Optimization of processes in textile industry: models and solution methods

Victor Claudio Bento de Camargo

Optimization of processes in textile industry: models and solution methods[§]

Victor Claudio Bento de Camargo

Advisor: **Profa. Dra. Franklina Maria Bragion de Toledo**
Advisor: **Prof. Dr. Bernardo Sobrinho Simões de Almada Lobo**

Doctoral dissertation submitted to the *Instituto de Ciências Matemáticas e de Computação* - ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*.

USP – São Carlos
November 2012

[§] Otimização de processos na indústria têxtil – modelos e métodos de solução
This work was financial supported by FAPESP (grant 2008/09953-2)

Ficha catalográfica preparada pela Seção de Tratamento
da Informação da Biblioteca Prof. Achille Bassi – ICMC/USP

Camargo, Victor Claudio Bento

C172o Optimization of processes in textile industry: models and solution methods / Victor Claudio Bento Camargo; orientadora Franklina Maria Bragion Toledo; orientador Bernardo Almada-Lobo. -- São Carlos, 2012.

117 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2012.

1. Planejamento da produção. 2. Dimensionamento e sequenciamento de lotes. 3. Programação inteira mista. 4. Matheurística. 5. Indústria têxtil. I. Toledo, Franklina Maria Bragion, orient. II. Almada-Lobo, Bernardo, orient. III. Título.

Victor Claudio Bento de Camargo

Optimization of processes in textile industry: models and solution methods

Submitted to Faculdade de Engenharia da Universidade do Porto in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Industrial Engineering and Management, supervised by Bernardo Sobrinho Simões de Almada-Lobo, Auxiliar Professor of Faculdade de Engenharia da Universidade do Porto and by Franklina Maria Bragion de Toledo, Assistant Professor of Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo.

Department of Industrial Engineering and Management
Faculdade de Engenharia da Universidade do Porto
2012

“Para ser grande, sê inteiro: nada
Teu exagera ou exclui.
Sê todo em cada coisa. Põe quanto és
No mínimo que fazes.
Assim em cada lago a lua toda
Brilha, porque alta vive”.

Ricardo Reis (Fernando Pessoa)

Acknowledgments

This thesis marks the end of a considerable academic effort and it also represents a personal success. It would be very selfish to say that this thesis has been the product of my endeavor only, I was never truly alone. I am very grateful to many individuals, for all their support, inspiration and understanding along the way, especially when one is willing to let go of everything.

I wish to thank my family for the support that they have given me. To my brother, André, and my sister, Thais, I love you! I have been away from them for many years and have missed many important moments of their lives, which I deeply regret. I will be eternally grateful to you for your understanding of my absence, along with my bad-tempered nature. To my parents, I am grateful for everything they have done for me throughout my whole life. Unfortunately, my mother is not here with us to see the whole thing finished. But, I am sure she is in someplace out there, always watching us closely. Mum, I dedicate this thesis to you.

I am tremendously thankful for all of the Rossignolo family. A family that received me as a member. I am especially indebted to Kadu, my brother's friend, for the support and strength he has given me. Thanks for sharing your expertise on the spinning production system and for the data that have largely enriched this project. It is not easy at all to put the importance of your help down on paper. Thank you very much for everything.

I would like to deeply thank to my Brazilian advisor, Franklina, and to my Portuguese advisor, Bernardo, for believing in me, their time and great help throughout the course of my studies. Thank you, Fran and Bernardo, you have been wonderful academic advisors and friends to me! I would also like to extend my thanks to other professors who were fundamental to my education. Special thanks also to my undergraduate professor Neide Bertoldi Franco, who pushed me to the operations research.

Thanks to my friends and colleagues that I had the pleasure to meet in Brazil, Portugal and around the world. Thank you all for the fellowship, the chats, the lunches, the dinners, the travels, the drinks and laughs. A GIANT thank you to all my true friends! I won't list, but you know who you are! I would be just another unhappy lonely PhD student without all of you.

Finally, I confess that this research would not have been (full time) possible without the financial support of FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo (grant 2008/09953-2) in Brazil and Erasmus Mundus External Cooperation Windows Programme in my period in Portugal. I also must express my gratitude to the Institute of Mathematics and Computer Sciences of the University of São Paulo (ICMC-USP) and the Faculty of Engineering of the University of Porto (FEUP).

Abstract

In the practice of a spinning industry, the operational decisions of the production planning are determined by the hierarchical solution of the lot-sizing and scheduling problem and the blending problem of the cotton bales. The tasks are: to define the size, sequence, timing and allocation of each production lot and to select which cotton bales are used for production. Each of these problems represents a large challenge in planning the production. However, in order to better represent the production environment and to reach lower production costs, process industries (as the spinning industry) are integrating more and more of the production sub-problems into the planning. The aim of this thesis is to propose novel mathematical models and solution methods to assist the decision maker to plan the production at the operational level. Three formulations for the synchronized two-stage lot sizing and scheduling are proposed. A new method based on mathematical programming and metaheuristics is also developed to solve this sub-problem. In addition, the integration of the lot sizing and scheduling with decisions related to the raw materials (cotton bales) is analyzed. The novel models represent a more realistic lot sizing and scheduling for the spinning industry and process industries of similar production environment. The solution method finds good solutions to the mentioned problem and outperforms other state-of-the-art methods incorporated in commercial softwares. Moreover, the method is general enough to solve other optimization problems. The integrated lot-sizing, scheduling and blending prove that constraints related to the yarn quality influence the costs and the feasibility of the production planning. The integrated planning of these operations approaches the system considering the constraint relationship and defines more realistic production plans.

Resumo

As decisões operacionais de produção em uma indústria de fiação são planejadas na prática determinando soluções dos sub-problemas de dimensionamento e sequenciamento de lotes e da mistura de fardos de algodão. As tarefas são: definir o tamanho, a sequência, o tempo e alocação de cada lote de produção e quais fardos de algodão devem ser utilizados na produção. Por si só, os sub-problemas representam grandes desafios no planejamento da produção. Entretanto, para melhor representar o ambiente produtivo e alcançar custos de produção mais baixos, indústrias de processo, como as de fiação, procuram integrar mais e mais seus sub-problemas de planejamento. O objetivo dessa tese é apresentar modelos matemáticos e métodos de solução para auxiliar a tomada de decisão no nível operacional do planejamento da produção. Três formulações matemáticas para o dimensionamento e sequenciamento de lotes em um sistema de dois estágios com produção sincronizada são propostas. Um novo método baseado em programação matemática e metaheurísticas é também desenvolvida para a solução desse sub-problema. Além disso, a integração das decisões relativas à matéria-prima (fardos de algodão) ao dimensionamento e sequenciamento de lotes é analisada. As novas formulações propostas representam de forma mais realista o problema de dimensionamento e sequenciamento de lotes da indústria de fiação e de indústrias de processo com ambiente produtivo similares. O método de solução encontra boas soluções para o problema e supera outros métodos similares presentes em *softwares* comerciais. Além disso, o método é geral o suficiente para a solução de outros problemas de otimização. O problema integrado de dimensionamento e sequenciamento de lotes e mistura comprovou que restrições relativas à qualidade dos fios influenciam os custos e viabilidade do planejamento da produção. O planejamento integrado dessas operações trata o sistema considerando restrições que se relacionam, definindo planos de produção mais realistas.

Contents

List of Figures

Chapter 1

Prologue

The spinning industries are capital intensive process businesses. Single-purpose installations are used to blend the raw cotton and to spin the yarns. Quality constraints occur in the synchronization between the cotton blending and the produced yarns. Moreover, a set of cotton bales must be selected to produce the blending that meets the quality specifications of the yarn produced. However, large investments in the installations are insufficient to ensure the quality of the produced yarns. Raw materials, production processes and their management are important aspects on the final product. Due to the little or no flexibility in the capacity usage, hard technological constraints and market characteristics, spinning industries suffer from various bottlenecks in the production efficiency. Nevertheless, an operative production planning can improve the performance of these companies.

The production planning problem can be divided into three hierarchical sub-problems: strategic, tactical and operational (Anthony, 1965). The time horizon in which they perform distinguishes each sub-problem. Strategic problem handles the time horizon of more than one year, that is, long term decisions. Tactical problem deals with few to several months, the medium term decisions. Of short term, the operational problem refers to decision of weeks, days or shifts.

Lot sizing and scheduling is one of the most important problems in planning the production. The aim is to determine the level, timing and allocation of each production lot in order to satisfy customer demand. The modeling of this production problem intends to determine plans by a systematic method with minimal costs. In addition to the common lot sizing and scheduling sub-problem, spinning industries gather raw material constraints. Usually, the classical blending problem defines the raw cotton usage in a production plan. These aforementioned sub-problems deal with short term decisions.

The research in production process modeling is incorporating more and more specificities of the industrial processes. The production planning representation becomes more realistic and able to improve the flexibility of the production operations and the productivity itself. Although the production planning be divided in subproblems and considered adequate in practice, not all of the technological constraints are taken into account. Thus, solutions are often unfeasible. An integrated view of the system should consider the relationship of the main requirements to design more realistic plans and to reduce global production costs. Moreover, a systemic method gives flexibility and contributes to raise the production efficiency of this important industrial sector.

The topic of this thesis is the optimization of processes in textile industries, in particular, the production planning process in the spinning industry. From a practice point of view, the subject matter assists the decision maker to accelerate the planning and to analyze different decision scenarios. From the academic perspective, this study can enrich the literature state-of-the-art and indicate directions to model and solve the planning problem in this kind of production environment.

1.1 Context

In this section, some specificities of the process industries are presented as recognized in the literature. Specifically, the focus turns to the spinning industry. Being part of the textile supply chain, the industrial environment of this particular process industry is discussed.

1.1.1 Process industries

Features and characteristics that define the process industry are exposed in some papers, such as (Fransoo, 1993), (Crama *et al.*, 2001) and (Kallrath, 2002). The definition of the process industry by the American Production and Inventory Control Society (APICS) is frequently referred. According to this definition, the process industries are "businesses that add value by mixing, separating, forming or chemical reactions. Processes may be either continuous or batch and usually require rigid process control and high capital investment" (Fransoo and Rutten, 1994). Businesses of textile goods, beverages, food, paper, chemicals, crude oil, rubber and plastic goods, metal alloys, pottery and glass are representative of this type of industry.

Production flow can be differentiated between batch or continuous processes. Industries with combined batch and continuous processes also exist and the flow is called semi-batch. The production flow type is commonly linked to the variety of produced items. Processes of continuous flow produce a reduced range of products each in a rather high volume. On the other hand, batch operation produces small quantities of a larger range. Not only, the range of different products is limited, but also the variety between products is relatively small. Consequently, the processing routes are quite similar for all products. In general, the added value of the products is significantly low. The wide part of the added value is performed by the processes that constitute the bottlenecks in the production system. These processes are considered dominant in the production plan and have a high-price installation.

The small number of production processes combined with low added value and low processing rates makes the raw materials playing a very important routine in production management activities for process industries. Its costs usually account for 60-90 per cent of the sale price (Crama *et al.*, 2001). The raw material for process industries is often provided by mining or agricultural industries. It can also be classified in main or secondary materials. In a textile industry, the fibers are the main materials, whereas, bobbins and tubes are secondary materials. The secondary materials are usually handled with the traditional bill of material (BOM) techniques and main materials as recipes. Recipes can be proceeded in a fixed set of ingredients or some attributes of the ingredients have values that characterize the final product. One can note that an important feature of recipes is that different ingredients or values can be used to obtain a final product. This feature admits the adding of constraints related to the costs and availability of the required ingredients for planning of the final product. The recipe design can be achieved by the well-known diet model, also called blending problem. By determining the blending, the proportion of each ingredient must be determined to achieve the recipe. Constraints related to the quality specifications of the final products and the ingredient availability must also be respected. The aims can be various: cheapest blending, blends that differ the minimum as possible between themselves, among others.

Recipes are described based on only some of the (essential) attributes of raw materials. Moreover, they represent product families, that is, a set of final products with the same characteristics of the essential attributes. Although the family variety is limited, the recipe management is vital for the production of final products. This fact corroborates with the representation of the raw material in the sale cost of the final product that can reach 90%, as mentioned before. Its management becomes a crucial but complex task for process industries. The problem affects the planning of the raw materials distribution for all the industrial environment respecting the availability to meet demand and quality of the recipes, as well as, the final products. In case the recipe quality fails, it may cause customer dissatisfaction, reorder and recycle. However, the control of the raw material inventory can minimize these issues.

The failures in achieving the quality specifications of the blends might be caused by the methodology to determine the ingredients, the absence of some ingredient in the inventory, among other factors.

Variations on the attribute values are intrinsic of the ingredients as the source of these raw materials is the nature. In a textile industry, the raw cotton presents different fiber lengths, which is an uncontrolled attribute. Therefore, the quality of the raw material delivered by suppliers is often not in accordance to the quality specified by the purchasing order. The difficulty to control the supply flows of raw materials causes high variation of the raw material inventory. So, process industries give often high importance for managing the main raw materials. Variations in raw material quality in the inventory may force to change the recipe proportions. In the spinning industry, variations in the color of the cotton bales in stock may impose variations in the blend proportions to fulfill the yarn specifications.

Moreover, process industries usually follow a hybrid pull and push production strategy. Although product demand is often high, it shows up a seasonality pattern. Thus, in order to reduce the machine idleness in some periods and to avoid the failure to fulfill high demand in other periods, process industries have adopted a hybrid pull and push system, that is, by mixing make-to-stock and make-to-order production strategies.

The characteristics related to technological constraints, production process, market particularity and in special the raw material importance are frequently mentioned to define the process industries.

1.1.2 Spinning industry

Spinning is a process industry part of the textile supply chain. The textile chain is composed by seven fundamental steps: farming; ginning; spinning; knitting; dyeing and finishing; cutting and sewing; and packaging and shipping. From the fields, the original cotton mass is ginned and the lint cotton is packed in pressed bales. Cotton bales are raw materials for the spinning in which the lints are processed to produce various types of yarns. Knitting industries work the yarns to produce crude cloth. Then, the cloth is dyed and finished. Finally, cutting and sewing industries produce dressing and other textile products. The shipping is presented in all the steps.

The textile supply chain is very important for the economy as it generates employment and increases regional development. For example, textile goods represented 10% of the Portuguese exportations in 2010 (ATP, 2012). In Brazil, there are currently 1.6 million employees in textile industries and spinning factories employ about 80,000 workers (ABIT, 2011). According to the annual report of the textile sector, Brazil plays a major role in the textile and apparel international market as it had a revenue of US$ 41.8 billion in 2010.

The chain is a mixed push and pull and the frontier is frequently viewed on the spinning industries. As cotton bales have quality variability controlled by the nature, ginning industries usually fail to fulfill the quality required of the orders. However, spinning industries must process any raw material to satisfy the knitting orders of hard quality requirements. Thus, due to the hard relationship between the raw material quality and the final products, the spinning production planning should take into account the raw material used to meet the yarn qualities.

The production system in a spinning industry can be defined as a procedure where fibers are processed and then used to make different types of yarns. An example of an industrial plant is viewed in Figure 1.1. In the beginning (level A), lints from various bales are mixed and blended together in order to make a uniform blend of fiber properties. The fiber blend is blown by air from a feeder through ducts to intermediate machines (level B) for cleaning and carding, which separate and align the fibers into a thin web. The web of fibers is then pulled through a conical device (level C), providing a soft called sliver. The sliver is pulled again and then twisted to make it tighter and thinner until it complies with the yarn specifications (level D). After spinning, the yarns are tightly wound around bobbins or tubes. The yarn packages are then ready for distribution.

The spinning process distinguishes among the type of spinning, such as open-end, ring, air jet and Vortex spinning. In open-end spinning, the yarn is produced directly from the sliver, and consequently, the drawing process presented in ring spinning is eliminated. The air jet and Vortex spinning systems have also eliminated the need for the drawing process.

As previously advertised, raw material is an important issue while planning the production. The spinning machines (level D), on which the yarns are produced, are supplied with a blend of fibers in sliver form by the opening-blending machine (level A) and the intermediate levels. The blending must comply with the quality specifications that support the yarn production. The specifications are usually defined by company policies and refer to cotton attributes. In this moment, one can see two situations related to the raw material: first, the production plan requires the synchronization between the blending and yarn production; second, the right selection of the cotton bales to ensure the quality specifications for production. In summary, in planning the production in the spinning industry, three problems must be taken into account: the lot sizing, the scheduling and the blending.

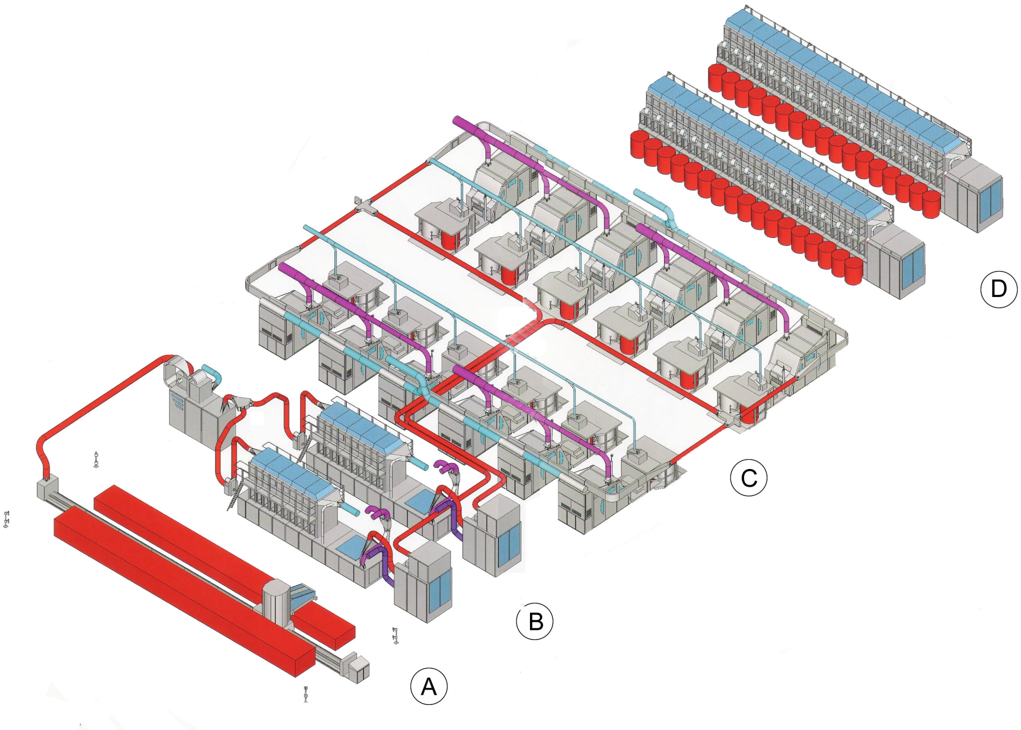

Figure 1.1: Illustrative plant of an open-end spinning adapted from (Trützschler, 2005).

1.2 Research objectives

This thesis addresses three main research objectives. The first objective is to study the production planning problem (specifically, the lot sizing and scheduling) in the spinning industry. The investigation can be done with a general view on the process industries, that include the spinning industry. The challenge is to model such problem concerning the representation of the continuous/batch production in a two-stage environment and the determination of the production lots that fulfill the demand. Moreover, the blending production needs to be synchronized in order to supply all the spinning machines in a feasible manner. We intend to propose new models that are more realistic by addressing scheduling features but maintaining the model general enough to approach the production planning in other similar production environments. Modeling development aims to assist the decision making. This approach also systematizes and allows for scenario simulations. Besides attempting to reduce costs, an analysis on the simulations can identify bottleneck processes or idle machines.

The second objective aims to define a powerful solution method to tackle the aforementioned lot-sizing and scheduling problem. We intend to give new insights into the current literature by developing a novel solution technique. Moreover, the contribution also relies on the matheuristics field as our purpose combines an exact solution method with a metaheuristic (Maniezzo *et al.*, 2010). The method should have more ability to cope with lot-sizing and scheduling problem than the traditional methods present in commercial softwares. Nevertheless, the method should be general enough to solve other problems not inked to this research subject.

Third, the integration value of the lot-sizing, scheduling and blending problems in the spinning industry will be investigated. As the raw material is an important routine in process industries, the cotton blending management can influence the production planning. Mathematical models will be provided to represent the integrated problem. A further qualitative analysis will bring some feelings about the integration of the problems and of the isolated approach.

We call the attention to the scarce research on the production planning in the spinning industry. Thus, we intend to give contributions to enrich the state of the art in this direction.

1.3 Outline

This thesis is organized as follows. The Prologue gives the thesis motivation, and context about the main themes and research objectives. Chapter 2 provides a somewhat self-contained introduction to the production planning in process industries. A general two-stage lot sizing and scheduling presented in many process industries is introduced. Three time-based scale formulations to represent the lot-sizing and scheduling problem in this industrial environment are developed. We outline the differences and similarities of the solutions generated by each model. Test results are reported for a range of instances. Remarks and outlook for further research related to the production planning models for process industries are addressed.

Chapter 3 studies the production planning problem in the spinning industry. We develop a new matheuristic as a solution method. A pseudo code of the method is provided together with some tools to accelerate the search for good solutions. Computational results concerning the comparison of the novel solution method against other matheuristics implemented in commercial softwares are also shown for its validation.

Chapter 4 provides an analysis of the value for combining the lot-sizing and scheduling decisions with the blending decisions. An integrated mathematical model is then provided. The validation is supported by the comparison to the hierarchical approach in which the lot-sizing and scheduling decisions are first defined in the upper level. In the bottom level, blending decisions are determined based on the lot-sizing and scheduling solution. A third approach that combines features from the integrated and hierarchical techniques is then developed.

Finally, Chapter 5 summarizes the contributions of this research. Moreover, some concluding remarks and future research directions are presented.

Chapter 2

The lot-sizing and scheduling problem in process industries

In this chapter, we propose three novel mathematical models for the two-stage lot-sizing and scheduling problem presents in many process industries. The problem shares a continuous or quasi-continuous production feature upstream and a discrete manufacturing feature downstream which must be synchronized. Different time-based scale representation are discussed. The first formulation encompasses a discrete-time representation. The second one is a hybrid continuous-discrete model. The last formulation is based on a continuous-time model representation. Computational tests with state-of-the-art MIP solver show that the discrete-time representation provides better feasible solutions in short running time. On the other hand, the hybrid model achieves better solutions for longer computational times and was able to prove optimality more often. The continuous-type model is the most flexible of the three for incorporating additional operational requirements, at a cost of having the worst computational performance. This chapter is strongly based on (Camargo *et al.*, 2012b).

2.1 Introduction

The production planning process addresses the development of a plan of production actions to transform raw material into final products. Involving diverse decisions, from the acquisition of resources and raw materials to the human resource management, it tries to meet customer satisfaction and industrial environment constraints in the most efficient and economical possible way. The production planning problem is typically of operational to tactical fashion, and is considered one of the most challenging subjects for industry (Drexl and Kimms, 1997). In essence, the major problems of operational production planning are the sizing and scheduling of the lots. On the one hand, the lot-sizing problem determines the timing and level of production to satisfy product demands over a time horizon. Setup, inventory and capacity related constraints must be respected at minimum cost. On the other hand, the scheduling problem establishes the sequencing and allocation of the lots. The integration of both problems in the same level aims to define better production plans than those generated from a hierarchical planning system in which the lot sizing is solved a priori and feeds the scheduling level. Typically, the key cost indicators include production, holding and setup costs.

In order to reflect all the possible specificities of industrial processes (e.g. the possibility of a machine configuration be carried from a period to the next or the requirement that all machines must produce similar items at the same time point, etc.), the research community is incorporating more and more scheduling decisions and features within lot-sizing models (Jans, 2009). This fact makes the production planning task more sophisticated and able to improve the flexibility of the production operations and the productivity itself.

In this chapter, we address the lot sizing and scheduling appearing at a two-stage production process that is common to a diverse range of industries. The first stage encompasses the continuous production of a common resource that is necessary for the discrete manufacturing on a set of parallel machines that occurs in the second stage. In each point in time, all machines process products of the same family, i.e. products of the same common resource. For instance, the beverage production planning problem (e.g. Ferreira *et al.* (2009)) clearly shows a two-stage production process, where the syrup is prepared in tanks in the first stage, and it is distributed to parallel bottling machines in the second stage; the production processes of the glass container industry and of the automated foundry industry (see by Almada-Lobo *et al.* (2008) and Santos-Meza *et al.* (2002), respectively), entail a furnace in the first stage supplying the fluid mass (glass or alloy) to a set of moulding machines; the manufacturing system of the spinning industry (e.g. Camargo *et al.* (2012a)) processes fibers (of different fiber blends) in a first step that are used downstream to produce different types of yarns on parallel machines. Beyond the similarity of the two-stage environment, these real-world problems share a continuous or quasi-continuous production feature upstream (in the first stage), and a discrete manufacturing feature downstream. In fact, the syrup, glass, alloy and fibers are usually handled as non-discrete products (expressed in liters, tonnes, etc), but the final products are discretizable in units (e.g. number of bottles, items, yarn packages, to mention a few). It is important to say that these real world problems also have their technological constraints.

All the aforementioned problems were formulated by mixed-integer programming. Nevertheless, despite the improvements seen in mathematical programming in the last 20 years in computational power (hardware) and in the quality of general purpose mixed-integer programming commercial solvers (software), the usage of adequate or tighter mathematical formulations are still mandatory to reduce the running times needed to solve them, and more important, to optimize more difficult instances of lot-sizing and scheduling problems of increasing sizes with special problem features found in practice. Despite of the continuous production nature of the first stage in the real-world problems mentioned before, the requirement to fulfill the product demands on given due dates may have induced the authors to deal with these two-stage lot-sizing and scheduling problems with discrete time models. Generally speaking, the choice of an appropriate time scale (either discrete or continuous) for production planning depends heavily on the inside dynamics of the production system that is planned for, as well as on the outside dynamics of the world the production system is embedded into (see Suerie (2005)). The differences between discrete time and continuous time representations for scheduling problems are described in Floudas and Lin (2004). But, this definition can be extended to the integrated lot-sizing and scheduling problem.

In discrete time representation, the real-word decisions and events that occur in continuous time, have to be translated into decisions and events occurring according to the discrete time scale. Here, the planning horizon is divided into a number of time periods (in some models, the periods are subdivided in micro-periods) and the beginning and ending of production lots are confined to the grid of the time periods. Moreover, external demand is assumed to take place at the end of each time bucket. There are two main types of models: large-bucket and small-bucket. In the former, the planning horizon is partitioned into a small number of lengthy time periods, and several products/setups may be produced/performed per period and machine. Such a period typically represents a time slot of one week or of one month. The capacitated lot-sizing problem (CLSP) is an example of large bucket problem (Drexl and Kimms, 1997), and it can be seen as an extension of the Wagner-Whitin model (Wagner and Whitin, 1958), to take into account capacity constraints. The standard CLSP does not schedule products within a period, and it has been extended over the years in several directions. For instance, the capacitated lot-sizing and scheduling problem (CLSD) includes sequencing decisions (Haase (1996) and Almada-Lobo *et al.* (2007)). On the other hand, in small-bucket models, the planning horizon is divided into many short periods (such as days, shifts or hours) – usually referred as micro-periods, in which at most one setup may be performed. Therefore, depending on the models, we are limited to producing at most one or two items per period. Such models are useful for developing short-term production schedules. Lot-sizing and scheduling decisions are taken simultaneously, as here a lot consists in the production of the same product over one or more consecutive micro-periods. This is the case of discrete lot-sizing and scheduling problem (DLSP), continuous setup lot-sizing problem (CSLP) and proportional setup lot-sizing problem (PLSP). In the DLSP, only one item can be produced in each micro-period and each machine either produces at full capacity or is idle (known as "all-or-nothing" production) – see Fleischmann (1990). The CSLP, presented by Karmarkar and Schrage (1985), is more flexible than the DLSP, relaxing the discrete production policy, as here lot sizes are continuous quantities up to capacity. By relaxing the "all-or-nothing" assumption of the DLSP, the CSLP wastes capacity in case a period capacity is not fully used. The PLSP (Drexl and Haase, 1995) is an attempt to avoid this drawback, by scheduling a second item in a period to use its remaining capacity. As other small-time bucket models, at most one setup may occur within a period. However, contrarily to the DLSP and the CSLP in which setups are performed at the beginning of a period, here the setup may take place at any point in time. A criticism to small-bucket models is that for real-world instances they require a prohibitive number of periods, especially if mathematical programming approaches are to be implemented. As opposed to the aforementioned models, the general lot-sizing and scheduling problem (GLSP), first proposed by Fleischmann and Meyr (1997), makes use of a two-level time structure to be more flexible. The planning horizon is divided into large buckets (also denoted as macro-periods), with a given length. Each macro-period is partitioned into a fixed number of non-overlapping micro-periods with variable length. The works (Meyr, 2000) and (Meyr, 2002) extend the standard GLSP to cope with sequence-dependent setup times and a parallel machine environment.

Nevertheless, the bucket-orientation of discrete time models raises some obstacles. The main one is to address continuous time settings, such as the need for synchronization of resources or for events that may cross over time boundaries (as setups or production lots) - see (Almeder and Almada-Lobo, 2011). The discrete-time model becomes more inadequate as the time scale of an individual production event increases in relation to the time scale of a planning period. In contrast to the discrete time representation, in the continuous time representation the production lots take place at any point in the planning horizon. It allows a continuous process starting and ending anywhere within the planning horizon, not only within the time period boundaries, overcoming this discrete time scheme limitation. Instead of being time-partitioned, the planning horizon is divided into a number of events (event-orientation). The batch sequencing problem (BSP, presented in Jordan and Drexl (1998)) is illustrated as a continuous time model equivalent to the discrete-time DLSP. The production lots are represented by jobs that need to be scheduled. In order to determine the lots and save setup costs, the consecutive jobs that produce the same product are grouped together to form one lot. Production problems adopting a continuous time representation are also found in papers addressing the chemical industries. An interesting study was developed by Karimi and McDonald (1997) that proposed two models to the single stage planning and scheduling of semi-continuous processes. The first is a pure-continuous model, whilst the second is a hybrid one.

However, continuous time representations often suffer from large integrality gaps due to poor linear programming (LP) relaxation (see Suerie (2005)). This stems from the fact that start (and end) times of tasks need to be coupled with time points, which is done by so-called big-M constraints. We highlight that no effective lower bound does not mean that a certain mathematical programming approach is less efficient on this type of models. In addition, inventory holding costs cannot be accounted for in continuous time model formulations easily. Hybridizing discrete and continuous time-oriented models is a possible direction to reap the maximum benefits from both types.

The goal of this chapter is to present new formulations for the two-stage semi-continuous lot-sizing and scheduling problem. The production environment consists of parallel machines that produce products sharing an upstream common resource. We develop three mathematical formulations for this problem based either on pure discrete-time and continuous-time representations, or on hybrid continuous-discrete time representation. First, we propose a formulation based on the GLSP considering a fixed time grid for both production stages. Second, we model the problem with a hybrid continuous-discrete model based on the CLSD. Here, the production slots may be of arbitrary lengths and depend on the time period and on the availability of the common resource. Last, we propose a continuous-time model in which production slots are of arbitrary lengths and independent of the time periods. The strengths and weaknesses of each model are analyzed. In the next section we introduce the problem definition. Sections 2.3, 2.4 and 2.5 present the three new model formulations. In Section 2.6, we show an illustrative example and outline the differences and similarities of the solutions generated by each model. A procedure to calculate a lower bound for some parameters required in the formulations is developed in Section 2.7. Computational results are reported in Section 2.8 for a range of instances. Conclusion and outlook for further research are addressed in Section 2.9.

2.2 Problem definition

In this study we consider a two-stage production environment. At the first stage, we plan the production of a common resource that is necessary for the discrete manufacturing of the second stage, for which we need to plan the production of several items on parallel machines. The common resource is produced either in batches or continuously. Figure 2.1 depicts the production flow, i.e., the relationship between the common resource and the final products. Each final product can only be produced in case the intermediate product manufactured in the first stage has the required attributes. For example, in the foundry industry, one item can be produced only if the brittleness and corrosion resistance required for it have attributes of the alloy melted in the first stage.

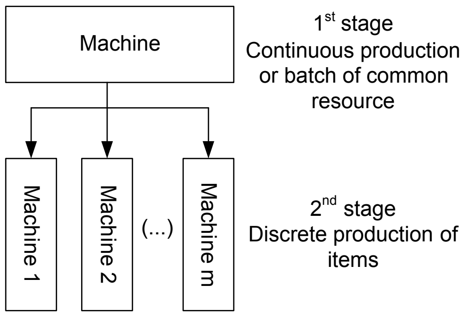

Figure 2.1: Illustrative two-stage production environment.

The underlying problem is considered as a two-stage lot sizing and scheduling. The production of the common resource in the first stage must be synchronized with the production process of the second stage. Without loss of generality, we consider for the first stage one machine and for the second stage, a set of parallel machines. The manufacturing system operates on continuous time. In order to model the problem, consider N final products to be scheduled on M parallel machines over a finite planning horizon split in T periods. Parallel machines are not unrelated as they are fed by a machine that produces the common resource. The product demands over each period are known and should be met on time without backlogging. A common resource batch (also called campaign) is generally used for the production of different products that compose a product family. Nevertheless, a product is only made using a specific resource batch with appropriated attributes to its product family. Therefore, one product cannot be manufactured in a given time period unless a required resource is available in that period. In each point in time, only one common resource batch can be processed upstream, thus all the machines should manufacture products that require the same resource. We consider the consumption, by the downstream stage, of semi-finished products from the upstream stage in the same unit of measurement, for example, liters, kilos, etc. But, it is always possible to consider different units of measurement since a factor can be used (on the produced quantities, the machine capacity or the product demands) to standardize them. Machines may have different processing rates to the same product; as a result, the common resource batch can be consumed in different speeds. So, the duration of common resource batches are not necessarily identical. Setup changeover from one product to another consumes machine capacity dependent of the sequence in which the products are processed. The setup can be carried from one period to the next. Without loss of generality, we consider null setup times and costs for the common resource, which is reasonable in some production environments. In Remark 1 of Section 2.3 we show how such feature can be incorporated in one of the models introduced in this chapter. A production plan defines the lot sizes and the setup sequence of both final products and common resources. Our objective is to find the plan that minimizes the sum of inventory and setup costs.

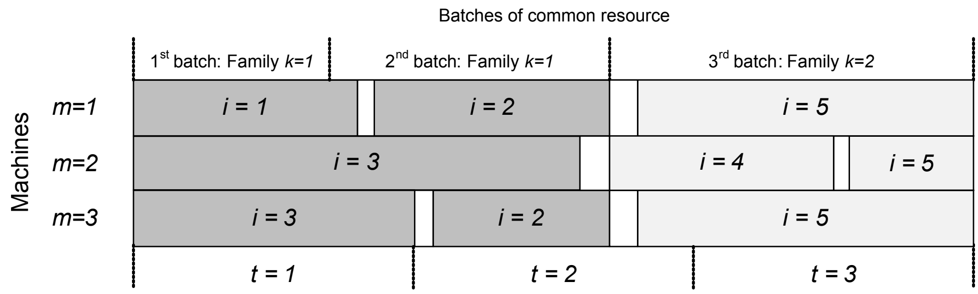

Figure 2.2: Illustrative example of a production plan.

Figure 2.2 represents an example of a production plan with one machine on the first stage and three parallel machines on the second, three periods and five demanded products. Products 1, 2 and 3 belong to the same family ($k = 1$), and products 4 and 5 form the second one. Note that the common resource batches are sequenced and they feed all parallel machines. Moreover, when two common resource batches required for the same family are sequenced consecutively, they can be viewed as only a bigger one (e.g., see the first and second batches of Figure 2.2). Since the machines operate on continuous time, the production lots are free to spread over periods, which is the case of product 3 on machine 2 in periods $t = 1$ and $t = 2$.

The great challenges of modeling such problem concerns on: 1) the representation of the continuous production; and 2) the determination of the production level that fulfills the demand in each time period. Furthermore, the determination of a production plan for the integrated lot sizing and scheduling problem with these characteristics is harder when the production of product families needs to be synchronized across all the machines. Moreover, the scheduling features of sequence-dependent setup times and setup carryover are also worth of mention.

The three models proposed in this chapter to represent this problem are detailed next. Afterwards, the models are compared and an illustrative example is used to detail the differences and similarities of the solutions.

2.3 MSGLSP - Multi-stage general lot-sizing and scheduling problem

Our first model is based on the general lot-sizing and scheduling problem (GLSP). As referenced before, the GLSP divides the planning horizon into T macro-periods. Each macro-period is sub-divided into a number of micro-periods in which only one product can be produced. Therefore, the sequence of production is obtained in a rather straightforward manner allowing the incorporation of sequence-dependent setups. In each macro-period we can have multiple setups, but retaining the small-bucket concept. The fundamental assumption is that a user-defined parameter restricts the number of micro-periods per period.

The first model makes use of the same micro-periods length across all machines (i.e., a fixed time grid). This feature allows for an easy synchronization between the first and second stages. Here, the maximum number of micro-periods (R) is fixed a priori for each period, however micro-periods have no pre-defined length. A micro-period is confined to be within a single period, and its length cannot exceed the length of its period, without loss of generality, we assume the length of each period to equal to one. This assumption enforces some parameters to be standardized accordingly.

An optimal solution involves the lot sizes and production schedule that respect the constraints with the minimum setup and holding costs. In order to present the model formulation clearly, notations are given as follows:

Indices	
$i = 1, \ldots, N$	products;
$k = 1, \ldots, K$	product families - that differ from each other on the common resource attributes;
$r = 1, \ldots, R$	micro-periods per period;
$m = 1, \ldots, M$	machines;
$t = 1, \ldots, T$	macro-periods.

Parameters	
h_{it}	holding cost for product i at the end of period t;
σ_{mij}	setup cost of a changeover on machine m from product i to j;
I_{i0}	initial inventory level of product i;
d_{it}	demand of the product i in period t;
s_{mij}	setup time of a changeover on machine m from product i to j;
p_{mi}	processing time on machine m of one unit of product i;
C	capacity (in units) of the common resource batch;
$S(k)$	set of products that belong to family k.

Variables	
I_{it}	inventory level of product i at the end of period t;
Z_{mijtr}	takes on 1, if there is a changeover on machine m from product i to product j in period t and micro-period r; 0 otherwise;
X_{mitr}	production quantity of product i on machine m in period t and micro-period r;
μ^s_{tr}	start time of micro-period r in period t;
μ^f_{tr}	end time of micro-period r in period t;
Y_{mitr}	takes on 1, if the machine m is set up for product i in period t and micro-period r; 0 otherwise;
U_{trk}	takes on 1, if the common resource processed in period t and micro-period r has attributes to attempt family k; 0 otherwise.

The formulation MSGLSP reads:

$$\textbf{Minimize} \quad \sum_{i=1}^{N}\sum_{t=1}^{T}(h_{it}\cdot I_{it}) + \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{r=1}^{R}(\sigma_{mij}\cdot Z_{mijtr}) \tag{2.1}$$

subject to

$$I_{i(t-1)} + \sum_{m=1}^{M}\sum_{r=1}^{R} X_{mitr} - I_{it} = d_{it} \quad \forall i,t \tag{2.2}$$

$$\mu^s_{t1} = t-1 \quad \forall t \tag{2.3}$$

$$\mu^f_{tR} = t \quad \forall t \tag{2.4}$$

$$\mu^s_{tr} \geq \mu^f_{t(r-1)} \quad \forall t, r>1 \tag{2.5}$$

$$\mu_{tr}^{f} - \mu_{tr}^{s} \geq \sum_{i=1}^{N}\sum_{j=1}^{N}(s_{mji} \cdot Z_{mjitr}) + \sum_{i=1}^{N} p_{mi} \cdot X_{mitr} \qquad \forall m, t, r \tag{2.6}$$

$$p_{mi} \cdot X_{mitr} \leq Y_{mitr} \qquad \forall m, i, t, r \tag{2.7}$$

$$Z_{mijtr} \geq Y_{mit(r-1)} + Y_{mjtr} - 1 \qquad \forall m, i, j, t, r > 1 \tag{2.8}$$

$$Z_{mijt1} \geq Y_{mi(t-1)R} + Y_{mjt1} - 1 \qquad \forall m, i, j, t > 1 \tag{2.9}$$

$$\sum_{i=1}^{N} Y_{mitr} = 1 \qquad \forall m, t, r \tag{2.10}$$

$$p_{mi} \cdot X_{mitr} \leq U_{trk} \qquad \forall m, i \in S(k), t, r, k \tag{2.11}$$

$$\sum_{k=1}^{K} U_{trk} \leq 1 \qquad \forall t, r \tag{2.12}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{N} X_{mitr} \leq C \cdot \sum_{k=1}^{K} U_{trk} \qquad \forall t, r \tag{2.13}$$

$$Y_{mitr} \in \{0,1\}\,;\; U_{trk} \in \{0,1\} \qquad \forall m, i, j, t, r, k \tag{2.14}$$

all other variables are non-negative and continuous. (2.15)

The objective function (2.1) attempts to minimize the sum of holding and sequence dependent setup costs. Constraints (2.2) represent the inventory balance equations. Constraints (2.3)-(2.6) deal with micro-periods. They define the start, length and end of each micro-period r. Constraints (2.3) block the starting time of the first micro-period to the beginning of the respective macro-period. Similarly, constraints (2.4) block the ending time of the last micro-period to the end of the respective macro-period. Note that, $\mu_{tR}^{f} - \mu_{t1}^{s} = 1$. Constraints (2.5) avoid the micro-periods overlapping. Constraints (2.6) define the lower bound of each micro-period length based on the production time and the setup time of all the machines. Note that (2.6) force a fixed grid of micro-periods across the machines. In this purpose, idle time intervals are allowed and are performed between periods (2.5) and intra periods (2.6).

The capacity limitation of the parallel machines is expressed implicitly. A machine works continuously during the period, thus p_{mi} can be understood as the fraction of the period consumed on machine m to produce one unit of product i. Constraints (2.3)-(2.6) ensure that $p_{mi} \cdot X_{mitr}$ respects this capacity limitation, i.e., the capacity machine, the processing rates and the setup times s_{mij} are normalized to 1. Constraints (2.7) ensure that production in a micro-period occurs only if the machine is set up for the respective product. Constraints (2.8) enforce a setup changeover if different products are produced in subsequent micro-periods within the same macro-period, whilst (2.9) perform the same function but linking two consecutive macro-periods. Requirements (2.10) establish that each machine is set up for one and only one product in a micro-period. Constraints (2.11) ensure that, in each micro-period, all the machines manufacture products of the same family. As only one machine is considered in the first stage, constraints (2.12) limit to at most 1 the number of families (or, in other words, the number of common resources of different attributes) to be produced per micro-period. Constraints (2.13) restrict the production in a micro-period by the maximum size of the common resource batch. Constraints (2.14) e (2.15) enforce the binary and non-negative requirements for different variables.

Remark 1. *We assumed in Section 2.2 that the changeover from one product family to another one in the first stage does not waste time. However, such feature can actually appear in some industrial environments. It can be represented in the MSGLSP by adding a binary variable λ_{trhk} that takes on 1, if a changeover occurs on batch of micro-period r in period t from product family h to k; 0 otherwise. The constraints (2.5) should be modified accordingly to represent the time wasted in a changeover of product families. In addition, a couple of constraints should be incorporated to enforce the setup based on the U_{trk} value. Moreover, setup costs for the first stage follows λ_{trhk} and could be added in the objective function. The same constraints could be easily extended to the other models.*

2.4 MSHLSP - Multi-stage hybrid lot-sizing and scheduling problem

The MSHLSP offers a hybrid continuous-discrete time representation. On one hand, the batches of common resource are free to start and end at any point of the planning horizon and are not attached to the period discretization. A maximum number of batches for the common resource can be used during the planning horizon, which is denoted by the parameter L. On the other hand, the production slots are confined to the common resource batches and bounded by the macro-period limits.

Unlike the MSGLSP, the MSHLSP determines, in the first stage, the sizes and sequence of the common resource batches and allows for production slots that are independent from one machine to another, at a cost of permitting a large number of dummy slots. Like the MSGLSP, here the usage of the common resource is synchronized. The production lots are sequenced in each period on every machine and synchronized across the machines, such that in any point in time all the products belong to the same family with attributes of the common resource being produced. Contrarily to the MSGLSP, the start and end times of each common resource batch are determined explicitly. Let $l = 1, \ldots, L$ be an upper bound on the number of common resource batches for the whole planning horizon. The alternative variables are defined below.

Variables	
Y_{mijtl}	takes on 1, if a changeover on machine m from product i to product j in period t using the lth common resource batch; 0 otherwise;
X_{mitl}	production quantity on machine m of product i in period t using the lth common resource batch;
U_{lk}	takes on 1, if the lth common resource batch has attributes to attempt the family k; 0 otherwise;
u_l^s	start time of lth common resource batch;
u_l^f	end time of lth common resource batch;
μ_{mitl}^s	start time for production on machine m of the product i in period t using the lth common resource batch;
μ_{mitl}^f	end time for production on machine m of the product i in period t using the lth common resource batch;
α_{mitl}	takes on 1, if the machine m is set up for product i in the period t when start processing the lth common resource batch; 0 otherwise;
P_{tl}^s	takes on 1, if the lth common resource batch starts before the end of period t (i.e., instant t); 0 otherwise;
P_{tl}^f	takes on 1, if the lth common resource batch ends after the start of period t (i.e., instant $t-1$); 0 otherwise.

For sake of understandability and flow of the explanation, constraints are grouped throughout the model presentation. Using the above notation, the MSHLSP can be formulated as follows.

$$\textbf{Minimize} \quad \sum_{i=1}^{N}\sum_{t=1}^{T}(h_{it} \cdot I_{it}) + \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{l=1}^{L}(\sigma_{mij} \cdot Y_{mijtl}) \tag{2.16}$$

subject to

$$I_{i(t-1)} + \sum_{m=1}^{M}\sum_{l=1}^{L} X_{mitl} - I_{it} = d_{it} \quad \forall i, t \tag{2.17}$$

The objective function (2.16) aims at minimizing the sum of holding and sequence dependent setup costs. Constraints (2.17) represent the production flow balance constraints. Below, requirements (2.18)-(2.20) refer to the first stage, and define the starting and ending times of the usage of each batch of the common resource. Constraints (2.18) assign at most one family to each batch of the common resource.

Constraints (2.19)-(2.20) establish the start and end times of availability of each common resource batch. It is important to emphasize that a common resource batch might be available in a given time interval, but not being used (i.e., might be empty). In fact, the production of the common resource is synchronized with the manufacturing of the second stage and the idle times are only allocated in the second stage plan.

$$\sum_{k=1}^{K} U_{lk} \leq 1 \quad \forall l \tag{2.18}$$

$$u_l^s \leq u_l^f \leq T \quad \forall l \tag{2.19}$$

$$u_l^f = u_{(l+1)}^s \quad \forall l < L \tag{2.20}$$

The second set of constraints concerns to the second stage. Constraints (2.21)-(2.23) define the production time of a product, which is defined by the time wasted on setting up the machine for production and the production itself. If this time length is zero, then there is no production nor setup of product i on machine m using the lth common resource in period t; such case is defined in (2.21) when $\sum_{j=1}^{N} Y_{mjitl} = 0$ and $X_{mitl} = 0$. The capacity of each time period can not be violated. Note that the constraints (2.21)-(2.22) are valid because the length of each period, the setup times and the production rates are normalized to 1. Idle times may occur between two consecutive production slots or within the production slots. Constraints (2.23) define the production sequence. Moreover, they cutoff any subtour in the sequence. Constraints (2.24) ensure production of a product in a period only if the machine is correctly configured for it. Note that we do not need to multiply the right-hand side of (2.24) by a big number, as p_{mi} normalizes X_{mitl}. Constraints (2.25) and (2.26) correspond to the product setup flow balance equations for a given common resource batch and time period, respectively, carrying the setup information over the following batch and period, respectively. From (2.27), each machine is set up to one and only one product when it starts processing a certain common resource batch. Constraints (2.28) forbids phantom set ups.

$$\mu_{mitl}^f - \mu_{mitl}^s \geq \sum_{j=1}^{N} (s_{mji} \cdot Y_{mjitl}) + p_{mi} \cdot X_{mitl} \quad \forall m, i, t, l \tag{2.21}$$

$$\sum_{i=1}^{N} \sum_{l=1}^{L} \left(\mu_{mitl}^f - \mu_{mitl}^s \right) \leq 1 \quad \forall m, t \tag{2.22}$$

$$\mu_{mitl}^s \geq \mu_{mjtl}^f + Y_{mjitl} - 1 \quad \forall m, i, j, t, l \tag{2.23}$$

$$p_{mi} \cdot X_{mitl} \leq \sum_{j=1}^{N} Y_{mjitl} + \alpha_{mitl} \qquad \forall m, i, t, l \tag{2.24}$$

$$\sum_{j=1}^{N} Y_{mjitl} + \alpha_{mitl} = \sum_{j=1}^{N} Y_{mijtl} + \alpha_{mit(l+1)} \qquad \forall m, i, t, l < L \tag{2.25}$$

$$\sum_{j=1}^{N} Y_{mjitL} + \alpha_{mitL} = \sum_{j=1}^{N} Y_{mijtL} + \alpha_{mi(t+1)1} \qquad \forall m, i, t \tag{2.26}$$

$$\sum_{i=1}^{N} \alpha_{mitl} = 1 \qquad \forall m, t, l \tag{2.27}$$

$$Y_{miitl} = 0 \qquad \forall m, i, t, l \tag{2.28}$$

The last group of requirements submits to integrate the constraints of the first and second stages, limiting variables u_l^s and u_l^f from the common resource batches to μ_{mitl}^s and μ_{mitl}^f from the production times of each product on each machine for the whole planning horizon. Let $P_{tl}^s = 0$ if the lth batch of the common resource starts not before the end of period t (i.e. instant t), $P_{tl}^s = 1$ otherwise. And, let $P_{tl}^f = 0$ if the availability of the lth common resource ends before the beginning of period t (i.e. instant $t-1$), $P_{tl}^f = 1$ otherwise. Figure 2.3 provides some examples to reinforce the explanation of the variables P_{tl}^s and P_{tl}^f. In cases 2.3.a, 2.3.b and 2.3.c, the common resource batch starts before the end of period t ($P_{tl}^s = 1$). In cases 2.3.a, 2.3.b and 2.3.d, the common resource batch finishes after the beginning of period t ($P_{tl}^f = 1$). Note that by definition $P_{tl}^s + P_{tl}^f > 0$ and constraints (2.29) are

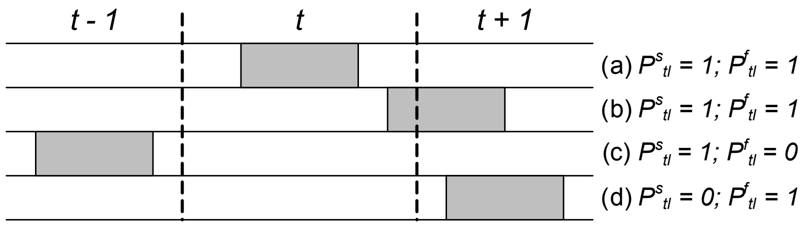

Figure 2.3: Illustrative examples for the P^s and P^f concept.

active when $P_{tl}^s = 0$ or $P_{tl}^f = 0$. In this case, $\mu_{mitl}^s = \mu_{mitl}^f$. Constraints (2.30)-(2.31) set lower bounds on the production times in case $P_{tl}^s = 1$, and respective upper bounds when $P_{tl}^f = 1$. Constraints (2.32) ensure that all production using the lth batch of the common resource occur only if the resource is available. From (2.33), all the productions relying on the same common resource batch are limited by its capacity. Constraints (2.34) admit only production of a product if the respective common resource is produced in the same time interval. Finally, (2.35)-(2.36) impose non-negativity and binary conditions.

$$\mu^f_{mitl} - T \cdot \left(P^s_{tl} + P^f_{tl} - 1\right) \leq \mu^s_{mitl} \leq \mu^f_{mitl} + T \cdot \left(P^s_{tl} + P^f_{tl} - 1\right) \qquad \forall m, i, t, l \qquad (2.29)$$

$$max\left(t - 1; u^s_l - T \cdot (1 - P^s_{tl})\right) \leq \mu^s_{mitl} \leq min\left(t; u^f_l + T \cdot \left(1 - P^f_{tl}\right)\right) \qquad \forall m, i, t, l \qquad (2.30)$$

$$max\left(t - 1; u^s_l - T \cdot (1 - P^s_{tl})\right) \leq \mu^f_{mitl} \leq min\left(t; u^f_l + T \cdot \left(1 - P^f_{tl}\right)\right) \qquad \forall m, i, t, l \qquad (2.31)$$

$$\sum_{t=1}^{T} \sum_{i=1}^{N} \left(\mu^f_{mitl} - \mu^s_{mitl}\right) \leq u^f_l - u^s_l \qquad \forall m, l \qquad (2.32)$$

$$\sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{t=1}^{T} X_{mitl} \leq C \cdot \sum_{k=1}^{K} U_{lk} \qquad \forall l \qquad (2.33)$$

$$p_{mi} \cdot X_{mitl} \leq U_{lk} \qquad \forall m, i \in S(k), t, l, k \qquad (2.34)$$

$$Y_{mijtl} \in \{0, 1\}\,;\; U_{lk} \in \{0, 1\}\,;\; P^s_{tl} \in \{0, 1\}\,;\; P^f_{tl} \in \{0, 1\} \qquad \forall m, i, j, t, l, k \qquad (2.35)$$

all other variables are non-negative and continuous. (2.36)

2.5 MSCont - Multi-stage continuous lot-sizing and scheduling problem

The third formulation is based on a continuous-time representation that avoids unnecessary time slot splitting. It requires an initial number of available common resource batches, but the sequence and duration of these batches are unknown previously. Moreover, the production slots and their start/end times may vary from machine to machine and are independent from each other. Opposed to the previous two models, slots are of arbitrary lengths and are independent of the time grid, i.e., a slot is not confined to be within one or more periods. It may cover one or more periods and can even extend to beyond the planning horizon.

The production quantity is described per period, which allows us to define the production portion of each slot to meet the demand of each period. Below, we introduce the additional variables and the model formulation.

The two-stage lot-sizing and scheduling model MSCont is then:

Variables	
μ^s_{mil}	start time for production on machine m of product i using the lth common resource batch;
μ^f_{mil}	end time for production on machine m of product i using the lth common resource batch;
Y_{mijl}	takes on 1, if there is a changeover on machine m from product i to product j using the lth common resource batch; 0 otherwise;
w^s_{mitl}	start time of production slot on machine m of product i in period t using the lth common resource;
w^f_{mitl}	end time of production slot on machine m of product i in period t using the lth common resource;
E_{mitl}	takes on 1, if there is a production slot on machine m to product i in period t using the lth common resource; 0 otherwise;

$$\textbf{Minimize} \quad \sum_{i=1}^{N}\sum_{t=1}^{T}(h_{it}\cdot I_{it}) + \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{l=1}^{L}(\sigma_{mij}\cdot Y_{mijtl}) \tag{2.37}$$

subject to

$$I_{i(t-1)} + \sum_{m=1}^{M}\sum_{l=1}^{L} X_{mitl} - I_{it} = d_{it} \qquad \forall i,t \tag{2.38}$$

$$\sum_{k=1}^{K} U_{lk} \leq 1 \qquad \forall l \tag{2.39}$$

$$u^f_l = u^s_{(l+1)} \qquad \forall l < L \tag{2.40}$$

$$u^s_l \leq \mu^s_{mil} \leq \mu^f_{mil} \leq u^f_l \leq T \qquad \forall m,i,l \tag{2.41}$$

$$\sum_{i=1}^{N}\left(\mu^f_{mil} - \mu^s_{mil}\right) \leq u^f_l - u^s_l \qquad \forall m,l \tag{2.42}$$

$$\mu^s_{mil} \geq \mu^f_{mjl} + T\cdot(Y_{mjil}-1) \qquad \forall m,i,j,l \tag{2.43}$$

$$\mu^f_{mil} - \mu^s_{mil} \geq \sum_{j=1}^{N}(s_{mji}\cdot Y_{mjil}) + \sum_{t=1}^{T}(p_{mi}\cdot X_{mitl}) \qquad \forall m,i,l \tag{2.44}$$

$$w^f_{mitl} \leq min\left(t;\ \mu^f_{mil}\right) \qquad \forall m,i,t,l \tag{2.45}$$

$$w^s_{mitl} \geq max\left(t-1;\ \mu^s_{mil} + \sum_{j=1}^{N}(s_{mji}\cdot Y_{mjil})\right) \qquad \forall m,i,t,l \tag{2.46}$$

$$p_{mi}\cdot X_{mitl} \leq min\left(E_{mitl};\ w^f_{mitl} - w^s_{mitl} + T\cdot(1-E_{mitl})\right) \qquad \forall m,i,t,l \tag{2.47}$$

$$p_{mi}\cdot X_{mitl} \leq \sum_{j=1}^{N} Y_{mjil} + \alpha_{mil} \qquad \forall m,i,t,l \tag{2.48}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{t=1}^{T} X_{mitl} \leq C \cdot \sum_{k=1}^{K} U_{lk} \qquad \forall l \qquad (2.49)$$

$$p_{mi} \cdot X_{mitl} \leq U_{lk} \qquad \forall m, i \in S(k), t, l, k \qquad (2.50)$$

$$\sum_{j=1}^{N} Y_{mjil} + \alpha_{mil} = \sum_{j=1}^{N} Y_{mijl} + \alpha_{mi(l+1)} \qquad \forall m, i, l < L \qquad (2.51)$$

$$\sum_{j=1}^{N} Y_{mijl} \leq 1 \qquad \forall m, i, l \qquad (2.52)$$

$$\sum_{j=1}^{N} Y_{mjil} \leq 1 \qquad \forall m, i, l \qquad (2.53)$$

$$\sum_{i=1}^{N} \alpha_{mil} = 1 \qquad \forall m, l \qquad (2.54)$$

$$Y_{mijl} \in \{0,1\}\,;E_{mitl} \in \{0,1\}\,;\; U_{lk} \in \{0,1\} \qquad \forall m, i, t, l, k \qquad (2.55)$$

all other variables are non-negative and continuous. (2.56)

The objective function (2.37) minimizes the sum of holding and sequence dependent setup costs. Constraints (2.38) represent the usual inventory equations. Constraints (2.39)-(2.41) deal with the common resource batches, defining their sequence. Constraints (2.39) associate the common resource batch with at most one product family. Constraints (2.40)-(2.41) establish the start and end times of availability for each common resource batch. Moreover, (2.41) ensure that the production slot of a product is compatible to the common resource required for the product available in that bucket. Constraints (2.42)-(2.46) deal exclusively with production slots. They define the start, sequence, length and end of each production slot according to the common resource batch and the period time. Constraints (2.42) limit the production of products to the capacity of the respective common resource batch. Constraints (2.43) set the production sequence. The minimum length of each production slot is imposed by (2.44). On the other hand, the part of the production slots in each time period is given by variables w^s_{mitl} and w^f_{mitl}. The term $\left(w^f_{mitl} - w^s_{mitl}\right)$ implicitly defines the production time of the slot allocated to period. The constraints (2.45) and (2.46) set bounds for variables w^s_{mitl} and w^f_{mitl}, which are limited by the window time period and/or the boundaries of the common resource. There are no constraints avoiding idle time and it can be represented between and within production slots.

Constraints (2.47)-(2.50) define bounds for the production of the products. Constraints (2.47) limit the production per period to the availability of the slot in each period. In case there is production, the variable E_{mitl} turns on 1 and the production amount is limited by $w^f_{mitl} - w^s_{mitl}$. Naturally, such scenario only occurs if the machine is set up for that product - see (2.48). Constraints (2.49) take into account the common resource batch capacity. Constraints (2.50) admit only production of products if the common resource is suitable for the respective product family.

Constraints (2.51) impose the setup flow balance and the machine configuration from one common resource batch into the next. Flow in and flow out of each product node are at the most one within the same common resource batch - constraints (2.52) and (2.53). At the beginning of the usage of a common resource batch, each machine is set up for one product (2.54). Constraints (2.55)-(2.56) are the domain requirements.

2.6 Illustrative example to compare the optimal solutions to the models

In order to illustrate the application of the models proposed in this chapter, we present a small example with data provided in Tables 2.1 and 2.2. The number of machines is $M = 3$ and the common resource capacity is $C = 18,000$ kilos. The machine 1 is set up for product $i = 1$ at the beginning of the planning horizon and the machines 2 and 3 are set up for product $i = 3$. The planning horizon entails $T = 3$ time periods. Products 1, 2 and 3 require the same attributes and belong to product family $k = 1$, whilst products 4 and 5 are part of product family $k = 2$. Inventory costs (h_{it}) are equal for all periods. Production rates (p_{mi}), setup costs and times (s_{mji} and σ_{mji}) do not vary between machines.

Table 2.1: Data parameters to the illustrative example.

Product	Family k	d_{it} $t=1$	$t=2$	$t=3$	p_{mi}	h_{it}
$i=1$	1	4,571	0	0	0.000175	0.2
$i=2$	1	0	3,684	0	0.000460	0.2
$i=3$	1	20,000	7,230	0	0.000100	0.2
$i=4$	2	0	200	500	0.001000	0.2
$i=5$	2	0	0	17,200	0.000164	0.2

In order to benchmark the solutions of the three models, the maximum number (R) of micro-periods on the MSGLSP formulation and the maximum number of available of common resources batches on the MSHLSP and MSCont formulations are larger enough not to cutoff the optimal solution.

Table 2.2: Setup times and setup costs (s_{mji} and σ_{mji}).

product	1	2	3	4	5
1	0.0/0.0	0.031/3.1	0.039/3.9	0.066/6.6	0.060/6.0
2	0.047/4.7	0.0/0.0	0.042/4.2	0.066/6.6	0.050/5.0
3	0.045/4.5	0.040/4.0	0.0/0.0	0.060/3.0	0.070/5.0
4	0.063/6.3	0.063/6.3	0.063/6.3	0.0/0.0	0.030/3.0
5	0.065/6.5	0.062/6.2	0.060/6.0	0.050/5.0	0.0/0.0

Figures 2.4, 2.5 and 2.6 illustrate the differences in slot designs of the three formulations. The vertical dashed lines denote important time instants of production - which can be a changeover from a common resource batch to another or the end of a period in the planning horizon. White rectangles represent the time wasted on setting up the machine for production. The different product families are represented in different grey scale tones. Slots with zero lengths are not shown in the Gantt charts. In Figure 2.4, there are 24 non-zero slots on the planning horizon, 8 on each machine, while in Figures 2.5 and 2.6, there are 18 and 13 non-zero slots on the planning horizon, respectively.

Figure 2.4 and Table 2.3 show the optimal solution to this instance when applying the MSGLSP model. Recall that for the MSGLSP, μ^s_{tr} and μ^f_{tr} represent the start and end times of micro-period r in period t; Z_{mijtr} takes on 1, if there is a changeover on machine m from product i to product j in micro-period r of period t and X_{mitr} and I_{it} denote the production and inventory variables.

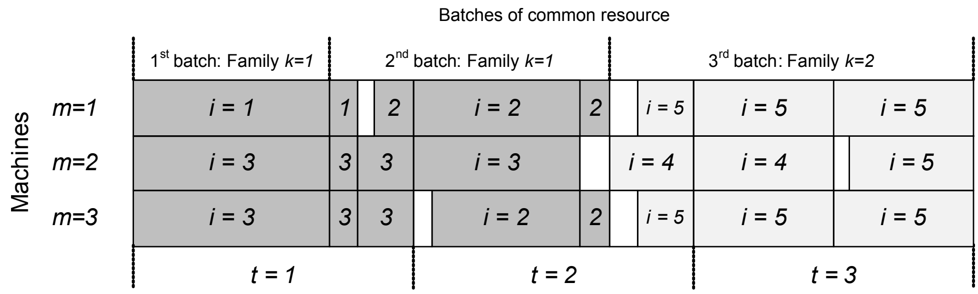

Figure 2.4: Optimal solution to MSGLSP of the illustrative example.

For MSHLSP, the most relevant non-zero solution values for the same instance are given in Table 2.4. Reminding that, μ^s_{mitl} and μ^f_{mitl} represent the start and end times to produce product i on machine m in period t using the lth common resource batch; Y_{mijtl} takes on 1, if there is a changeover on machine m from product i to product j in period t, using the lth common resource batch; α_{mitl} equals to 1, if the machine m is set up for product i in the period t using the lth common resource batch; X_{mitr} and I_{it} denote the production and inventory variables.

Table 2.3: Optimal solution from MSGLSP model.

	$t = 1$	$t = 2$	$t = 3$
Fixed micro-periods	$\mu^s_{11} : \mu^f_{11} = 0 : 0.7$ $\mu^s_{12} : \mu^f_{12} = 0.7 : 0.8$ $\mu^s_{13} : \mu^f_{13} = 0.8 : 1$	$\mu^s_{21} : \mu^f_{21} = 1 : 1.7$ $\mu^s_{22} : \mu^f_{22} = 1.7 : 1.8$ $\mu^s_{23} : \mu^f_{23} = 1.8 : 2$	$\mu^s_{31} : \mu^f_{31} = 2 : 2.5$ $\mu^s_{32} : \mu^f_{32} = 2.5 : 3$ $\mu^s_{33} : \mu^f_{33} = 3 : 3$
$m = 1$	$Z_{11111} = 1$ $X_{1111} = 4000$ $Z_{11112} = 1$ $X_{1112} = 570$ $Z_{11213} = 1$ $X_{1213} = 366.6$	$Z_{12221} = 1$ $X_{1221} = 1571.7$ $Z_{12222} = 1$ $X_{1222} = 130.4$ $Z_{12523} = 1$ $X_{1523} = 1018.3$	$Z_{15531} = 1$ $X_{1531} = 2945.1$ $Z_{15532} = 1$ $X_{1532} = 3152.4$
$m = 2$	$Z_{23311} = 1$ $X_{2311} = 7000$ $Z_{23312} = 1$ $X_{2312} = 1000$ $Z_{23313} = 1$ $X_{2313} = 2000$	$Z_{23321} = 1$ $X_{2321} = 7230$ $Z_{23422} = 1$ $Z_{24423} = 1$ $X_{2423} = 217$	$Z_{24431} = 1$ $X_{2431} = 483$ $Z_{24532} = 1$ $X_{2532} = 2969.5$
$m = 3$	$Z_{33311} = 1$ $X_{3311} = 7000$ $Z_{33312} = 1$ $X_{3312} = 1000$ $Z_{33313} = 1$ $X_{3313} = 2000$	$Z_{33221} = 1$ $X_{3221} = 1484.8$ $Z_{32222} = 1$ $X_{3222} = 130.4$ $Z_{32523} = 1$ $X_{3523} = 1017.1$	$Z_{35531} = 1$ $X_{3531} = 2945.1$ $Z_{35532} = 1$ $X_{3532} = 3152.4$
Stock	$I_{21} = 366.6$	$I_{42} = 17$ $I_{52} = 2035.4$	
Value of solution: 509.9			

Figure 2.5 illustrates graphically the solution detailed in Table 2.4.

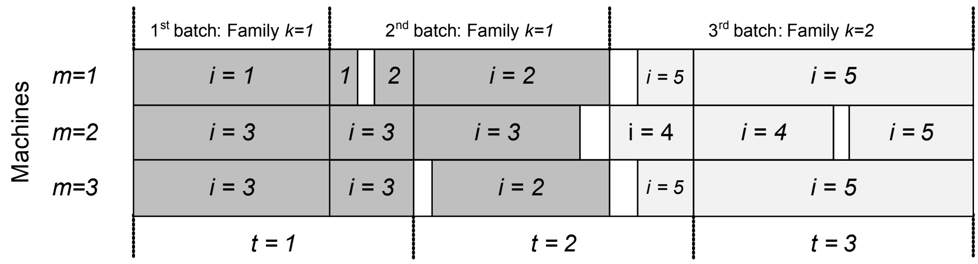

Figure 2.5: Optimal solution to Model MSHLSP of the illustrative example.

Finally, Table 2.5 presents the optimal solution to MSCont. Evoking that, μ^s_{mil} (μ^f_{mil}) represents the start time (end time) on machine m to produce product i using the lth common resource batch; w^s_{mitl} (w^f_{mitl}) is the start time (end time) of production slot on machine m of product i in period t using the lth common resource; Y_{mijl} and α_{mitl} denote the setup-related variables and X_{mitr} and I_{it} the production and inventory variables. The solution presented in Table 2.5 is illustrated in Figure 2.6.

The differences between the three models on the time structure are well exposed

Table 2.4: Optimal solution from MSHLSP model.

	$t=1$	$t=2$	$t=3$
$m=1$	$\alpha_{1111}=1$ $\mu^s_{1111}:\mu^f_{1111}=0:0.7$ $X_{1111}=4000$ $\alpha_{1112}=1$ $\mu^s_{1112}:\mu^f_{1112}=0.7:0.8$ $X_{1112}=571$ $\mu^s_{1212}:\mu^f_{1212}=0.8:1$ $Y_{1122}=1$ $X_{1212}=366.6$	$\alpha_{1222}=1$ $\mu^s_{1222}:\mu^f_{1222}=1:1.8$ $X_{1222}=1702.2$ $\alpha_{1223}=1$ $\mu^s_{1523}:\mu^f_{1523}=1.8:2$ $Y_{1253}=1$ $X_{1523}=1017.1$	$\alpha_{1533}=1$ $\mu^s_{1533}:\mu^f_{1533}=2:3$ $X_{1533}=6097.6$
$m=2$	$\alpha_{2311}=1$ $\mu^s_{2311}:\mu^f_{2311}=0:0.7$ $X_{2311}=7000$ $\alpha_{2312}=1$ $\mu^s_{2312}:\mu^f_{2312}=0.7:1$ $X_{2312}=3000$	$\alpha_{2322}=1$ $\mu^s_{2322}:\mu^f_{2322}=1:1.7$ $X_{2322}=7230$ $\mu^s_{2422}:\mu^f_{2422}=1.7:1.8$ $Y_{2342}=1$ $\alpha_{2422}=1$ $\mu^s_{2423}:\mu^f_{2423}=1.8:2$ $X_{2423}=217$	$\alpha_{2433}=1$ $\mu^s_{2433}:\mu^f_{2433}=2:2.5$ $X_{2433}=483$ $\mu^s_{2533}:\mu^f_{2533}=2.5:3$ $Y_{2453}=1$ $X_{2533}=2969.5$
$m=3$	$\alpha_{3311}=1$ $\mu^s_{3311}:\mu^f_{3311}=0:0.7$ $X_{3311}=7000$ $\alpha_{3312}=1$ $\mu^s_{3312}:\mu^f_{3312}=0.7:1$ $X_{3312}=3000$	$\alpha_{3322}=1$ $\mu^s_{3222}:\mu^f_{3222}=1:1.8$ $Y_{3322}=1$ $X_{3222}=1615.2$ $\alpha_{3221}=1$ $\mu^s_{3523}:\mu^f_{3523}=1.8:2$ $Y_{3253}=1$ $X_{3523}=1018.3$	$\alpha_{3533}=1$ $\mu^s_{3533}:\mu^f_{3533}=2:3$ $X_{3533}=6097.6$
Stock	$I_{21}=366.6$	$I_{42}=17$ $I_{52}=2035.4$	
Value of solution: 509.9			

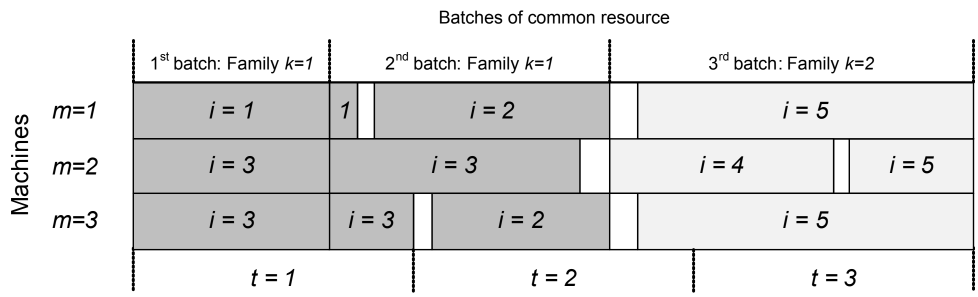

Figure 2.6: Optimal solution to Model MSCont of the illustrative example.

by the second lot of product 3 on machine 2 in MSCont of Figure 2.6. In MSHLSP and MSGLSP, this lot has to be divided into small slots. In fact, the main advantage of the MSCont formulation is that the machines have the production slots independent of the time periods boundaries, but limited to the common resource batch. Another advantage provided by the independence of time periods is that this model allows for setup crossover, i.e., a setup changeover can be initialized in

Table 2.5: Optimal solution from MSCont model.

	$t=1$	$t=2$	$t=3$
$m=1$	(0) $\mu^s_{111} : \mu^f_{111}$ (0.7) $\mu^s_{112} : \mu^f_{112}$ (0.8) $\mu^s_{122} : \mu^f_{122}$ (1.8) $\mu^s_{153} : \mu^f_{153}$ (3)		
	$\alpha_{111}=1$ $w^s_{1111} : w^f_{1111} = 0 : 0.7$ $X_{1111}=4000$ $\alpha_{112}=1$ $w^s_{1112} : w^f_{1112} = 0.7 : 0.8$ $X_{1112}=571$ $w^s_{1212} : w^f_{1212} = 0.8 : 1$ $Y_{1122}=1$ $X_{1212}=366.6$	$w^s_{1222} : w^f_{1222} = 1 : 1.8$ $X_{1222}=1702.2$ $\alpha_{123}=1$ $w^s_{1523} : w^f_{1523} = 1.8 : 2$ $Y_{1253}=1$ $X_{1523}=1017.1$	$w^s_{1533} : w^f_{1533} = 2 : 3$ $X_{1533}=6097.6$
$m=2$	(0) $\mu^s_{231} : \mu^f_{231}$ (0.7) $\mu^s_{232} : \mu^f_{232}$ (1.7) $\mu^s_{242} : \mu^f_{242}$ (1.8) $\mu^s_{243} : \mu^f_{243}$ (2.5) $\mu^s_{253} : \mu^f_{253}$ (3)		
	$\alpha_{231}=1$ $w^s_{2311} : w^f_{2311} = 0 : 0.7$ $X_{2311}=7000$ $\alpha_{232}=1$ $w^s_{2312} : w^f_{2312} = 0.7 : 1$ $X_{2312}=3000$	$w^s_{2322} : w^f_{2322} = 1 : 1.7$ $X_{2322}=7230$ $Y_{2342}=1$ $\alpha_{242}=1$ $w^s_{2423} : w^f_{2423} = 1.8 : 2$ $X_{2423}=217$	$w^s_{2433} : w^f_{2433} = 2 : 2.5$ $X_{2433}=483$ $w^s_{2533} : w^f_{2533} = 2.5 : 3$ $Y_{2453}=1$ $X_{2533}=2969.5$
$m=3$	(0) $\mu^s_{331} : \mu^f_{331}$ (0.7) $\mu^s_{332} : \mu^f_{332}$ (1) $\mu^s_{322} : \mu^f_{322}$ (1.8) $\mu^s_{353} : \mu^f_{353}$ (3)		
	$\alpha_{3311}=1$ $w^s_{3311} : w^f_{3311} = 0 : 0.7$ $X_{3311}=7000$ $\alpha_{3312}=1$ $w^s_{3312} : w^f_{3312} = 0.7 : 1$ $X_{3312}=3000$	$w^s_{3222} : w^f_{3222} = 1 : 1.8$ $Y_{3322}=1$ $X_{3222}=1615.2$ $\alpha_{321}=1$ $w^s_{3523} : w^f_{3523} = 1.8 : 2$ $Y_{3253}=1$ $X_{3523}=1018.3$	$w^s_{3533} : w^f_{3533} = 2 : 3$ $X_{3533}=6097.6$
Stock	$I_{21}=366.6$	$I_{42}=17$ $I_{52}=2035.4$	
Value of solution: 509.9			

a period and it can be finished in the following. Its disadvantage is that it becomes more challenging to model scheduling constraints with more complicated structures compared to its discrete time counterparts (MSGLSP and MSHLSP).

On the other hand, the MSGLSP model provides a clean representation of the problem. However, the number of micro-periods is large and can influence significantly the solution efficiency. The main advantage of the MSGLSP is providing the reference time grid for all machine operations. This quality renders the possibility of formulating the various scheduling constraints (and possibly others not expressed here) in a relatively straightforward manner as well as the synchronization of the shared common resource.

Table 2.6 lists a few important industrial features namely the setup carry over, the setup crossover and the start and end times for the common resource batches, which are addressed or not by each of the three models presented here. From these comparison, the MSHLSP and MSCont formulations approach more industrial features than MSGLSP. For instance, the explicit determination of each common resource batch and the setup crossover periods for the MSCont model. Moreover, all proposed formulations allow for setup carry over between periods and batches.

Table 2.6: Comparison between features of the formulations.

	MSGLSP	MSHLSP	MSCont
Setup carry over between periods	✓	✓	✓
Setup carry over between batches	✓	✓	✓
Setup crossover between periods	χ	χ	✓
Start and end times of each common resource batch	χ	✓	✓

2.7 Determining the maximum number of batches for the common resource

Kim *et al.* (2010) explain that the maximum number of setups allowed on a machine influence the size of formulation for lot-sizing and scheduling problems. The authors define a minimum value to the maximum number of setups allowed on a machine. This figure should be small for solvability reason, but not too small to delete the minimum cost feasible solution. In our problem, L is defined as the maximum number of batches of the common resource on the planning horizon. This parameter is important for models MSHLSP and MSCont. We should determine a number of batches able to process the required demand over the planning horizon which is restricted by the production rates of the products.

As there is demand for several products, we must allow for the production of K different attribute batches of the common resource to meet the set of product families. Let $min_{m \in M}\,(p_{mi})$ as the smallest processing time of one unit of product i among M machines and ZL_k be the cumulative summation of the needed capacity to manufacture all products of family k. The maximum number of batches of the common resource on the planning horizon can be determined by the procedure of Algorithm 1 as follows:

Technically, the total capacity of a common resource batch can be used partially during the process, but it may need more manpower and generate raw material waste. So, it makes sense to have a production plan in which each batch is consumed by almost exactly C kilos. The capacity of the common resource process is related to the L value when we use this proposed method to determine a maximum number

Algorithm 1: Procedure L_Lower_Bound

begin

Sort one list $\mathcal{L}$ of products with family k associated and $min_{m \in M}(p_{mi})$ in increasing order;

$T' = M \cdot T$;

$ZL_k = 0 \; \forall k$;

for i=$\mathcal{L}(1)$ **to** $\mathcal{L}(N)$ **do**

$$ZL_k = ZL_k + min\left(\frac{T'}{min_{m \in M}(p_{mi})}; \sum_{t=1}^{T} d_{it}\right);$$

$$T' = T' - min\left(\frac{T'}{min_{m \in M}(p_{mi})}; \sum_{t=1}^{T} d_{it}\right) \cdot min_{m \in M}(p_{mi});$$

$$L = max\left(K; \sum_{k=1}^{K} \left\lceil \frac{ZL_k}{C} \right\rceil\right).$$

of common resource batches on planning horizon. A small value for L can ensure a better use of raw material and a lower number of variables to model our problem.

2.8 Computational results

In order to compare the performance and flexibility of the models, they were implemented in ILOG OPL language, and solved by the mixed-integer programming solver CPLEX 12.1. The computational experiments were performed on an IBM personal computer with two 2.4GHz CPUs and 2GB of random access memory under Windows 7.

The instances used to conduct the computational experiments described below are based on real world datasets from a spinning industry. In this exemplar of the process industry, the common resource is a blend of fibers that, in mass form, supplies several spinning machines. The final products are yarn packages belonging to two yarn families. The size of these instances is of practical relevance for this industry. The number of products N range from 4 to 6, the number of machines M is 3 and 4, the number of periods T equal to 5. This planning horizon corresponds to the working days of a typical week. The inventory and setup costs are estimated based on the opportunity cost of one yarn package. Cut denotes the approximated capacity utilization of the downstream machines. For every set N/M, Cut takes the values of 0.6, 0.7 and 0.8. The initial setup state for all machines is randomly generated. The maximum number of the common resource batches along the planning horizon is determined using the procedure described previously in Section 2.7. Furthermore, in order to ensure a fair comparison, we add in the MSCont the following constraints to avoid the occurrence of setup crossover between periods:

$$\lceil \mu^s_{mil} + \epsilon \rceil - \mu^s_{mil} \geq \sum_{j=1}^{N} (s_{mji} \cdot Y_{mjil}) \, , \; \forall m, i, l,$$

where $\epsilon \geq 0$ is a small number.

Tables 2.7 and 2.8 present the results for each instance of the class $M = 3$ and $M = 4$, respectively, considering a time limit of one hour. The column *UB* indicates the best feasible solution found. Column *gap (%)* reports the optimality gap, considering the best lower bound provided by the respective model at the end of the run. *LP* denotes the solution from the linear relaxation of the model (the root node of the branch-and-bound tree). And, *nodes* indicates the number of nodes explored in the branch-and-bound tree. Each instance name refers to its parameters set. For example, `rw06-N4M3T5` is an instance with a capacity utilization (Cut) equal to 0.6, a number of products (N) to 4, a number of machines (M) to 3 and a number of periods (T) to 5.

From Tables 2.7 and 2.8, we can see that after one hour of computation time, MSGLSP model has found feasible solutions for 17 instances and reports the best solutions for 17 out of 18 instances. When compared to MSHLSP and MSCont, with regards to the optimality gap, the MSGLSP beats the other two for every instance but `rw07-N4M3T5` and `rw08-N4M3T5` instances, where MSHLSP model outperforms the others. Actually, for 4 machines-instances, the MSHLSP and MSCont can solve only 3 out of 9 and 2 out of 9, respectively. Regarding the linear relaxation of the three models, from Tables 2.7 and 2.8, one may infer that MSHLSP is at least as strong as MSGLSP, and MSGLSP at least as strong as MSCont. Nevertheless, we would like to stress that such condition is dependent upon the number of micro-periods used for the MSGLSP and the number of batches for the other two models. The MSCont does not seem to be competitive for generating lower and upper bounds. In 1 hour test, the CPLEX solver explores more nodes on the MSGLSP formulation than on the MSHLSP model. The linear programming of each node of the MSGLSP's tree is more demanding. On the other hand, it seems that MSHLSP does not need to explore as many nodes as MSGLSP to provide almost equivalent upper bounds and sometimes better lower bounds. Because of this fact, we have tried to analyze the convergence of MSHLSP with respect to MSGLSP and MSCont for longer running times.

Tables 2.9 and 2.10 show those experiments for one day of running time. For 3 machine-instances, MSHLSP model delivered provably optimal solutions for 5 out

Table 2.7: Results for the three models on instances with $M = 3$ after one hour.

		UB	gap (%)	LP	nodes
rw06-N4M3T5	MSGLSP	**36.47***	**19.45%**	4.51	1,175,900
	MSHLSP	36.95	24.32%	4.51	88,800
	MSCont	**36.47***	54.46%	4.17	549,800
rw06-N5M3T5	MSGLSP	**39.16***	**27.33%**	5.03	714,300
	MSHLSP	39.67	41.69%	5.03	39,400
	MSCont	58.40	74.84%	4.71	368,000
rw06-N6M3T5	MSGLSP	**56.27**	**60.18%**	5.91	39,000
	MSHLSP	-	-	10.01	9,000
	MSCont	-	-	5.57	33,000
rw07-N4M3T5	MSGLSP	**50.10***	40.04%	4.76	300,200
	MSHLSP	**50.10***	**23.10%**	7.43	91,300
	MSCont	**50.10***	49.52%	4.17	616,900
rw07-N5M3T5	MSGLSP	**44.22***	**41.18%**	4.96	196,500
	MSHLSP	47.76	49.83%	4.96	30,000
	MSCont	-	-	4.65	177,700
rw07-N6M3T5	MSGLSP	**113.64**	**87.38%**	3.98	38,000
	MSHLSP	-	-	7.63	1,600
	MSCont	-	-	3.67	10,700
rw08-N4M3T5	MSGLSP	**78.46**	53.00%	11.21	479,800
	MSHLSP	79.81	**43.17%**	17.22	23,600
	MSCont	89.15	69.05%	4.17	267,100
rw08-N5M3T5	MSGLSP	**112.89**	**88.05%**	3.20	157,600
	MSHLSP	-	-	3.26	13,000
	MSCont	-	-	2.93	110,000
rw08-N6M3T5	MSGLSP	-	-	3.58	37,000
	MSHLSP	-	-	8.30	3,600
	MSCont	-	-	3.30	13,100

*Optimal solution - as proved in Table 2.9.

of 9 instances. And it is quite expressive the small number of explored nodes by the MSHLSP to prove optimality. It can achieve better solutions without accumulate too much information of the branch-and-bound tree. These results made also clear the difficulty of MSGLSP on proving the solution optimality. By comparing Tables 2.7 and 2.9, it is worth of note the MSGLSP can deliver optimal solutions just after one hour. The troublesome of this model is really the weak lower bound and here is where the MSHLSP can play a role. The MSCont formulation is the most flexible of the three for incorporating additional operational features, but it is by far the worst of the three in terms of performance.

Our results provide compelling evidence in favor of MSGLSP model on generating good solutions in shorter time in spite of the trouble on proving their optimality. On the other hand, the MSHLSP model shows features of a large bucket problem dealing with better lower bounds. Of course, this advantage depends on the parametrization choice, but it can find provably optimal solutions faster than the other two. Finally, despite of the flexibility, the results showed the MSCont model is clearly dominated

Table 2.8: Results for the three models on instances with $M = 4$ after one hour.

		UB	gap (%)	LB	nodes
rw06-N4M4T5	MSGLSP	**46.79**	**34.18%**	5.22	847,000
	MSHLSP	49.09	43.16%	4.43	25,000
	MSCont	50.52	64.83%	3.68	312,400
rw06-N5M4T5	MSGLSP	**50.30**	**45.74%**	5.33	92,100
	MSHLSP	64.00	61.77%	5.45	15,600
	MSCont	60.89	80.53%	3.81	167,600
rw06-N6M4T5	MSGLSP	**48.10**	**59.37%**	5.39	51,000
	MSHLSP	-	-	5.46	2,700
	MSCont	-	-	4.96	10,900
rw07-N4M4T5	MSGLSP	**58.54**	**52.24%**	4.35	513,800
	MSHLSP	78.30	70.90%	4.43	27,300
	MSCont	-	-	3.09	158,900
rw07-N5M4T5	MSGLSP	**78.92**	**71.39%**	5.48	71,400
	MSHLSP	-	-	5.54	8,200
	MSCont	-	-	3.99	107,400
rw07-N6M4T5	MSGLSP	**55.40**	**65.79%**	5.42	58,000
	MSHLSP	-	-	5.47	2,000
	MSCont	-	-	5.05	11,000
rw08-N4M4T5	MSGLSP	**114.02**	**66.72%**	4.50	176,400
	MSHLSP	-	-	4.55	6,800
	MSCont	-	-	3.25	37,500
rw08-N5M4T5	MSGLSP	**111.77**	**73.16%**	4.57	272,300
	MSHLSP	-	-	4.66	1,600
	MSCont	-	-	3.35	9,700
rw08-N6M4T5	MSGLSP	**91.03**	**79.72%**	5.85	43,000
	MSHLSP	-	-	5.92	1,100
	MSCont	-	-	5.55	10,600

by its counterparts. After verifying the performance of the models on real data based instances, we extend our analysis on a set of 150 feasible instances that were randomly generated based on hints from practice. Given a set of parameters N (number of products), K (number of families), M (number of machines) and T (number of periods), the remaining data is determined as follows:

- all products are randomly assigned to a product family based on the uniform distribution;
- the set of products of the same family that might be allocated to each machine is chosen randomly;
- the setup times (with triangle inequalities respected) and the processing times for all products on each machine are generated from the normal distribution;
- all costs are derived from the opportunity cost per product unit;
- the demand of a product and the capacity of the first-stage machine are chosen

Table 2.9: Comparison between the models on running some instances with $M = 3$.

		UB	gap (%)	nodes
rw06-N4M3T5	MSGLSP	**36.47**	9.51%	31,750,345
	MSHLSP	**36.47**	**0.00%**	695,715
	MSCont	**36.47**	30.33%	37,407,000
rw06-N5M3T5	MSGLSP	**39.16**	23.58%	5,253,925
	MSHLSP	**39.16**	**0.00%**	978,774
	MSCont	**39.16**	41.46%	3,048,000
rw06-N6M3T5	MSGLSP	**56.03**	50.16%	715,700
	MSHLSP	64.15	**42.35%**	137,100
	MSCont	58.91	56.07%	948.800
rw07-N4M3T5	MSGLSP	**50.10**	18.98%	8,652,675
	MSHLSP	**50.10**	**0.00%**	437,610
	MSCont	**50.10**	33.22%	16,982,300
rw07-N5M3T5	MSGLSP	**44.22**	18.74%	5,556,946
	MSHLSP	**44.22**	**0.00%**	463,164
	MSCont	**44.22**	46.27%	4,658,628
rw07-N6M3T5	MSGLSP	**82.73**	**74.41%**	1,381,000
	MSHLSP	-	-	44,900
	MSCont	-	-	424,000
rw08-N4M3T5	MSGLSP	72.88	30.10%	13,259,988
	MSHLSP	**72.79**	**0.00%**	1,163,906
	MSCont	75.74	42.89%	3,508,937*
rw08-N5M3T5	MSGLSP	**73.34**	**73.82%**	3,860,992
	MSHLSP	-	-	90,155
	MSCont	-	-	2,765,024
rw08-N6M3T5	MSGLSP	**84.54**	**79.02%**	1,211,000
	MSHLSP	-	-	59,500
	MSCont	-	-	346,100

*Solver stopped on out-of-memory status before the stop-criterium.

with respect to the capacity requirements.

Now, the number of products N were four, five and seven, which can belong to 2 and 3 product families K. In addition, the number of machines M considered is three, five and seven, the number of periods T was equal to 5 and the number of micro-periods R for the MSGLSP was fixed to 6. The maximum number of the common resource batches along the planning horizon is determined by the procedure described in Section 2.7. For each combination N/M/K, ten different instances were randomly generated.

Table 2.11 shows the results for the random instances after 1 hour of running time. For each type of instance, we present the average optimality gaps (gap (%)) and linear relaxation values (LB), as well as the number of instances out of 10 for which CPLEX found feasible solutions. Using this dataset, clearly, the MSGLSP finds more feasible solutions than the other two models. Only two sets of instances were not solved by the MSGLSP. On the other hand, the MSHLSP is able to solve

Table 2.10: Comparison between the models on running some instances with $M = 4$.

		UB	gap (%)	nodes
rw06-N4M4T5	MSGLSP	**46.79**	19.27%	12,070,588
	MSHLSP	**46.79**	**12.95%**	1,307,600
	MSCont	47.19	30.44%	9,304,627
rw06-N5M4T5	MSGLSP	**50.30**	**42.84%**	192,874*
	MSHLSP	52.41	43.56%	47,500
	MSCont	54.02	74.65%	376,400
rw06-N6M4T5	MSGLSP	**45.91**	**52.69%**	268,900
	MSHLSP	53.13	59.74%	55,000
	MSCont	-	-	165,800*
rw07-N4M4T5	MSGLSP	**58.54**	39.67%	3,917,200
	MSHLSP	**58.54**	**22.64%**	1,292,231
	MSCont	59.93	77.08%	900,600
rw07-N5M4T5	MSGLSP	**60.08**	**58.22%**	1,552,765*
	MSHLSP	75.17	60.11%	26,700
	MSCont	-	-	147,756*
rw07-N6M4T5	MSGLSP	**51.34**	**50.50%**	1,115,700
	MSHLSP	-	-	20,300
	MSCont	-	-	722,000
rw08-N4M4T5	MSGLSP	**105.99**	**63.56%**	7,019,774
	MSHLSP	-	-	48,784*
	MSCont	-	-	513,689*
rw08-N5M4T5	MSGLSP	**103.10**	**71.05%**	941,200*
	MSHLSP	-	-	44,287*
	MSCont	-	-	1,928,540
rw08-N6M4T5	MSGLSP	**73.94**	**69.93%**	1,483,900
	MSHLSP	-	-	25,100
	MSCont	-	-	601,000

*Solver stopped on out-of-memory status before the stop-criterium.

few instances from four different sets, whereas the MSCont is only able to solve one instance. Regarding the linear relaxation, again, the MSHLSP seems to be at least as strong as the MSGLSP, which seems to be at least as strong as the MSCont. In general, we can state that the findings for the real-world instances are still valid for the artificial ones.

2.9 Final remarks

In this chapter, the two stage lot-sizing and scheduling problem usually present in process industries was tackled. All machines on the second stage must manufacture products of the same product family in each point time which requires a first-stage resource batch that satisfies quality attributes. Thus, the synchronization between the first and second stages is an important issue in modeling. This matter is frequently modeled by the authors using a discrete time representation. The discrete approach is justified by the requirements for meeting product demands

Table 2.11: Comparison between the models on running random instances. Grouping classes.

			MSGLSP			MSHLSP			MSCont		
N	M	K	gap (%)	Solved	LB	gap (%)	Solved	LB	gap (%)	Solved	LB
4	3	2	44.84	10 / 10	25.65	16.85	1 / 10	25.74	-	0 / 10	1.78
5	3	2	41.22	9 / 10	23.09	18.31	1 / 10	23.18	48.95	1 / 10	1.77
7	3	2	60.44	7 / 10	18.17	-	0 / 10	18.83	-	0 / 10	2.15
5	3	3	54.02	9 / 10	21.03	42.51	2 / 10	21.03	-	0 / 10	1.54
7	3	3	69.76	5 / 10	13.04	63.75	1 / 10	13.05	-	0 / 10	1.83
4	5	2	44.46	10 / 10	38.85	-	0 / 10	38.91	-	0 / 10	2.18
5	5	2	53.43	9 / 10	34.03	-	0 / 10	34.08	-	0 / 10	2.26
7	5	2	72.70	2 / 10	27.24	-	0 / 10	27.28	-	0 / 10	2.13
5	5	3	70.31	6 / 10	27.75	-	0 / 10	28.12	-	0 / 10	2.32
7	5	3	70.24	2 / 10	25.35	-	0 / 10	25.66	-	0 / 10	2.84
4	7	2	54.34	10 / 10	45.39	-	0 / 10	46.00	-	0 / 10	2.65
5	7	2	52.08	4 / 10	50.52	-	0 / 10	51.49	-	0 / 10	3.13
7	7	2	-	0 / 10	45.16	-	0 / 10	45.89	-	0 / 10	3.13
5	7	3	74.39	1 / 10	40.20	-	0 / 10	40.61	-	0 / 10	2.88
7	7	3	-	0 / 10	26.85	-	0 / 10	27.23	-	0 / 10	3.19

on given due dates and for determining the inventory level. For that problem, we have presented three different time-based scale formulations. First, the MSGLSP, based on a discrete time representation, makes use of a fixed grid to synchronize the two stages. Second, the MSHLSP has a hybrid discrete-continuous representation where the lot sizing and scheduling are determined for the first and second stages, respecting the synchronization between them. Last, the MSCont gives production slots that can spread over periods, but retaining the production quantity per period.

On one hand, we have the MSHLSP and MSCont achieving a suitable approximation of the original problem where the common resource is produced in batches. The evidence can be viewed in the control of the common resource by the MSHLSP and MSCont constraints. On the other hand, the MSGLSP can represent the problem where the common resource has continuous production. In this model, no specific constraints define the start or end of each common resource. From the computational experiments, the MSGLSP seems to be the best option when the decision maker needs a feasible solution in a short time. However, if the decision maker looks for solution quality or some confidence on the optimality gap, the MSHLSP formulation provides better solutions for longer running times. Despite delivering the worst performance, MSCont in its basic form is the most flexible of the three to incorporate setup-related features.

In conclusion, the integration of problems such as lot sizing and scheduling on the two stage environment that incorporates more and more specificities of industrial processes such as synchronized production is nowadays a challenging area, as also stressed in (Clark *et al.*, 2011). Overall, the development of valid inequalities can tighten the model relaxations and allow to solve harder instances. Finally, we hope that these models, specifically some of their ingredients, help the researchers on approaching their problems with each additional specificity and encourage others to work on it.

In the next chapter, the production planning problem in the spinning industry is analyzed. The MSGLSP incorporates some specificities of the spinning process and a reformulation to make it stronger. We also introduce the solution method HOPS - Hamming-Oriented Partition Search to deal with the problem.

Chapter 3

HOPS - Hamming-Oriented Partition Search for production planning in the spinning industry

In this chapter, we investigate a two-stage lot-sizing and scheduling problem in a spinning industry. The MSGLSP model, proposed in Chapter 2, is adapted to incorporate some features of this industry. A new hybrid method called HOPS (Hamming-Oriented Partition Search) is proposed to solve the problem, which is a branch-and-bound based procedure that incorporates a fix-and-optimize improvement method. An innovative partition choice for the fix-and-optimize is developed. The computational tests with generated instances based on real data show that the HOPS is a good alternative for solving mixed-integer problems with recognized partitions such as the lot-sizing and scheduling case. This chapter is strongly based on (Camargo *et al.*, 2012a).

3.1 Introduction

As discussed in Chapter 1, the textile industry is very important for the economy as it generates employment and increases regional development. The production system of the spinning industry is typically characterized by a multi-stage process, with single or parallel machines on each stage. In general, these machines have unsimilar production characteristics, such as processing speed and changeover times. The setups depend on the production sequence of products in the machines. The main products are made of natural or synthetic fibers, but the most commonly used raw material is cotton. Production orders or lots must be found in order to satisfy customer demand in the most cost effective manner. Therefore, in the spinning industry there is a need for simultaneous size and sequence production lots. The scheduling integration with the lot-sizing problem attempts to define better production plans than those generated from the solution of these problems in a hierarchical way. The lot sizing and scheduling is considered the major issue of operational production planning (Drexl and Kimms, 1997) and it is also present in other industrial processes such as, e.g., foundry (Araujo *et al.*, 2007), glass container (Almada-Lobo *et al.*, 2008), soft drinks (Ferreira *et al.*, 2009) and beverages (Guimarães *et al.*, 2012).

The research reported in the literature on production planning in the textile industry is rather scarce. Silva and Magalhaes (2006) focus on the lot-sizing and scheduling problem of an acrylic fiber industry. The fiber manufacturing process appears upstream the spinning in the textile supply chain. Nevertheless, three studies address the production planning problem downstream the spinning. Serafini and Speranza (1992) focus on the weaving process, Karacapilidis and Pappis (1996) tackle the integrated process of the warp making, starching and weaving machines and Pimentel *et al.* (2011) address the knitting process. To the best of our knowledge, there is little research concerning production planning in the spinning process. Nevertheless, a weak point regarding the industry efficiency is the difficulty to quickly devise good production plans, and overcoming this weakness can increase the competitiveness of companies.

Solution methods for this type of problems rely on MIP solvers or (meta)heuristics. Usually, the MIP solvers are exact methods that aim at finding provably optimal solutions. In some cases, the running time may be prohibitive. On the other hand, the (meta)heuristics provide feasible solutions faster, but the optimality proof is (often) not guaranteed. Recent research focuses on problem-oriented methods (James and Almada-Lobo, 2011), such as relax-and-fix (Dillenberger *et al.*, 1994) and fix-and-optimize (Sahling *et al.*, 2009) heuristics. Variables belonging to a problem partition are fixed to a certain value to create a reduced problem of an easier solution. These problem-oriented methods carry problem aspects into the solution procedure, as the variable partitions are inherent to the problem.

State-of-the-art MIP solvers incorporate heuristics within the exact methods (e.g. relaxation induced neighborhood search (Danna *et al.*, 2005), local branching (Fischetti and Lodi, 2003) and some variants) to find better solutions quickly. These hybrid methods follow the variable fixing ideas. Each branch-and-bound node provides information for the decision variables and then, a reduced problem is created by fixing some variables based on this information. In case the reduced problem returns a solution, it is also a solution to the original problem. The new solution can be injected into the branch-and-bound tree as a new upper bound. These improvement strategies can accelerate the optimality proof. The cross fertilization between exact and (meta)heuristic methods is called matheuristics (Maniezzo *et al.*, 2010).

In this chapter, we propose a mathematical model that integrates the lot-sizing and scheduling decisions of a spinning industry taking into account the synchronization between the first and second stages. The model is based on the multi-stage general lot-sizing and scheduling problem - MSGLSP. Equally important, the exact approach called HOPS (Hamming-Oriented Partition Search) is proposed as a new matheuristic based on mathematical programming to solve this problem. It involves the problem solution with the branch-and-bound method combined with a problem-oriented procedure that injects new and better upper bounds in the original problem. The hamming distance between the solutions found when exploring the branch-and-bound tree is used to define the problem partitions of stabilized values. The stabilized partitions are fixed to create the reduced problems and then, their solutions can be injected into the branch-and-bound tree of the original problem. This improvement procedure assists for a clever guidance to refine solutions. An instance generator is designed to conduct computational experiments in an extensive dataset. The HOPS validation in production planning in the spinning industry is also supported by real-world data based on instance settings.

Section 3.2 describes the production problem of the spinning industry which is then formulated by adapting the MSGLSP. Section 3.3 gives an overview of fixing variable matheuristics. In Section 3.4, the HOPS algorithm is introduced. Section 3.5 sets benchmarks for the results of HOPS compared to those obtained by other state-of-the-art mathematical program-based approaches available in commercial software. Some concluding remarks are given in Section 3.6.

3.2 The production system

The manufacturing system in a spinning industry can be defined as a procedure where fibers are processed and then used to make different types of yarns. This procedure is described by the sequence of processes represented in Figure 3.1. In the beginning, lints from various bales are mixed and blended together in order to make a uniform blend of fiber properties. The fiber blend is blown by air from a feeder through ducts to intermediate machines for cleaning and carding, which separate and align the fibers into a thin web. The web of fibers is then pulled through a conical device, providing a soft called sliver. The sliver is pulled again and then twisted to make it tighter and thinner until it complies with yarn specifications.

After spinning, the yarns are tightly wound around bobbins or tubes. The yarn packages are then ready for distribution.

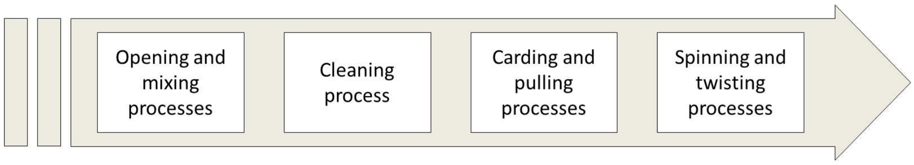

Figure 3.1: Sequence of processes in the production system of a spinning industry.

The spinning process distinguishes among the type of spinning, such as open-end, ring, air jet and Vortex spinning. In open-end spinning, the yarn is produced directly from the sliver, and consequently, the drawing process presented in ring spinning is eliminated. The air jet and Vortex spinning systems have also eliminated the need for the drawing process.

This study deals with open-end spinning where all the intermediate processes can work in a synchronized mode with the opening and mixing processes. The capacities of the first processes are sufficient to maintain the spinning process working. Therefore, the last process - spinning - is considered the production bottleneck.

In order to detail the production features, consider N yarns to be scheduled on M parallel machines over a finite planning horizon with given length T. The demand of yarns is known beforehand for each period of the planning horizon that should be met in case capacity is sufficient. Despite the production line operating 24 hours, on a 7 day week basis, delays can occur when demand for yarns is high, and so backlogs must be represented in the model.

A yarn can not be manufactured in a given time period unless a fiber blend that ensures its quality is also processed in this period. Therefore, planning of the first-stage product is also required. A fiber blend is generally used in several yarns, but a yarn is only made using a fiber blend that respects its quality limits. In each point in time, only one fiber blend can be processed in this kind of production line, thus all the machines must produce yarns that require the same quality of fibers. Production dependency between the two stages is represented in Figure 3.2.

The machines may differ in the processing rates of the same yarn, thus the fiber blend can be consumed at different speeds. A setup changeover from one yarn to another consumes capacity time which is dependent on the sequence in which the yarns are processed. The setup for yarns can be carried from one period to the next. The setup changeover for the fiber blend can be considered null as another fiber blend is immediately available for later use in the production. This fact reinforces the statement that the fiber blend sub-process is not the production bottleneck.

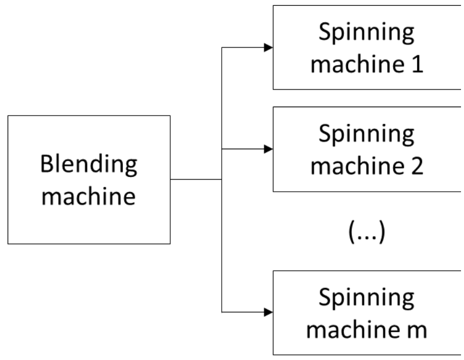

Figure 3.2: Diagram of two-stage production environment in a spinning industry.

The objective of devising the production plan is to determine the size and sequence the lots on all machines in order to minimize the sum of costs of backlogs, inventories and changeovers. An example of a production plan for the spinning case is illustrated in Figure 3.3. Five yarns of two different qualities of fibers are scheduled on three parallel machines over three planning periods. The dark gray rectangles represent the time wasted on setting up the machine for yarn production. The first two blends are dedicated to the same product family, that is, the set of yarns that requires the same fiber blend quality. On the other hand, the third blend is dedicated to a different product family.

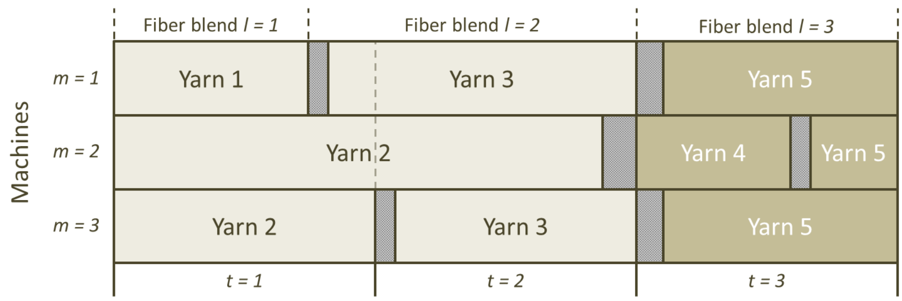

Figure 3.3: An illustrative production plan.

3.2.1 Mathematical model

In this section a MIP formulation is presented for the problem described before which is adapted from the Multi-Stage General Lot sizing and Scheduling Problem (MSGLSP) - introduced in Chapter 2. Backlogging must be taken into account as it is a common feature in the spinning industry. The MSGLSP divides the planning horizon into T periods. Each period is sub-divided into a number of micro-periods in which only one yarn can be produced as a small-bucket model (Meyr, 2002).

Therefore, the sequence of production is obtained in a rather straightforward manner allowing for the incorporation of sequence-dependent setups. The fundamental assumption is that a user-defined parameter restricts the number of micro-periods per period and, consequently, is an upper bound on the number of setups. However, the micro-periods have no predefined length. A micro-period is confined to be within a single period, and its length can not exceed the length of its period. Without loss of generality, we assume the length of each period to be equal to one. This assumption ensures some parameters to be standardized accordingly. The model makes use of the same micro-periods length across all machines (that is, a fixed time grid). This feature allows for an easy synchronization between the fiber blend (first stage) and the yarns produced by the spinning machine (second stage).

An optimal solution involves sizing and scheduling the production lots, respecting the constraints with the minimum setup, backlogging and holding costs. The solution to the general lot-sizing and scheduling problem - GLSP (Fleischmann and Meyr, 1997) and its extensions (e.g. MSGLSP) from a MIP solver is usually restricted to small size instances - see (Santos and Almada-Lobo, 2012). However, improvements such as reformulations provide tighter bounds and can speed up the problem solution or even partially overcome these limitations. For the lot-sizing problem, the SPL - simple plant location reformulation (Krarup and Bilde, 1977) is one of them - see (Gao *et al.*, 2008) for a comparative study. Hereafter, the SPL is used to reformulate the MSGLSP. Moreover, a stronger set of equations (3.8)-(3.10) that ensures a setup changeover in consecutive periods and micro-periods provides a much better alternative for the MSGLSP. Such constraints were first proposed for the DLSP - discrete lot sizing and scheduling problem (Belvaux and Wolsey, 2001).

In order to clearly present the model formulation, the notation is given as follows:

Indices	
$i,\ j = 1,\ \ldots,\ N$	yarns;
$k = 1,\ \ldots,\ K$	blend that ensures the quality of the family $S(k)$ of yarns;
$m = 1,\ \ldots,\ M$	spinning machines;
$r = 1,\ \ldots,\ R$	micro-periods per period;
$t, t' = 1,\ \ldots,\ T$	time periods.

The stronger formulation - sMSGLSP reads:

$$\textbf{Minimize} \quad \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{r=1}^{R}\sum_{t'=1}^{T} c_{itt'} \cdot X_{mitrt'} + \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{r=1}^{R} \sigma_{mij} \cdot Z_{mijtr} \quad (3.1)$$

Parameters	
$c_{itt'}$	cost for yarn i produced in period t to meet demand in period t';
σ_{mij}	setup cost of a changeover in machine m from yarn i to j;
d_{it}	demand of yarn i in period t;
C	maximum capacity (kg) per loading of the bale laydown machine - fiber blend process;
p_{mi}	processing time in machine m of one kilogram of yarn i;
s_{mij}	setup time of a changeover in machine m from yarn i to j;
$S(k)$	set of yarns that belong to family k.

Variables	
$X_{mitrt'} \geq 0$	production quantity in machine m of yarn i in micro-period r of period t to meet demand in period t';
Z_{mijtr}	takes on 1, if there is a changeover in machine m from yarn i to yarn j in period t and micro-period r; 0 otherwise;
$\mu^s_{tr} \geq 0$	start time of micro-period r in period t;
$\mu^f_{tr} \geq 0$	end time of micro-period r in period t;
$Y_{mitr} \in \{0,1\}$	takes on 1, if machine m is set up for yarn i in period t and micro-period r; 0 otherwise;
$U_{trk} \in \{0,1\}$	takes on 1, if fiber blend processed in period t and micro-period r meets the quality to produce yarns of the family k; 0 otherwise.

subject to

$$\sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{r=1}^{R} X_{mitrt'} = d_{it'} \qquad \forall i, t' \tag{3.2}$$

$$\mu^s_{t1} = t - 1 \qquad \forall t \tag{3.3}$$

$$\mu^f_{tR} = t \qquad \forall t \tag{3.4}$$

$$\mu^s_{tr} \geq \mu^f_{t(r-1)} \qquad \forall t, r > 1 \tag{3.5}$$

$$\mu^f_{tr} - \mu^s_{tr} \geq \sum_{i=1}^{N}\sum_{j=1}^{N}(s_{mji} \cdot Z_{mjitr}) + \sum_{i=1}^{N}\sum_{t'=1}^{T} p_{mi} \cdot X_{mitrt'} \qquad \forall m, t, r \tag{3.6}$$

$$X_{mitrt'} \leq d_{it'} \cdot Y_{mitr} \qquad \forall m, i, t, r, t' \tag{3.7}$$

$$Y_{mitr} = \sum_{j=1}^{N} Z_{mjitr} \qquad \forall m, i, t, r \tag{3.8}$$

$$Y_{mit(r-1)} = \sum_{j=1}^{N} Z_{mijtr} \qquad \forall m, i, t, r > 1 \tag{3.9}$$

$$Y_{mi(t-1)R} = \sum_{j=1}^{N} Z_{mijt1} \qquad \forall m, i, t > 1 \tag{3.10}$$

$$\sum_{i=1}^{N} Y_{mitr} = 1 \qquad \forall m, t, r \tag{3.11}$$

$$X_{mitrt'} \leq d_{it'} \cdot U_{trk} \qquad \forall m, i \in S(k), t, r, t', k \tag{3.12}$$

$$\sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{t'=1}^{T} X_{mitrt'} \leq C \cdot \sum_{k=1}^{K} U_{trk} \qquad \forall t, r \tag{3.13}$$

$$\sum_{k=1}^{K} U_{trk} \leq 1 \qquad \forall t, r \tag{3.14}$$

$$Y_{mitr} \in \{0,1\}\,;\; U_{trk} \in \{0,1\} \qquad \forall m, i, j, t, r, k \tag{3.15}$$

all other variables are non-negative and continuous. (3.16)

The objective function (3.1) is to minimize the sum of backlogging, holding and sequence dependent setup costs. Here, the amount produced before the delivery date is considered as inventory, given by $X_{mitrt'}$ in case $t < t'$. On the other hand, $X_{mitrt'}$ when $t > t'$ is the amount produced after the delivery date, that is, backlogging orders. Similarly, the costs $c_{itt'}$ refer to holding costs in case $t < t'$ and to backlogging costs in case $t > t'$. For $t = t'$, $c_{itt'}$ equals zero. This type of model is known as the *disaggregate formulation* (Brahimi *et al.*, 2006). Constraints (3.2) attempt to satisfy the demand, taking into account inventories and backlogging. Constraints (3.3)-(3.5) define the start and end times of each micro-period r and, therefore, its length. Constraints (3.3) block the starting time of the first micro-period to the beginning of the respective macro-period. Likewise, constraints (3.4) block the ending time of the last micro-period to the end of the respective macro-period. It should be highlighted that $\mu^f_{tR} - \mu^s_{t1} = 1$. This condition will be clear when the capacity limits are discussed later on. Constraints (3.5) avoid the micro-periods overlapping. The capacity limitation of the parallel spinning machines is implicitly expressed. A spinning machine works continuously during the period, therefore p_{mi} can be understood as the fraction of the period consumed on machine m to produce one kilogram of yarn i. Constraints (3.6) ensure that $p_{mi} \cdot X_{mitr}$ respects this capacity limitation, that is, the machine capacity, the processing rates and the setup times are normalized to 1. Moreover, they define the lower bound on each micro-period length based on the production and setup times of all machines. Furthermore, the same requirements determine the upper bound on the amount produced in each micro-period, and they set a fixed time grid of micro-periods for the machines. Observe that the model allows for idle time intervals which may take place within a micro-period or between micro-periods.

Constraints (3.7) ensure that production of a yarn in a micro-period occurs only if the machine is set up for it. Equations (3.8)-(3.10) link the setup variables to the changeover variables. By means of constraints (3.8), machine m can only produce yarn i in micro-period r of period t if and only if a changeover from another yarn j to yarn i takes place at the beginning of micro-period ($i = j$ is possible). Similarly, (3.9) and (3.10) make sure that if a changeover from yarn i to another yarn j takes place in a given micro-period, then the preceding micro-period has been set up for yarn i. Requirements (3.11) establish that each machine is set up for one and only one yarn in each micro-period. In each micro-period, all the machines produce yarns of the same family stated by (3.12). Constraints (3.13) restrict the production in a micro-period to the maximum size of the bale laydown. As only one machine is considered in the first stage, constraints (3.14) limit to one the number of families (or, in other words, the number of fiber blends) to be produced per micro-period. Constraints (3.15) and (3.16) express the variable domains.

Figure 3.4 illustrates a production plan based on the sMSGLSP variables. Three groups of active variables are shown. Recall that variables μ^s_{tr} and μ^f_{tr} indicate the start and finish of each micro-period. The fiber blend produced in each micro-period that supplies all the spinning machines is expressed by U_{trk}. Finally, variables Y_{mitr} represent which yarn is prepared for production. Naturally, the variables defining production quantities, inventory and backlogging levels, as well as the setup changeover are associated in a straightforward manner with the variables depicted in Figure 3.4.

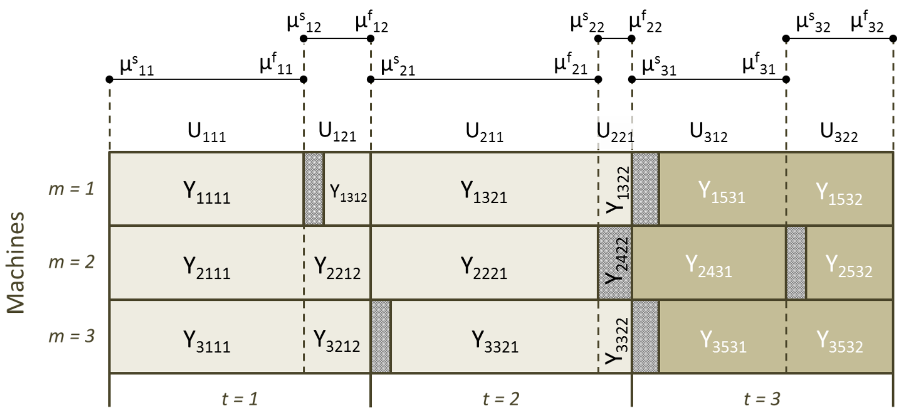

Figure 3.4: An illustrative production plan based on the decision variables.

For instance, the plan for machine 1 consists of six micro-periods and the yarn production sequence of 1-3-3-3-5-5. The sequences for machines 2 and 3 are 1-2-2-4-4-5 and 1-2-3-3-5-5, respectively. Note that a micro-period can allocate time for production and for changeover. Moreover, idle time is not illustrated in the example, but it may also occur within the micro-periods.

3.3 Solution methods

As a first attempt to address the sMSGLSP, one may rely on a MIP solver. However, as reported in Chapter 2, it is a hard problem and the exact methods embedded in such solvers can only find feasible solutions for small-sized instances. The MIP solver does not often find solutions for MSGLSP concerning medium and large instances, relevant in practice. Exact methods try to find an optimal solution and prove its optimality, but the respective running time increases exponentially with the instance size. On the other hand, a (meta)heuristic method can find a good solution in a limited time in spite of sacrificing the guarantee of finding optimal solutions. Hence, when choosing between an exact method or a (meta)heuristic method to solve this problem, one should consider the best option to ensure optimality and acceptable running times.

Combinations of exact and (meta)heuristic methods have been studied in depth by researchers. These combinations, usually referred to as hybrid metaheuristics, try to cross-fertilize different optimization strategies. Puchinger and Raidl (2005) classify the hybrid metaheuristic into two main groups. First, the collaborative combinations are algorithms that exchange information, but no algorithm is incorporated into another. This class can be sub-divided into those having sequential execution, intertwined or parallel executions. The second group contains the integrative combinations, in which the algorithm is a dependent embedded component of another algorithm. These are sub-divided into metaheuristics that incorporate exact algorithms or exact algorithms that incorporate metaheuristics. This second group of hybrid methods is also called Matheuristics (Maniezzo *et al.*, 2010).

From this matheuristics class, various approaches adopt the strategy of fixing a variable set to certain values in order to create a reduced problem. This approach decreases the number of free variables which makes the problem easier to solve. Soft and hard variable fixing methods have been developed to construct and improve solutions. On one hand, the soft variable fixing concept are characterized for the non-imposition of the set variables which are fixed. Fischetti *et al.* (2005) introduce the feasibility pump, in which a feasible solution is obtained from successive roundings based on the problem linear relaxation. As a soft fixing method, the roundings are done in a non-imposed variable set. The Local Branching - LB (Fischetti and Lodi, 2003) is an iterative local search method solving a reduced problem. The reduced problem is created by introducing a local branching constraint based on an incumbent solution. This constraint imposes that the difference of the 0-1 variables between the incumbent solution and an improved solution (measured by the Hamming distance) is limited to a given positive integer parameter. Therefore, the variables are soft fixed.

On the other hand, hard fixing methods impose which variables are fixed. The relax-and-fix heuristic as proposed in (Dillenberger *et al.*, 1994; Araujo *et al.*, 2007; Beraldi *et al.*, 2008; Ferreira *et al.*, 2009) solves the lot-sizing and scheduling problem as a sequence of partially relaxed sub-MIPs. Each sub-MIP retains the variables of some time-partitions with integrality constraints and the following partitions are relaxed. The integer variables are progressively fixed at their optimal values obtained in earlier iterations. Usually, relax-and-fix is used as a construction-heuristic. The fix-and-optimize heuristic, such as those presented in (Sahling *et al.*, 2009; Helber and Sahling, 2010; James and Almada-Lobo, 2011), is an iterative improvement method. At each iteration, the integer variables are fixed at their best value previously found, except for a limited set of integer variables. It relates to the original problem, but by fixing part of the solution, just a partition of the search space is explored. It requires two elements: the first is the original problem to be solved; the second is a partition scheme that can generate new solutions from a current incumbent solution. Besides the traditional time-partitions, James and Almada-Lobo (2011) implement product and machine-based partitions. However, the choice of which partition is to be optimized still requires attention. Problem-oriented methods, such as the relax-and-fix and fix-and-optimize heuristics, bring elements of the problem into the solution procedure, in these cases, partitions of the problem.

Another two hard fixing matheuristics, the Relaxation Induced Neighborhood Search - RINS (Danna *et al.*, 2005) and the Distance Induced Neighborhood Search - DINS (Ghosh, 2007) are used as improvement procedures within the branch-and-bound method in an effort to obtain a good feasible solution quickly, maintaining the optimality guarantee - advantage of the exact method. They share the core of large neighborhood search running under the branch-and-bound method. Moreover, their execution depends on the existence of a relaxation and/or an incumbent solution. In the RINS and DINS, a set of the decision variables is fixed to some bound and only the remaining variables are optimized by the MIP solver. If the MIP solver finds an improved solution using the reduced problem, it becomes the new incumbent of the original problem. The RINS idea is to occasionally devise a subproblem at any node of the branch-and-bound tree that corresponds to a neighborhood of an incumbent solution. The variables which have the same values in the incumbent and in the current solution of the linear relaxation are fixed and the reduced problem is then solved. Similarly, the DINS fixes the variables using a metric between the variables of the linear relaxation solution at the current node and the variables of the best known feasible solution. Hard fixing is done by RINS fixing every variable that has the same value as the linear relaxation. Finally, DINS plays around with hard and soft fixing as it fixes variables with an imposed and non-imposed way. A motivation for using this kind of matheuristics is also to produce a good upper bound early in the solution procedure and to prune as much of the branch-and-bound tree as possible. These algorithms exploit the fact that smaller problems can be easily generated from the original problem and can often be efficiently solved by exact methods such as MIP solvers. As mentioned before, the reduced problem can be defined in different ways and its solution may give better upper bounds to the original problem.

For our problem, the backlogging feature allows the MIP solver to provide a feasible solution quickly. Therefore, matheuristics that improve the upper bound, such as RINS, seem to be a good alternative to solve the sMSGLSP that includes backlogging. Bearing this in mind, a novel MIP improvement heuristic is proposed in the next section.

3.4 HOPS - Hamming-Oriented Partition Search

Our approach is oriented by problem features to select which set of decision variables is to be fixed in order to create the reduced problem. Following the problem-oriented methods (e.g. (James and Almada-Lobo, 2011)), the selection of the variable set is based on problem partitions clearly recognized from the model, such as periods, micro-periods, machines or products. The choice is performed following a ranking that identifies the most promising partitions to be optimized. The procedure to determine the potential of a partition to be improved is designated as Partition Attractiveness Measure, providing a clever partition choice. This idea resembles the consistent variables as defined by (Glover, 1977) to define the variables that repeatedly assume a particular value. In the partition case, it can be extended to the variable set that belongs to the partition. Thus, following the variable fixing concept, the HOPS is classified as a hard fixing method since it creates the reduced problem by freezing the values of certain variables.

To sum up, we bring and compile ideas to propose a new hybrid heuristic called HOPS (Hamming-Oriented Partition Search) that incorporates interesting strategies to solve large problems by reducing them to smaller and more tractable subproblems.

In order to make understandability easier, the algorithm scheme is divided into three parts. The first part illustrates the main method to be incorporated in the branch-and-bound method (Section 3.4.1). The function that defines the metric to choose which variable to be fixed is then introduced (Section 3.4.2). Finally, we show the necessary steps to fix the decision variables in order to create the reduced problem and to solve it (Section 3.4.3).

3.4.1 The HOPS core

Algorithm 2 presents the pseudo code that must be embedded in the branch-and-bound scheme. The HOPS - Hamming-Oriented Partition Search may be performed after the branch-and-bound processes each node. Let P be the problem to be solved by the MIP solver with HOPS:

$$P = \textbf{Minimize} \quad \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{r=1}^{R}\sum_{t'=1}^{T} c_{itt'} \cdot X_{mitrt'} + \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{r=1}^{R} \sigma_{mij} \cdot Z_{mijtr}$$
$$\textbf{subject to} \quad (3.2) - (3.16).$$

The search for improvements is performed every time a new incumbent x^0 is found throughout the exploration of the branch-and-bound tree. As explained before, x^0 can be improved by freeing one partition of this solution and optimizing the reduced problem. The partition choice relies on a ranking based on the frequency of the variable changes - belonging to that partition - in the past iterations. In other words, one can determine the attractiveness of each partition to be optimized by calculating a measure for the stability of the 0-1 variables throughout the search. Intuitively, one may say that every feasible solution found along the branch-and-bound tree contains parts of the optimal solution (or even the entire one). Therefore, there are partitions with stable variable values that might be part of the optimal solution and do not need to be optimized iteratively. On the other hand, unstable variables should be more frequently re-optimized to promote the convergence to the optimal solution. Some elements are required in the procedure and must be defined: let $\mathbf{z}$ be the smooth array that saves the history of the solutions obtained along the branch-and-bound tree. The array storage values are obtained using previous values of each decision variable. Each new feasible solution is incorporated into $\mathbf{z}$ with an exponential smoothing factor α. The aim is to smoothly integrate the solutions without loosing their memory. $\mathbf{z}$ is called smooth array as the most recent solutions are given more weight than the others, and the respective weights decay exponentially as their recency becomes lower. In order to create the reduced problem, the partitions to be considered by HOPS are listed in set $\mathcal{A}$. For sMSGLSP the example, $\mathcal{A}$ can contain machine, period, micro-period and/or product partitions. The partition selection is based on the Partition Attractiveness Measure H that determines the size of the distance of the new incumbent of each partition to the array $\mathbf{z}$.

Algorithm 2: HOPS

Data: Problem P, incumbent solution x^0, smooth array $\mathbf{z}$, parameter α, set of partitions $\mathcal{A}$, and Partition Attractiveness array H.
Result: A new feasible solution $\bar{x}$ to P.
1 **begin**
2   **while** *Branch-and-bound active* **do**
3   **if** x^0 *is a new incumbent to* P **then**
4       $H \longleftarrow PartitionAttractivenessMeasure(x^0, \mathbf{z}, \alpha, \mathcal{A})$
5       $\bar{x} \longleftarrow$ *Fix-and-Optimize*$(P, H, x^0, \mathcal{A})$
6   **if** $\bar{x}$ *is feasible* **then**
7   inject $\bar{x}$ in the branch-and-bound as a new incumbent

The HOPS core consists of two main procedures. First (line 4), in case the new incumbent found by the branch-and-bound is the best solution known, the attractiveness for optimization (H) of each partition is determined. Second (line 5), the reduced problem is then created based on the original problem, the new incumbent solution x^0 and the partitions to be fixed following the Partition Attractiveness Measure. The reduced problem is also solved by a MIP solver. In case a feasible solution to the reduced problem is found, then it is injected into the branch-and-bound tree of the original problem P as the new incumbent and the new upper bound. Therefore, when solving the reduced problem, better upper bounds are sometimes obtained for P. It is important to note that the optimal solution of the reduced problem is at least as good as the best solution obtained so far. By fixing variables of the partitions that usually have no (or few) changes along the branch-and-bound exploration, one may construct good partial solutions and the optimization of the remaining part of the solution may improve the upper bound faster.

3.4.2 Partition attractiveness measure

The choice of which variables should be fixed is the main difference between HOPS and other methods from the literature. RINS and DINS fix variables without considering the relationship between them. In contrast, our approach looks for the inherent variable relationship presented in the problem. The relationship is expressed as partitions of the problem, such as periods, products or machines. Let $\mathcal{A}_k$ be the index set of variables that belong to partition k. For the sMSGLSP example, in case only the machine partition is considered, $\mathcal{A} = \{\text{machine } m^1,\ \text{machine } m^2,\ \dots,$ $\text{machine } m^M\}$, where *machine* m^1 denotes the set of decision variables related to machine 1. This setting is applied in case other partitions are considered simultaneously with the machine partition. The procedure to define the attractiveness measure for each partition to be optimized is illustrated in Algorithm 3. Let $\mathcal{S} = \left\{j : x_j^0 = 1\right\}$ denote the index set of binary variables with a non-zero value of a new solution x^0 found by the branch-and-bound. Thus, $\mathcal{A}_k \cap \mathcal{S}$ are the variables of the partition k where $x_j^0 = 1$. $\mathcal{A}_k \setminus \mathcal{S}$ are those where $x_j^0 = 0$.

Algorithm 3: PartitionAttractivenessMeasure

Data: Incumbent solution x^0, smooth array $\mathbf{z}$, parameter α, and set of partitions $\mathcal{A}$.

Result: Partition Attractiveness array H updated.

1 **begin**

2   **foreach** *partition* $\mathcal{A}_k \in \mathcal{A}$ **do**

3         $H(\mathcal{A}_k) = \dfrac{\displaystyle\sum_{j \in (\mathcal{A}_k \cap \mathcal{S})} \left(x_j^0 - z_j\right) + \sum_{j \in (\mathcal{A}_k \setminus \mathcal{S})} z_j}{|\mathcal{A}_k|}$

4   **foreach** j **do**

5         $z'_j \longleftarrow \alpha \cdot x_j^0 + (1 - \alpha) \cdot z_j$

6         $z_j \longleftarrow z'_j$

The Partition Attractiveness Measure algorithm ranks the partitions every time a new feasible solution is encountered. Moreover, the history of found solutions (smooth array) must be updated. The first step (lines 2 and 3) returns the information on how much each variable value has changed compared to the feasible solution history. To this end, the hamming distance between the new incumbent and the smooth array $\mathbf{z}$ is used, and by grouping them one can quantify and rank the partitions with more changes. It is important to note that the same variable may belong to more than one partition. This grouping strategy defines the stable partitions, that is, the average hamming distance of all variables belonging to a partition defines its stabilization level. By means of comparing the solution history and the new solution, an attractiveness measure for each partition optimization is defined. The variable values of those partitions that have no (or few) changes are considered changeless, while the other part must be stabilized. Finally, the feasible solution history in smooth array z is updated with the new solution x^0 (lines 4, 5 and 6). The linear combination creates the history where each new x^0 has an exponential smoothing factor α over z. For example, let $\mathbf{z} = \{0, 0, 1, 0, 0.6, 0.4, 0, 0\}$ be the smooth array that must be updated with the new solution $x^0 = \{0, 0, 1, 0, 0, 1, 0, 1\}$ and exponential smoothing factor $\alpha = 0.6$. The new smooth array is $\mathbf{z'} = \{0, 0, 1, 0, 0.24, 0.76, 0, 0.6\}$.

3.4.3 Fix-and-Optimize

The aim of this section is to introduce a heuristic that optimizes the reduced problems. Similar to the fix-and-optimize heuristic used in James and Almada-Lobo (2011) or Sahling *et al.* (2009), our design creates the reduced problem $(P1)$ from the original one (P), fixing all but a predefined partition of the solution. The solution to reduced problem $P1$ respects all the constraints of the original problem. In addition, the objective function value is restricted to be lower than the upper bound $(f(x^0))$ and greater than the lower bound $(Z(P))$ in the original problem. This implies that a better solution either for P is obtained or the reduced problem delivers an infeasibility status. Last but not least, the variables are fixed, leaving out those belonging to the chosen partition. Algorithm 4 describes the pseudo code.

Algorithm 4: Fix-and-Optimize

Data: Problem P, Partition Attractiveness array H, incumbent solution x^0, and set of partitions $\mathcal{A}$.
Result: A new feasible solution $\bar{x}$ to P.

begin
  Define the partition $\mathcal{A}_k$ from $\mathcal{A}$ that must be improved
  Let ϵ be a small number
  Let $P1$ be a reduced problem from P with additional constraints:
    $x \longleftarrow x^0 \ \forall \ x \notin \mathcal{A}_k$
    $f(x) \leq f(x^0) - \epsilon$
    $f(x) \geq Z(P)$
  Solve$(P1)$
  if *a new solution* $\bar{x}$ *is obtained from* $P1$ **then**
    return a new feasible solution $\bar{x}$ to P
  else
    return $\emptyset$

Obviously, even using the Partition Attractiveness Measure to choose the better portions to be optimized, the selection of the partition variables requires a strong trade-off. The number of variables of the freed partition directly influences the solution improvement. In fact, it plays with diversification and intensification mechanisms. On the other hand, too large partitions might result in excessive running times to solve the reduced problem by the MIP solver. In order to avoid this scenario, the MIP solver is aborted when a certain time limit is reached and the best solution (if available) is considered.

3.4.4 Accelerators

A set of six features is presented in this section to boost the HOPS method over its core. Strategies to mitigate the computational effort and additional HOPS invocations are described next.

HOPS node invocation

By default the HOPS is initially invoked when the first MIP feasible solution is found. It can also be called at every node of the tree or several consecutive times at the same node. Each invocation involves the improvement attempt of one partition. Parameter $nl \geq 1$ indicates the number of nodes explored by the branch-and-bound between two consecutive invocations. In case the HOPS is invoked without a new feasible solution since the last call, the history of solutions found remains unchanged. Therefore, the procedure to update the smooth array $\mathbf{z}$ is skipped.

Unambitious choice

The choice of which partition to be fixed is based on its partition attractiveness measure. The higher the value of the attractiveness measure, the higher the instability of the partition and the respective priority to be fixed first. Preliminary tests showed a slight better performance of HOPS in case the partition choice is not pure greedy, that is, the selection is made from a restricted candidate list of the partitions based on the value of the attractiveness measure.

Prohibited partition list

With an incumbent solution x^0 at hand, the same reduced problem is generated consecutively if partition $\mathcal{A}_k$ does not change. By designing a prohibited partition list, the method invests computational efforts only in the unexplored partitions. Each explored partition is incorporated in the list avoiding the future generation of the same reduced problem. In case a new incumbent solution is found, the prohibited partition list is emptied.

Infeasibility invocation

Constraints limiting the upper and lower bounds in the reduced problem may accelerate its solution. On one hand, it can return an improved solution quickly. On the other hand, the reduced problem can be too tight and may return an infeasibility status.

We expect that each invocation returns one new improved solution, thus an infeasibility invocation is designed to generate another reduced problem in case an infeasibility response is returned in a minimum time.

Partition oscillating growth

The *prohibited partition list* and the *infeasibility invocation* allow for a quicker exploration of partition set $\mathcal{A}$. In case of failure to improve the incumbent solution, the partitions can be combined in pairs, triples or larger clusters to expand the search space. In case an improved solution is found, the partition combinations move back and the prohibited partition list is again emptied.

HOPS turn on/off

HOPS is turned on when the MIP exploration tree finds its first feasible solution. In case the *partition oscillating growth* does not retract, HOPS can be turned off. This adaptive tool allows the branch-and-bound to invest more time in exploring the original problem tree. It may speed up the lower bound improvement and/or find a new incumbent solution for the problem. In case of a new incumbent solution, the HOPS is turned on.

3.5 Computational results

In this section we present the computational tests performed to validate the HOPS method. The experimental results are compared to the results of the RINS and LB. These options for comparison aim to analyze the HOPS front results against other hard and soft variable fixing strategies.

Most of the practical cases present difficulties to obtain stable static data. The real-world case deliveries relative low number of instances and statical parameters. Therefore, to demonstrate the HOPS performance with respect to the solution quality over the problem, it is important to provide the ground for systematic testing. An instance generator was designed to supply an extensive set of instances that reflects the characteristics of real-world cases.

3.5.1 Problem instance generation

By drawing an instance generator, the parameters can be varied systematically. Moreover, it can generate a large number of problem instances with specific desired properties. Here, the problem is generated with a constructive procedure. Period after period, product setups are assigned to each machine, and then, the maximum level for production of each product is determined. The other main points to generate the instances are randomly designed, however their proportions are similar to the real data. Given a set of parameters N (number of yarns), K (number of families), M (number of machines) and T (number of periods), the remaining data is determined as follows:

- all yarns are randomly assigned to a product family;

- the set of yarns of the same family that might be allocated to each machine is randomly chosen;
- the setup times (with triangle inequalities respected) and the processing times for all yarns on each machine are randomly generated;
- all costs are derived from the opportunity cost per yarn package unit;
- the demand of a yarn and the capacity of the first-stage machine are chosen with respect to the capacity requirements.

In our tests, the evaluated instances have the following parameters. The number of products N is four, five, seven and nine, which can belong to two and three product families K. The number of spinning machines M considered is three, five, seven and nine. The number of periods T is equal to five and the number of micro-periods R is fixed to three. The generator was constructed to provide feasible solutions for the case without backlogging, that is, the whole yarn demand can be satisfied in the planning horizon. In order to evaluate the backlogging feature (presented in real-world cases), factors $df = \{1.0; 1.5; 2.0\}$ are adopted to inflate the yarn demand. For each combination $M/N/K/df$, ten different instances were randomly generated, making a total set $\mathcal{P}$ of $n_p = 840$ instances.

3.5.2 Algorithms tested

The MIP solver CPLEX has the RINS implemented and activated by default in version 12.1. Similarly, the Local Branching is part of the CPLEX package, but it must be activated by setting the parameter `IloCplex::LBHeur` to 1. HOPS was implemented in C++ and uses the CPLEX to solve the original and reduced problems. In order to ensure a fair comparison, RINS was disabled from CPLEX when running the LB and HOPS by setting the parameter `IloCplex::RINSHeur` to -1. No other CPLEX parameter was changed.

All computational tests were conducted on an Intel Xeon (2 GHz) with 5 Gb of random access memory, running under Linux. In all cases, the methods were unable to compute the optimal solution within a time limit of one CPU hour for each problem instance. Thus, the comparison is focused on the best solution found and the respective optimality gap considering the time limit of one hour.

Parameters for HOPS and its accelerators were defined by preliminary tests and are provided next. The reduced problems are created by freeing a variable set of the incumbent solution. This variable partition must be clearly recognized in the problem. For the sMSGLSP, the variable set of micro-periods and machines were used in the HOPS computational tests. The partition attractiveness, which determines which partition is freed to create the reduced problem, is measured from the smooth array **z**. In these tests, a new feasible solution x^0 is incorporated into the smooth array with an exponential smoothing factor $\alpha = 0.4$. The accelerator *Partition oscillating growth* is implemented allowing for the period partition implicit usage when the micro-period partition size increases. For the *HOPS node invocation*, the parameter to define the nodes explored between two consecutive invocations is set to $nl = 500$. The parameter of minimum time to solve a reduced problem required in *Infeasibility invocation* is set to 5 seconds. Nevertheless, the maximum time spent on searching for an improved upper bound is defined to be 36 seconds. Lastly, the HOPS is turned off after exploring the partition list twice without finding a new solution. A variation of the parameters used in the computational tests to validate HOPS was also tested. However, the parameter calibration was not the priority of this work. Its emphasis is rather on the presentation a of new matheuristic.

Despite CPLEX being able to provide the optimal solution (if it exists) for a problem, in some cases it is not possible with reasonable time. For the sMSGLSP problem and the instances provided in this validation, the optimality proof is not complete within the time limit. It is well-known that the newest CPLEX versions have non-deterministic exploration of the tree nodes and the truncated results may be divergent. Thus, each instance was run five times to dilute any possible dissonant solution. The results reported below are the average of the five runs.

The instances are grouped into three different sets following the factors that determine the yarn demand (df). Three classes are presented: Class A details the instances where $df = 1.0$, Class B and Class C lay out the instances where $df = 1.5$ and $df = 2.0$, respectively. This division makes the effectiveness of the different methods visible in three different yarn demand scenarios.

Three measures are used to evaluate the performance of the methods. After one hour of running time, the upper bound (best feasible solution found) and its respective optimality gap are collected. In the tables below, the ten different instances of each class $M/N/K/df$ are grouped together. On each line, "$\#UB$" means the number of instances for which the method found the best upper bound. In case the best result was found by two methods, it is written up for both. Similarly, "$\#GAP$" refers to the number of instances for which the method found the smallest relative gap. The relative gap provided by CPLEX is defined as $GAP_{p,s} = 1 - LB_{p,s}/UB_{p,s}$, where $LB_{p,s}$ is the lower bound and $UB_{p,s}$ the upper bound for the instance $p \in \mathcal{P}$ found by the method s after 1 hour of running time. The third measure, "$r_{p,s}$" depicts the aggregated relative performance of each method for each instance class and refers to the ratio of the solution found by each method over the best solution reported by the 3 methods all together. For the HOPS example, $r_{p,HOPS} = UB_{p,HOPS}/min(UB_{p,HOPS}, UB_{p,RINS}, UB_{p,LB})$. This metric is an average for the ten instances of each class. In the last line, "*Total #UB*" presents the percentage of instances in which the method found the best solution. "*Total #GAP*" is the percentage of instances for which the method reported the minimum gap. Subsequently, "*Total* $r_{p,s}$" is the ratio average of the 280 instances.

From the results shown in Table 3.1, HOPS achieves the best solution for 55% of 280 instances belonging to Class A. The "*Total* $r_{p,HOPS}$" value of 1.041 indicates that the solution values found by HOPS are, on average, 4.1% worse than the best solutions. On the other hand, the solutions from RINS and LB are, on average, 20.6% and 41.8% worse than the best solutions. The HOPS optimality gaps are as good as the RINS gaps. It is important to note that as the problem dimension increases, the superiority of HOPS over the other two methods seems to be clearer. For every instance of class 9/9/3/1.0, HOPS presents the best solution and gap.

Table 3.2 shows the results of Class B in which yarn demands are increased by factor $df = 1.5$. For this class, backlogging may occur to satisfy the demand constraints. Once again, HOPS reached the best objective value for 45% of 280 instances against 42% by RINS. Nevertheless, the optimality gap provided by HOPS is the best for 85% of the instances. These results indicate that the HOPS finds good solutions quicker and spends more efforts on improving the lower bound. In terms of solution quality, both HOPS and RINS found solutions not worse than 0.8% (on average) to the best solutions. Clearly, the LB method yields a poor relative performance for the whole set instances.

Finally, the results from Class C are shown in Table 3.3. Similarly to Class B, the solution qualities for both HOPS and RINS are equivalent and deviate on average 0.2% from the best solutions. However, HOPS outperforms RINS and LB to find the best solutions (53%) and the best optimality gaps (94%).

Below, in order to provide a general comparison for classes A, B and C, the performance profiles (Dolan and Moré, 2002) are plotted to compile the information from Tables 3.1, 3.2 and 3.3. Given the 840 instances of set $\mathcal{P}$ and $s \in \mathcal{S} = [HOPS,\ RINS,\ LB]$ methods under evaluation, as referred to before, the performance ratio is determined as:

$$r_{p,s} = \frac{UB_{p,s}}{min\,(UB_{p,HOPS};\ UB_{p,RINS};\ UB_{p,LB})}.$$

In addition, let $\rho_s\,(\tau)$ be the proportion of instances from $\mathcal{P}$ that were solved by the

Table 3.1: Comparison between HOPS, RINS and LB for Class A.

M	N	K	HOPS #UB	HOPS $r_{p,s}$	HOPS #GAP	RINS #UB	RINS $r_{p,s}$	RINS #GAP	LB #UB	LB $r_{p,s}$	LB #GAP
3	4	2	7	1.016	4	4	1.013	6	5	1.023	7
3	5	2	4	1.018	2	5	1.010	4	3	1.026	7
3	7	2	5	1.038	1	1	1.068	5	4	1.030	5
3	9	2	4	1.017	2	5	1.026	5	1	1.056	3
5	4	2	6	1.034	3	2	1.043	6	2	1.032	3
5	5	2	4	1.010	3	4	1.031	6	2	1.067	1
5	7	2	7	1.003	4	3	1.096	5	0	1.158	4
5	9	2	6	1.031	6	3	1.173	3	1	1.285	2
7	4	2	6	1.026	2	1	1.049	4	3	1.044	7
7	5	2	3	1.055	3	6	1.050	6	1	1.097	2
7	7	2	7	1.023	8	3	1.092	4	0	1.376	0
7	9	2	5	1.085	4	4	1.187	5	1	1.481	1
9	4	2	4	1.030	2	3	1.058	6	3	1.071	4
9	5	2	4	1.023	3	4	1.082	5	2	1.072	4
9	7	2	6	1.083	6	3	1.100	4	1	1.484	1
9	9	2	5	1.027	4	5	1.138	6	0	2.198	1
3	5	3	5	1.016	2	2	1.051	5	5	1.012	6
3	7	3	7	1.025	3	0	1.130	3	3	1.048	4
3	9	3	1	1.034	3	1	1.040	5	8	1.016	6
5	5	3	5	1.020	3	3	1.081	4	2	1.076	5
5	7	3	6	1.153	4	4	1.093	7	0	1.300	0
5	9	3	5	1.082	4	3	1.296	3	2	1.267	3
7	5	3	7	1.017	6	2	1.128	3	1	1.368	2
7	7	3	5	1.079	7	2	1.451	3	3	1.798	3
7	9	3	6	1.132	6	4	1.324	5	0	2.772	0
9	5	3	7	1.024	8	3	1.110	4	0	1.389	0
9	7	3	8	1.041	8	2	2.244	2	0	3.057	0
9	9	3	10	1.000	10	0	2.593	0	0	3.099	0
Total			55%	1.041	43%	29%	1.206	44%	19%	1.418	29%

method s with relative performance within a factor $\tau \geq 1$, that is, $\rho_s(\tau)$ is expressed as:

$$\rho_s(\tau) = \frac{|\mathcal{P} : r_{p,s} \leq \tau; p \in \mathcal{P}|}{|\mathcal{P}|}.$$

Figure 3.5 summarizes the comparison of the performance over set $\mathcal{P}$. By taking into account all 840 instances, HOPS found the best solution for 50.9% of the instances, RINS for 35.8% of the class and LB for 16.6%. With regards to solution robustness, HOPS has proved to be by far the best. For 95% of the instances, HOPS provides solutions that dist less than 7.9% of the best solutions. On the other hand, RINS and LB solutions are up to 26.0% and 66.6% away from the best solutions, respectively. In the worst case, when $\rho_s(\tau)$ is equal to 100, there is a problem p such that $r_{p,HOPS} = 1.79$ (that is, HOPS is 79% away from the best solution). For the RINS and LB cases, $r_{p,RINS}$ achieves 6.55 and $r_{p,LB}$ achieves 8.88 (not shown in the trucated graphic).

Table 3.2: Comparison between HOPS, RINS and LB for Class B.

			HOPS			RINS			LB		
M	N	K	#UB	$r_{p,s}$	#GAP	#UB	$r_{p,s}$	#GAP	#UB	$r_{p,s}$	#GAP
3	4	2	7	1.002	9	5	1.002	4	1	1.004	3
3	5	2	7	1.002	8	4	1.002	6	2	1.005	5
3	7	2	7	1.000	10	1	1.011	2	4	1.004	4
3	9	2	5	1.004	7	3	1.006	3	2	1.007	3
5	4	2	2	1.004	9	4	1.004	3	5	1.002	4
5	5	2	4	1.004	9	2	1.002	1	4	1.001	1
5	7	2	4	1.004	8	4	1.002	4	2	1.016	1
5	9	2	5	1.008	9	5	1.005	1	0	1.034	1
7	4	2	3	1.004	8	4	1.004	4	3	1.004	1
7	5	2	7	1.003	10	0	1.008	0	3	1.015	0
7	7	2	7	1.008	10	3	1.015	1	0	1.034	0
7	9	2	4	1.013	9	6	1.011	3	0	1.047	0
9	4	2	3	1.007	9	5	1.006	0	2	1.007	1
9	5	2	5	1.007	9	3	1.014	4	2	1.016	0
9	7	2	2	1.013	8	7	1.003	2	1	1.027	1
9	9	2	3	1.021	8	6	1.006	3	1	1.064	0
3	5	3	4	1.001	9	5	1.004	0	2	1.004	4
3	7	3	4	1.004	10	2	1.012	1	4	1.006	3
3	9	3	4	1.008	6	4	1.006	2	2	1.012	4
5	5	3	4	1.005	9	5	1.01	0	1	1.016	2
5	7	3	2	1.021	7	7	1.004	3	1	1.026	1
5	9	3	3	1.017	7	7	1.003	4	0	1.025	1
7	5	3	7	1.006	9	1	1.015	3	2	1.016	1
7	7	3	5	1.010	9	3	1.007	3	2	1.032	1
7	9	3	6	1.006	10	4	1.008	0	0	1.055	0
9	5	3	3	1.013	8	6	1.008	3	1	1.024	0
9	7	3	2	1.015	7	7	1.009	4	1	1.081	0
9	9	3	6	1.010	8	4	1.034	2	0	1.141	0
Total			45%	1.008	85%	42%	1.008	24%	17%	1.026	15%

HOPS is indeed more robust and can provide better solutions than RINS and LB. Moreover, Figure 3.6 illustrates the solution value evolution along the search for HOPS, RINS and LB on a particular instance of Class C. We recall that the backlogging feature benefits the CPLEX to provide a quick feasible solution. All methods start from the same solution at node 0. HOPS is fast in improving the solution not only at the beginning of the run (before 600 seconds), but also throughout the one hour of computational time.

3.6 Final remarks

This chapter presents an integrated industrial problem at a spinning industry. The first-stage process deals with cotton blending and the second stage yarn production. The first-stage production must be synchronized with the second-stage

Table 3.3: Comparison between HOPS, RINS and LB for Class C.

			HOPS			RINS			LB		
M	N	K	#UB	$r_{p,s}$	#GAP	#UB	$r_{p,s}$	#GAP	#UB	$r_{p,s}$	#GAP
3	4	2	6	1.000	9	5	1.000	6	3	1.000	5
3	5	2	5	1.001	9	5	1.001	3	1	1.000	4
3	7	2	7	1.000	8	2	1.002	5	2	1.001	5
3	9	2	5	1.002	9	5	1.001	5	1	1.002	3
5	4	2	5	1.000	9	3	1.000	1	4	1.000	1
5	5	2	6	1.001	10	2	1.002	2	2	1.001	2
5	7	2	5	1.002	9	3	1.001	3	2	1.004	3
5	9	2	5	1.001	8	3	1.002	2	2	1.004	2
7	4	2	6	1.001	10	3	1.001	3	1	1.002	3
7	5	2	6	1.001	9	3	1.000	6	1	1.003	3
7	7	2	4	1.002	9	5	1.003	1	1	1.008	1
7	9	2	4	1.005	9	5	1.003	4	1	1.014	0
9	4	2	5	1.001	10	3	1.002	4	2	1.005	1
9	5	2	7	1.001	10	2	1.002	1	1	1.005	1
9	7	2	6	1.001	10	3	1.004	1	1	1.011	0
9	9	2	3	1.003	10	6	1.003	4	1	1.018	1
3	5	3	6	1.001	10	1	1.002	4	3	1.001	3
3	7	3	7	1.001	10	2	1.004	3	1	1.003	1
3	9	3	7	1.000	10	2	1.004	3	1	1.005	1
5	5	3	8	1.000	10	1	1.003	3	1	1.006	2
5	7	3	4	1.003	10	3	1.003	2	3	1.008	2
5	9	3	4	1.004	10	5	1.004	6	1	1.011	2
7	5	3	5	1.002	10	5	1.002	1	0	1.006	2
7	7	3	4	1.005	10	5	1.002	2	1	1.010	2
7	9	3	4	1.007	9	5	1.005	2	1	1.027	1
9	5	3	8	1.002	10	2	1.005	1	0	1.006	0
9	7	3	4	1.010	8	5	1.004	4	1	1.014	2
9	9	3	2	1.014	7	8	1.002	6	0	1.044	0
Total			53%	1.002	94%	36%	1.002	31%	14%	1.008	19%

product, in product quality terms. Moreover, the problem becomes harder with the occurrence of sequence-dependent setup features. Mathematical modeling aims to determine the production plan of the first and second stages with a minimum setup changeover, inventory and backlogging costs while trying to meet the required demand due dates.

This two-stage lot-sizing and scheduling problem is approached by a new matheuristic. The HOPS - Hamming-Oriented Partition Search is an exact method incorporating a heuristic in which a fix-and-optimize procedure finds better feasible solutions to boost the branch-and-bound procedure. The Partition Attractiveness Measure is used to choose the variable set that is considered unstable and can potentially improve an incumbent solution. The new solution is injected into the branch-and-bound tree and it can accelerate the optimality proof.

A large set of instances assuming proportions which are similar to real-world data

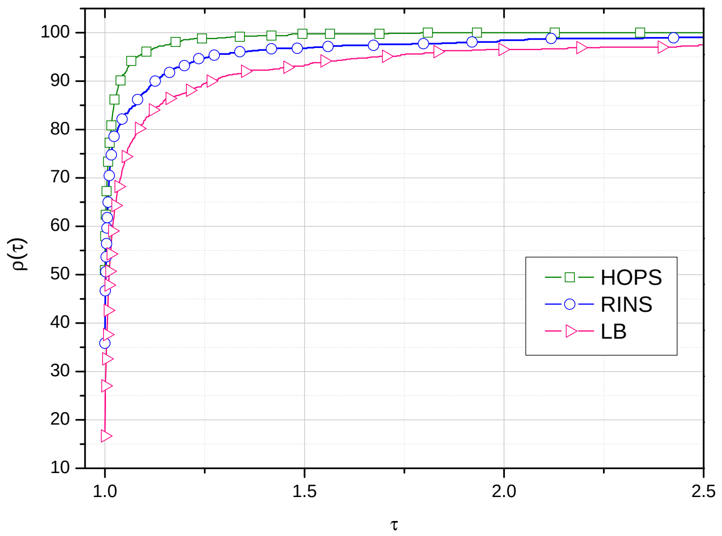

Figure 3.5: Performance profiles.

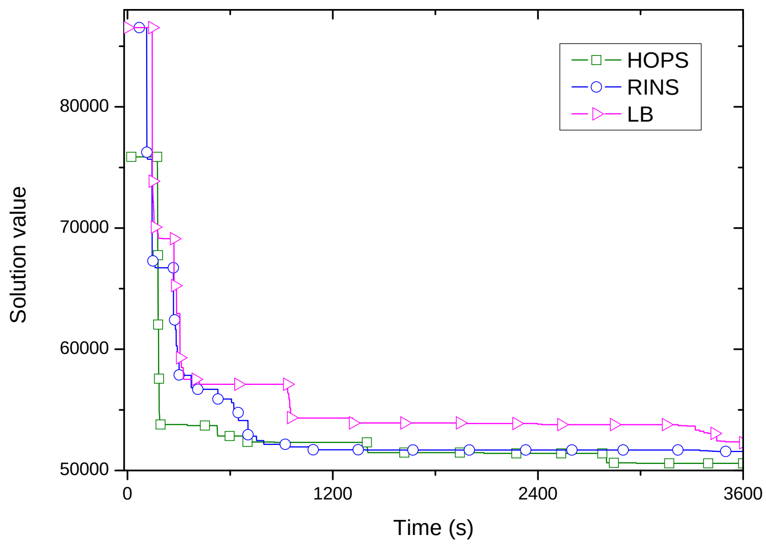

Figure 3.6: Solution quality evolution over time for a Class C example.

was designed to validate the new matheuristic. A comparison of HOPS against commercial implementations of improvement matheuristics shows that it outperforms both RINS and LB in obtaining good feasible solutions within a certain time limit. The computational results support the argument that in the context of problems addressing production backlogging, such exact methods with improvement procedures are good options for providing better solutions. The backlogging feature makes it easier to find an initial solution, which is necessary to trigger the method. HOPS results show that a problem-oriented improvement procedure can provide better solutions for problems with a known partition structure (in this case, machine, period, micro-period and product).

HOPS was designed to solve the production planning in the spinning industry. Nevertheless, some of the ideas can be applied to different lot-sizing and scheduling problems or to other problems not inked to this research subject. The main requirement is to recognize "natural" partitions of the problem. We believe HOPS may serve as a guidance for solving many applications.

In the next chapter, the production planning problem incorporates blending decisions of the cotton bales. Some approaches are analyzed to discuss the relevance of the integration. The integrated model is compared against a hierarchical approach in which the lot-sizing and scheduling decisions are taken a priori and feed the blending problem.

Chapter 4

Integrated lot-sizing, scheduling and blending decisions in spinning industry

This chapter presents a study of the relevance of coordinating production and blending planning in a spinning industry. The considered scenario concerns a particular plant that produces a number of package yarns over a planning horizon. Each yarn type is produced using a blending of a set of cotton bales that must contain attributes to ensure the quality of the produced yarns. A set of cotton bales of the inventory is selected. Two approaches to manage these operations are compared, one in which the production scheduling and the blending problems are solved in a hierarchical way, and another in which they are integrated within a single model. The two approaches are applied to real-based case and the respective performances are analyzed. Furthermore, a third procedure that combines features from the hierarchical and integrated approaches is proposed. The latter can better deal with the lot sizing, scheduling and blending in the spinning industry.

4.1 Introduction

As previously discussed in Chapter 3, the production planning problem arising at a spinning industry aims to determine the size and sequence of the yarn production lots. Moreover, a set of cotton bales must be determined to provide a fiber blend that ensures the quality attributes to produce the demanded yarns. These decisions involve the lot-sizing and scheduling, and blending problems. The lot-sizing and scheduling problem determines the timing, level and sequence of production to satisfy yarn demands over a time horizon. To represent the production environment, setup, inventory and capacity related constraints must be respected. On the other hand, the blending problem determines the set of cotton bales used on each blending to supply the spinning machine to produce the yarns. These problems appear at a two-stage production system in which the fiber blend is produced in the first stage (opening-blending machine) and the yarns are produced in the second stage on several spinning machines. There is a quality relationship between the first and second stages. The yarns have attribute specifications that can be achieved by an appropriated blend of fibers (El Mogahzy, 2004). The specifications are related to the fiber attributes, such as grade color, trash percentage area, fiber length and others. In order to achieve the yarn requirements, each blend must be set by a group of cotton bales in which the fibers ensure the specification. The yarns belong to product families that differ in the attributes. Two blend loadings for the same yarn family should have a minimal difference of their attributes. Otherwise, in case of a high attribute variability, yarns can present different unwanted features. This variation causes production problems at the next level of the textile supply chain. Clearly, these constraints show the dependence of the raw cotton blending in the yarn production. Crama *et al.* (2001) also point out the high importance of dealing the raw material together with the production planning in the process industries.

The integrated lot-sizing and scheduling problem has received attention in the literature in light of its relevancy to real world problems. This integrated approach is discussed in Chapter 2 for process industries and in Chapter 3 for the spinning industry. In practice, the lot sizing and scheduling and blending are hierarchically determined. First, the lot-sizing and scheduling decisions are taken. Then, the loading by loading of the blend is preformed to meet the quality specifications. This strategy can be considered myopic as it does not take into account the attribute variation in the stored bales that can influence future blending decisions. The integrated production-blending problem aims to draw a production plan with a better control on the attribute variation.

Jans and Degraeve (2008) call the attention for the research growth on modeling the production process with specific features presented in real world problems. Some studies integrating more than one process of the production chain emphasize the benefits achieved with this integrated approach. Poltroniere *et al.* (2008) model an integrated lot-sizing and cutting stock problem for the paper industry and solve it with the help of two heuristics in a decomposed manner. The aim is to reduce the waste cost in the cutting process by drawing good jumbo production plans. The integrated lot-sizing and cutting stock problem is study of other two papers. Gramani *et al.* (2009) achieve production plans with reduction of setup and cutting costs but the inventory costs increase. Furthermore, Malik *et al.* (2009) conclude that an improved customer service levels and low total costs can be obtained by this integrated lot-sizing and cutting stock approach. Chandra and Fisher (1994) investigate the value of coordinating production and distribution planning. The integration can achieve the reduction from 3-20 per cent in total operating cost. On the other hand, Amorim *et al.* (2012) investigate the integrated production-distribution problem of the perishable goods. The shelf-life is discussed as part of the objective which influences the production and distribution decisions. Santos and Almada-Lobo (2012) propose a synchronized model to the pulp and paper production and digester control. The authors highlight that some bottlenecks at the pulp and paper mills can be identified when the industrial process is approached in an integrated way.

To the best of our knowledge, there are no existing methods which have been developed to solve the integrated lot-sizing, scheduling and blending problem. Three approaches to manage this operation are compared in this chapter: one in which the production scheduling and blending problems are solved separately (in a hierarchical way), another in which they are coordinated within a monolithic model, and the last is a partial integrated approach. In our hierarchical design, the production schedule that minimizes the setup, inventory and backlogging costs is determined to meet the yarn demand per period. Then, the cotton bales from the inventory are selected to ensure the quality of the blend loadings scheduled previously. The integrated model attempts to simultaneously define the production plan and the cotton bales for each blend loading. In other words, the quality of the produced yarns using each blend is taken into account when drawing the production plan. Third, the partial integrated approach first solves the production scheduling along with a few blending constraints and the cotton bales are then selected in a second step.

The comparison between the approaches is made on a real-based instance problem. The result analysis indicates that a feasible production planning can be obtained only if blending related constraints are taken into account. Our work provides a systemic method to assist the decision maker on planning the production and enduring the yarn quality.

The next section introduces the problem context in the spinning industry. The integrated lot sizing, scheduling and blending is defined in Section 4.3. The developed integrated model is also shown. An analysis to compare the integrated and hierarchical approaches are reported in Section 4.4. Section 4.5 introduces a partial integrated approach that carries features from both integrated and hierarchical ones. Remarks and future works are addressed in Section 4.7.

4.2 The spinning industry

As discussed before, a spinning industry can produce different types of yarns with specifications that are determined by the customers. The breach of yarns specifications causes customer dissatisfaction and loss of product price and productive efficiency of the spinning. The success of a spinning industry can be measured in the quality of the produced yarns and its manufacturing costs, and therefore both criteria can determine the competitiveness of a company. The yarn manufacturing costs are measured by the raw material and production process costs. According to Admuthe and Apte (2009), the raw material costs can reach 70 per cent of the cost price.

In an open-end spinning, as studied in this chapter, the raw cotton is purchased in bales. The production flow to transform the raw cotton to yarns is depicted in Figure 4.1. These two stages require the quality synchronization between the fiber blended at the first stage and the yarns spun at the second stage.

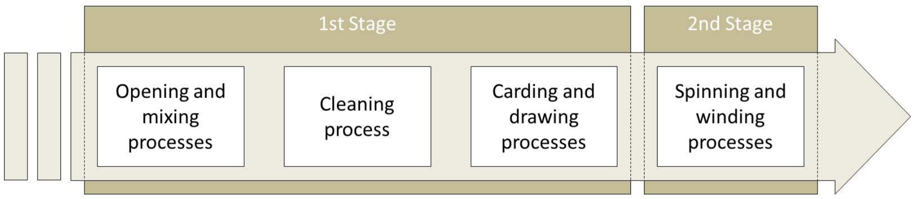

Figure 4.1: Sequence of processes and stages considered in the yarn production.

The production should be determined by a plan that informs the machine and the time instant in which each demanded yarn must be produced. By respecting the stage synchronization, the planning must contemplate informations about the fiber blend, that is, which yarn family is to be produced by each blend loading and its starting and finishing times. With the blend sequence at hand, the decision maker must select the cotton bales to compound the fiber blend that ensures the quality specification for each yarn attribute. A mathematical model to represent this integrated lot-sizing, scheduling and blending problem is discussed in the next section.

4.3 Integrating the lot sizing, scheduling and blending

This section proposes a mathematical model for the integrated lot-sizing, scheduling and blending problem in the spinning industry. The problem solution draws up the production plan in which the blend loadings are sequenced and their required qualities are determined. Moreover, the yarn amount (and its sequence) is established to be produced on each machine obeying the blending quality. The other part of the plan must define the cotton bales selected to be the ingredient of the blend loading. The planning aims to meet the required quality for the yarns and should minimize production costs and the attribute variability between blends. The mixed-integer programming presented below is an integration of the lot-sizing and scheduling problem and the blending problem. The lot-sizing and scheduling constraints are based on the multi-stage hybrid lot-sizing and scheduling model proposed in (Camargo *et al.*, 2012b). The blending constraints match some ideas from (Zago, 2005).

4.3.1 Lot-sizing and scheduling problem definition

The spinning can produce different types of yarns on several parallel machines. The production plan must define the production for each yarn in each period over a finite planning horizon. The demand for the yarns is known and should be met in case capacity is sufficient. Delays can occur if demand for yarns is high, and so backlogs must be represented in the model.

The production of an yarn in a given time period imposes that a fiber blend that ensures its quality is also processed in that bucket. Thus, the planning of the second stage requires the planning of the first stage. The yarns belong to families related to the required blend and its quality specifications. A fiber blend is generally used in several yarns, but a yarn is made of only one fiber blend. In each point in time, only one fiber blend can be processed on this kind of production line, thus, all the machines must produce yarns of the same family.

The machines may differ in the processing rates of the same yarn, thus the fiber blend can be consumed in different speeds. A setup changeover from one yarn to another consumes capacity time dependent on the sequence in which the yarns are processed. The setup for yarns can be carried from one period to the next. The setup changeover for the fiber blend can be considered null as another fiber blend is immediately available for later use in the production.

The simple plant location reformulation (Krarup and Bilde, 1977) of the multi-stage hybrid lot-sizing and scheduling constraints from (Camargo *et al.*, 2012b) is developed in Section 4.3.3. The problem solution draws up the production plan in which the blend loadings are sequenced and their required qualities are determined. Moreover, with synchronized quality, the yarn amount (and its sequence) to be produced on each machine is established.

4.3.2 Blending problem definition

The raw material used in the yarn production can be natural fibers or synthetic fibers. This issue is focused on a spinning industry in which the single raw material used is cotton. The cotton is purchased in bales from different vendors and each bale has its own attributes, which directly influence the quality and specification of the yarns produced. In practice, a yarn specification can be achieved by the blend of cotton fibers even if the bales have fibers of different attributes. Such fiber attributes have being gathered using the HVI (High Volume Instruments) system that systematically classifies the cotton fibers. Color grade, fiber length, micronaire, strength, length uniformity index, and trash percent area are some of the attributes given by an HVI system when classifying of cotton bales. If the decision maker knows the attributes required for yarn production (and the manufacturing process tolerances), and has at his disposal cotton bales with different attributes, it is possible to select the best combination of bales at low cost. In case these informations are not available, the decision maker is induced to purchase cotton bales of high quality at high cost to ensure that production requirements are met. On the other hand, in case the decision maker elects to purchase lower-cost cotton bales that have uncertain quality, the uncontrolled attributes in the cotton blend may increase the number of broken yarns during the production process. This fact, besides reducing the yarn quality, increases the demand for staff to correct the failures caused by disruptions. Moreover, the high number of disruptions decreases the usage of the potential maximum speed in the spinning process. All these factors can raise the manufacturing costs that may achieve the yarn maximum selling price, making the whole process economically unfeasible. By skirting these difficult issues, successful yarn producers improve their gains by means of a proper selection and blend of the raw materials and an optimal adjustment of machinery (Fryer *et al.*, 1996).

The blending problem is defined by (El Mogahzy, 2004) and (El Mogahzy *et al.*, 2004) as the process to combine different fiber attributes to achieve an homogeneous blend. Its importance is enhanced through a fiber statistical analysis in (El Mogahzy, 2004) that testifies the attributes that affect the yarn quality. The attribute specification at a blending must respect a predefine quality range. The definition of which attributes are crucial to achieve the yarn specifications is not consensual. Waters and Philips (1961) consider fiber length, strength, fineness, yarn count, and grade index the essential attributes. On the other hand, Kang *et al.* (2000) looks at fiber length, strength and fineness; and Harpa (2008) notice tenacity, fiber length, fineness and fiber uniformity as crucial. As one can see, the authors (consequently, the industries) are not harmonized to determine the attributes necessary to define the yarn quality. Nevertheless, supported by empirical analysis, El Mogahzy *et al.* (2004) validate the yarn production using a controlled variation of different attributes. By knowing this propriety, a quality blend can be replicated avoiding high variability between blends. One of the most common spinning problems that often result from high variability between blending is the so-called fabric *barré*. The problem is also described by the periodic variation in the weft direction, that is, yarns of the same type with color difference can produce a bicolor single piece of cloth. Thus, it shows the importance of minimizing the variation of the quality attributes between two consecutive fiber blends. According to El Mogahzy (2005), controlled blends can *reduce* the attribute variation related to the raw material.

The issue is clearly viewed in Figure 4.2 adapted from (Camargo *et al.*, 2012b). The yarn 2 is produced on machine $m = 2$ using the fiber blends $l = 1$ and $l = 2$. In theory, when two fiber blends with the same quality attributes are scheduled consecutively, they can be considered as a bigger one. However, besides the fiber blends $l = 1$ and $l = 2$ meeting the quality specification, the attributes can differ from one fiber blend to another. The difference of attributes can cause serious problems on the supply chain downstream. Therefore, while controlling the quality requirements, the blending problem can have different goals as reducing the raw material cost and minimizing the attribute variability between blends, among others.

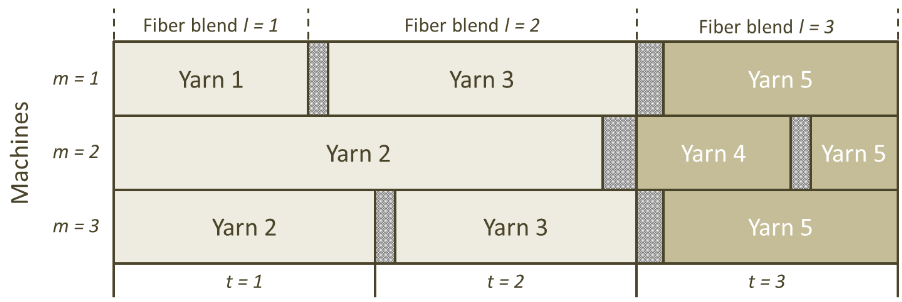

Figure 4.2: Illustrative production plan a spinning industry.

In practice, the bale selection is performed by the planner looking exclusively to the last loading, that is, the decision maker tries to repeat the last loading of the same quality. In case any bale fails to repeat the last loading, another bale with similar attribute is selected to compose the blending. It is important that the vital attributes of a blend loading resembles a previous loading of the same blending. This control seeks to maximize the efficiency of the spinning machines and to produce yarns of the demanded quality. However, this selection decision ignores future loadings by applying a short-term greedy selection.

A common strategy to allow for the repetition of the quality of a blend is to keep in the inventory the percentage of each attribute within the composition. By doing so, the attribute variations of new cotton bales are minimized as they are smoothly distributed across the loadings. Very few works have attempted to develop mathematical models of such blending problem to cotton bales. Greene *et al.* (1965) propose an initial modeling in which the objective is to minimize the raw material costs and decision variables are linear, that is, part of a cotton bale can be selected to mount the blend. Besides the classical quality constraints, Zago (2005) develops a formulation that considers the entire bale, and consequently, the decision variables are constrained by integer values. Ideas from the Zago's formulation are applied to our problem. The point of this chapter is to select among the wide number of stored bales (with different attributes) a set that meets the quality specification and allows for the reproduction with minimum (or no) variation.

4.3.3 The integrated formulation

In this section, the integrated lot-sizing, scheduling and blending problem is described. The indices, parameters and decision variables are defined as follows.

Indices	
$i = 1, \ldots, N$	yarn types;
$k = 1, \ldots, K$	yarn families - yarn set with the same blend quality;
$m = 1, \ldots, M$	spinning machines;
$t = 1, \ldots, T$	periods;
$l = 1, \ldots, L$	blend loadings available in the planning horizon;
$g = 1, \ldots, G$	color grade (fiber attribute).

Parameters	
$c_{itt'}$	cost of producing in period t one kilogram of yarn i to fulfill the demand of period t';
σ_{mij}	setup cost of a changeover on machine m from yarn i to yarn j;
r_l	cost of one kilogram of residue of the lth blend loading;
d_{it}	demand (kilograms) of yarn i in period t;
s_{mij}	setup time of a changeover on machine m from yarn i to yarn j;
p_{mi}	production time of one kilogram of yarn i on machine m;
C	opening-and-blending machine capacity (kilograms) for one blend loading;
$S(k)$	yarn set associated to the same family k;
w	weight of one cotton bale;
ϵ	small number;
eo_g	initial bales inventory level of color grade attribute g;
qeo	total amount of bales in the initial inventory;

q	amount of cotton bales for a blend loading: $q = \lceil C/w \rceil$;
sp_{gk}	maximum percentage of the color grade attribute g in the blend of type k;
ip_{gk}	minimum percentage of the color grade attribute g in the blend of type k;
fo_{gk}	amount of bales with color grade g used in the last blend loading of type k;
v_{gk}	maximum variation allowed to the color grade attribute g in the blend of type k (in percentage).

Variables	
$X_{mitlt'}$	production (kilograms) on machine m of yarn i in period t using the lth blend loading to meet demand in period t';
Y_{mijtl}	takes 1, if a changeover occurs on machine m from yarn i to j in period t using the lth blend loading; 0 otherwise;
R_l	raw cotton residue (kilograms) of the lth blend loading;
U_{lk}	takes 1, if the lth blend loading satisfies the quality to yarn family k; 0 otherwise;
u_l^s	starting time of the lth blend loading;
u_l^f	finishing time of the lth blend loading;
μ_{mitl}^s	production starting time on machine m of yarn i in period t using the lth blend loading;
μ_{mitl}^f	production finishing time on machine m of yarn i in period t using the lth blend loading;
α_{mitl}	takes 1, if machine m is set-up for production of yarn i in period t using the lth blend loading; 0 otherwise;
P_{tl}^s	takes 1, if the lth blend loading starts before the end of period t (that is, instant time t); 0 otherwise;
P_{tl}^f	takes 1, if the lth blend loading ends after the begin of period t (that is, instant time $t-1$); 0 otherwise;
$\mathcal{A}_g$	variation (in percentage) of bales with color grade g in the inventory;
$\mathcal{H}_{gk}$	variation of bales with color grade g between blends of type k;
D_k	number of blend loadings of type k needed for the production;
B_k	takes 1, if one or more blend loadings of type k is used; 0 otherwise;
Qef	total amount of bales in the final inventory;
Ef_g	inventory level of bales with color grade g;
F_{gk}	amount of bales with color grade g used in the blend loading of type k.

$$\textbf{Minimize} \quad \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{l=1}^{L}\sum_{t'=1}^{T} c_{itt'} \cdot X_{mitlt'} + \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{l=1}^{L} \sigma_{mij} \cdot Y_{mijtl}$$
$$+ \sum_{l=1}^{L} r_l \cdot R_l + \sum_{g=1}^{G} \mathcal{A}_g + \epsilon \cdot \sum_{k=1}^{K}\sum_{g=1}^{G} \mathcal{H}_{gk} \tag{4.1}$$

Subject to:

Lot-sizing and scheduling constraints:

$$\sum_{k=1}^{K} U_{lk} \leq 1 \qquad \forall l \tag{4.2}$$

$$u_l^s \leq u_l^f \leq T \qquad \forall l \tag{4.3}$$

$$u_l^f = u_{(l+1)}^s \qquad \forall l < L \tag{4.4}$$

$$\sum_{m=1}^{M} \sum_{t=1}^{T} \sum_{l=1}^{L} X_{mitlt'} = d_{it'} \qquad \forall i, t' \tag{4.5}$$

$$\mu_{mitl}^f - \mu_{mitl}^s \geq \sum_{j=1}^{N} (s_{mji} \cdot Y_{mjitl}) + p_{mi} \cdot \sum_{t'=1}^{T} X_{mitlt'} \qquad \forall m, i, t, l \tag{4.6}$$

$$\sum_{i=1}^{N} \sum_{l=1}^{L} \left(\mu_{mitl}^f - \mu_{mitl}^s \right) \leq 1 \qquad \forall m, t \tag{4.7}$$

$$\mu_{mitl}^s \geq \mu_{mjtl}^f + Y_{mjitl} - 1 \qquad \forall m, i, j, t, l \tag{4.8}$$

$$\sum_{t'=1}^{T} p_{mi} \cdot X_{mitlt'} \leq \sum_{j=1}^{N} Y_{mjitl} + \alpha_{mitl} \qquad \forall m, i, t, l \tag{4.9}$$

$$\sum_{j=1}^{N} Y_{mjitl} + \alpha_{mitl} = \sum_{j=1}^{N} Y_{mijtl} + \alpha_{mit(l+1)} \qquad \forall m, i, t, l < L \tag{4.10}$$

$$\sum_{j=1}^{N} Y_{mjitL} + \alpha_{mitL} = \sum_{j=1}^{N} Y_{mijtL} + \alpha_{mi(t+1)1} \qquad \forall m, i, t \tag{4.11}$$

$$\sum_{i=1}^{N} \alpha_{mitl} = 1 \qquad \forall m, t, l \tag{4.12}$$

$$\sum_{j=1}^{N} Y_{mjitl} \leq 1 \qquad \forall m, i, t, l \tag{4.13}$$

$$\sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{t'=1}^{T} X_{mitlt'} + R_l = C \cdot \sum_{k=1}^{K} U_{lk} \qquad \forall l \tag{4.14}$$

$$X_{mitlt'} \leq d_{it'} \cdot U_{lk} \qquad \forall m, i \in S(k) \quad \forall t, l, t', k \tag{4.15}$$

$$\sum_{t=1}^{T} \sum_{i=1}^{N} \left(\mu_{mitl}^f - \mu_{mitl}^s \right) \leq u_l^f - u_l^s \qquad \forall m, l \tag{4.16}$$

$$\mu_{mitl}^f - T \cdot \left(P_{tl}^s + P_{tl}^f - 1 \right) \leq \mu_{mitl}^s \leq \mu_{mitl}^f + T \cdot \left(P_{tl}^s + P_{tl}^f - 1 \right) \qquad \forall m, i, t, l \tag{4.17}$$

$$max\,(t-1; u_l^s - T\cdot(1-P_{tl}^s)) \le \mu_{mitl}^s \le min\left(t; u_l^f + T\cdot\left(1-P_{tl}^f\right)\right) \qquad \forall m,i,t,l \quad (4.18)$$

$$max\,(t-1; u_l^s - T\cdot(1-P_{tl}^s)) \le \mu_{mitl}^f \le min\left(t; u_l^f + T\cdot\left(1-P_{tl}^f\right)\right) \qquad \forall m,i,t,l \quad (4.19)$$

integrating constraints:

$$D_k = \sum_{l=1}^{L} U_{lk} \qquad \forall\ k \quad (4.20)$$

blending constraints:

$$\left|\frac{eo_g}{qeo} - \frac{Ef_g}{Qef}\right| = \mathcal{A}_g \qquad \forall\ g \quad (4.21)$$

$$D_k \le L\cdot B_k \qquad \forall\ k \quad (4.22)$$

$$(ip_{gk} - 1 + B_k)\cdot q \le F_{gk} \le (sp_{gk} + 1 - B_k)\cdot q \qquad \forall\ g,\ k \quad (4.23)$$

$$\left|\frac{F_{gk} - fo_{gk}}{q}\right| \le \mathcal{H}_{gk} + (1-B_k) \qquad \forall g,\ k \quad (4.24)$$

$$\mathcal{H}_{gk} \le v_{gk} \qquad \forall\ g,\ k \quad (4.25)$$

$$\sum_{g=1}^{G} F_{gk} = q\cdot B_k \qquad \forall\ k \quad (4.26)$$

$$Qef = qeo - \sum_{k=1}^{K} q\cdot D_k \quad (4.27)$$

$$Ef_g = eo_g - \sum_{k=1}^{K} D_k\cdot F_{gk} \qquad \forall\ g \quad (4.28)$$

$$B_k \in \{0,1\}\,;\ F_{gk} \in \mathbb{Z}_+ \qquad \forall\ g,\ k \quad (4.29)$$

$$Y_{mijtl} \in \{0,1\}\,;\ U_{lk} \in \{0,1\}\,;\ P_{tl}^s \in \{0,1\}\,;\ P_{tl}^f \in \{0,1\} \qquad \forall m,i,j,t,l,k \quad (4.30)$$

all other variables are non-negative and continuous. (4.31)

Objective function (4.1) aims to minimize the backlogging, inventory, changeover and residue costs and the variations of the attributes in the inventory and between blends. Here, the amount produced before the delivery date is considered as inventory, given by $X_{mitrt'}$ in case $t < t'$. On the other hand, $X_{mitrt'}$ for $t > t'$ is the amount produced after the delivery date, that is, backlogging orders. Similarly, the production costs $c_{itt'}$ refer to holding costs in case $t < t'$ and to backlogging costs in case $t > t'$. For $t = t'$, $c_{itt'}$ equals to zero, as production of period t meets demand of the same period. Variable $\mathcal{A}_g$ holds the inventory variation of bales with color grade g. Similarly, $\mathcal{H}_{gk}$ holds the variation of bales with color grade g between blends of type k. The constraint group (4.2)-(4.4) defines the schedule of the blend loadings. Each blend loading is constrained (4.2) to meet the quality for at most one yarn family. Constraints (4.3) and (4.4) avoid the overlapping of the starting and finishing blend loadings. Note that a blend allocation to a loading is allowed without production. Naturally, it generates residues that are detected in the second-stage requirements. The constraint (4.5)-(4.13) define the second-stage production system. The equations (4.5) are the stock balance constraints. Requirements (4.6) establish the time used to set up the machine and to produce the yarn. Observe that, together with (4.7), the production slots are confined to a period of size one. The confinement is made by normalizing X_{mitl} with p_{mi}. Similarly, s_{mji} refers to the period fraction wasted to set up the machine m. Constraints (4.6) and (4.7) allow for machine idle times inter and between slots. The production slots sequences and the sub-sequence removal are defined by (4.8). The flow of the machine setup is guaranteed with (4.9)-(4.13). Production is ensured (4.9) by setting up the machine or carrying a previous configuration over periods or blends. The setup carried over blend loadings and periods is represented in (4.10) and (4.11), respectively. The constraint group (4.14)-(4.19) integrates the first and second stage constraints. Constraints (4.14) define the all-or-nothing production for the blending machine. Specifically, the total amount of the blend is used either for production or for residue. Requirements (4.15) allow for yarn production that belongs to the family of the blend loading in case the latter exists. Constraints (4.16)-(4.19) define the useful production slots, that is, when $P^s_{tl} = P^f_{tl} = 1$. The reader is referred to Section 2.4 for additional information about the P variable concept. Constraints (4.17) are active when $P^s_{tl} = 0$ or $P^f_{tl} = 0$. In this case, the lth blend loading does not occur in period t and $\mu^s_{mitl} = \mu^f_{mitl}$, forbidding the production in that slot. Constraints (4.18)-(4.19) define lower bounds on the production slot starting time when $P^s_{tl} = 1$, and upper bounds on the slot ending time when $P^f_{tl} = 1$.

Constraints (4.20) aim to integrate the lot-sizing and scheduling constraints (4.2)-(4.19) with the blending constraints (4.21)-(4.28). They count the number of blend loadings of each type k needed to meet the production plan.

The next constraint group (4.21)-(4.28) relates to the blending problem. They promote the cotton bale selection that ensures the quality specification. Constraints (4.21) allocate the variation of the composition percentage of each attribute in the inventory by means of the variables $\mathcal{A}_g$. In Constraints (4.22), B_k takes on one, if one or more blend loadings of the type k are used. Constraints (4.23) determine quality limits for the attribute g of the blend type k. Constraints (4.24) account for the variation of each attribute g between blends that attempt the same yarn family. Constraints (4.25) limit the difference of the number of bales to an user predefined value. The total amount of bales of each blend loading is accounted by (4.26). The total amount of bales consumed over the planning horizon is determined by (4.27).

Similarly, equations (4.28) define the number of stored bales of each attribute at the planning horizon end. Constraints (4.29)-(4.31) enforce the binary, integrality and non-negative requirements for the variables.

The aforementioned model is non-linear due to requirements (4.21) and (4.28). The following remark shows how such feature can be tackled.

Remark 2. *As D_k and F_{gk} are decision variables, constraints (4.21) and (4.28) are non-linear. Given the equations (4.28) and the assumption $Qef = qeo - \sum_{k=1}^{K} q \cdot D_k$, the requirements (4.21) can be rewritten as:*

$$\sum_{k=1}^{K} D_k \cdot \left| F_{gk} - q \cdot \frac{eo_g}{qeo} \right| = \bar{\mathcal{A}}_g \qquad \forall\ g, \tag{4.32}$$

where $\bar{\mathcal{A}}_g$ denotes the deviation between the planed F_{gk} and the expected usage of the attribute g.

Let $Z_k^{a^k}$ be one, if $D_k = a^k$; and 0 otherwise, where $a^k = 0,\ \ldots,\ L$. That is, the integer number D_k is codified as a binary summation as follows:

$$D_k = \sum_{a^k=0}^{L} a^k \cdot Z_k^a \qquad \forall\ k; \tag{4.33}$$

$$\sum_{a^k=0}^{L} Z_k^{a^k} = 1 \qquad \forall\ k. \tag{4.34}$$

Finally, (4.32) are defined as:

$$\begin{aligned} & a^1 \cdot \left| F_{g1} - q \cdot \frac{eo_g}{qeo} \right| + a^2 \cdot \left| F_{g2} - q \cdot \frac{eo_g}{qeo} \right| + \ldots + a^K \cdot \left| F_{gK} - q \cdot \frac{eo_g}{qeo} \right| \\ & \leq \bar{\mathcal{A}}_g + \left(K - Z_1^{a^1} - Z_2^{a^2} - \ldots - Z_K^{a^K} \right) \cdot M \qquad \forall\ g,\ a^1,\ a^2,\ \ldots,\ a^K. \end{aligned} \tag{4.35}$$

Remarks 3 and 4 consider generalizations for the previous formulation. Extensions to deal with raw cotton residue and to manage the quality of more than one fiber attribute can be represented as follows.

Remark 3. *We assume in model (4.1)-(4.31) that the opening-and-blending machine capacity is completely used. It is represented in constraints (4.14) by adding a variable R_l to accommodate the non-used cotton. The costs for this residue are properly added in the objective function.*

Without loss of generality, the residue can also be used for make-to-stock[1] *pro-*

[1]For a reading about combination of make-to-stock and make-to-order production in process industries, see (Soman *et al.*, 2004).

duction. This specificity can appear in practice. Thus, we define X^{MTS}_{mitl} *to be the make-to-stock production in machine* m *of yarn* i *in period* t *using the* l*th blend loading. On the other hand,* $X^{MTO}_{mitlt'}$ *represents the make-to-order production as previously discussed. A couple of constraints should be incorporated to enforce these production cases:*

$$X_{mitl} = X^{MTS}_{mitl} + \sum_{t'=1}^{T} X^{MTO}_{mitlt'} \qquad \forall\ m,\ i,\ t,\ l; \tag{4.36}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{t=1}^{T} X^{MTS}_{mitl} = R_l \qquad \forall\ l. \tag{4.37}$$

The total production is used in constraints (4.36) for make-to-stock and for make-to-order decisions. Constraints (4.37) link the raw material residue to the make-to-stock production. For this assumption, some constraints must be changed accordingly to represent the make-to-order and make-to-stock productions. Constraints (4.5), (4.6) and (4.9) are respectively replaced by:

$$\sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{l=1}^{L} X^{MTO}_{mitlt'} = d_{it'} \qquad \forall i, t'; \tag{4.38}$$

$$\mu^{f}_{mitl} - \mu^{s}_{mitl} \geq \sum_{j=1}^{N}(s_{mji} \cdot Y_{mjitl}) + p_{mi} \cdot X_{mitl} \quad \forall m, i, t, l; \tag{4.39}$$

$$p_{mi} \cdot X_{mitl} \leq \sum_{j=1}^{N} Y_{mjitl} + \alpha_{mitl} \qquad \forall m, i, t, l. \tag{4.40}$$

Constraints (4.38) ensure the fulfilling of the yarn demand. Requirements (4.39) and (4.40) accommodate the new production variables X_{mitl}*. Naturally, the objective function have to be updated.*

Remark 4. *In case more than one fiber attribute must be managed, some constraints and variables have to be changed in the model. For instance, the company policy required to control the color grade* ($g = 1, \ldots, G$) *and fiber length* ($b = 1, \ldots, B$) *attributes. The inventory level variable is reformulated to incorporate the fiber length attribute* (Ef_{gb}) *and to inform the number of bales with color grade* g *and fiber length* b *in the inventory at the end of the planning horizon. Similar reformulations are done in variables* F_{gbk} *and parameters* eo_{gb} *and* fo_{gbk}*. Moreover, variables* $\mathcal{A}_g$ *and* $\mathcal{H}_{gk}$ *are specific for the color grade attribute and must be replicated for the fiber length, as well as the parameters* v_{gk}*,* sp_{gk} *and* ip_{gk}*. With these assumptions, it is rather straightforward to accommodate in the model constraints related to additional* *attributes.*

4.4 Model validation analysis

The validation for the integrated lot sizing, scheduling and blending is supported on the comparison with a hierarchical approach. The integrated model can be decoupled into a lot-sizing and scheduling sub-model and the blending sub-model by ignoring the constraints (4.20): $D_k = \sum_{l=1}^{L} U_{lk},\ \forall\ k$. Figure 4.3 illustrates the steps for a hierarchical solution of the problem.

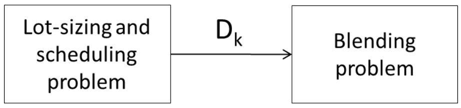

Figure 4.3: Step flow for the hierarchical production-scheduling and blending.

First, the hierarchical production planning determines the lot sizing and scheduling. The sequence of the blends defines the number of blends needed to carry out the production, that is, D_k is a input data for the blending definition. Thus, the blending solution gives the set of bales that satisfy the quality specifications for the yarns. The lot-sizing and scheduling formulation reads as:

Lot sizing and scheduling:

$$\textbf{Minimize} \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{l=1}^{L}\sum_{t'=1}^{T} c_{itt'} \cdot X_{mitlt'} + \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{l=1}^{L} \sigma_{mij} \cdot Y_{mijtl} + \sum_{l=1}^{L} r_l \cdot R_l$$

Subject to:

(4.2) – (4.19)

$$Y_{mijtl} \in \{0,1\}\,;\ U_{lk} \in \{0,1\}\,;\ P^s_{tl} \in \{0,1\}\,;\ P^f_{tl} \in \{0,1\} \qquad \forall m,i,j,t,l,k$$

all other variables are non-negative and continuous.

Moreover, the blending problem is formulated as follows:

Blending:

$$\textbf{Minimize} \quad \sum_{g=1}^{G} \mathcal{A}_g + \epsilon \cdot \sum_{k=1}^{K}\sum_{g=1}^{G} \mathcal{H}_{gk}$$

Subject to:

(4.21) – (4.28)

$B_k \in \{0, 1\}\,;\ F_{gk} \in \mathbb{Z}_+ \qquad\qquad \forall\ g,\ k$

all other variables are non-negative and continuous.

The comparison of the integrated and hierarchical approaches is undertaken with a data instance based on the real-world case. The dataset of the cotton bale inventory, blending specifications and the settings of the latest applied loadings are real. The raw material inventory is composed by 3,952 cotton bales. In addition, the company policy considers four attributes as crucial to define the blending: supplier, color, leaf grade and short fiber index (SFI), respectively, with 18, 3, 2 and 2 different possible values. The other parameters related to the production environment are generated as described in Section 3.5, but their proportions are similar to the real data. The number of yarns N is five and they belong to two product families K. Moreover, K is the number of different types of blends. The number of spinning machines M is three. The number of periods T is equal to five and the maximum number of blend loading along the planning horizon L is fixed to six. The opening-blending machine has the capacity set to 100 bales of 200 kilograms. In order to evaluate the influence of the yarn orders on the quality of the produced blends, factors df equal to 1.0 and 1.5 are adopted to the yarn demand. Instances of $df = 1.0$ are generated so that machine capacity is enough to produce the total demand for yarns.

The models are generated in OPL language and solved by the CPLEX mixed-integer solver version 12.1. Tests have been conducted on an Intel Xeon computer at 2 GHz with 20 GB of RAM. The running time was limited to one hour for all tests.

Several limits to the variability between blends (VBB) are checked. VBB means v_{gk} in the mathematical formulation. However, it equally limits all attribute variation. The comparison of the integrated and hierarchical approaches relies on the production costs and on the variation of the attributes in the inventory. Figures 4.4 and 4.5 must be analyzed together. As one can note, independent on the VBB limit, production costs given by the hierarchical approach are constant. The production plan is defined at the first step (lot-sizing and scheduling problem) and do not consider any information about the blending requirements. For VBB limits greater than 0.07, the results of the integrated model resembles those from the hierarchical approach. On the other hand, the integrated approach find solutions with higher production costs for VBB limited to 0.07 or less. By looking to the variation of the attributes in the inventory (Figure 4.5), the hierarchical approach is not able to find solutions at the blending step when the VBB is limited to 0.07 or less.

By analyzing both approaches, in case the company policies requires yarn production with hard quality constraints, the hierarchical approach corresponds to a trial and error approach. In case D'_k provides an unfeasible solution, a new lot-sizing and scheduling solution is requested with an additional constraint to avoid D'_k. However, the integrated approach deliveries the optimal D_k respecting the hard quality constraint. Blending requirements are met waiving the best production decision. This fact can be viewed in Figure 4.4, where high costs represent the backorder values.

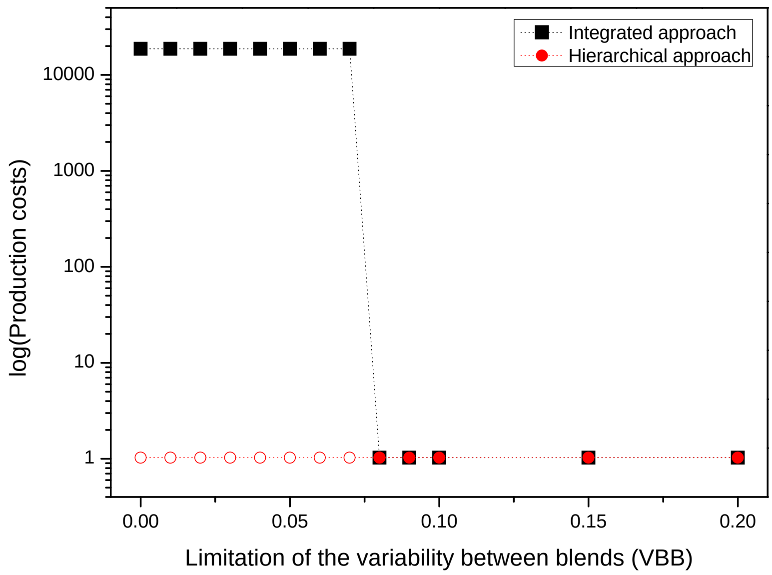

Figure 4.4: Integrated versus hierarchical approach - production cost behavior.

As one can see in Figure 4.5, the hierarchical approach deliveries blends without variation of the attributes in the inventory if D_k is feasible. The integrated approach should also delivery blends without variation. However, the values reported by the integrated approach can be explained. The blending decisions have low weight in the objective function of the integrated model, thus they have low priority in the optimization process. This high difference between decisions in the objective function aims to focus the optimization on decisions with higher weight. Decisions of lower weight are used as tiebreaker of solutions with similar decisions of higher weight. Theoretically, CPLEX solver is able to give the optimal solution. However, decisions of lower weight have little influence over the optimality gap. Solvers should be set with low relative MIP gap tolerance to find the decisions of lower weight of optimal values. Nevertheless, there is an inherent trade off between optimality proof and running time. Besides the solution values of the integrated approach shown in Figure 4.5 are in the MIP gap tolerance, clearly, the values are not the optimal blends. The optimality is not covered by the relative MIP gap tolerance.

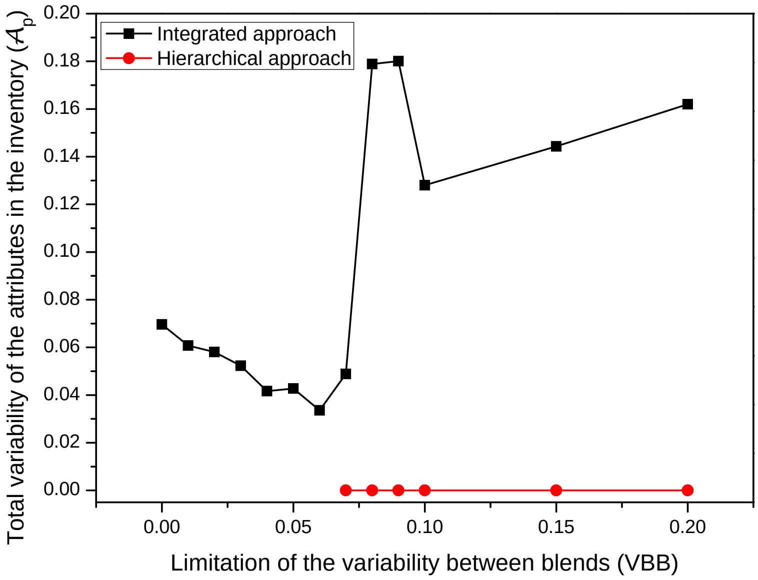

Figure 4.5: Integrated versus hierarchical approach - attribute variability in the inventory.

In addition, it is worth noting in Figure 4.5 that the hierarchical approach only deliveries blends without variation of the attributes in the inventory by chance. The cotton bale inventory has quality enough to ensure the minimum variation between blends VBB less than 0.07 without variation of the attributes in the inventory. Then, solutions in case VBB is limited to 0.08 are dominated. It is legitimate that the variation of the attributes in the inventory increases when the limitation of the variability between blends (VBB) decreases.

The best of both worlds, the integrated and hierarchical approach, can better achieve the objectives defined for the integrated lot sizing, scheduling and blending.

4.5 Partial integrated approach for the production scheduling and blending decisions

The partial integrated approach for coordinating production and blending planning attempts to carry features from the aforementioned integrated and hierarchical approaches. The aim is to provide the lot-sizing and scheduling problem with blending constraints to foster production plans with hard quality requirements. On the other hand, specific decisions to minimize the attribute variation can be found by the blending model without the optimality gap hurdle of the integrated approach. The partial integrated approach for the production scheduling and blending is written as following:

$$\textbf{Minimize} \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{l=1}^{L}\sum_{t'=1}^{T} c_{itt'} \cdot X_{mitlt'} + \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{l=1}^{L} \sigma_{mij} \cdot Y_{mijtl} + \sum_{l=1}^{L} r_l \cdot R_l \tag{4.41}$$

Subject to:

Lot-sizing and scheduling constraints: (4.2) – (4.19)

Some blending constraints: (4.22) – (4.26)

$$B_k \in \{0,1\}\,;\ F_{gk} \in \mathbb{Z}_+ \qquad \forall\ g,\ k$$

$$Y_{mijtl} \in \{0,1\}\,;\ U_{lk} \in \{0,1\}\,;\ P^s_{tl} \in \{0,1\}\,;\ P^f_{tl} \in \{0,1\} \qquad \forall m,i,j,t,l,k$$

all other variables are non-negative and continuous.

As one can see, the partial integrated production and blending planning takes into account some of the blending requirements. Minimal variations between blending are ensured but those related to the attribute variation in the inventory are skipped. In the same way as done in the hierarchical approach, the blending sequence ($\sum_{l=1}^{L} U_{lk} = D_k$) defines the input data for the blending model. However, unlike the hierarchical one, the partial integrated approach provides feasible parameters D_k. Thus, the blending solution is able to give the set of bales that satisfies the quality specifications as the feasibility is tested in advance. Note that the blending model is the same as that of the hierarchical approach:

Blending:

$$\textbf{Minimize} \quad \sum_{g=1}^{G} \mathcal{A}_g + \epsilon \cdot \sum_{k=1}^{K}\sum_{g=1}^{G} \mathcal{H}_{gk}$$

Subject to:

(4.21) – (4.28)

$B_k \in \{0,1\}\,;\;\; F_{gk} \in \mathbb{Z}_+ \qquad\qquad\qquad\qquad \forall\; g,\; k$

all other variables are non-negative and continuous.

The partial integrated approach is also analyzed. Figure 4.6 depicts both the production costs and the variation of the attributes in the inventory. Note that the green star points refer to production costs and the blue triangle points to the total variability of the attributes in the inventory.

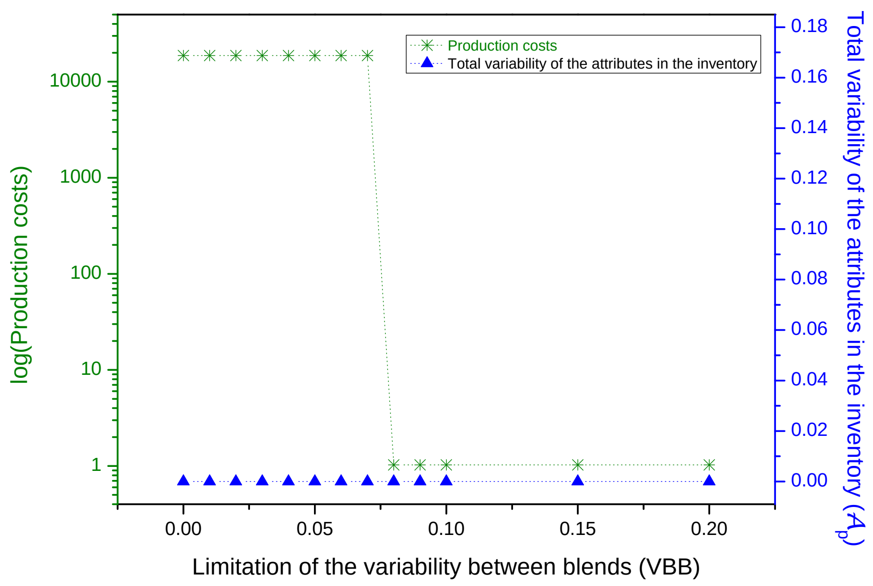

Figure 4.6: The partial integrated approach - attribute variability in the inventory and production cost behavior.

From the results, the partial integrated approach can determine feasible production plans with hard quality specifications and can define the best set of cotton bales. As one can see, the production costs resemble those delivered by the integrated approach and the attribute variation caused in the inventory is optimal as that from the monolithic blending problem (compare Figure 4.6 against Figures 4.4 and 4.5).

4.6 Illustrative example

The usage of the models is illustrated by a small example provided in Tables 4.1, 4.2 and 4.3. The raw material inventory is composed of 730 cotton bales. Two types of fiber blends are managed in the example. The inventory and quality limits for the attribute color grade is defined in Table 4.3, as well as the latest blend loadings. The maximum variation allowed by the company policies to the color grade between blends is 7%.

The number of machines is $M = 3$ and the common resource capacity is $C = 20,000$ kilos. The machine 1 is set up for product $i = 1$ at the beginning of the planning horizon and the machines 2 and 3 are set up for product $i = 2$. The planning horizon entails $T = 3$ time periods. Products 1, 2 and 3 require the same attributes and belong to product family $k = 1$, whilst products 4 and 5 are part of product family $k = 2$. Inventory ($c_{itt'}|t < t'$) and backlogging costs ($c_{itt'}|t > t'$) are equal for all periods. Production rates (p_{mi}), setup costs and times (s_{mji} and σ_{mji}) do not vary between machines.

Table 4.1: Data parameters.

Product	Family k	d_{it} $t=1$	$t=2$	$t=3$	p_{mi}	$c_{itt'}$ $t<t'$	$t>t'$
$i=1$	1	4,666	0	0	0.000150	0.0017	0.17
$i=2$	1	20,000	10,247	0	0.000097	0.0017	0.17
$i=3$	1	0	4,067	0	0.000540	0.0017	0.17
$i=4$	2	0	0	3,243	0.000185	0.0017	0.17
$i=5$	2	0	0	15,815	0.000141	0.0017	0.17

Table 4.2: Setup times and setup costs (s_{mji} / σ_{mji}).

Product	1	2	3	4	5
1	0.0/0.0	0.031/3.1	0.039/3.9	0.066/6.6	0.060/6.0
2	0.047/4.7	0.0/0.0	0.042/4.2	0.066/6.6	0.050/5.0
3	0.045/4.5	0.040/4.0	0.0/0.0	0.060/3.0	0.070/5.0
4	0.063/6.3	0.063/6.3	0.063/6.3	0.0/0.0	0.030/3.0
5	0.065/6.5	0.062/6.2	0.060/6.0	0.050/5.0	0.0/0.0

Hereafter, the lot-sizing ans scheduling solution of the hierarchical approach is illustrated. The most relevant non-zero solution values for the instance are given in Table 4.4. Reminding that, μ^s_{mitl} and μ^f_{mitl} represent the start and end times to produce product i on machine m in period t using the lth common resource batch; Y_{mijtl} takes on 1, if there is a changeover on machine m from product i to product j

Table 4.3: Data parameters of the raw material inventory to the illustrative example.

		Color grade			
		White	Light spotted	Spotted	Tinged
Initial inventory (eo_g)		230	250	200	50
Number of bales used in latest loading (fo_{gk})	$k = 1$	90	10	0	0
	$k = 2$	60	18	20	2
Minimum limit (ip_{gk})	$k = 1$	0.9	0	0	0
	$k = 2$	0.6	0	0	0
Maximum limit (sp_{gk})	$k = 1$	1	0.1	0	0
	$k = 2$	1	0.25	0.2	0.02

in period t, using the lth common resource batch; α_{mitl} equals to 1, if the machine m is set up for product i in the period t using the lth common resource batch; $X_{mitrt'}$ denote the production variables. As one can see, a production plan is determined. The required blend loadings are $U_{11} = 1$, $U_{21} = 1$ and $U_{32} = 1$. These values are inputted in the blending problem by the parameters $D_k = \{2, 1\}$. However, the blending solution is infeasible, that is, the cotton bale inventory is inadequate to ensure the required quality limits of the production plan.

Table 4.4: Optimal solution from the MSHLSP model to the hierarchical approach.

	$t = 1$	$t = 2$	$t = 3$
$m = 1$	$\alpha_{1111} = 1$ $\mu^s_{1111} : \mu^f_{1111} = 0 : 0.7$ $X_{11111} = 4667$ $Y_{11312} = 1$ $\mu^s_{1312} : \mu^f_{1312} = 0.7 : 1$ $X_{13122} = 474.41$	$\alpha_{1322} = 1$ $\mu^s_{1322} : \mu^f_{1322} = 1 : 2$ $X_{1322} = 1851.9$	$\alpha_{1333} = 1$ $Y_{13533} = 1$ $\mu^s_{1533} : \mu^f_{1533} = 2 : 3$ $X_{1533} = 6595.7$
$m = 2$	$\alpha_{2211} = 1$ $\mu^s_{2211} : \mu^f_{2211} = 0 : 0.7$ $X_{22111} = 7217.9$ $\alpha_{2212} = 1$ $\mu^s_{2212} : \mu^f_{2212} = 0.7 : 1$ $X_{22121} = 2472.8$	$\alpha_{2222} = 1$ $Y_{22322} = 1$ $\mu^s_{2322} : \mu^f_{2322} = 1 : 1.94$ $X_{2322} = 1774$ $Y_{23422} = 1$ $\mu^s_{2422} : \mu^f_{2422} = 1.94 : 2$	$\alpha_{2433} = 1$ $\mu^s_{2433} : \mu^f_{2433} = 2 : 2.6$ $X_{2433} = 3243$ $Y_{24533} = 1$ $\mu^s_{2533} : \mu^f_{2533} = 2.6 : 3$ $X_{2533} = 2482.7$
$m = 3$	$\alpha_{3211} = 1$ $\mu^s_{3211} : \mu^f_{3211} = 0 : 0.7$ $X_{32111} = 7217.9$ $\alpha_{3212} = 1$ $\mu^s_{3212} : \mu^f_{3212} = 0.7 : 1$ $X_{32121} = 3091.3$	$\alpha_{3222} = 1$ $\mu^s_{3222} : \mu^f_{3222} = 1 : 2$ $X_{3222} = 10247$	$\alpha_{3233} = 1$ $Y_{32533} = 1$ $\mu^s_{3533} : \mu^f_{3533} = 2 : 3$ $X_{3533} = 6737.6$
Stock level	$X_{13122} = 474.41$		
Residue level	$R_1 = 474.41$	$R_2 = 116.93$	$R_3 = 942$
Value of solution: 33.24			

However, in case the lot-sizing, scheduling and blending decisions are integrated, a feasible production plan is found. Table 4.5 illustrates the most relevant non-zero solution values for the instance. As one can see, the production plan takes into account the quality limitation and retards the production of same orders. The required blend loadings are $U_{11} = 1$ and $U_{22} = 1$. In addition, Table 4.6 illustrates the final inventory of cotton bales. The number of bales of each attribute is also defined. The variation of each color grade is given in percentage.

Table 4.5: Optimal production plan to the integrated approach.

	$t = 1$	$t = 2$	$t = 3$
$m = 1$	$\alpha_{1111} = 1$ $\mu^s_{1111} : \mu^f_{1111} = 0 : 0.94$ $X_{1111} = 4667$ $Y_{11511} = 1$	$\alpha_{1521} = 1$ $\mu^s_{1522} : \mu^f_{1522} = 1.77 : 2$ $X_{1522} = 1630.6$	$\alpha_{1532} = 1$ $\mu^s_{1532} : \mu^f_{1532} = 2 : 3$ $X_{1532} = 7092.2$
$m = 2$	$\alpha_{2211} = 1$ $\mu^s_{2211} : \mu^f_{2211} = 0 : 0.934$ $X_{2211} = 5540.2$ $Y_{22411} = 1$		$\alpha_{2432} = 1$ $\mu^s_{2432} : \mu^f_{2432} = 2 : 3$ $X_{2432} = 3243$
$m = 3$	$\alpha_{3211} = 1$ $\mu^s_{3211} : \mu^f_{3211} = 0 : 0.95$ $X_{3211} = 9793.8$ $Y_{32511} = 1$		$\alpha_{3532} = 1$ $\mu^s_{3532} : \mu^f_{3532} = 2 : 3$ $X_{3532} = 7092.2$
Stock level		$X_{15223} = 1630.6$	
Backlogging level	$X_{22431} = 4666$ $X_{22432} = 10247$ $X_{23432} = 4067$		
Residue level	$R_1 = 0$	$R_2 = 942$	$R_3 = 0$
Value of solution: 724664.2			

Table 4.6: Optimal blending solution to the integrated approach.

		Color grade			
		White	Light spotted	Spotted	Tinged
Final inventory (Ef_g)		87	215	180	48
Used bales (F_{gk})	$k = 1$	90	10	0	0
	$k = 2$	53	25	20	2
Variation between blends	$k = 1$	0	0	0	0
($\mathcal{H}_{gk}$)	$k = 2$	0.7	0.7	0	0

In summary, when the production is planned without take into account blending decisions, an unfeasible plan may be executed.

4.7 Final remarks

In this chapter, we have discussed the importance of integrating the production and blending planning problem dealing with attribute variability constraints.

Besides the traditional lot-sizing and scheduling decisions, the planning should determine which cotton bales must be part of the blend (the raw material). The problem is to determine the production level and sequence for the yarns and the cotton blends. It provides informations on the number of blends and cotton bales needed for production across the planning horizon. With this information, the blending problem must define which stored bales should be used to ensure the yarn quality and to keep a raw material inventory able to reproduce the blends with minimum attribute variation. A mathematical model for the integrated lot-sizing, scheduling and blending problem is proposed. The basic idea behind this approach is to simultaneously optimize decision variables of different functions that have traditionally been optimized sequentially. The integrated model validation is supported by the comparison with its hierarchical approach. The hierarchical approach is proposed such that the lot-sizing and scheduling problem is solved a priori. Then, the blending problem defines the set of cotton bales that meets the quality requirements.

From the results, we can note the influence of the raw material inventory on the lot-sizing and scheduling decisions. In case that the product quality has to be controlled, it is reasonable to suggest the integration of the lot-sizing, scheduling and blending decisions. Moreover, the partial integrated approach incorporates some blending constraints that ensure the yarn quality when planning the production in the first phase. In addition, the partial approach has the accuracy of the blending phase to find the set of cotton bales with minimal variations in the inventory and between blends.

In conclusion, the production planning without taking into account the raw material requirements can often be the source of production problems. It can generate impractical plans of the quality requirement position or cause customer dissatisfaction and loss of product price in case of the yarns specifications breach. We believe our study has shown that, under the restrict quality conditions, the value of coordinating production and blending can be extremely important.

This work should be felt as a qualitative research in this challenging field. However, further research toward the multi-objective decisions can assist the decision maker to define the production plan and cotton blends with the best trade off between the attribute variability and production costs. Moreover, the analysis over the integrated problem might help the purchasing department on how to define the orders for cotton bales. The purchase can also be oriented by the production plan to keep the attribute variability in the inventory and to indirectly minimize the impossibility of repeating the blends.

Chapter 5

Conclusions and future research

The spinning industry is an example of a process industry. Besides its importance (for both the economy and textile supply chain), little research has been conducted on the respective production planning. The topic of this thesis is the optimization of processes in textile industries, in particular, the production planning process in the spinning industry. The production process is characterized as a two-stage synchronized system. At the first stage, the continuous (or batch) production supply several machines of the second stage to produce discrete units. The production must be synchronized on the two stages as the quality produced in the second stage depends on the quality of the first-stage product. The production planning should assist for short-term decisions.

First, the overall lot-sizing and scheduling procedure in process industries is studied. The industrial environment shows similarities among various process industries, such as: spinning, metal alloys, beverage, glass container and others. Three time-based scale formulations are proposed to represent this kind of production system: MSGLSP (multi-stage general lot-sizing and scheduling problem), MSHLSP (multi-stage hybrid lot-sizing and scheduling problem) and MSCont (multi-stage continuous lot-sizing and scheduling problem). Based on a discrete time representation, the MSGLSP model synchronizes the production in first and second stages by a fixed grid across the machines. The MSHLSP model has a discrete-continuous representation, the production slots are independent of machines but respect the stage synchronization. The MSCont model has event-oriented features, the production slots can spread over periods keeping the synchronization between the stages. The models can be considered general enough to represent the production in the spinning industry as well as in other process industries with similar environment. Computational tests with state-of-the-art MIP solver show that the MSGLSP provides better feasible solutions in short running time. On the other hand, the MSHLSP achieves better solutions for longer computational times and was able to prove optimality more often. The MSCont is the most flexible of the three for incorporating additional operational requirements, at a cost of having the worst computational performance.

The lot-sizing and scheduling problem analysis is then narrowed to the spinning industry. To solve this problem, a new matheuristic called HOPS (Hamming-Oriented Partition Search) is proposed. HOPS is considered an upper bound improvement method that uses the hard variable fixing strategy on the variable partitions that can be easily recognized in the problem. With the solutions found along the branch-and-bound tree, variable partitions are measured based on changes of their variable values. The partitions are considered unstable if they yield too many changes. This measure is obtained by the hamming distance between the incumbent solution and a metric that represents the history of the search. For each feasible solution found, a reduced problem is created by fixing all variables except those belonging to the partitions with more unstable values. Upper bounds for the original problem can be delivered by the reduced problem solution. In summary, we develop a method combining mathematical-programming and heuristic tools. HOPS is compared to RINS (relaxation induced neighborhood search) and LB (local branching) - well-known matheuristics implemented in commercial MIP solvers. An instance set is generated with proportions based on real data. Results report that HOPS outperforms RINS and LB, mainly on instances considered large. In addition, the HOPS is general enough to solve other lot-sizing and scheduling problems and other problems in which variable partitions can be recognized.

With consensus, process industries give great importance to the raw material management. In order to analyze it for the spinning industry, an integration of the lot-sizing, scheduling and blending planning is considered. Here, the blending problem aims to select the cotton bales that satisfy the quality specifications of the yarn production planning. Three approaches are proposed to represent the combined problem: integrated, hierarchical and a partial integrated, each one with its respective mathematical formulation. An analysis on the value of the integration indicates that the production planning must incorporate constraints related to raw materials. The combined decisions are required to ensure feasible production plans in case any yarn quality requirement has to be managed and, consequently, the produced fiber blends. This study corroborates and reinforces the importance of taking into account the the raw material management while planning the production.

More than proposing approaches to speed up the production planning in spinning industries and enriching the literature, this work is in line with research gaps pointed out in the literature: lacunas on addressing lot sizing and scheduling considering backlogging and sequence-dependent setups (Zhu and Wilhelm, 2006); multiple stages with parallel machines and synchronization of resources (Clark *et al.*, 2011); and problem-oriented methods (James and Almada-Lobo, 2011).

Further researches are opened in the topic of the thesis in different directions. Solution methods, as HOPS, can also be applied to solve the integrated lot-sizing, scheduling and blending model. Furthermore, HOPS can be the central target of the whole study. An additional component is an interesting improvement for boosting the lower bound of the problem. The unstable variable measure can be applied to indicate the variable to be branched. As indicated before, HOPS has features allowing for its usage to solve other problems not studied in this thesis.

Extensions to the presented mathematical models are natural evolutions of the research. Process industries as also spinning industries are moving from push to pull strategy, that is, the production is focused in make-to-order. As the inventory of final products aims to be null, the industrial layout is frequently losing the inventory space. Thus, the formulations should reflect the management of the final product inventory when planning the production.

Companies can have multiple spinning plants. However, yarn orders are asked for the companies. Thus, the decision about which plant should produce each order is a company decision. On the other hand, larger spinning industries can have hundreds of parallel spinning machines. However, a great number of opening-blending machines are also needed to supply the spinning machines. In this type of industry, the plant can be viewed as of small size, that is, a large-size industry can be handled as a multiple plant extension. In the multiple plants case, the management of the cotton bale delivery can be applied in order to always keep the variability of the several inventories under control. This strategy can also be taken into account in the production planning.

Further research into the variability minimization in the quality of the yarns produced by different blends poses some challenges. It is interesting to minimize the number of blends used to produce a customer demand. The order production planning is proposed in Furtado *et al.* (2011) and can be reformulated to the spinning case. The expected results are orders produced by the same blending without attribute variation, reducing serious problems on the supply chain downstream.

Equally important, the integrated lot-sizing, scheduling and blending problem allows for an analysis of the cotton bale costs in the yarn production. A quantitative study on the partial integrated approach can consider the cost of each cotton bale in the objective function (besides the constraints related to the blending).

Last, but not least, a decision support tool is more than mathematical models and solution methods. It also consists of a friendly user interface to manage data, an analytical tool and a presentation tool. Although it seems more practical-oriented work than an academic one, it involves questions related to how to bring the practice to the academy and vice versa. Concluding, several extension ideas are provided in this thesis and their development will properly represent more and more complex environments.

References

ABIT (2011). Brazilian textile and apparel industry association. http://www.abit.org.br (25/01/2012).

Admuthe, L. S. and Apte, S. D. (2009). Optimization of spinning process using hybrid approach involving ANN, GA and linear programming. *Proceedings of the 2nd Bangalore Annual Compute Conference*, pages 1–4.

Almada-Lobo, B., Klabjan, D., Carravilla, M. A., and Oliveira, J. F. (2007). Single machine multi-product capacitated lot sizing with sequence-dependent setups. *International Journal of Production Research*, **45**, 4873–4894.

Almada-Lobo, B., Oliveira, J. F., and Carravilla, M. A. (2008). Production planning and scheduling in the glass container industry: A VNS approach. *International Journal of Production Economics*, **114**, 363–375.

Almeder, C. and Almada-Lobo, B. (2011). Synchronisation of scarce resources for a parallel machine lotsizing problem. *International Journal of Production Research*, **49**, 7315–7335.

Amorim, P., Günther, H. O., and Almada-Lobo, B. (2012). Multi-objective integrated production and distribution planning of perishable products. *International Journal of Production Economics*, **138**, 89–101.

Anthony, R. N. (1965). Planning and control systems: A framework for analysis. Technical report, Harvard University Graduate School of Business.

Araujo, S. A., Arenales, M. N., and Clark, A. R. (2007). Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. *Journal of Heuristics*, **13**, 337–358.

ATP (2012). Portuguese textiles & clothing association. http://www.atp.pt (30/06/2012).

Belvaux, G. and Wolsey, L. A. (2001). Modelling practical lot-sizing problems as mixed-integer programs. *Management Science*, **47**, 993–1007.

Beraldi, P., Ghiani, G., Grieco, A., and Guerriero, E. (2008). Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. *Computers & Operations Research*, **35**, 3644–3656.

Brahimi, N., Dauzere-Peres, S., Najid, N. M., and Nordli, A. (2006). Single item lot sizing problems. *European Journal of Operational Research*, **168**, 1–16.

Camargo, V. C. B., Toledo, F. M. B., and Almada-Lobo, B. (2012a). HOPS - Hamming-Oriented Partition Search for production planning in the spinning industry. *submitted to European Journal of Operational Research*.

Camargo, V. C. B., Toledo, F. M. B., and Almada-Lobo, B. (2012b). Three time-based scale formulations for the two-stage lot sizing and scheduling in process industries. *Journal of the Operational Research Society*, **63**, 1613–1630.

Chandra, P. and Fisher, M. L. (1994). Coordination of production and distribution planning. *European Journal of Operational Research*, **72**, 503 – 517.

Clark, A. R., Almada-Lobo, B., and Almeder, C. (2011). Editorial on lotsizing and scheduling: industrial extensions and research opportunities. *International Journal of Production Research*, **49**, 2457–61.

Crama, Y., Pochet, Y., and Wera, Y. (2001). A discussion of production planning approaches in the process industry. CORE Discussion Papers 2001042, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE).

Danna, E., Rothberg, E., and Pape, C. L. (2005). Exploring relaxation induced neighborhoods to improve MIP solution. *Mathematical Programming*, **102**, 71–90.

Dillenberger, C., Escudero, L. F., and Zhang, A. W. W. (1994). On practical resource allocation for production planning and scheduling with period overlapping setups. *European Journal of Operational Research*, **75**, 275–286.

Dolan, E. and Moré, J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, **91**, 201–213.

Drexl, A. and Haase, K. (1995). Proportional lotsizing and scheduling. *International Journal of Production Economics*, **40**, 73–87.

Drexl, A. and Kimms, A. (1997). Lot sizing and scheduling - survey and extensions. *European Journal of Operational Research*, **99**, 221–235.

El Mogahzy, Y. (2004). An integrated approach to analyzing the nature of multi-component fiber blending - part I: Analytical aspects. *Textile Research Journal*, **78**, 701–712.

El Mogahzy, Y. (2005). Specific approaches for cutting manufacturing cost and increasing profitability using the EFS system. In *EFS System Conference Presentations*.

El Mogahzy, Y., Farag, R., Abdelhady, F., and Mohamed, A. (2004). An integrated approach to analyzing the nature of multicomponent fiber blending - part II: Experimental analysis of structural and attributive blending. *Textile Research Journal*, **74**, 767–775.

Ferreira, D., Morabito, R., and Rangel, S. (2009). Solution approaches for the soft drink integrated production lot sizing and scheduling problem. *European Journal of Operational Research*, **196**, 697–706.

Fischetti, M. and Lodi, A. (2003). Local branching. *Mathematical Programming*, **98**, 23–47.

Fischetti, M., Glover, F., and Lodi, A. (2005). The feasibility pump. *Mathematical Programming*, **104**, 91–104.

Fleischmann, B. (1990). The discrete lot sizing and scheduling problem. *Management Science*, **44**, 337–348.

Fleischmann, B. and Meyr, H. (1997). The general lotsizing and scheduling problem. *OR Spectrum*, **19**, 11–21.

Floudas, C. A. and Lin, X. (2004). Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers & Chemical Engineering*, **28**, 2109–21129.

Fransoo, J. C. (1993). *Production control and demand management in capacitated flow process industries*. Ph.D. thesis, Technische Universiteit Eindhoven.

Fransoo, J. C. and Rutten, W. G. (1994). A typology of production control situations in process industries. *International Journal of Operations & Production Management*, **14**, 47–57.

Fryer, L. F., Rost, J. P., and Lord, P. R. (1996). Effects of cotton fiber blending and processing on HVI mensurements - part I. *Textile Research Journal*, **66**, 349–357.

Furtado, M. G., Vergine, E., Camargo, V. C. B., and Toledo, F. M. B. (2011). Um modelo para o problema de planejamento da produção de pedidos em fundições de pequeno porte. In *Annals of the 43rd Simpósio Brasileiro de Pesquisa Operacional*, pages 1255–1262.

Gao, L. L., Altay, N., and Robinson, E. P. (2008). A comparative study of modeling and solution approaches for the coordinated lot-size problem with dynamic demand. *Mathematical and Computer Modelling*, **47**, 1254–1263.

Ghosh, S. (2007). DINS, a MIP improvement heuristic. In M. Fischetti and D. Williamson, editors, *Integer Programming and Combinatorial Optimization*, volume 4513 of *Lecture Notes in Computer Science*, pages 310–323. Springer Berlin / Heidelberg.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decisions Science*, **8**, 156–166.

Gramani, M. C. N., França, P. M., and Arenales, M. (2009). A lagrangian relaxation approach to a coupled lot-sizing and cutting stock problem. *International Journal of Production Economics*, **119**, 219–227.

Greene, J. H., Chatto, K., Hicks, C. R., Cox, C. B., Baynham, T. E., and Baynam, T. E. J. (1965). Linear programming - cotton blending and production allocation. Technical report, International Business Machines Corporation (IBM); The Journal of Industrial Engineering.

Guimarães, L., Klabjan, D., and Almada-Lobo, B. (2012). Annual production budget in the beverage industry. *Engineering Applications of Artificial Intelligence*, **25**, 229 – 241.

Haase, K. (1996). Capacitated lot-sizing with sequence dependent setup costs. *OR Spectrum*, **18**, 51–59.

Harpa, R. (2008). Applications of multicriteria analysis in cotton mill. *Textile Research Journal*, **78**, 421–426.

Helber, S. and Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, **123**, 247–256.

James, R. J. and Almada-Lobo, B. (2011). Single and parallel machine capacitated lotsizing and scheduling: New iterative MIP-based neighborhood search heuristics. *Computers & Operations Research*, **38**, 1816–1825.

Jans, R. (2009). Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. *Journal on Computing*, **21**, 123–136.

Jans, R. and Degraeve, Z. (2008). Modeling industrial lot sizing problems: a review. *International Journal of Production Research*, **46**, 1619–1643.

Jordan, C. and Drexl, A. (1998). Discrete lotsizing and scheduling by batch sequence. *Management Science*, **44**, 698–713.

Kallrath, J. (2002). Planning and scheduling in the process industry. *OR Spectrum*, **24**, 219–250.

Kang, B. C., Park, S. W., Koo, H. J., and Jeong, S. H. (2000). A simplified optimization in cotton bale selection and laydown. *Fibers and Polymers*, **1**, 55–58.

Karacapilidis, N. I. and Pappis, C. P. (1996). Production planning and control in textile industry: A case study. *Computers in Industry*, **30**, 127–144.

Karimi, I. A. and McDonald, C. M. (1997). Planning and scheduling of parallel semicontinuous processes. 2. short-term scheduling. *Industrial & Engineering Chemistry Research*, **36**, 2701–2714.

Karmarkar, U. S. and Schrage, L. (1985). The deterministic dynamic product cycling problem. *Operations Research*, **33**, 326–345.

Kim, S., Han, J., Lee, Y., and Park, E. (2010). Decomposition based heuristic algorithm for lot-sizing and scheduling problem treating time horizon as a continuum. *Computers & Operations Research*, **37**, 302–314.

Krarup, J. and Bilde, O. (1977). *Optimierung bei Graphenthe-oretischen und Ganzzahligen Probleme*, chapter Plant location, set covering and economic lot sizes: an O(mn) algorithm for structured problems, pages 155–180. Birkhauser Verlag, Basel.

Malik, M., Qiu, M., and Taplin, J. (2009). An integrated approach to the lot sizing and cutting stock problems. In *Industrial Engineering and Engineering Management, 2009. IEEM 2009. IEEE International Conference on*, pages 1111 –1115.

Maniezzo, V., Stützle, T., and Voß, S. (2010). *Matheuristics - Hybridizing Metaheuristics and Mathematical Programming*. Springer.

Meyr, H. (2000). Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. *European Journal of Operational Research*, **120**, 311–326.

Meyr, H. (2002). Simultaneous lotsizing and scheduling on parallel machines. *European Journal of Operational Research*, **139**, 277 – 292.

Pimentel, C., Alvelos, F. P., Duarte, A., and de Carvalho, J. M. V. (2011). Exact and heuristic approaches for lot splitting and scheduling on identical parallel machines. *International Journal Manufacturing Technology and Management*, **22**, 39–57.

Poltroniere, S. C., Poldi, K. C., Toledo, F. M. B., and Arenales, M. N. (2008). A coupling cutting stock-lot sizing problem in the paper industry. *Annals of Operations Research*, **157**, 91–104.

Puchinger, J. and Raidl, G. R. (2005). Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, volume 3562 of *Lecture Notes in Computer Science*, pages 113–124. Springer Berlin / Heidelberg.

Sahling, F., Buschkhl, L., Tempelmeier, H., and Helber, S. (2009). Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research*, **36**, 2546–2553.

Santos, M. O. and Almada-Lobo, B. (2012). Integrated pulp and paper mill planning and scheduling. *Computers & Industrial Engineering*, **63**, 1–12.

Santos-Meza, E., Santos, M. O., and Arenales, M. N. (2002). A lot-sizing problem in an automated foundry. *European Journal of Operational Research*, **139**, 490–500.

Serafini, P. and Speranza, M. G. (1992). Production scheduling problems in a textile industry. *European Journal of Operational Research*, **58**, 173–190.

Silva, C. and Magalhaes, J. M. (2006). Heuristic lot size scheduling on unrelated parallel machines with applications in the textile industry. *Computers & Industrial Engineering*, **50**, 76–89.

Soman, C. A., van Donk, D. P., and Gaalman, G. (2004). Combined make-to-order and make-to-stock in a food production system. *International Journal of Production Economics*, **90**, 223–235.

Suerie, C. (2005). Time continuity in discrete time models: new approaches for production planning in process industries. Publications of darmstadt technical university, institute for business studies (bwl), Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL).

Trützschler (2005). *Manual - Tecnologia da fibra - Trützschler*.

Wagner, H. and Whitin, T. (1958). Dynamic version of the economic lot size model. *Management Science*, **5**, 89–96.

Waters, W. and Philips, J. (1961). The effect of blending cottons of dissimilar fiber proprieties upon spinning performance and yarn quality. *Textile Research Journal*, **31**, 608–618.

Zago, G. (2005). Otimização da composição da matéria prima para uma indústria têxtil de grande porte. Technical report, São Paulo.

Zhu, X. and Wilhelm, W. E. (2006). Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE Transactions*, **38**, 987–1007.