 1298–1315.

———. "A mixed-integer programming formulation for the general capacitated lot-sizing problem." *Computers and Chemical Engineering* 32, no. 1-2: (2008) 244–259.

Takahashi, K., & H. Fujii. "New concept for batchwise specialty chemicals production plant." *Instrumentation and Control Engineering* 1, no. 2: (1991) 19–22.

The Mathworks, Inc. *MATLAB: The Language of Technical Computing: MATLAB Notebook User's Guide*. Natick, MA, USA: The Mathworks Inc., 1998.

Timpe, C. H., & J. Kallrath. "Optimal planning in large multi-site production networks." *European Journal of Operational Research* 126: (2000) 422–435.

Tsiakis, P., & L. G. Papageorgiou. "Optimal production allocation and distribution supply chain networks." *International Journal of Production Economics* 111, no. 2: (2008) 468–483.

Van Dam, P., G. Gaalman, & G. Sierksma. "Scheduling of packaging lines in the process industry: An emplirical investigation." *International Journal of Production Economics* 30-31: (1993) 579–589.

Van Hentenryck, P. *Constraint satisfaction in logic programming*. Cambridge, MA, USA: MIT PRESS, 1989.

———. "Constraint and integer programming in OPL." *INFORMS Journal on Computing* 14, no. 4: (2002) 345–372.

Venditti, L., D. Pacciarelli, & C. Meloni. "A tabu search algorithm for scheduling pharmaceutical packaging operations." *European Journal of Operational Research* 202: (2010) 538–546.

Verderame, P. M., & C. A. Floudas. "Operational planning framework for multisite production and distribution networks." *Computers and Chemical Engineering* 33: (2009) 1036–1050.

Wang, K., T. Löhl, M. Stobbe, & S. Engell. "A genetic algorithm for online-scheduling of a multiproduct polymer batch plant." *Computers and Chemical Engineering* 24, no. 2-7: (2000) 393–400.

Werner, F., & A. Winkler. "Insertion techniques for the heuristic solution of the job shop problem." *Discrete Applied Mathematics* 58: (1995) 191–211.

Wilkinson, S. J., A. Cortier, N. Shah, & C. C. Pantelides. "Integrated production and distribution scheduling on a european-wide basis." *Computers and Chemical Engineering* S20: (1996) S1275–S1280.

Wilkinson, S. J., N. Shah, & C.C. Pantelides. "Aggregate modelling of multipurpose plant operation." *Computers and Chemical Engineering* S19: (1995) S583–S588.

Wolsey, L. A. *Integer Programming*. Series in Discrete Mathematics and Optimization. New York, USA: Wiley-Interscience, 1998.

Wu, J., & X. He. "A new model for scheduling of batch process with mixed intermediate storage policies." *Journal of the Chinese Institute of Chemical Engineers* 35: (2004) 381–387.

Xia, Q., & S. Macchietto. "Routing, scheduling and product mix optimization by minimax algebra." *Chemical Engineering Research and Design* 72: (1994) 408–414.

Yee, K. L., & N. Shah. "Scheduling of fast-moving consumer goods plants." *Journal of the Operational Research Society* 48: (1997) 1201–1214.

———. "Improving the efficiency of discrete-Time scheduling formulations." *Computers and Chemical Engineering* S22: (1998) S403–S410.

Zhang, X., & R. W. H. Sargent. "The optimal operation of mixed production facilities - a general formulation and some approaches for the solution." *Computers and Chemical Engineering* 20: (1996) 897–904.

Zweben, M., & M. S. Fox. *Intelligent scheduling*. San Francisco, USA: Morgan Kaufmann, 1994.

# Author Index