Luis Filipe Ribeiro dos Santos Guimarães

U. PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
DEPARTAMENTO DE ENGENHARIA E GESTÃO INDUSTRIAL

Advanced Production Planning Optimization in the Beverage Industry

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 Almada Lobo, Assistant Professor of Faculdade de Engenharia da Universidade do Porto and Diego Klabjan, Professor of Industrial Engineering and Management Sciences, Northwestern University

Department of Industrial Engineering and Management
Faculdade de Engenharia da Universidade do Porto
2013

This research was partially supported by the PhD grant SFRH/BD/62010/2009 awarded by the Portuguese Foundation for Science and Technology and by the private company Unicer Bebidas, S.A. under the projects *Study and development of quantitative methods to support decision making in the beverage industry* and *Research and development of quantitative methods for operations management*.

“Failing to plan is planning to fail.”
Alan Lakein

Acknowledgments

I am thankful to Unicer, SA for giving me the opportunity to work on their extremely interesting planning problems. In particular, I would like to acknowledge Carlos César for believing in analytics from the very start of our collaboration. I would also like to take this occasion to express my thankfulness to all Unicer's staff for allowing me to improve my knowledge on the beverage industry by sharing their expertise, in particular to Benjamin Herbin, Maria João Pissarra, Pedro Alves, Pedro Ribeiro, Carlos Caiado, Jorge Gomes, Ricardo Lírio, Ana Ferreira, Jorge Monteiro e Merceiano Roque.

A special thanks goes to all the friends I have met in the I201, Chicago and Brazil. I really enjoy the time we spent together both in the office and outside. I have no doubt I made friends for life. I owe a big thank you especially to Pedro, Victor and Bill. I also would like to thank my friends outside the research area, especially Ricardo, the good moments I spent in their company and away from research that were fundamental to my mental health.

I would like to express a word of appreciation and gratitude to my co-advisor Diego Klabjan. First for hosting me during my stay in the Northwestern University. Second, for the time spent in our weekly meetings afterwards, most often very early in the morning for him. Despite the time zone difference, Diego could always find some free time to fit our meetings and this has greatly contributed to the quality of this thesis.

I am much indebted to my advisor Bernardo for believing in me. From the days of IMAGE, it has always been a true pleasure to work with him. He instigated my interest in Operations Research (OR) in the last year of my master, first in the course of Quantitative Methods for Decision Support and later during the internship I did at Unicer as part of my dissertation. Besides OR, he is much responsible for my improvement in skills such as writing, oral presentations and management, although I know there is still a long path to be walked. I am grateful for all the opportunities he has given to me and for his friendship.

I wish to thank my family for all they have done for me during my life, today I carry a little bit of each one them inside me. Thank you Mãe, Pai, Hugo, Avô and Avó. I know the several times I went abroad were particularly difficult for both of us, your support and motivation were crucial. Unfortunately, my grandmother is not among us anymore to see her grandson graduate, so I would like to send her a special message - "Um beijinho Avó!".

Most importantly, I would like to acknowledge destiny to make Teresa and I pursue a PhD at the same time and place. If it was not for that I could probably never met the most incredible person I know. She has a rare combination of kindness, beauty and determination. She is highly responsible for making me more methodic in my work following her example and I can only aspire to payback all the help she has given me during the PhD in hers. Since I am not good with words I chose to adapt some from Elton Jonh:

(...)

If I was a sculptor, but then again, no
Or a man who makes potions in a traveling show
I know it's not much but it's the best I can do
My gift is my "thesis" and this one's for you

(...)
I hope you don't mind
I hope you don't mind that I put down in words
How wonderful life is while you're in the world!

"Your Song" by Elton Jonh

Abstract

Nowadays companies look at their supply chains as a strategic asset. An efficiently managed and streamlined supply chain can lead to significant cost reductions and to increased customer satisfaction, resulting in an important competitive advantage. Production planning plays a decisive role in the supply chain performance, as inefficient production plans contribute to the increase of costs and lead times.

Motivated by the production planning complexity of the beverage industry, we aim to study and develop quantitative tools to support managers in their decision-making process. The research objectives are aligned with two of the major production planning challenges present in this industry. First, we address the recent loss of operational efficiency in the filling lines due to the increasing number of final products (and consequently setups) by studying the industry short-medium term production planning and scheduling. Second, we tackle the long term planning to face the challenges of demand seasonality and of the coordination of a multi-plant supply chain.

The contributions of this thesis constitute breakthroughs on a scientific front by developing new mathematical models and state-of-the-art solution approaches combining mathematical programming and metaheuristics. The new models comprehend improvements in terms of computational performance and include new and more realistic features from real-world problems faced by the industry. In addition, the solution approaches explore in innovative ways the combination of exact and approximate methods.

A case study from a major Portuguese beverage company constitutes an additional motivation, allowing to assess the practical relevance of our scientific contributions by testing them in real-world instances. This also enables us to make contributions to the practice of operations research by giving insights on how to deploy advanced quantitative methods in the practice of decision making.

Although the beverage industry is our primary focus, other semi-continuous process industries that share common features to this one can also benefit from our contributions.

Resumo

Atualmente as empresas consideram as suas cadeias de abastecimento um ativo estratégico. Uma gestão eficiente e flexível da cadeia pode levar a uma redução significativa de custos e aumento da satisfação dos clientes, resultando numa importante vantagem competitiva. Neste contexto, o planeamento da produção desempenha um papel crítico, uma vez que planos de produção ineficientes contribuem decisivamente para o aumento dos custos e dos prazos de entrega.

Motivados pela complexidade inerente ao planeamento da produção na indústria das bebidas, pretendemos estudar e desenvolver métodos quantitativos para apoiar os gestores na sua tomada de decisão. Em particular, os objetivos desta dissertação estão alinhados com dois dos grandes desafios que atualmente a indústria enfrenta. O primeiro prende-se com a recente perda de eficiência operacional das linhas de produção, consequência do número crescente de produtos finais que implica um aumento dos tempos de preparação. Neste âmbito, pretendemos atuar no planeamento e sequenciamento da produção a curto-médio prazo. O segundo desafio está relacionado com a sazonalidade da procura e com a necessidade de coordenar os diferentes centros de produção, abordado no planeamento da produção de longo prazo.

As contribuições desta dissertação constituem avanços científicos através do desenvolvimento de novos modelos matemáticos e de novos métodos de solução combinando programação matemática e metaheurísticas. Os novos modelos apresentam melhorias em termos de desempenho computacional e incluem também novas funcionalidades que representam de forma realista problemas enfrentados nesta indústria. Por outro lado, os métodos de solução exploram de forma inovadora a combinação de abordagens exatas e aproximadas.

Um caso de estudo numa empresa de bebidas portuguesa constitui uma motivação adicional para esta dissertação e permite avaliar a relevância prática das contribuições científicas através de testes com dados reais. Adicionalmente, o caso de estudo permite expandir as contribuições para o domínio prático, explorando formas inteligíveis de aplicar métodos quantitativos na tomada de decisão.

Embora a indústria das bebidas seja o foco principal desta dissertação, outras indústrias com processos de produção semi-contínuos com características comuns podem também beneficiar das nossas contribuições.

Résumé

Actuellement, les entreprises regardent leurs chaînes d'approvisionnement, un atout stratégique. Une gestion efficiente et souple de la chaîne peut conduire à une réduction significative des coûts et une satisfaction accrue des clients, ce qui entraîne un avantage concurrentiel significatif. Dans ce contexte, la planification de la production joue un rôle essentiel puisque des plans de production inefficaces contribuent décisivement à la hausse des coûts et des délais de livraison.

Motivés par la complexité inhérente à la planification de la production dans l'industrie des boissons, nous avons l'intention d'étudier et de développer des méthodes quantitatives pour appuyer les gestionnaires dans leur prise de décision. En particulier, les objectifs de cette thèse sont alignés avec deux des principaux défis actuels de l'industrie. Le premier s'attache avec la perte récente de l'efficacité de fonctionnement des chaînes de production, le résultat du nombre croissant de produits finals ce qui implique l'augmentation du temps de préparation. Ainsi, nous avons l'intention d'opérer dans la planification et l'ordonnancement de la production à court /moyen terme. Le deuxième défi est lié à la saisonnalité de la demande et au besoin de coordonner les différents centres de production, adressé à la planification de la production à long terme.

Les contributions de cette thèse constituent des progrès scientifiques à travers du développement de nouveaux modèles mathématiques et de nouvelles méthodes de solution combinant programmation mathématique et méta heuristiques. Les nouveaux modèles présentent d'améliorations en termes de performance de calcul et comprennent aussi de nouvelles fonctionnalités qui représentent d'une façon réaliste des problèmes rencontrés dans cette industrie. En outre, les méthodes de solution exploitent de manière innovante la combinaison des approches exactes et approximatives.

Une étude de cas dans une entreprise de boissons portugaise est une motivation supplémentaire pour cette thèse et permet d'évaluer la pertinence pratique des contributions scientifiques à travers des tests avec des données réelles. De plus, l'étude de cas permet d'étendre les contributions dans le domaine pratique, en explorant des façons intelligibles d'appliquer des méthodes quantitatives dans la prise de décision.

Bien que l'industrie des boissons soit l'objectif principal de cette dissertation, d'autres industries avec des procédés de production semi-continus avec des caractéristiques communes peuvent également profiter de nos contributions.

Contents

Chapter 1

Motivation and overview

Supply chain planning and optimization presents a wide range of challenges in both expansion and depression economic cycles. Companies that excel at supply chain planning by trading-off cost and service can best optimize their margins and create a significant competitive advantage.

At the very heart of supply chain optimization we may find production planning problems. These important and challenging planning problems attempt to find the most efficient way of acquire, use and allocate production resources in order to transform raw material into finished products to satisfy customer demands. Efficiency focus can be based on a time and/or cost scale and decisions cross several decision levels - from high-level strategic work that is embarked upon every few years, passing by the tactical plans created for the following months, and ending in the weekly/daily planning that takes place at the operational level. Plans usually address work force leveling, overtime requirements determination, lot sizes and production sequencing.

With the recent advances in IT and the generalization of Enterprise Resource Planning systems (ERP), the data relevant to production planning problems became widely available in most of the companies making the use of decision support systems relying on analytical models easier. Advanced planning systems (APS) appear to explore this valuable information, providing quantitative methods that enable the automation and optimization of the entire supply chain, including production planning and scheduling tasks. However, the planning characteristics arising in different industries raise special requirements that these standard tools are not always able to respond. The complexity of the manufacturing systems and the misalignment with the relevant features to include can often lead to unsuccessful implementations of those systems. In some industries the approaches in which APS are based upon do not translate the reality accurately enough creating plans that are either useless or require excessive managers manual corrections. Nevertheless, one advantage that these systems do have is their modular architecture open to include new modules and new algorithms. Thus, the new planning paradigm emerging from the use of APS provides the ideal framework to deploy operations research developments in practice.

This thesis is the result of problem-driven research motivated by the production planning problems arising in the beverage industry. The beverage industry, as other fast moving consumer goods industries, operates in an extremely aggressive market with great pressure towards operational efficiency both in terms of delivery requirements and cost performance. An efficient management of production resources can guarantee customer satisfaction in the most cost-efficient manner helping companies to fulfill their goals. This dissertation focuses its attention on the unresolved challenges faced by the beverage industry concerning its production planning and aims to develop quantitative tools to support managers in their decision-making process. A case study from a Portuguese beverage company constitutes an additional motivation for our study, allowing to assess the practical value of the scientific contributions. The unsatisfactory answers by the current APS installed at the company concerning production planning are addressed as part of this thesis research objectives.

This thesis seeks to constitute breakthroughs in two main axes: scientific and practical grounds, and mathematical formulations and solution approaches. The new mathematical formulations should comprehend improvements in terms of computational performance and include new and more realistic features from real-world problems faced by the industry. On the other hand, solution approaches should explore the combination of traditional metaheuristics with exact methods in new and innovative ways. The motivation behind the use of hybrid heuristics is twofold. First, we aspire to achieve important findings in the field of the matheuristics, which have proven to give excellent responses to hard optimization problems by taking advantage of the computational efficiency of modern commercial solvers. Second, we aim to show the potential of their application to solve real-world problems, since these heuristics can more easily be adapted to cope with model extensions or to address different optimization problems that arise in practice. All the formulations and algorithms are to be tested using the case study and should be extensible to other similar consumer good industries or other industries facing similar problems.

To achieve our research goals, we follow a hierarchical approach by tackling first the challenges in short-medium term planning and progressively moving towards larger planning horizons and including additional decisions related to the distribution process.

The remainder of this chapter is organized as follows. Section 1.1 presents the beverage industry and its supply chain highlighting the main planning challenges (in Section 1.1.1). The planning framework of the industry is presented in Section 1.1.2. By crossing the main planning challenges and the planning framework we define our research goals in Section 1.2. Finally, 1.3 contains a description of each of the chapters composing this dissertation, where the main contributions made are described in more detail.

1.1. Beverage industry

The beverage industry is a sub-sector of the food industry, the second largest sector in the European manufacturing industry in terms of added value. Products manufactured by the beverage industry include: bottled water, juice, sparkling and still drinks, syrups, nectars, ready-to-drink and regular teas and coffees, dairy drinks, energy drinks, sports drinks, fruit powders, and alcoholic drinks such as beer, wine, cider and spirits. Markets worldwide are strongly affected by cultural differences, especially in Europe, resulting in the appearance of small to medium size companies that are specialized in local products and/or local brands. For these companies, the efficiency of their operations is crucial to survive in the market.

Nowadays the use of advanced manufacturing technology allows beverage manufacturers to increase production and become increasingly streamlined and efficient, favoring its expansion. The resulting sector's competition has forced companies to expand their product portfolio, translated in a huge number of packaging sizes, customized package, prints and labels, composition of ingredients and flavors, as well as in new products, which raises the need for efficient production planning.

1.1.1 The supply chain

We detail next the main features of a typical beverage company supply chain for a better understanding of the problems arising in practice. Most of the description that follows is valid for a vast range of companies competing in the sector or even for companies in the fast moving consumer goods industry, nevertheless specific constraints and features may not be covered. The type of supply chain is analyzed according to Meyr and Stadtler (2005). The attention is focused on the *production type*, *distribution type* and *sales type*. The procurement function is ignored since production outsourcing is not usually an option for these companies and materials procurement has virtually no influence in production decisions. Therefore, in the context of the thesis this function can be neglected.

Production

The production process of beer and soft drinks shares some common features with other semi-continuous process industries (Kallrath (2002); Kopanos et al. (2011)):

- multi-product equipment,
- sequence-dependent setup times and cleansing costs,
- divergent bill of materials,
- multi-stage production, with a known bottleneck,
- combined batch production in the upper stages, with continuous operations at the downstream stage (semi-continuous process).

In particular, beer and soft drinks encompass two main production stages: liquid production (stage I) and liquid bottling (stage II). Stage I in the beer production process is composed by three steps, namely brewing, fermentation/maturation and filtration, see Figure 1.1. This process is far more complex than soft drink production (a single process), especially due to long lead times in fermentation/maturation. Bottling or filling lines are responsible for stage II and sequentially: wash, fill, seal, label and pack bottles, cans or kegs in a continuous process .

Stages I and II of the production process occur within the same plant, being most of the times physically separated. Typically, a single unit of stage I supplies a series of parallel filling lines, with buffers between the two stages. The bottling stage is often the main bottleneck of the entire production process. For the beer case, this is explained by additional buffers between the different main processes of stage I and also by their flexibility in terms of processing times. Moreover, the high number of final items that have to be manufactured in stage II correspond to a few different types of beer (or syrups for the soft drinks) in stage

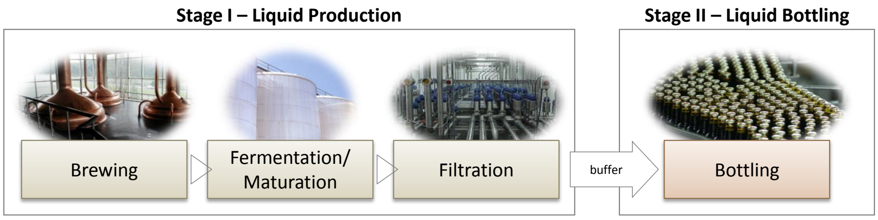


Figure 1.1: Main stages of the beer production process

I, since SKU (Stock Keeping Unit) differences may rely on a different container, label or package affecting only stage II.

Filling lines are usually divided according to their technological aspects (e.g. filling lines for kegs are unable to fill bottles or cans). Even so, filling lines are relatively flexible and a certain product can often be assigned to several alternative lines, even within the same plant, but with throughput rates (measured in terms of liters per hour) that might differ substantially.

Each filling line can only produce a single product at any time, being adjusted to fill a certain liquid, container type and size, and final package. A product changeover may involve several mechanical adjustments in the filling line and possibly a cleansing step. Since the setup costs and times are dependent on the production sequence, we say to be in the presence of sequence dependent changeover times and costs. In recent years, the market pressure has led companies to increase the number of products, along with less stock, by delivering products more frequently. This has leveraged the appearance of more setups which consume the scarce available production time (capacity) and has reduced substantially the operational efficiency of filling lines.

Filling lines operate on a shift basis and their capacity can be translated into the number of hours available for production. Overtime may not be an option as some of the filling lines operate around the clock. Investing in new or more flexible lines is also problematic and has to be carefully studied because it greatly increases fixed costs. Yet, such changes would only produce effects in the long term, thus short-medium term capacity can be considered constant.

Distribution

A typical beverage industry company has one or more plants relatively close to the geography of demand, in order to avoid transfer costs, which otherwise would assume an important percentage of the total cost. This cost reduction strategy is particularly important for standard products with high volume and excludes beverages such as still and sparkling water that have to be produced near the source. Nevertheless, some product specialization is possible, aiming to achieve better throughput rates or standard quality requirements in stage I due to larger production batches whose process is easier to control. In face of these characteristics production planning can not be conducted considering only one plant at a time, as it would ignore the potential benefits of coordination.

Plants, distribution platforms and customers form a three echelon distribution network. The first echelon is composed by the production sites and the second one by distribution platforms. Customers in the downstream echelon can be supplied by both upstream echelons.

Sales

Sales of beer and soft drinks have high seasonality and variability. Beer and soft drinks consumption peaks at Easter and Christmas, but Summer is by far the highest point in terms of sales. Moreover, there is an increase of sales in the second half of each month. Product demand is also affected by other sources of variability, such as brand management and clients commercial policy. Some of the most important customers of these companies are large retail chains with extremely aggressive marketing strategies that require almost instantaneous response from suppliers. These sales characteristics stress an almost constant and scarce production capacity which leads the industry to work on a make-to-stock basis.

1.1.2 Planning framework

Production planning in the beverage industry is a complex process. Not only there are several complex processes and multiple stakeholders involved, but also the increasing competitiveness of the market forces companies to enlarge their product portfolio and responsiveness posing new challenges and raising the need for decision support tools to help managers.

To face the constraints described above and the different nature of decisions, production planning in the beverage industry is made at several hierarchical levels with different aims and planning horizons. Although decisions are strongly dependent, it is virtually impossible to sustain a single decision model for the entire decision-making process as it would be extremely hard to maintain, solve and interpret. Moreover, market dynamics also determine that high detailed plans for a distant future are in most occasions useless. Production planning decisions are made in a hierarchical process composed of three levels (Fleischmann and Meyr (2003)):

- **Long term planning (strategic):** assesses investments in the installed capacity, trying to balance capacity with demand for a planning horizon of 12 up to 36 months (from 1 up to 3 years), based on a monthly bucket. Product mix, closing and opening of new plants or filling lines are among the options studied. Naturally, at this level of decision, distribution planning has a strong impact and is often performed simultaneously or with some degree of interaction. One of the main goals is to evaluate the current network and to perform what-if analysis based on scenarios defined by managers.

- **Medium term planning (tactical):** decisions at this level consider filling lines production lot sizing and overtime utilization (whenever possible) focusing at cost efficiency. Here planning horizons commonly span from 4 to 12 weeks. Transportation quantities among the major locations in the supply chain may also be tackled.

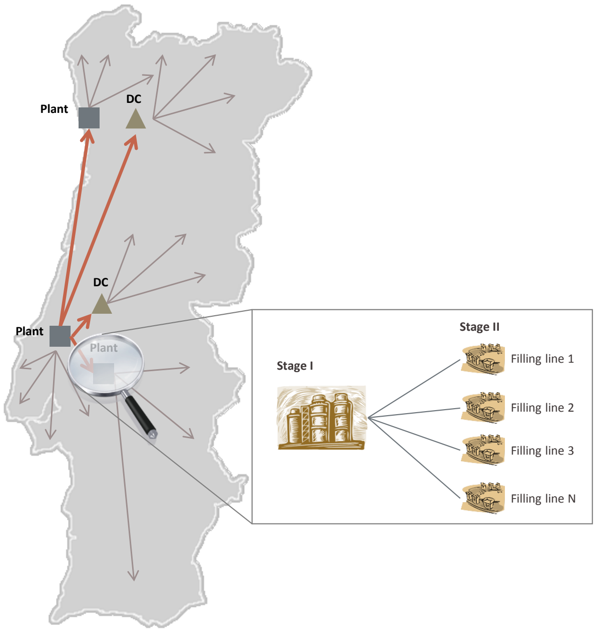


Figure 1.2: Supply chain of the case study

- **Short term planning (operational):** is the lower level of the hierarchy. It schedules operations in the available resources looking at a very short planning period, typically a day to one week. The objective is to define a schedule for the lots defined in the tactical level looking at one filling line at a time. Mostly the plan is oriented to the minimization of the sum of the sequence-dependent setup times and the sum of the tardinesses, being the due dates also defined by the previous planning level.

Planning is performed in a rolling horizon approach. Only a few periods in the begin of the planning horizon are actually executed and plans are updated as the horizon is rolled forward. Furthermore, the output of an upper level constitutes an important input for the following level.

1.1.3 Case Study

Our case study company holds many nationally very popular brands of beer, soft drinks, and mineral and sparkling water. Production sites are spread around the country, accounting for 8 plants and more than 20 filling lines. Mineral and sparkling water plants are located near a water source, while other production sites are responsible for beer and soft drink production. The scope of our study is the planning of beer and soft drinks that has to be done simultaneously since these product types share common production resources. The planning of water is rather straightforward and, therefore, it will be disregarded here.

Figure 1.2 shows the main product flows in the supply chain of the company considering only one plant as the origin. We do not represent all the flows and distribution centers, denoted as (DC), for the sake of the figure clarity. The product flows will be discussed in more detail in Chapter 6. The figure also depicts the production process inside one plant. Considering only the beer and soft drink production, the company has 3 plants dedicated to these products, with a total of 14 filling lines. The plants are located near the areas with highest demand (Porto and Lisbon).

In the case study, long term production planning occurs once every month to conduct an evaluation of the adequacy of the capacity to future demand (12 months), investigating the need for additional shifts or to build more seasonal stock. Its inputs are then used to update the planning data of mid-term planning, which in turn is accomplished every week for the following 6 weeks, immediately followed by the short-term planning just for the next week in the horizon.

1.2. Research objectives

As mentioned before, the thesis is motivated by problem-driven operations research. Its ultimate goal is to tackle beverage industry production planning problems with quantitative tools to support decision-making process. Still, other semi-continuous process industries share common features with beverage industry and can also benefit from this project. Meanwhile, besides aiming to make important contributions to the current state-of-the-art in terms of formulations and solutions approaches for all known problems, we are also motivated in applying operations research in practice.

The research objectives of the thesis are directly aligned with two of the major planning challenges faced by beverages companies: (O.1) setup times and costs dependent on the production sequence tackled at the short term planning and (O.2) the seasonal demand and the coordination of the multi-plant supply chain approached in the long term planning.

O.1 - Short-medium term planning

The lot sizing problem consists in the determination of the production lots to satisfy demand at minimum cost. Lot sizing models enclose the trade-off between setup and holding costs. On the other hand, production scheduling focuses on the allocation of resources to execute tasks at different time points, hence scheduling implies assigning and sequencing jobs. Pure lot sizing models do not sequence products in each period and implicitly require a setup for an item in every period in which it is produced. These models often define the inputs for scheduling decisions which are taken afterwards for each period separately following an hierarchical approach. By allocating unnecessary setups such strategy does not perform well in practice (Porkka et al. (2003)). This close relationship between lot sizing and scheduling makes it imperative that both decisions are made simultaneously (Drexl and Kimms (1997)). The first group of research objectives arises when looking at the production planning problem at the beverage industry integrating lot sizing and scheduling problems due to the sequence dependent setups. We will call hereafter this planning horizon a short-medium term planning for two main reasons: (1) the traditional planning horizon of the short term planning will be expanded due to the lot sizing decisions, (2) yet, not all the decisions of a medium term level will be covered, such as defining transfer quantities among locations of the supply chain.

To tackle this problem we have defined the following research objectives:

1. *Study the different modeling alternatives for lot sizing and scheduling and compare their efficiency when solving real-world size instances*

   Several approaches are known to introduce sequencing decisions in lot sizing models. The goal of this research line is to analyze their performance when solving large size instances as the ones that appear in practice. The outcome related to this study is a clear identification of the modeling alternatives more suitable for being used under different planning features.

2. *Develop modeling alternatives for lot sizing and scheduling*

   Besides studying the existent mathematical formulations of lot sizing and scheduling, we also seek to contribute with new model formulations. The goal is to develop models computationally more efficient, enhancing their potential to be used in mathematical programming-based heuristics.

3. *Develop a new mathematical programming-based heuristic and compare it to the state-of-the-art solution approaches to the problem*

   Despite the attention given to the modeling of lot sizing with sequence dependent setups, there is still work to be done in terms of solution procedures to a problem known to be NP-hard. Such solution procedure would allow to integrate medium term decisions in short-term planning performed by companies in beverage and related industries, in order to achieve plans that use more efficiently the available capacity. We expect to obtain breakthroughs in state-of-the-art solution techniques, especially in matheuristics (Maniezzo et al. (2010)). It is our goal to obtain insights into the way mathematical formulations of the aforementioned research line can be combined with metaheuristics to develop efficient tools for solving this hard problem. In addition, the new mathematical programming-based heuristics should be easily adapted to cope with model extensions or to address similar optimization problems.

4. *Design and develop a practical rolling horizon method for short-medium term planning*

   Much of the research in lot sizing and scheduling problems has concerned just the optimization of its static version without taking into account the fact that the optimal solutions are usually applied on a rolling horizon fashion. In this planning approach only the production decisions related to the first period or periods (depending on the planning frequency) are implemented after which the horizon is rolled forward and the model/method is solved once more with updated data.

   The expected scientific results of this objective are new formulations and solution methods for lot sizing and scheduling taking advantage of the knowledge apprehended with the previous research issues, but also embedding the principles of rolling planning.

O.2 - Long term planning

Long term planning in the presence of geographically disperse production sites involves decisions concerning: product specialization, inventory location and capacity investments. As revisions in the supply chain configuration or client supply strategy may completely change the demand allocated to plants, it is vital to tackle distribution decisions at this level to ensure a holistic view of the supply chain. Therefore, and in contrast with the research objectives defined for shorter horizons, at the long term planning level we study both distribution and production planning.

The goals at this planning level are often to estimate the adequacy of resource capacity to demand forecasts, analyze supply chain behavior to future requirements or to study changes in the supply chain design. Although detailed plans are not the purpose, more realistic and accurate models translate into more potential for cost saving solutions. A key issue at this level is also the managerial acceptance of the solutions made available.

A substantial motivation for this study came from the disappointing answer given by the APS system running at the case study company for this planning level. The plans provided were viewed as unrealistic, since they did not capture correctly some of the operational constraints and decisions that were regarded as essential.

Hence, to close this gap we traced this group of research objectives as follows:

1. *Study the crucial operational features to be considered in mathematical formulations for the long term planning in the beverage industry*

   We intend to give new insights into current literature by formulating novel industrial extensions to tackle this topic. It is our goal that the models developed correctly represent the reality of the case study. The scarce literature devoted to this line of research neglects important production/distribution process features creating aggregate solutions with limited value to the current company practice.

   From the production planning perspective we expect to understand the benefits of coordinating different production facilities across the supply chain. This may potentially breach the gap discussed in the literature review, between the line of research dedicated to incorporate operational features into the models and the one dedicated towards integrating other supply chain decisions in lot sizing models.

   On the distribution planning side, the goal is to perceive the impact of the supply flexibility arising in the case study and the way it can be explored to lower supply chain costs.

2. *Design and develop new solution approaches to tackle long term planning in the beverage industry*

   Most of the work for long term planning is based on mathematical formulations and few solution approaches are suggested. Strategic decisions are traditionally based on what-if analysis. With additional features to be incorporated into the models as a result of the previous research objective, it is most likely that state-of-the-art optimization engines may face difficulties to generate good quality solutions in the reasonable computation time expected for such analysis. Therefore, we aim to develop matheuristics to tackle both the production and distribution problems. Our case study company constitutes an important test, as it can validate our approaches both in terms of model formulation and solution methodology.

1.3. Thesis Synopsis

The main chapters of this thesis consist of a collection of papers that seek to answer the research objectives defined in the previous section. Chapters 2, 3 and 4 are aligned with the objectives pursued for the short-medium term planning, and Chapters 5 and 6 comprise answers to the challenges identified for the long term planning. In this section, we overwiew the main aspects covered and the most substantial contributions associated with each of them.

Chapter 2 proposes a two-dimensional classification framework to survey and classify the main modeling approaches to integrate sequencing decisions in discrete time lotsizing and scheduling models. This is aligned with the first research objective defined for the short-medium term planning. We also perform extensive computational experiments to assess the performance of various models, in terms of running times and upper bounds, when solving real-word size instances. This allow us to identify the most suitable models to use in this planning step. In the meantime, we also present a new formulation for the problem using commodity flow based subtour elimination constraints, which gives very good results in terms of the trade-off between solution quality and computational effort. Our contributions are the new classification framework to classify modeling approaches to lot sizing and scheduling with sequencing decisions, the new commodity flow based formulation, and the computational tests. The latter present an evaluation of the pros and cons of the different modeling techniques, comparing models which, to the best of our knowledge, had never been compared, before.

Chapter 3 discusses a novel mathematical programming based approach to deliver superior quality solutions for the single machine capacitated lot sizing and scheduling problem with sequence dependent setup times and costs (CLSD). The matheuristic is built over an also new mathematical model, which explores the idea of scheduling products based on the selection of known production sequences. The hybrid heuristic integrates the pricing principles in well known MIP-based heuristics and conducts a partial exploration of distinct neighborhood structures to avoid local entrapment, by selecting neighbors on a rule base scheme. We show its potential by using benchmark instances with distinct features, for which the heuristic maintained a very good performance. The main contributions lay in two fronts: mathematical formulations and solution approach. In terms of mathematical formulations both the sequence based MIP model and the formulation that combines a compact and an extended formulation within a single model are new to the literature. And on the algorithmic front, we create new MIP-based construction and improvement heuristics using column generation.

In Chapter 4, in line with the current case study practice, the mathematical formulations to the CLSD are adapted aiming their use on a rolling horizon planning basis. The objective is to explore the principles of rolling planning to develop efficient approaches to the problem. The planning horizon is decomposed in two parts: a detailed section in the beginning of the horizon considering production sequences explicitly, followed by a second section where a rough plan gives an estimation of future costs and potential capacity shortages. In this context, we review the simplifications strategies proposed in the literature for the undetailed horizon and introduce a new model that incorporates the setup loss in the future periods capacity based on the loss observed in the detail section. Besides this new model, an important main contribution is an iterative method designed to improve the accuracy of the approximate parameters used in the simplified formulations. The method is modular and can be applied to refine the estimation of distinct parameters arising in the different models, giving very interesting results in our computational experiments.

Chapters 5 and 6 address the two research objectives established for the long term planning of production and distribution, respectively.

In Chapter 5, driven by the long term production planning task at the case study, a new mathematical formulation assigns and schedules production lots in a multi-plant environment, where each plant has a set of filling lines, and considers final product transfers between the plants. Furthermore, we develop a hybrid algorithm that explores sensitivity analysis to guide a partial neighborhood search embedded in a Variable Neighborhood Search scheme. We show that the new algorithm can substantially improve the current business practice, and it is more competitive than state-of-the-art commercial solvers and other VNS variants. Results indicate a cost reduction of up to 40% in practice, estimated in about 1.2M euros in 2011. The main contributions of this chapter are the more realistic formulation for the long term production planning in the beverage industry and the new algorithm. In particular, it should be highlighted the ideas enclosed in the sensitivity analysis guided search to partially explore large neighborhoods.

Chapter 6 follows the work done in the previous chapter to further detail the solutions in terms of the distribution process. We discuss the design, development and implementation of an operations research (OR)-based approach to support managers of the case study to take their tactical decisions concerning distribution planning. We propose an innovative model to grasp the operational complexity in an higher planning level and a mathematical programming-based heuristic to achieve good quality solution in acceptable running times. This chapter is much dedicated in engaging the main issues of implementing this OR-based solution at the company. We show the steps to its implementation and report a potential cost reduction of up to 2M euros per year compared to the company's plan in 2012. The main contributions are the methodology followed to incorporate operational detail in the new MIP model and the implementation insights.

Finally, Chapter 7 summarizes the work and suggests directions for future research.

Figure 1.3 summarizes in a graphical way the main contributions of each chapter. Chapter's contributions are evaluated along the two main axes of the thesis: scientific and practical grounds and, mathematical formulations and solution approaches. To place the boxes we weight their relevance from the practical and scientific point of view, and from the mathematical formulation and solution approach.

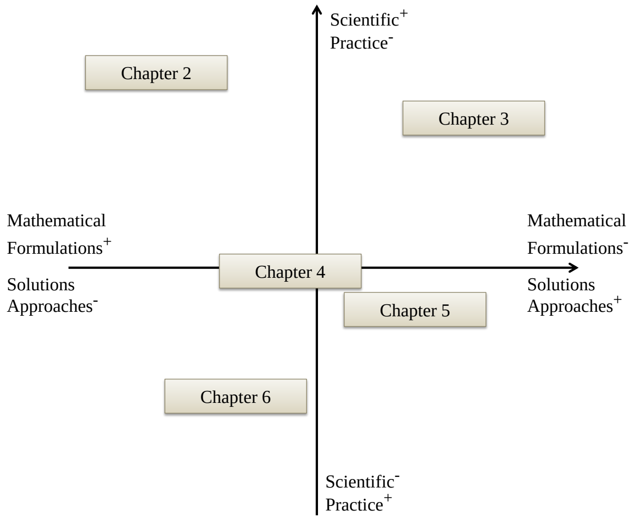


Figure 1.3: Summary of contributions

Bibliography

A. Drexl and A. Kimms. Lot sizing and scheduling - survey and extensions. *European Journal of Operational Research*, 99(2):221–235, 1997.

B. Fleischmann and H. Meyr. Planning hierarchy, modeling and advanced planning systems. In S. Graves and A. de Kok, editors, *Supply Chain Management: Design, Coordination and Operation*, volume 11 of *Handbooks in Operations Research and Management Science*, pages 455 – 523. Elsevier, 2003.

J. Kallrath. Planning and scheduling in the process industry. *OR Spectrum*, 24(3):219–250, 2002.

G. M. Kopanos, L. Puigjaner, and M. C. Georgiadis. Production scheduling in multiproduct multistage semicontinuous food processes. *Industrial & Engineering Chemistry Research*, 50(10):6316–6324, 2011.

V. Maniezzo, T. Stützle, and S. Voß. *Matheuristics*, volume 10 of *Annals of Information Systems*. Springer US, 2010.

H. Meyr and H. Stadtler. Types of supply chains. In *Supply chain management and advanced planning*, pages 65–80. Springer, 2005.

P. Porkka, A. P. J. Vepsäläinen, and M. Kuula. Multiperiod production planning carrying over set–up time. *International Journal of Production Research*, 41(6):1133 – 1148, 2003.

CHAPTER 2

Modeling lotsizing and scheduling problems with sequence dependent setups

Modeling lotsizing and scheduling problems with sequence dependent setups

Luis Guimarães* · **Diego Klabjan†** · **Bernardo Almada-Lobo***

Submitted to *European Journal of Operations Research*, 2013

Abstract Several production environments require simultaneous planing of sizing and scheduling of sequences of production lots. Integration of sequencing decisions in lotsizing and scheduling problems has received an increased attention from the research community due to its inherent applicability to real world problems. A two-dimensional classification framework is proposed to survey and classify the main modeling approaches to integrate sequencing decisions in discrete time lotsizing and scheduling models. Computational experiments are conducted to assess the performance of various models, in terms of running times and upper bounds, when solving real-word size instances. We also present a new formulation for the problem using commodity flow based subtour elimination constraints.

Keywords Production Planning · Lotsizing and Scheduling · Mathematical Programming

2.1. Introduction

Several companies face the problem of timing and sizing production lots over a given planning horizon. Additionally, in many of these production environments, switching between production lots of two different products triggers operations, such as machine adjustments and cleansing procedures. These setup operations, which are dependent on the sequence, consume scarce production time and may cause additional costs due to, for example, losses in raw materials or intermediate products. Consequently, the production sequence must be

*INESC TEC, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal
†Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA

explicitly embedded in the lot definition and scheduling. Lot sizing determines the timing and level of production to satisfy deterministic product demand over a finite planning horizon. Sequencing establishes the order in which lots are executed within a time period, accounting for the sequence–dependent setup times and costs. Integration of these two problems enables the creation of better production plans than those obtained when solving the two problems hierarchically by inducing the solution of the lotsizing problem in the scheduling level. Production plans are created with the objective of minimizing the overall costs consisting mainly of stock holding and setups, while satisfying the available capacity in each time period from which the expenditure in setup times is deducted.

This production scenario is present in many process industries, in which an efficient use of the available capacity is key to stay competitive in the current market environment. In the beverage industry sequence dependent setups occur in bottling lines when switching between two products that differ in the container size and/or container shape and/or liquid type. Another case comes from the glass container industry, in which costly changeovers are incurred in molding lines due to differences in the container mold and/or in the glass color among products. Similarly, in automated foundries time and cost expenditures in setups are dependent on the sequence of changes both in the alloy type and piece molds triggered at casting machines. The problem of production sequencing is also important in the textile industry on spinning facilities. The planned production sequence of yarn packages define the required setups to change the fiber blend and also provoke adjustments in yarn machines. More real world examples are present in chemicals, drugs and pharmaceuticals, pulp and paper, animal nutrition, among other industries.

From a research perspective, the aforementioned problems belong to the field of lotsizing and scheduling problems (LS). LS models are usually expressed in the form of mixed integer programming (MIP) formulations. The advances observed in mathematical programming in recent years combined with the increase in computational power (hardware) and in the quality of general purpose mixed-integer programming commercial solvers (software) allowed sequence independent LS problems to be solved efficiently using exact methods for reasonable size instances. However, the development of tighter mathematical formulations is still mandatory to reduce the running times needed to solve LS instances with sequencing decisions, particularly when dealing with real world constraints and problem sizes. As a result, both the complexity and inherent applicability to real world problems caused an increased enthusiasm from the research community to tackle LS problems with sequencing decisions. This interest is shown in the reviews by Drexl and Kimms (1997); Zhu and Wilhelm (2006); Jans and Degraeve (2008); Quadt and Kuhn (2008) and especially by the recent special issue Clark et al. (2011). Researchers have been incorporating additional scheduling decisions and features into LS models to improve their realism and potential applicability. However, none of the aforementioned reviews focuses on modeling techniques to integrate sequencing decisions in LS models and their impact on the solution quality achieved.

In this paper we first propose a framework to classify discrete time models for LS with sequencing decisions using two main sequencing dimensions: technique and time structure. Only the most relevant models in each class are reviewed to show their main features and to highlight the differences among them. Besides reviewing the models present in the literature we also introduce a new polynomial-sized model formulation to the problem which uses commodity flow based constraints to eliminate disconnected subtours and allows for multiple lots of the same product within each time period.

The performance of the models reviewed in the context of the framework and also of the new formulation is assessed by solving large size instances of the problem using a mixed-integer programming commercial solver. During the computational experiments we analyze the trade-offs present in these different modeling approaches. First, we study the correlation between the complexity introduced by allowing more general sequences (e.g. product repetition) and the solution quality obtained when a time limit is imposed. Second, we compare the use of an exponential number of constraints and variables against the use of compact model formulations. We focus on running times and upper bounds since our goal is to test the capability of providing solutions to instances of real-world size. In addition, many solution procedures for LS combine heuristics with exact methods, such as the progressive interval heuristics and the 'exchange' (fix-and-optimize) improvement heuristic, which rely on the solution of a series of sub-MIPs, also depend on the generation of good upper bounds. Hence, this assessment of the formulations can potentially contribute to the identification of the potentially most efficient MIP formulations to be used in these hybrid methods.

Our contributions are as follows. We present a new classification framework to classify modeling approaches to LS with sequencing decisions. The new framework is used to survey and classify the different modeling approaches present in the literature grouping models into classes. We also introduce a new commodity flow based formulation to integrate sequencing decisions in discrete time LS models. Finally, the extensive computational results present an evaluation of the pros and cons of the different modeling techniques, comparing models which, to the best of our knowledge, had never been compared. This enabled us to pinpointing the most efficient models in the several contexts studied.

The remainder of this paper is organized as follows. Section 2.2 presents the proposed classification framework for the modeling approaches. In Section 2.3 we describe the problem under study and all the assumptions made. Following the classes defined in our framework Sections 2.4 and 2.5 present the reviewed models, as well as introduce the new formulation proposed herein. Computational experiments assessing the models performance are shown in Section 2.6. Finally, Section 2.7 is devoted to final remarks, where conclusions from this work and some potential future research directions are highlighted.

2.2. Modeling sequence-dependent setups

In this section, we introduce a framework to classify the discrete time modeling approaches existing in the literature for LS with sequencing decisions. The framework is organized along two main sequencing dimensions: technique and time structure (see Figure 2.1). A class is defined by the technique and time structure used, e.g. product oriented macro period (PO-MP) models.

The sequence of production lots in a machine can be categorized following the definitions given by Kang et al. (1999): a *production-sequence* refers to the sequence of products

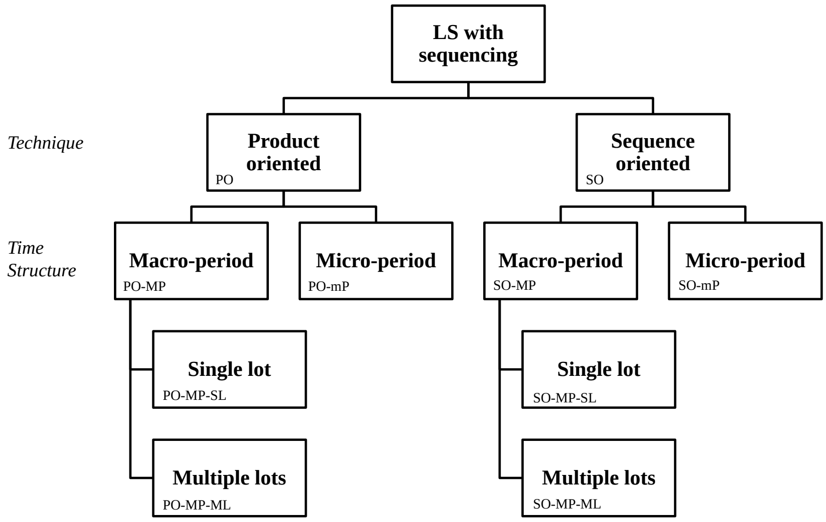
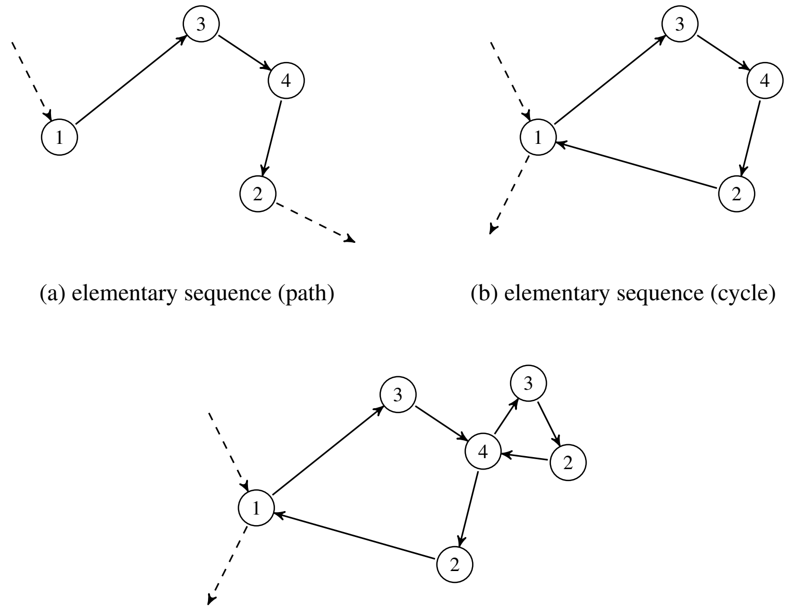


Figure 2.1: Proposed classification framework

being produced on the machine over the entire planning horizon and a *period-sequence* denotes the sequence of setup states within a time period. In discrete time models for LS with sequencing decisions a production-sequence decomposes into period-sequences, hence the term sequence will be used hereafter to refer to period-sequences. The first dimension used for classification regards the technique used to capture sequencing decisions. Two main approaches are distinguished: product oriented (PO) and sequence oriented (SO) formulations. When using a PO technique, sequences are explicitly defined by the MIP model, while in SO formulations the MIP model prescribes for each period a sequence from a pre-determined set of sequences, i.e. the model selects one sequence from the set.

Consider the representations of sequences depicted in Figure 2.2. By definition a sequence is a connected direct graph where each node i represents a production lot of product i and arc (i, j) indicates a setup from product i to product j. Additionally, the dashed arcs identify the first (input arc) and the last (output arc) production lots in the sequence, i.e. the initial and final setup state of the machine. A SO formulation corresponds to the selection of a connected graph (sequence) to be applied in each time period, thus it does not require additional constraints to ensure the connectivity of the setup decisions. On the other hand, a PO formulation operates on the selection of arcs (setups) to be performed in each time period, hence the so-called disconnected subtour elimination constraints, which can be of an exponential size, are often required to ensure the connectivity of the subgraph induced by setup decisions. This is a major difference between these two approaches and explains why sequence oriented based formulations are easier to model. However, this potential advantage has the drawback of the number of possible sequences (decision variables) growing exponentially with the number of products present in the problem instance.



(a) elementary sequence (path)

(b) elementary sequence (cycle)

(c) non-elementary sequence

Figure 2.2: Examples of possible sequences.

The second dimension of the framework classifies the models concerning the time structure used to capture sequencing decisions. Time discretization of LS models usually follows exogenous criteria such as demand forecast granularity to partition the planning horizon into several time periods, also called macro-periods. When the macro-period structure is adopted to capture sequencing decisions, in each of these time periods more than one setup is allowed. Sequencing decisions in macro-period (MP) models are made through decision variables similar to those of routing problem formulations and require subtour elimination constraints to correctly represent sequences. On the other hand, some models create a second level in the time structure by dividing each macro-period into more than one micro-period. The assumption in micro-period (mP) models is that at most one setup is performed per micro-period and, thus, the production-sequence comes for free directly from the setup state changes among adjacent micro-periods. In the scope of sequencing decisions on the number of micro-periods limit the maximum number of setup operations allowed in each macro-period.

To illustrate an example consider the sequence {1-3-4-2} shown in Figure 2.2a which defines a production lot sequence in a given time period. A mP model would require at least 4 micro-periods to describe this sequence corresponding to the 4 setup states of the machine. Suppose the number of micro-periods was set to 5; the sequence can be captured by defining the setup state in each of the micro-periods as (1)(3)(4)(2)(2) and then changes among adjacent micro-periods capture the setups performed. To the contrary, a MP model would select setups (1-3) (3-4) (4-2) to establish the sequence depicted in the example.

Classifying discrete time LS models according to the type of time partition has been commonly accepted and used in the community, which groups models into large and small bucket. In large bucket models the planning horizon is partitioned into a small number of long time periods representing, in most cases, a week or month. To the contrary, in small bucket models the planning horizon is divided into a large number of short periods (e.g. days, shifts or hours). Our classification according to the time structure is inspired and closed related to the established terminology. However, there is a clear difference in the classification of models having a multi-level time structure. The established classification only applies to the top level of the time structure, while our framework classifies models according to the level in which the sequencing decisions are captured. Hence, we chose to select different names for this dimension to avoid misinterpretations. In example, same multi-level time structure models capture inventory balance decisions in the top level and sequencing decisions in the lower level, in this case the established classification differs from ours.

MP models can be further divided according to the number of production lots of each product allowed to start within a time period into single lot (SL) and multiple lots (ML) models, giving origin to subclasses. Usually, setups obey the triangle inequality with respect to both the setup time and costs, i.e. it is more efficient to change directly between two products than via a third product. Under this setting in any optimal solution, at most one setup for each product per time period is performed (single lot). Nevertheless, in some industries, contamination occurs when changing from one product to another implying additional cleansing operations. If a 'cleansing' or shortcut product can absorb contamination while being produced and therefore replacing the cleansing operations, non-triangular setups appear. Thus, allowing multiple lots of each product per time period can potentially reduce setup times and costs. The need for multiple lots can also come from industries where production batches are bounded or of fixed size. This distinction is made since tackling multiple lots in the same period is a non trivial extension to most models. Furthermore, minimum lot size is important in the case of non-triangular setups to avoid fictitious setups via empty product lots (zero production). Figures 2.2a and 2.2b are examples of sequences which can be obtained by using MP-SL models, while the sequence illustrated in Figure 2.2c can only be achieved using a MP-ML formulation.

Table 2.1 presents the models which will be reviewed in the following sections in each of the classes defined.

2.3. Problem definition

To describe the deterministic LS problem addressed here, consider N products indexed by $i, j = 1, \dots, N$ to be produced on a single capacitated machine over a finite planning horizon of T periods, indexed by $t = 1, \dots, T$. The following data is associated with this problem:

- d_{it} demand of product i in period t (units),
- h_{it} holding cost of one stock unit of product i in period t (cost/unit),
- cap_t capacity of the machine in period t (time),

Table 2.1: Classification of lotsizing and scheduling models with sequencing decisions

Time Structure			Technique: Product oriented - PO		Technique: Sequence oriented - SO	
Macro-period MP	Single lot MP-SL		SDR	Smith-Daniels and Ritzman (1988)	HK	Haase and Kimms (2000)
			H	Haase (1996)		
			AL1	Almada-Lobo et al. (2007)		
			AL2	Almada-Lobo et al. (2007)		
	Multiple lots MP-ML		BW	Belvaux and Wolsey (2001)	GKAL	Guimarães et al. (2013)
			MCAL	Menezes et al. (2011)		
			MCF	Sarin et al. (2011)		
			SCF	this paper		
Micro-period mP			GLSP	Fleischmann and Meyr (1997)	KANG	Kang et al. (1999)
			GLSP^{NF}	Wolsey (1997)		
			CC	Clark and Clark (2000)		

p_{it} processing time of product i in period t (time/unit),

b_{it} the maximum amount of product i that can be produced in period t (units),

sc_{ij} cost incurred to set up the machine from product i to product j (cost),

st_{ij} time needed to set up the machine from product i to product j (time),

m_i minimum lot size of product i (units).

Before presenting the modeling approaches, we introduce the assumptions to clearly define the problem tackled.

- Stockouts are not accepted, which is a common setting in deterministic demand environments.
- Initial inventory is considered to be zero. Nevertheless, both, the consideration of initial inventory and stockouts, are relatively straightforward extensions.
- The setup state is carried over among adjacent periods. Moreover, the setup state is preserved over idle time.
- Setup crossovers are not allowed, which forces setup operations to be performed within the time period, i.e. without spanning to the following period.
- The machine configuration at the beginning of the planning horizon is not defined and thus it is a decision.
- No structure is imposed either on setup times and costs, or on their relation.
- Minimum lot sizes are imposed to avoid fictitious setups via empty production lots whenever non-triangular setups exist.
- In the presence of minimum lot sizes we assume that at least one setup is performed in every time period.

The validity of the assumptions made for setup crossovers and minimum lot sizes rely on the fact that these models are designed by considering that several products can be produced per period (e.g. a week, a month). Hence, not allowing for setup crossovers or assuming at least one setup per period should not exclude high quality production plans.

2.4. Product oriented formulations

2.4.1 Micro-period models

The mP formulations rely on the division of the time periods into several micro-periods. Drexl and Kimms (1997) survey models based on this partition highlighting the different assumptions. We discuss the most general of such models, the General Lotsizing and Scheduling Problem - GLSP (Fleischmann and Meyr (1997), Meyr (2002), Meyr (2000)). The GLSP embeds a two-level time structure being the upper lever composed by the macro-periods and the lower level devised by the division into micro-periods of each macro-period. Hence, the GLSP is usually referred as a large bucket or hybrid model, as opposed to the small bucket models, namely the Discrete Lotsizing and Scheduling Problem (Fleischmann (1994)), the Continuous Setup Lotsizing Problem (Almada-Lobo et al. (2010)) and the Proportional Lotsizing and Scheduling Problem (Drexl and Haase (1995)), which assume a fixed micro-period duration and a single level time structure. In the GLSP the micro-period length is a decision in the optimization process, thus it potentially allows for better solutions than the small bucket models. We introduce the following decision variables to model the GLSP:

X_{in} quantity of product i produced in micro-period n,

I_{it} stock of product i at the end of period t,

Y_{in} =1 if the machine is set up for product i in micro-period n,

T_{ijn} if a changeover from product i to product j is performed at the beginning of subperiod n.

Additionally, let $A_t = \{1, \ldots, l_t\}$ be the set of micro-periods n belonging to time period t and l_t the maximum number of lots allowed in time period t. The GLSP model is as follows:

$$\text{GLSP} \qquad \min \sum_{i,t} h_{it} \cdot I_{it} + \sum_{i,j,n} sc_{ij} \cdot T_{ijn} \tag{2.1}$$

$$\text{s.t.} \qquad I_{i,t-1} + \sum_{n \in A_t} X_{in} = d_{it} + I_{it} \qquad \forall\, i,\, t, \tag{2.2}$$

$$\sum_{i,n \in A_t} p_{it} \cdot X_{in} + \sum_{i,j,n \in A_t} st_{ij} \cdot T_{ijn} \leq cap_t \qquad \forall\, t, \tag{2.3}$$

$$X_{in} \leq b_{it} \cdot Y_{in} \qquad \forall\, i,\, t,\, n \in A_t, \tag{2.4}$$

$$\sum_i Y_{in} = 1 \qquad \forall\, n, \tag{2.5}$$

$$T_{ijn} \geq Y_{i,n-1} + Y_{jn} - 1 \qquad \forall\, i,\, j,\, n, \tag{2.6}$$

$$X_{in} \geq m_i \cdot (Y_{in} - Y_{i,n-1}) \quad \forall i, n, \tag{2.7}$$

$$X, I, T \geq 0, \quad Y \in \{0,1\}. \tag{2.8}$$

The objective function (2.1) minimizes the total sum of holding and setup costs. Inventory balance constraints (2.2) satisfy demand either from initial inventory or production within the current period. The total period's production of each product is obtained by summing up the productions in the different micro-periods. Inequalities (2.3) ensure that the total production time plus the required setup time does not exceed the available capacity. The correct relation between production quantities and the machine setup state in each micro-period is expressed by (2.4), while (2.5) enforce a single setup state per micro-period. Constraints (2.6) trace changeovers throughout the planning horizon. Minimum lotsizes are introduced by constraints (2.7) to prevent empty lots and thus an incorrect evaluation of setup times and cost if the setup matrix does not obey the triangle inequality.

Figure 2.3 graphically represents the same sequence depicted in Figure 2.2c using the network interpretation of the GLSP and dividing a macro-period into 8 micro-periods. Essentially, it corresponds to a path in a directed graph where nodes are possible setup states in each micro-period and arcs connecting setup states are changeovers.

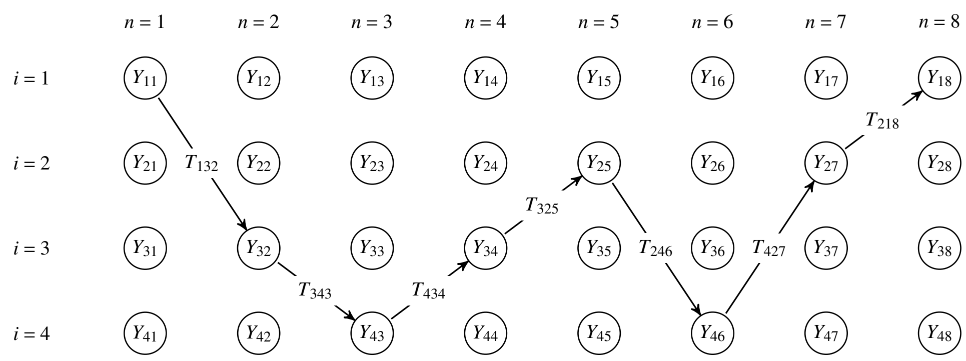


Figure 2.3: An example of sequence {1-3-4-3-2-4-2-1} using a PO-mP formulation

As suggested by the previous figure, changeover constraints (2.6) can be reformulated as a shortest path or network flow (NF) problem from the first micro-period to the last. This yields a substantially tighter model as shown in Wolsey (1997). The strength of the (GLSP^{NF}) reformulated model comes from the fact that, when no other constraints are present, the extreme points of the relaxed problem are integer solutions. The reformulated constraints are stated as:

$$\sum_j T_{ijn} = Y_{i,n-1} \qquad \forall i, n, \tag{2.9}$$

$$\sum_i T_{ijn} = Y_{jn} \qquad \forall j, n. \tag{2.10}$$

Constraints (2.9) force a changeover from the product produced in the previous micro-period at the beginning of the current micro-period. Similarly, constraints (2.10) force a changeover to the product being produced in each micro-period. Note that is possible to have a changeover from product i to itself and thus these constraints act as setup state conservation constraints allowing production lots to span over multiple micro-periods. The complete GLSP^{NF} model reads (2.1) - (2.5) and (2.7) - (2.10).

Exploring the same idea of fixing the maximum number of changeovers in each time period, the model of Clark and Clark (2000) (CC) is closely related to the GLSP and GLSP^{NF}. In this formulation, setup state decision variables Y are dropped and the integrality of T is imposed to account for the changeovers and setup state in each micro-period. The CC formulation is as follows:

$$\text{CC} \qquad \min (2.1)$$

$$\text{s.t.} \qquad (2.2) - (2.3)$$

$$\sum_{i,j} T_{ji0} = 1 \tag{2.11}$$

$$X_{in} \leq b_{it} \cdot \sum_{j} T_{jin} \qquad \forall\, i, t, n \in A_t, \tag{2.12}$$

$$\sum_{j} T_{ji,n-1} = \sum_{j} T_{ijn} \qquad \forall\, i, n, \tag{2.13}$$

$$X_{in} \geq m_i \cdot \sum_{j \neq i} T_{jin} \qquad \forall\, i, n, \tag{2.14}$$

$$X, I \geq 0, \quad T \in \{0,1\}. \tag{2.15}$$

In this formulation, constraint (2.11) defines the initial setup state of the machine. Constraints (2.12) guarantee that production of a given product only occurs if the machine is set up at the beginning of the micro-period, which can be either by an actual setup or via conservation of the previous setup state. Flow constraints (2.13) simultaneously keep track of changeovers and machine configuration state. These constraints have a similar structure as constraints (2.9) - (2.10) also capturing the network flow interpretation of the model. Minimum lotsizes (2.14) are again imposed to address non-triangular setups. In Appendix 2.A we show that the CC formulation is stronger than the original GLSP formulation.

2.4.2 Macro-period models

The problem of extending the traditional capacitated lotsizing problem (CLSP) to account for sequencing decisions is known as the CLSD. To formulate the CLSD we introduce the binary decision variables Z_{it} which equals one if the machine is set up for product i at the beginning of period t capturing the setup state conservation among adjacent periods. We also update the definition of variables T_{ijt} to be the number of changeovers from product i to product j in time period t and include q_{it} as an upper bound on the number of setups to product i in period t. A general CLSD model formulation reads:

$$\text{CLSD} \qquad \min \sum_{i,t} h_{it} \cdot I_{it} + \sum_{i,j,t} sc_{ij} \cdot T_{ijt} \tag{2.16}$$

$$\text{s.t.} \quad I_{i,t-1} + X_{it} = d_{it} + I_{it} \qquad \forall i, t, \tag{2.17}$$

$$\sum_i p_{it} \cdot X_{it} + \sum_{i,j} st_{ij} \cdot T_{ijt} \leq cap_t \qquad \forall t, \tag{2.18}$$

$$X_{it} \leq b_{it} \cdot \left(\sum_j T_{jit} + Z_{it} \right) \qquad \forall i, t, \tag{2.19}$$

$$\sum_i Z_{it} = 1 \qquad \forall t, \tag{2.20}$$

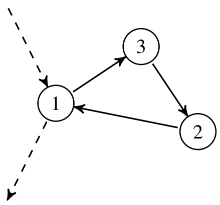
$$Z_{it} + \sum_j T_{jit} = \sum_j T_{ijt} + Z_{i,t+1} \qquad \forall i, t, \tag{2.21}$$

$$X, I \geq 0, \quad Z \in \{0,1\}, T_{ijt} \in \{0, \ldots, q_{jt}\}, \tag{2.22}$$

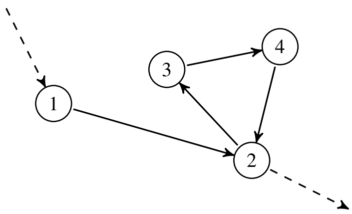
$$\{(i,j) : T_{ijt} > 0\} \text{does not include disconnected subtours} \, \forall t. \tag{2.23}$$

As in mP models the objective function (2.16) minimizes the total expenditure in holding and setup costs. Constraints (2.17) and (2.18) express, respectively, the common inventory balance and capacity constraints from large bucket formulations. Production is linked by the machine setup state through constraints (2.19); production may only occur if a setup is carried over from the previous period or at least one setup is performed in the period. Constraints (2.20) guarantee that the machine is set up for a single product in the beginning of each time period. Machine configuration is traced in (2.21) which ensures a balanced flow of setups. If there are no setups in period t the machine configuration is carried to period $t+1$. On the other hand, for each product i three cases may appear: more input than output setups, more output than input setups and equal number of input and output setups. The first case forces the machine to be set up for product i in the beginning of the next period. Similarly, in the second case the machine must be set up for product i in the beginning of period t. Finally, the third case happens when the product is neither the first nor the last in the sequence, or no setup occurs in the period.

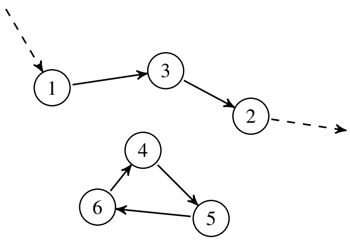
The formulation would be incomplete without constraints (2.23) which prevent disconnected subtours to create feasible integer solutions. Subtours are sequences that start and end at the same setup state. Figure 2.4 illustrates examples of subtours that can appear in a solution for the CLSD without constraints (2.23) and classified according to the notation of Menezes et al. (2011). An alpha subtour (see Figure 2.4a) is defined by $Z_{it} = Z_{i,t+1} = 1$ and $\sum_{i,j} T_{ijt} \geq 1$, meaning that at least one setup is performed in time period t and the machine configuration at the beginning and end of the period is the same. Figure 2.4b depicts the case of a connected subtour created by the existence of two production lots for product 2. Figures 2.4c and 2.4d show examples of disconnect subtours, i.e. sequences that are not connected to the main sequence. This class of subtours can be divided into simple disconnected subtours (Figure 2.4c, subtours forming a single cycle) and complex disconnect subtours (Figure 2.4d, subtours formed by multiple connected cycles). Note that only disconnected subtours should be prevented as alpha or simple connected subtours can be part of feasible integer solutions. Next we discuss several strategies to define constraints (2.23) giving origin to the different PO-MP models, starting with single lot formulations and later advancing to multiple lot versions.



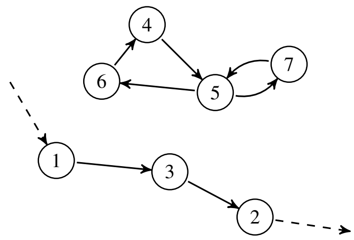
(a) main sequence as an alpha subtour - {1-3-2-1}



(b) main sequence with a simple connected subtour - {2-3-4-2}



(c) main sequence with a simple disconnect subtour - {4-5-6-4}



(d) main sequence with a complex disconnect subtour - {4-5-7-5-6-4}

Figure 2.4: Examples of possible subtours.

2.4.2.1 Single lot

In the scope of single lot formulations by definition $q_{it} = 1$ for every i, t and, therefore, T_{ijt} are defined as binary, and connected and complex disconnect subtours are automatically excluded. We start by reviewing models which adopt constraints similar to those of the Miller, Tucker and Zemlin (MTZ) formulation of the Asymmetric Traveling Salesman Problem (ATSP). The first is the Smith-Daniels and Ritzman (SDR) formulation (Smith-Daniels and Ritzman (1988)), that uses C_{it} to map the completion time of the production lot of product i in period t. The subtour elimination constraints which replace (2.23) are stated as:

$$C_{jt} \geq C_{it} + st_{ij} + p_{jt} \cdot X_{jt} - M \cdot (1 - T_{ijt}) \qquad \forall i, j \neq i, t, \tag{2.24}$$

$$C_{it} \geq p_{it} \cdot X_{it} \qquad \forall i, t. \tag{2.25}$$

Constraints (2.24) guarantee that if product j follows product i in the sequence its completion time is greater than the completion time of product i plus the time required for setting up the machine and production. The completion time of the first production lot of each time period has to be also imposed, as done by (2.25). The authors did not present the above constraints as subtour elimination constraints, nevertheless they do translate into an accurate formulation of the problem. The SDR formulation can be further tighten by defining an upper bound of $C_{it} \leq cap_t$ to all completion times and by setting $M = st_{ij} + cap_t$. One of the advantages of computing the completion times comes from the possibility of imposing time windows to production lots or to synchronize parallel resources in multi-machine settings or multi-stage processes.

Another formulation based on MTZ type constraints is presented by Haase (1996) (H) that uses decision variables V_{it} to capture the order in which production lots are processed in each time period and to eliminate subtours. Constraints (2.23) are defined as:

$$V_{jt} \geq V_{it} + 1 - N \cdot (1 - T_{ijt}) \qquad \forall i. j \neq i, t. \tag{2.26}$$

The above MTZ-based constraints (2.24) and (2.26) define sequences as paths. Therefore if a given product i is carried over from period $t-1$, i.e. $Z_{it} = 1$, no setup can be performed to it during time period t. This follows from the fact that both sets of constraints imply $\sum_j T_{jit} + Z_{it} \leq 1$ and thus eliminate alpha subtours. Later Almada-Lobo et al. (2007) extended (2.26) in order to include alpha subtours. The reformulated constraints (2.26) present in Almada-Lobo et. al (AL1) are as follows:

$$V_{jt} \geq V_{it} + 1 - N \cdot (1 - T_{ijt}) - N \cdot Z_{it} \qquad \forall i, j \neq i, t. \tag{2.27}$$

In the same work the authors proposed an alternative (AL2) formulation using an exponential number of constraints and also proved that AL2 is stronger than AL1. Due to the assumption is at most one setup per time period for each product, the following constraints hold:

$$\sum_j T_{ijt} \leq 1 \qquad \forall i, t, \tag{2.28}$$

$$\sum_j T_{jit} \leq 1 \qquad \forall i, t. \tag{2.29}$$

Then constraints (2.27) are replaced by:

$$\sum_{i \in S, j \notin S} T_{ijt} + \sum_{i \in S} Z_{i,t+1} \geq \sum_j T_{jkt} \qquad \forall t, k \in S, S \subseteq \mathcal{N}. \tag{2.30}$$

Being $\mathcal{N}$ the set of all products. In case of a potential simple disconnected subtour S the left-hand side of constraints (2.30) equals 0 while the right-hand side is at least 1, thus violating the inequality. Note that as shown in the cited work these constraints are still valid for alpha subtours.

2.4.2.2 Multiple lots

Considering multiple production lots of the same product within each time period is a non-trivial extension. To start, one has to deal with integer T_{ijt} variables, and, moreover, connected subtours are now allowed in sequences. The first formulation comes from Belvaux and Wolsey (2001) and resembles the prize collecting ATSP. Let $Y_{it} \in \{0, \ldots, q_{it}\}$ be the number of setups to product i in period t. The Belvaux and Wolsey (BW) formulation is obtained by introducing new setup balance constraints (2.31)-(2.32) which replace (2.21) and the disconnect subtours elimination constraints listed as:

$$Z_{it} + \sum_j T_{jit} = Y_{it} \qquad \forall i, t, \tag{2.31}$$

$$\sum_j T_{ijt} + Z_{i,t+1} = Y_{it} \qquad \forall i, t, \tag{2.32}$$

$$\sum_{i,j \in S} T_{ijt} \le \sum_{i \in S} Y_{it} - \frac{1}{q_{kt}} \cdot Y_{kt} \qquad \forall t, k \in S, S \subseteq \mathcal{N}. \tag{2.33}$$

Subtour elimination constraints (2.33) build on the idea from the Dantzig, Fulkerson and Johnson (DFJ) formulation of the ATSP. When product k from the subset selected is in the sequence, these constraints establish that the number of changeovers within the subset has to be less than the cardinality of the subset. The only exception occurs when one of the products is the first product in the sequence, in this case the number of setups within the group may equal the subset's cardinality. The complete formulation for BW reads (2.16) - (2.20), (2.31) - (2.33) and the updated variable domain.

The formulation by Menezes et al. (2011) (MCAL) also has an exponential number of inequalities to eliminate disconnected subtours. Let us define G_{it} to be a binary variable which takes the value one if the machine is set up for product i at least once in period t. The following constraints replace (2.23) to correctly model the problem:

$$\sum_j T_{jit} + Z_{it} \ge G_{it} \qquad \forall i, t, \tag{2.34}$$

$$\sum_j T_{jit} + Z_{it} \le q_{it} \cdot G_{it} \qquad \forall i, t, \tag{2.35}$$

$$\sum_{i \in S, j \notin S} T_{jit} + \sum_{i \in S} Z_{it} \ge 1 - M \cdot \sum_{i \in S} (1 - G_{it}) \qquad \forall t, k \in S, S \subseteq \mathcal{N}, |S| \ge 2. \tag{2.36}$$

Both (2.34) and (2.35) establish the correct relationship between the product setup state, changeovers and the initial machine configuration. In order to be active, inequalities (2.36) require all the products in subset S to be produced in time period t. When active, it imposes that the number of changeovers coming from products not belonging to the subset and/or the machine initial setup configuration to products within the subset has to be greater than one, thus connecting the subset. To reduce the number of inequalities required the authors also introduced *a priori* set of constraints that prevent simple disconnected subtours. For that propose let us define binary decision variable Q_{ijt} which equals 1 if at least one changeover from product i to product j is performed in period t. The additional constraints are as follows:

$$T_{ijt} \ge Q_{ijt} \qquad \forall i, j, t, \tag{2.37}$$

$$T_{ijt} \le q_{jt} \cdot Q_{ijt} \qquad \forall i, j, t, \tag{2.38}$$

$$V_{jt} \ge V_{it} + 1 - M \cdot (1 - Q_{ijt}) - M \cdot \left(\sum_k T_{kit} + Z_{it} - Q_{ijt} \right) \qquad \forall i, j \ne i, t. \tag{2.39}$$

The complete MCAL model is (2.16) - (2.22), (2.34) - (2.39) and the variable domain definition.

Commodity flow based formulations

The last PO-MP formulations can be called *commodity flow* formulations and are also inspired in models of the ATSP. Disconnected subtours are eliminated with additional decision variables representing commodity flows through a network where the nodes are products, arcs represent the selected setups in the current solution and the flow has to satisfy conservation constraints. We consider two different formulations: single commodity flow (SCF) and multi-commodity flow (MCF).

The SCF model below is a new contribution of this work. The continuous variables F_{ijt} represent the flow of the commodity from node i to node j in period t. An artificial node indexed by 0 is introduced to capture the setup carryover acting as the source of the flow. Disconnected subtours are eliminated by ensuring the connectivity of the graph induced by non-zero T's. For this purpose, the following constraints enforce the existence of a path from the source to each product in the sequence:

$$\sum_{j} F_{0jt} = \sum_{j} G_{jt} \qquad \forall\, t, \tag{2.40}$$

$$\sum_{j=0}^{N} F_{jit} = G_{it} + \sum_{j} F_{ijt} \qquad \forall\, i, t. \tag{2.41}$$

$$F_{0it} \leq N \cdot Z_{it} \qquad \forall\, i, t, \tag{2.42}$$

$$F_{ijt} \leq N \cdot T_{ijt} \qquad \forall\, i, j, t. \tag{2.43}$$

Constraints (2.40) force the commodity flow to leave the source. The total flow amount is required to be equal to the number of products being produced in the period which is equivalent to the number of paths needed. The flow balance constraints are expressed by (2.41) which ensure that a unitary flow is sent to every selected node, corresponding to a path from the source to every product being produced in the time period. Both (2.42) and (2.43) impose an upper bound on the amount of flow traversing the arcs. Constraints (2.42) impose that the flow can only leave the source to the first product in the sequence, while (2.43) guarantee that the flow only traverses arcs in the current solution.

The MCF model was proposed by Sarin et al. (2011). The connectivity of the graph induced by the setups selected is preserved by forcing the existence of a path linking the source node to every product in the sequence only using the arcs selected. MCF uses the flow of N commodities to generate the subtour elimination constraints. Therefore, superscript k in flow variables represents the commodity being considered. These F variables are defined if arc (i, j) is used in the path from the source to product k. The set of subtour elimination constraints is given by:

$$\sum_{j} F^k_{0jt} = G_{kt} \qquad \forall k, t, \tag{2.44}$$

$$\sum_{j=0, j\neq k}^{N} F^k_{jit} = \sum_{j} F^k_{ijt} \qquad \forall k, i \neq k, t \tag{2.45}$$

$$\sum_{j=0}^{N} F^k_{jkt} = G_{kt} \qquad \forall k, t \tag{2.46}$$

$$F^k_{0jt} \leq Z_{jt} \qquad \forall k, j, t, \tag{2.47}$$

$$F^k_{ijt} \leq T_{ijt} \qquad \forall k, i, j, t. \tag{2.48}$$

Constraints (2.44) force the flow of commodity k to leave the source only if product k is part of the sequence. The flow conservation of commodity k is preserved by (2.45), while (2.46) force the path to end in product k. The flow can only use arcs in the current solution as imposed by (2.47) and (2.48).

Contrarily to SCF in which the flow variable in arc (i, j) represents the number of paths using this arc, in MCF the flow explicitly defines whether arc (i, j) is in the path from the source to product k, thus explicitly establishing the paths used. In fact, SCF is an aggregation of MCF which results that MCF provides a tighter relaxation (Öncan et al. (2009)).

2.5. Sequence oriented formulations

A different approach to model the sequencing decisions within the CLSP is to use a collection of pre-defined sequences which establish the items to be produced and their order. Associated with a given sequence s the following parameters are defined:

- $\widehat{sc}_s$ setup cost incurred if sequence s is selected,
- $\widehat{st}_s$ setup time incurred if sequence s is selected,
- g_{is} =1 if product i is present in sequence s,
- f_{is} =1 if product i is first in sequence s,
- l_{is} =1 if product i is last in sequence s.

As opposed to a PO technique, SO formulations do not explicitly define changeovers using decision variables T_{ijt}, but prescribe a set of changeovers by assigning values to W_s which equals one if sequence s is selected for production or zero otherwise.

Next we discuss several approaches to formulate the CLSP with sequencing decisions using a SO technique.

2.5.1 Micro-period models

SO-mP formulations rely on the same principle as PO formulations of dividing the time periods into smaller segments. The idea behind the model of Kang et al. (1999) is to divide every sequence into a pre-defined number (S_t^{max}) of *split-sequences*. Let L_t be the set of split-sequences belonging to time period t and R_r be the set of sequences s which are available to schedule production in split-sequence r. Parameter Br^{max} defines the maximum number of products in a split-sequence and it is imposed that products can not repeat in a split-sequence. The number of lots in each sequence is limited to $S_t^{max} \times Br^{max}$, which mimics the partition of time periods into micro-periods in PO-mP models. Before introducing the split-sequence model we need to redefine, in the context of this model, parameter g_{is} to equal one only in the case product i is present in sequence s in every but the last position, i.e. $g_{is} + f_{is} \leq 1$ for every i, s. The following additional decision variables are also introduced:

Y_{ir} =1 if product i is in the sequence selected for split-sequence r,

E_{ir} =1 if split-sequence r is empty and product i was the last product produced.

The model of Kang et al. (1999) using split-sequences is herein referred as KMT and it reads:

$$\text{KMT} \qquad \min \sum_{i,t} h_{it} \cdot I_{it} + \sum_{s} \widehat{sc}_s \cdot W_s \tag{2.49}$$

$$\text{s.t.} \qquad I_{i,t-1} + X_{it} = d_{it} + I_{it} \qquad \forall i, t, \tag{2.50}$$

$$\sum_i p_{it} \cdot X_{it} + \sum_{r \in L_t, s \in R_r} \widehat{st}_s \cdot W_s \leq cap_t \qquad \forall t, \tag{2.51}$$

$$\sum_{s \in R_1} W_s = 1, \tag{2.52}$$

$$\sum_{s \in R_r} f_{is} \cdot W_s - \sum_{s \in R_{r-1}} l_{is} \cdot W_s = E_{i,r-1} - E_{ir} \qquad \forall i, r \geq 2, \tag{2.53}$$

$$\sum_{s \in R_r} g_{is} \cdot W_s = Y_{ir} \qquad \forall i, r, \tag{2.54}$$

$$X_{it} \leq b_{it} \cdot \sum_{r \in L_t} Y_{ir} \qquad \forall i, t, \tag{2.55}$$

$$X_{it} \geq m_i \cdot \sum_{r \in L_t} Y_{ir} \qquad \forall i, t, \tag{2.56}$$

$$X, I, W, E \geq 0, \quad Y \in \{0, 1\}. \tag{2.57}$$

Objective function (2.49) minimizes the total expenditure in holding costs and setup costs incurred from sequence selection. Constraints (2.50) represent the classical inventory balance constraints. Note that production is not divided among the different split-sequences as one may expect in mP models. Capacity constraints are expressed by (2.51) in which the total setup time within the period is obtained by summing the setup times incurred in each split-sequence. Sequence selection in split-sequences is ruled by (2.52) and (2.53).

The first constraint imposes that a sequence must be chosen for the first split-sequence. Constraints (2.53) link sequence selection among adjacent split-sequences, moreover, they also preserve setup state when empty split-sequences occur. The relationship between product setup state in each split-sequence and sequence selection is guaranteed by (2.54). Finally, (2.55) and (2.56) define the bounds on production in each period according to the product setup state in the corresponding split-sequences.

2.5.2 Macro-period models

SO-MP formulations use sequences to determine sequencing decisions. Let S_t be the set of available sequences to schedule products on the machine in period t. We also group these formulations by the number of production lots allowed for each product within a single time period.

2.5.2.1 Single lot

The model proposed in Haase and Kimms (2000) (HK) is obtained by adding to the constraints below the requirements (2.49) and (2.50).

$$\sum_i p_{it} \cdot X_{it} + \sum_{s \in S_t} \widehat{st}_s \cdot W_s \leq cap_t \qquad \forall t, \tag{2.58}$$

$$\sum_{s \in S_t} W_s = 1 \qquad \forall t, \tag{2.59}$$

$$\sum_{s \in S_t} f_{is} \cdot W_s = \sum_{s \in S_{t-1}} l_{is} \cdot W_s \qquad \forall i, t, \tag{2.60}$$

$$X_{it} \leq b_{it} \cdot \sum_{s \in S_t} g_{is} \cdot W_s \qquad \forall i, t, \tag{2.61}$$

$$X, I \geq 0, \quad W \in \{0, 1\}. \tag{2.62}$$

The new capacity constraints are expressed in (2.58). The use of a single sequence in each period is ensured by (2.59), while (2.60) guarantee setup carry-over by linking the first and last products of consecutive time periods. The last set of constraints (2.61) only allows production for products in the sequence selected. Similarly to the original paper, the model HK is consider a single lot model. Nevertheless, it is important to note that by changing the definition of g_{is} to the number of times product i appears in sequence s, the model tackles multiple lots of the same product.

2.5.2.2 Multiple lots

The model GKAL proposed in Guimarães et al. (2013) determines a sequence selection by choosing among the several setup states of each product in each period. For this purpose two additional parameters are required:

e_{is} =1 if the machine is ever set up for product i in sequence s,

a_{is} the number of setups performed to product i in sequence s.

We also need to introduce binary decision variables U_{it} to capture if at least one setup is performed to product i in period t. The overall GKAL models contains (2.49), (2.50), (2.58) together with:

$$\sum_{s \in S_t} f_{is} \cdot W_s = Z_{it} \qquad \forall\, i, t, \tag{2.63}$$

$$\sum_{s \in S_t} l_{is} \cdot W_s = Z_{i,t+1} \qquad \forall\, i, t, \tag{2.64}$$

$$\sum_{i} Z_{it} = 1 \qquad \forall\, t, \tag{2.65}$$

$$\sum_{s \in S_t} e_{is} \cdot W_s = U_{it} \qquad \forall\, i, t, \tag{2.66}$$

$$\sum_{s \in S_t} a_{is} \cdot W_s = Y_{it} \qquad \forall\, i, t, \tag{2.67}$$

$$X_{it} \leq b_{it} \cdot (U_{it} + Z_{it}) \qquad \forall\, i, t, \tag{2.68}$$

$$X, I, W \geq 0, \quad Z, U \in \{0, 1\}, \quad Y_{it} \in \{0, \ldots, q_{it}\}. \tag{2.69}$$

The first two set of constraints (2.63) and (2.64) link the machine's initial configuration in each period with the first and last product in the selected sequence, and also establish the setup carry-over. Constraints (2.65) state that the machine is set up for exactly one product at the beginning of each time period. Product setup decisions are linked with sequence selection through constraints (2.66) and (2.67). Requirements (2.68) ensure for each period that a product is only produced in the case the machine is properly set-up. Such a configuration might have been carried over from the previous period or resulted from a setup in that period.

2.6. Computational tests results

In this section we present the results of our computational study to assess the performance of the reviewed formulations. We support our comparison by measuring both the quality of the upper bounds and running times obtained on an extensive set of instances. This instance set captures different characteristics of real world problems, such as the existence or not of non-triangular setups. The problem sets allow to test the performance of models under a variety of conditions, e.g. testing single lot formulations of non-triangular instances, or testing multiple lot formulations on triangular instances, contributing to the evaluation of the models flexibility.

Most of the formulations are straightforward implementations when using an optimization software package, however, this excludes the models having an exponential number of constraints or variables. Explicitly implementing these formulations would lead to intractable models, therefore constraints and variables are dynamically generated and added to the models as needed.

We apply a row generation algorithm to manage the number of subtour elimination constraints in formulations AL2, BW and MCAL. First we solve a partial model formulation without any of the subtour elimination constraints. Feasibility of the optimal solution is checked by searching for potential disconnected subtours. If no such a tour is identified the solution is feasible and also optimal, otherwise the corresponding violated subtour elimination constraints are generated and added to the model. We repeat the process of solving the model and generating additional constraints until no subtours appear in the optimal solution of the incumbent model.

To deal with the large number of variables (sequences) present in models KMT, HK and GKAL we have followed a column generation approach. The aim of the column generation algorithm is to identify a set of sequences to use in each time period. At each iteration the algorithm solves the model's linear relaxation (LP) restricted to a limited set of sequences and tries to price out new sequences to be included. The subproblems arising during the column generation process are defined in Appendix 2.B. An important issue is the basis initialization, as the initial set of sequences provided to the model may not include a feasible solution. Hence, we apply a two-phase approach in which the first phase aims to find a feasible LP solution to the problem, while the second phase seeks to find an optimal LP solution. Consider the additional artificial variables I_{i0} defining the initial stock on hand. During phase I of our column generation algorithm the model's objective function is changed to $\sum_{i,j} I_{i0}$ which is a measure of the infeasibility of the current solution. As soon as the sum of the artificial initial stock is zero a feasible LP has been found and the algorithm advances to phase II recovering the original objective function. When the column generation algorithm stops, the integrality constraints are restored and the model is solved as a MIP over the sequences encountered during column generation to find a feasible solution to the original problem.

Contrarily to the use of a commercial solver to solve polynomial sized formulations or the use of our row generation approach to treat exponential number of constraints, the technique used to solve models with an exponential number of variables does not guarantee optimality. However, since we are limiting the running time to one hour in all experiments it is also not guaranteed that the other methods can prove the solution optimality or even find a feasible solution.

Two versions of the models are tested, the original formulation presented in the body of the paper and the facility location reformulation (FL), originally proposed by Krarup and Bilde (1977) for the single-item problem. The reformulation redefines production variables as X_{itl} that determines the quantity of product i produced in period t (or micro-period s) to satisfy demand in period l, simultaneously capturing the production and stock held at each period. The objective is to test, under the different model types, the effects both on the solution quality and efficiency of a formulation which is known to give tight lower bounds. We omit the complete formulations here since each one of them is a straightforward extension of the original formulation.

In the following subsections results are divided according to the time structure of the models. In the first benchmark we explore the effect of the different formulations in mP models. We do not compare micro and macro-period models since their comparison from previous studies (Menezes et al. (2011)) has already been established and it can also be known from the results of the following subsection that mP models struggle even when the instances are of medium size. The second benchmark is fully dedicated to MP models. We start by assessing the models on a set of instances obeying the triangle inequality and later the formulations are tested on a non-triangular instance set obtained by modifying these instances and adding minimum lot size requirements.

All computations were performed on Intel @ 2.40 GHz processing units with 4 GB of random access memory using the Linux operating system. All formulations and algorithms were implemented in C++ using the ILOG Concert Technology and compiled with a gcc compiler. To solve mixed integer and linear programming models we used IBM ILOG CPLEX 12.4 with all runs having a limit of one hour or running out of memory. Furthermore, when solving the MIP problems we always tested all the possible MIP solution emphasis strategies available in CPLEX, namely: default (balance of feasibility and optimality), feasibility, optimality, best bound and hidden feasibility. In each benchmark the results of each model correspond to the best search strategy for that model considering all benchmark's instances.

2.6.1 Micro-period models

To compare the efficiency of the reviewed mP formulations we rely on the well known TV instances of Fleischmann (1994). The instances are relatively small sized with eight products and eight periods. Problems only differ in terms of the machine capacity and setup matrix. We present the results for instances with a capacity utilization (measured as $\sum_i d_{it}/cap_t$) of 97%, 76% and 64%, problems TV11, TV13 and TV14 and setup matrices S1, S2, S3 and S4. Setup costs of both S1 and S3 are uniformly distributed in the interval of $[0,600]$ and $[0,300]$, respectively. Matrix S2 is obtained by randomly selecting values from the set $\{0,100,200,\ldots,600\}$. Finally, entries of S4 mimic a situation that often occurs in pratice when setups can be grouped into major setups, changeovers between products of diferent families ($sc_{ij} = 500$), and minor setups, changeovers among products of the same family ($sc_{ij} = 100$). Note that only S4 obeys the triangle inequality and no setup times and miminum lot sizes are considered. A total of 12 different problems were solved by combining the capacity utilization with setup matrices.

The number of micro-periods in the original and reformulated versions of GLSP, GLSP^{NF} and CC was set to $N+2$ which corresponds to the maximum number of $N+2$ setups per time period. To conduct a fair comparison, the two KMT models use $S_t^{max} = 2$ and $Br^{max} = 5$, defining the same number of maximum setups per period.

The comparison of the several mP models is shown in Figure 2.5. On the horizontal axis we have the mean running time in seconds and on the vertical axis the mean deviation from the best known solution, which are actual optimal values available from Menezes et al. (2011). Hence, the closer from the bottom left corner of the chart the more effective/efficient the model is. The original formulations are depicted as squares and the FL reformulations as circles.

Let us first discuss the results obtained by GLSP. Both the original and reformulated versions often exceed the available memory during the tree search performed by CPLEX,

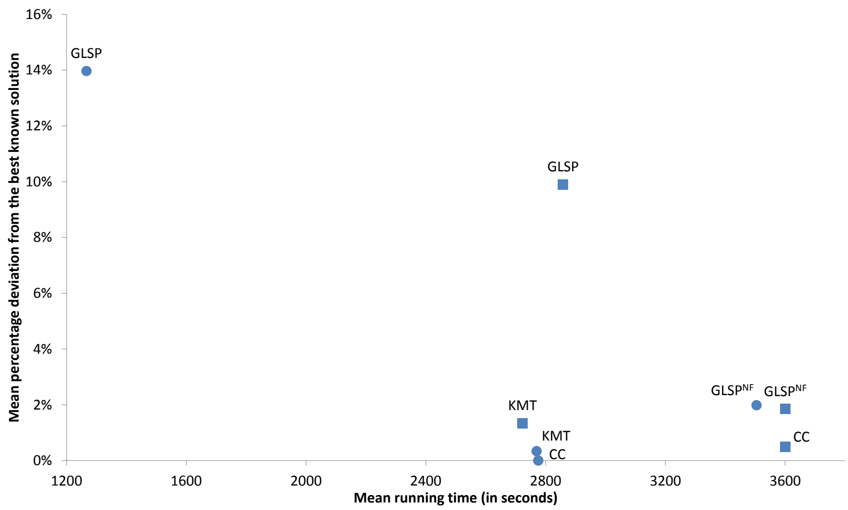


Figure 2.5: Performance comparison of mP formulations

which is reflected in the low running times. The memory limit is exceeded in 3 (out of 12) instances by the original version and in 10 instances by the reformulation. In line with this fact, the GLSP has the worst performance among the mP models in terms of final solution quality as the tree search is often prematurely stopped. Considering PO-mP original formulations, the CC model has the best performance and the network reformulation of the GLSP clearly improves its solution quality, besides improving memory consumption. Note that the original versions of GLSP^{NF} and CC consume all the available running time in the tree search, thus not being able to prove optimality in any of the instances. In terms of original formulations, the SO model KMT has the best performance among all delivering superior results in terms of solution quality and running times.

Reformulating the models using FL yields an improvement in the solution quality of most formulations. There is a clear negative influence in the tractability of the GLSP formulation, which more often (and sooner) reaches the maximum memory allowed for the run with impact in the quality of the solution obtained. Results of CC suffer a boost, which comes from the fact that this model can now prove solution optimality in 3 problems. The gains from the GLSP^{NF} are less significant. Finally, in spite of an improvement in the solution quality, the FL reformulation of KMT requires longer running times. Overall, the FL reformulation of CC presents the overall best results in terms of mP models. We also highlight the fact that less than %0.1 of KMT running time is spent in identifying the sequences (the actual column generation algorithm) to use in the MIP, which in turn often consumes the remaining of the available running time. A detailed view of all the results is presented in Table 2.2 in Appendix 2.C.

2.6.2 Macro-period models

2.6.2.1 Triangular instances

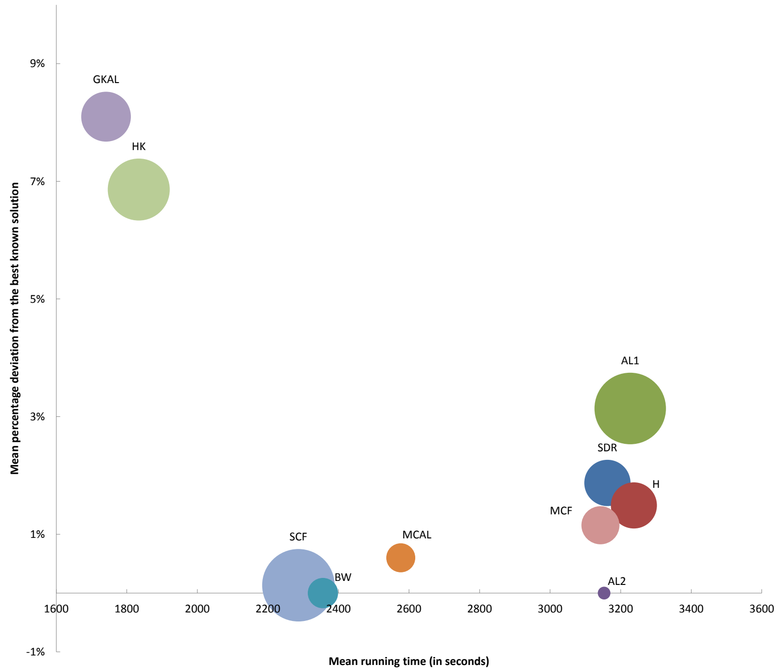
The MP models are first tested on a problem set obeying the triangle inequality and available from James and Almada-Lobo (2011) which adapts the approach of Almada-Lobo et al. (2007) to consider different values for capacity utilization over the planning horizon. All data parameters are generated from a uniform distribution. Product demand ranges between 40 and 59 units per period, holding costs between 2 and 9 cost units per period and setup times vary between 5 and 10 time units. Setup costs are made proportional to setup times by using a cost factor θ. The processing time is equal to all products and set to one time unit. To define machine capacity two parameters Cut and $CutVar$ are used. Cut establishes the target machine utilization over the entire planning horizon and $CutVar$ controls the maximum deviation from the target capacity utilization in each period. Moreover, it is ensured that the cumulative capacity utilization in any period does not exceed Cut in order to ensure problem feasibility.

Instances are classified into problem types according to the five-tuple (N, T, Cut, $CutVar$, θ). We use a total of 160 instances, 10 different instances for each one of the 16 problem types created by combining the following values for the parameters: $N \in \{15, 25\}$, $T \in \{10, 15\}$, $Cut \in \{0.6, 0.8\}$, $CutVar = 0.5$ and $\theta \in \{50, 100\}$. For further details on the instance generator the reader is referred to the cited works.

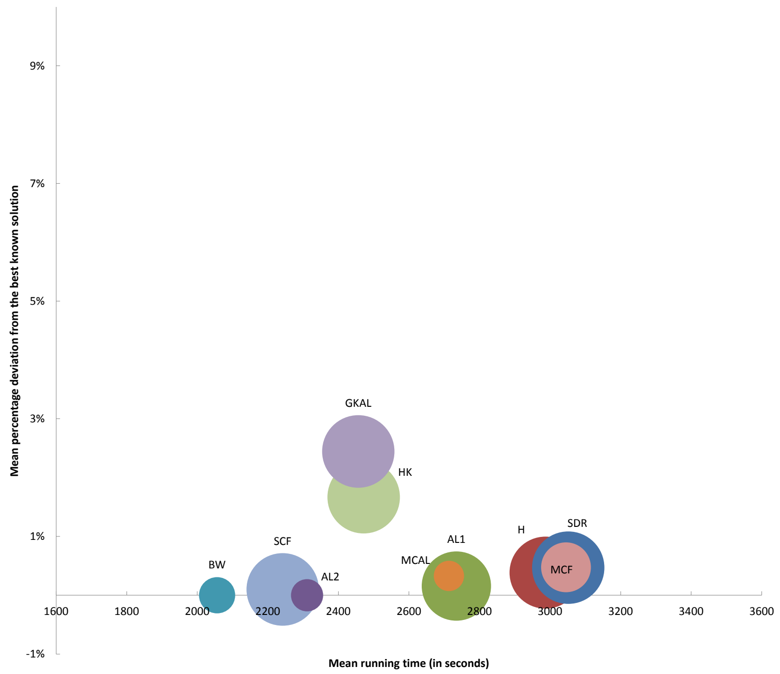
Figure 2.6 presents the comparison of several MP models on the triangular instance set separated into the original formulation and the reformulation. The size of each circle accounts for the number of instances in which the model is able to provide at least one feasible solution. Hence, the larger the circle the more problems are solved. The horizontal and vertical axis measure the mean running time in seconds and the mean deviation from the best known solution, respectively. The mean deviation only takes into consideration deviations in problems with feasible solutions. The best known solution is the best objective function among the solutions of all the models, including the reformulated versions, and is often the provably optimal solution to the problem (81 out of 160 instances).

Concerning the original formulations, it is clear that PO-MP-SL models have a poor performance in terms of the running time. We highlight three results: (1) SDR, H and AL1 exceed the memory limit in 39, 32 and 28 instances, respectively, explaining the mean running time below one hour, since their rarely prove optimality (3, 4 and 8 instances, respectively); (2) the extra flexibility of allowing alpha subtours introduced by model AL1 causes the model to have a higher mean deviation and longer running time, in comparison to SDR and H, but also allows the model to provide a feasible solution to barely all the problem instances; (3) model AL2 has a very poor performance in the number of problems solved, only 28 out of the 160 instances, and with similar running times.

PO-MP-ML models with an exponential number of constraints perform better than single lot models both in terms of average deviation and running time, especially BW which proves solution optimality in 67 cases with an average running time of 628 seconds. However, they are not as competitive as PO-MP-SL in providing a feasible solution to the problems.



(a) Original formulation



(b) FL reformulation

Figure 2.6: Performance comparison of MP formulations on the triangular instance set

Commodity flow based models present distinct behaviors; MCF performance is close to the single lot models, while SCF exhibits an excellent trade-off between the solution quality and efficiency, while providing a feasible solution to every instance in the set. The major difference between these two formulations relies on the size of the models created, especially as the number of products and periods increases, e.g. for a 25 product 15 time period instance the MCF model has a total of 255,401 variables and 256,906 constrains while the SCF model has a total of 21,401 variables and 13,546 constraints. As a result, CPLEX spends the total running time trying to solve the root node LP relaxation of MCF in 73 out of the 80 instances having 25 products. On the other hand, SCF requires much lower computational time and the LP bound is only slightly worse when compared to MCF. The remaining time is used for branching to prove solution optimality or to find a very good integer feasible solution.

In relation to SO based formulations, HK and GKAL, both present a good performance with respect to the number of problems with feasible solutions found and average running times, but fail in terms of the mean deviation. As expected, the sequences found during the solution of the root node LP relaxation of these models are not enough to achieve superior quality integer solutions and further branching would be required to improve the solution quality. However, tests show that there is still computational time available to carry on with the solution improvement if compared to SCF which also provides feasible solutions to a large number of problems. It is also important to note that HK performs better than GKAL. The reason behind this difference has to do with an easier MIP model resulting from a more restricted set of sequences in the model (no sequences with alpha subtours are allowed in HK).

Similarly to the results for the mP models, the FL reformulation improves the mean deviation to the best known solution in every model. In the single lot models it also helps to reduce running times and increase the number of problems in which these models provide a feasible solution. This effect is particularly evident in model AL2. In multiple lot models the effect is less pronounced, nevertheless mean running times of BW and MCF decrease, with the latter able to identify a feasible solution in a larger number of problems. Furthermore, CPLEX is able to solve the root node LP relaxation of the MCF reformulation on an higher number of instances, only 21 remain unsolved. Regarding SCF the reformulation appears to have no impact at all, as results are almost equal to the original version. Both column generation based models exhibit the same behavior when solved with the reformulation, an increase in the number of problems with a feasible solution (both provide a feasible solution to every problem) and in the mean running times, while the deviation is greatly reduced. The increase in the running times is explained by the larger model which has to be solved at each iteration of the column generation algorithm and also by the fact that the better LP bound forces a larger number of iterations. The time spent in solving the root note LP relaxation increases from 160 and 185 seconds in the original models to 915 and 520 seconds in the reformulation for HK and GKAL, respectively. This in turn allows the identification of better sequences to construct integer feasible solutions and thus reduces the mean deviation from the best known solution.

Overall, both versions of SCF exhibit the overall best trade-off providing superior quality feasible solutions to all the problems in reasonable running times. Moreover, for the two versions of SCF the final MIP gap is on average less than %0.1 emphasizing the quality of the solutions provided. We also draw attention to BW which finds the largest number of optimal solutions, a total of 80 in less than 520 seconds with the FL reformulation. The full results on this instance set is shown in Table 2.3 in Appendix 2.C.

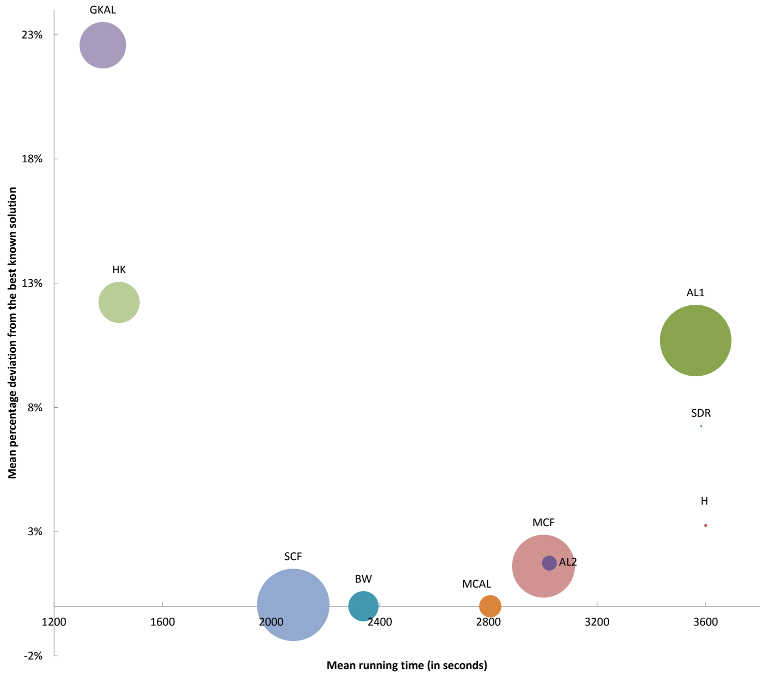
2.6.2.2 Non-triangular instances

To test the MP models on large instances disobeying the triangle inequality, we use the problem set designed by Guimarães et al. (2013) that is based on the set of triangular instances described in the previous section. To induce non-triangular setup matrices in the original problem set, the authors modify setup times of a subset of products called hereafter shortcut products. For each one of the k shortcut products new setup times st_{ik} and st_{ki} were randomly generated from a uniform distribution between 2 and 4, while setup costs remain proportional to setup times using the cost factor θ. A total of 2 and 3 shortcut products are introduced in instances with 15 and 25 products, respectively. The problem set is composed by the same 160 problems grouped in 16 problem types, but with the setup matrices modified. A minimum lot size of 25 units was also introduced to avoid having fictitious setups at optimal solutions.

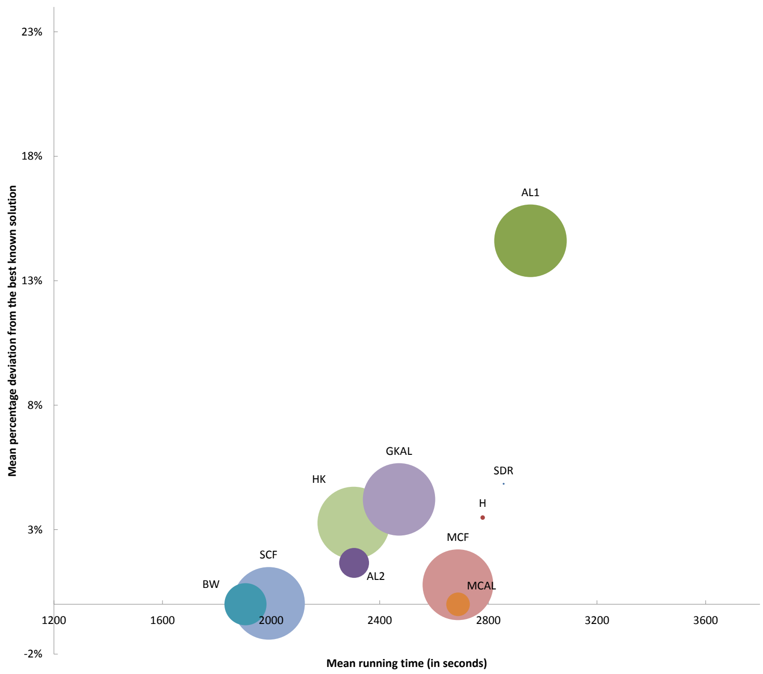
Figure 2.7 shows the comparison of several MP models for the non-triangular instance set in the same format as the comparison made for the triangular set. In this set a total of 98 solutions were proved to be optimal.

Models SDR and H and their respective reformulations fail to identify a feasible solution in more than 90% of the problems, and the tractability of the reformulated models is an important issue since the memory limit is reached in 79 cases for the SDR and 72 for the H model. In this problem set extra flexibility of AL1 and AL2 pays-off and the models are able to provide feasible solutions to a larger number of instances when compared to the other PO-MP-SL formulations. In particular AL1 which finds at least a feasible solution to practically all the problems, although the solution quality deteriorates quickly with the increase of the problem size. Note that the solutions found by AL2 are optimal if we consider a maximum of one production lot for each product per time period, however these solutions are on average 1.8% off from the best known solution when considering the possibility of several production lots.

Both multiple lot models with an exponential number of constraints reveal a similar performance as for triangular problems. Nonetheless, BW and MCAL find provably optimal solutions in higher mean running times, and BW still performs better than MCAL. The two commodity flow models have a high performance regarding the generation of feasible solutions to the problems. MCF performs much better on this set of instances with respect to the number of problems with a feasible solution and mean running time. Even so, it is important to add that the size of MCF models is still an issue as for 21 instances (hard instances with 25 products and $\theta = 100$) CPLEX is unable to solve the root node LP relaxation in less than one hour. Luckily, after solving the root node LP relaxation MCF is often able to provide a feasible solution in less than 5 nodes. Consistent with the results for the triangular set, SCF presents the best trade-off between mean deviation, mean running time and number of problems with a feasible solution. Moreover, its difference to BW in



(a) Original formulation



(b) FL reformulation

Figure 2.7: Performance comparison of MP formulations on the non-triangular instance set

terms of running times decreases in this set when considering the reformulated version, as for the original it is clearly better.

Mean deviation from the best known solution of SO-MP models considerably worsens, especially in the original formulation suggesting a lower quality of the relaxation in the presence of non-triangular setups and minimum lotsizes. Applying the FL reformulation to these models results in the same performance change as in the triangular set, increasing the running times and feasible solutions and decreasing the mean deviation. Again, GKAL shows a bigger improvement, but is still not enough to match HK's performance. Table 2.4 in Appendix 2.C details these results.

2.7. Conclusions

In this paper a two-dimensional framework is proposed to review and classify the different modeling approaches to incorporate sequencing decision in lotsizing and scheduling models. The framework uses the sequencing technique and time structure dimensions to divide the approaches into classes. The most relevant models in each class are reviewed to present their main features and differences, especially in the underlying assumptions. From this study emerged an important contribution which is a new polynomially sized formulation to the problem using commodity flow based subtour elimination constraints.

We perform extensive computational experiments to compare the performance of the different formulations with respect to the ability of providing quality solutions in limited running time, under different features of the problem. The benchmark sets solved present instances with the case of triangular setups and non-triangular setups. The results pointed the potential best formulation to use under each scenario. Our findings indicate that sequence oriented models, i.e. models having the sequences defined explicitly, appear an interesting tool for micro-period models. For macro-period models the new formulation proposed yields the best trade-off between solution efficiency, efficacy and feasibility in all problem settings. We also show that tightening the formulation using a reformulation of the production variables results in the improvement of the solution quality and in an increase in the number of problems for which models can find at least one feasible solution. Nevertheless, this effect is more evident in models with weaker original formulations. The study also suggested that models requiring a cutting plane generation algorithm can be an interesting solution to the problem if combined with an approach to generate valid integer feasible solutions during their search. Moreover, models explicitly defining the sequences have to be properly integrated with sophisticated column generation algorithms to be able to take advantage of their natural ability of providing feasible integer solutions.

Our insights also point out that the literature related to the Asymmetric Traveling Salesman Problem can be an important source of ideas to develop more efficient models and methods to this problem. Ongoing research has already used this relationship which originated some of the most relevant models in lotsizing and scheduling with sequencing decisions, however there is still a vast opportunity. Quite important is also the extension of these models to different real-world aspects as the use of parallel machines and the presence of multi-level production environments since the increase in the model size may have an important effect on the models performance. Finally, taking into account that production planning is often performed in practice on a rolling horizon basis, it is worthwhile investigating how to adapt these models to fit this reality.

Bibliography

B. Almada-Lobo, D. Klabjan, J. F. Oliveira, and M. A. Carravilla. Single machine multi-product capacitated lot sizing with sequence-dependent setups. *International Journal of Production Research*, 45(20):4873 – 4894, 2007.

B. Almada-Lobo, D. Klabjan, M. Carravilla, and J. Oliveira. Multiple machine continuous setup lotsizing with sequence-dependent setups. *Computational Optimization and Applications*, 47(3):529–552, 2010.

E. Balas. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636, 1989.

G. Belvaux and L. A. Wolsey. Modelling practical lot-sizing problems as mixed-integer programs. *Management Science*, 47(7):993–1007, 2001.

A. Clark, B. Almada-Lobo, and C. Almeder. Lot sizing and scheduling: industrial extensions and research opportunities. *International Journal of Production Research*, 49(9): 2457–2461, 2011.

A. R. Clark and S. J. Clark. Rolling-horizon lot-sizing when set-up times are sequence-dependent. *International Journal of Production Research*, 38(10):2287–2307, 2000.

A. Drexl and K. Haase. Proportional lotsizing and scheduling. *International Journal of Production Economics*, 40(1):73 – 87, 1995.

A. Drexl and A. Kimms. Lot sizing and scheduling - survey and extensions. *European Journal of Operational Research*, 99(2):221–235, 1997.

B. Fleischmann. The discrete lot-sizing and scheduling problem with sequence-dependent setup costs. *European Journal of Operational Research*, 75(2):395 – 404, 1994.

B. Fleischmann and H. Meyr. The general lotsizing and scheduling problem. *OR Spektrum*, 19(1):11–21, 1997.

L. Guimarães, D. Klabjan, and B. Almada-Lobo. Pricing, relaxing and fixing under lot sizing and scheduling. *European Journal of Operational Research*, (0):–, 2013. doi: 10.1016/j.ejor.2013.04.030. Available in Press.

K. Haase. Capacitated lot-sizing with sequence dependent setup costs. *OR Spectrum*, 18 (1):51–59, 1996.

K. Haase and A. Kimms. Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. *International Journal of Production Economics*, 66(2):159 – 169, 2000.

R. J. James and B. Almada-Lobo. Single and parallel machine capacitated lotsizing and scheduling: New iterative MIP-based neighborhood search heuristics. *Computers & Operations Research*, 38(12):1816 – 1825, 2011.

R. Jans and Z. Degraeve. Modeling industrial lot sizing problems: a review. *International Journal of Production Research*, 46(6):1619–1643, 2008.

S. Kang, K. Malik, and L. J. Thomas. Lotsizing and scheduling on parallel machines with sequence-dependent setup costs. *Management Science*, 45(2):273–289, 1999.

J. Krarup and O. Bilde. Plant location, set covering and economic lot size: An O(mn) algorithm for structured problems. In *Optimierung bei Graphentheoretischen und Ganzzahligen Probleme*, pages 155 – 180. BirkhauserVerlag, Berlin, 1977.

A. Menezes, A. Clark, and B. Almada-Lobo. Capacitated lot-sizing and scheduling with sequence-dependent, period-overlapping and non-triangular setups. *Journal of Scheduling*, 14(2):209–219, 2011.

H. Meyr. Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. *European Journal of Operational Research*, 120(2):311–326, 2000.

H. Meyr. Simultaneous lotsizing and scheduling on parallel machines. *European Journal of Operational Research*, 139(2):277 – 292, 2002.

T. Öncan, I. K. Altinel, and G. Laporte. A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3): 637 – 654, 2009.

D. Quadt and H. Kuhn. Capacitated lot-sizing with extensions: a review. *4OR: A Quarterly Journal of Operations Research*, 6(1):61–83, 2008.

S. Sarin, H. Sherali, and L. Yao. New formulation for the high multiplicity asymmetric traveling salesman problem with application to the Chesapeake problem. *Optimization Letters*, 5(2):259–272, 2011.

V. L. Smith-Daniels and L. P. Ritzman. A model for lot sizing and sequencing in process industries. *International Journal of Production Research*, 26(4):647–674, 1988.

L. A. Wolsey. MIP modelling of changeovers in production planning and scheduling problems. *European Journal of Operational Research*, 99(1):154 – 165, 1997.

X. Zhu and W. E. Wilhelm. Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE Transactions*, 38(11):987–1007, 2006.

Appendix 2.A Relationships among product oriented small bucket models

Proposition 1. *The CC formulation is stronger than the GLSP formulation.*

Proof. Let (X, I, T) be an optimal solution to the LP relaxation of CC. We define $Y_{jn} = \sum_i T_{ijn}$ for every j, n and show that (X, I, T, Y) is a feasible solution to the LP relaxation of (2.1) - (2.8) with the same objective value. Constraints (2.2), (2.3), (2.4) hold by definition.

Summing (2.13) over all i we obtain

$$\sum_{i,j} T_{ji,n-1} = \sum_{i,j} T_{ijn} \qquad \forall n.$$

This together with (2.11) implies

$$\sum_{i,j} T_{jin} = 1 \qquad \forall n \tag{2.70}$$

which is equivalent to $\sum_j Y_{jn} = 1$ and thus (2.5).

To show (2.7), observe that (2.13) implies

$$T_{iin} \le \sum_j T_{ji,n-1}.$$

Now we get

$$Y_{in} - Y_{i,n-1} = \sum_j T_{jin} - \sum_j T_{ji,n-1} \le \sum_j T_{jin} - T_{iin} = \sum_{j,j\neq i} T_{jin},$$

which immediately implies (2.7).

Finally, note that (2.6) is equivalent to

$$1 \ge Y_{i,n-1} + Y_{in} - T_{ijn} = \sum_k T_{ki,n-1} + \sum_t T_{tjn} - T_{ijn} = \tag{2.71}$$

$$\sum_s T_{isn} + \sum_t T_{tjn} - T_{ijn} = \sum_s T_{isn} + \sum_{t,t\neq i} T_{tjn} \tag{2.72}$$

where (2.71) follow by (2.13).

Constraint (2.70) implies

$$1 \ge \sum_{s,t} T_{tsn} = \sum_s \sum_{t,t\neq i} T_{tsn} + \sum_s T_{isn} \ge \sum_s T_{isn} + \sum_{t,t\neq i} T_{tjn},$$

which is identical to (2.72). This in turn shows (2.6). □

Appendix 2.B Subproblem formulation

The subproblem arising in each time period in models KMT, HK and GKAL resembles the prize collecting traveling salesman problem introduced by Balas (1989). Network $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ consists of node set $\mathcal{V} = \mathcal{N} \cup \{0, N+1\}$ and arc set $\mathcal{A}$. Node 0 is the source and node $N+1$ the sink while the remaining nodes represent products (see Figure 2.8). The source and the sink are used to identify the starting and ending products of the sequence, hence an arc connecting the source to a product means a carry over from the previous period and, similarly, an arc connecting a product to the sink represents a carry over to the next period. Travel costs c_{ij} are incurred for traversing arcs (i, j) and a prize ρ_i for including node i in the walk. The objective is to find the minimum cost walk through the network from the source to the sink.

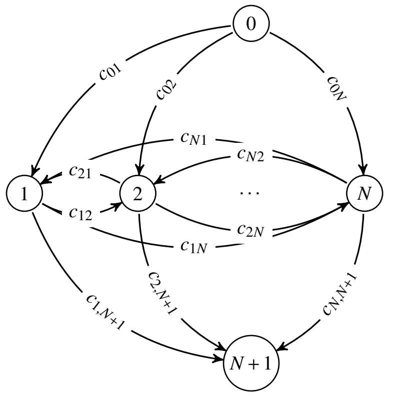


Figure 2.8: Network representation of the subproblem

To mathematically state the subproblem, we introduce integer decision variables χ_{ij} representing the number of times arc (i, j) is traversed. Furthermore, additional decision variables $y_i^{'}$ equal to 1 in case node i is part of the walk. The MIP model for the subproblem in time period t is as follows.

$$(sub_t) \qquad \min \sum_{i,j \in \mathcal{V}} c_{ij} \cdot \chi_{ij} - \sum_{i \in \mathcal{N}} \rho_i \cdot y_i^{'} \tag{2.73}$$

$$\text{s.t.} \qquad \sum_{j \in \mathcal{V}} \chi_{ji} = \sum_{j \in \mathcal{V}} \chi_{ij} \qquad \forall i \in \mathcal{N}, \tag{2.74}$$

$$\sum_{j \in \mathcal{N}} \chi_{0j} = 1, \tag{2.75}$$

$$\sum_{j \in \mathcal{N}} \chi_{j,N+1} = 1, \tag{2.76}$$

$$y_i^{'} \leq \sum_{j \in \mathcal{V}} \chi_{ji} \qquad \forall i \in \mathcal{N}, \tag{2.77}$$

$$y_i^{'} \geq \sum_{j \in \mathcal{V}} \chi_{ji} \qquad \forall i \in \mathcal{N}, \tag{2.78}$$

$$\chi_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}. \tag{2.79}$$

Here c_{ij} and ρ_i are derived from pricing equation. Here c_{ij} and ρ_i are derived from the pricing equation not listed in herein. Objective function (2.73) minimizes the cost of the traversed arc minus the prizes collected from the scheduled products (visited nodes). Constraints (2.74) balance in- and out-flow of each product. The source and sink nodes must be connected to a product, guaranteed by requirements (2.75) and (2.76), representing the first and last products in the sequence. The last two sets of constraints (2.77)-(2.78) represent the logical connections between node variables.

The model for the subproblems is, however, still incomplete, as a solution for (2.73)-(2.79) permits disconnected subtours. To eliminate such subtours we use single-commodity-flow type constraints. Consider decision variables f_{ij} as the commodity flow traversing arc (i, j), which is constrained to be less than or equal to the number of products. The following constraints are added to sub_t to prohibit disconnected subtours:

$$f_{ij} \le N \cdot \chi_{ij} \qquad \forall i \in \mathcal{V}, j \in \mathcal{N}, \tag{2.80}$$

$$\sum_{j \in \mathcal{N}} f_{0j} = \sum_{j \in \mathcal{N}} y'_j, \tag{2.81}$$

$$\sum_{j \in \mathcal{V}} f_{ji} = y'_i + \sum_{j \in \mathcal{N}} f_{ji} \quad \forall i \in \mathcal{N}. \tag{2.82}$$

Constraints (2.80) ensure that flows only traverse the arcs in the solution. Constraints (2.81)-(2.82) require that flow variables of the commodity describe a path from the source to every node in the sequence defined by arc variables. In detail, constraints (2.81) force a flow equal to the number of products in the sequence to leave the source and constraints (2.82) impose flow conservation for each node in the graph.

This model serves as the basis for formulating the subproblems arising in the column generation algorithm of KMT, HK and GKAL. In the case of HK the model correctly defines the sequences to be created (in fact, $\rho_i = 0$ for all i). However, both KMT and GKAL require some adjustments.

In the case of KMT the pricing forces the last setup to be explicitly known, hence we introduce variables χ'_{ij} which equal one if the changeover from product i to product j is the last in the sequence to be built (product j is the last in the sequence). Note that contrarily to χ_{ii} which is always zero, χ'_{ii} may equal one, which captures the case of setup preservation between consecutive split-sequences. If we let c'_{ij} be the cost of the last changeover, the model is adapted as follows:

$$(sub_t^{KMT}) \quad \min \quad \sum_{i,j \in \mathcal{V}} c_{ij} \cdot \chi_{ij} + \sum_{i,j \in \mathcal{N}} c'_{ij} \cdot \chi'_{ij} \tag{2.83}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{V}} \chi_{ji} + \sum_{j \in \mathcal{N}} \chi'_{ji} = \sum_{j \in \mathcal{V}} \chi_{ij} + \sum_{j \in \mathcal{N}} \chi'_{ij} \quad \forall i \in \mathcal{N}, \tag{2.84}$$

$$\sum_{j \in N} \chi_{ij} \le 1 \qquad \forall i \in \mathcal{N} \tag{2.85}$$

$$\sum_{j \in V} \chi_{ji} \leq 1 \qquad \forall i \in \mathcal{N} \tag{2.86}$$

$$\sum_{i,j \in \mathcal{N}} \chi_{ij} \leq Br^{max} - 1 \qquad \forall i, j \in \mathcal{N}, \tag{2.87}$$

$$\sum_{j \in \mathcal{N}} \chi'_{ji} \leq \chi_{i,N+1} \qquad \forall i \in \mathcal{N}, \tag{2.88}$$

$$\sum_{i,j \in \mathcal{N}} \chi'_{ij} = 1, \tag{2.89}$$

$$(2.75) - (2.78), (2.80) - (2.82),$$

$$\chi_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}, \tag{2.90}$$

$$\chi'_{ij} \in \{0,1\} \qquad \forall i, j \in \mathcal{N}. \tag{2.91}$$

The last changeover variables are introduced into objective function (2.83) and similarly to HK, no prizes appear in the pricing equations. Setup conservation constraints (2.84) now include the last changeover. The maximum number of Br^{max} in the sequence is guaranteed by allowing up to Br^{max} - 1 setups to take place through constraints (2.87). The correct linking between the last setup and final machine configuration is ensured by (2.88) and constraints (2.89) impose the last changeover to take place.

To address the subproblem associated with model GKAL, we have to accept solutions using the same arc more than once. Moreover, it is also required to capture if a node is visited, i.e. if it follows another node other than the source, as the prize ρ_i is only incurred in this case. We introduce integer decision variables y_i which equals 1 if node i is visited and 0 otherwise. The MIP model for the subproblem in time period t is as follows.

$$(sub_t^{GKAL}) \qquad \min \sum_{i,j \in \mathcal{V}} c_{ij} \cdot \chi_{ij} + \sum_{i \in \mathcal{N}} \rho_i \cdot y_i \tag{2.92}$$

$$\text{s.t.} \qquad \sum_{j \in \mathcal{N}} \chi_{ji} \geq y_i \qquad \forall i \in \mathcal{N}, \tag{2.93}$$

$$\sum_{j \in \mathcal{N}} \chi_{ji} \leq q_{it} \cdot y_i \qquad \forall i \in \mathcal{N}, \tag{2.94}$$

$$y'_i \leq y_i + \chi_{0i} \qquad \forall i \in \mathcal{N}, \tag{2.95}$$

$$2 \cdot y'_i \geq y_i + \chi_{0i} \qquad \forall i \in \mathcal{N}, \tag{2.96}$$

$$(2.75) - (2.76), (2.80) - (2.82),$$

$$\chi_{ij} \in \mathbb{N} \qquad \forall (i,j) \in \mathcal{A}, \tag{2.97}$$

$$y_i \in \{0,1\} \qquad \forall i \in \mathcal{N}. \tag{2.98}$$

Objective function (2.92) minimizes the cost of the traversed arc minus the prizes collected from the scheduled products (visited nodes). Constraints (2.93) and (2.94) enforce the logical relationship between the arcs traversed and nodes visited. The difference between y_i and y'_i relies on the fact that the latter equals to one also if the product is scheduled immediately after the source (first in the sequence, not representing an actual setup into it).

Constraints (2.95)-(2.96) represent the logical connections between node variables y and $y^{'}$.

Appendix 2.C Results Tables

Table 2.2: Summary of results for the mP models. The first row in each model corresponds to the original formulation and the second to the FL reformulation.

	Mean deviation (%)				Mean running time (secs)			
Problem	GLSP	GLSP^{NF}	CC	KMT	GLSP	GLSP^{NF}	CC	KMT
TV11-S1	19.1	2.4	0.0	0.0	3600.1	3600.4	3600.2	3600
	23.9	0.4	0.0	0.0	716.5	3600.3	3600.4	3600
TV11-S2	5.7	0.0	0.0	0.0	3600	3600.2	3600.2	66.3
	8.1	0.0	0.0	0.0	716	3600	487.7	226.9
TV11-S3	6.0	0.0	0.9	0.0	3600.1	3600.3	3600.3	3600
	26.7	4.7	0.0	0.0	563.4	3600.6	3600.4	3600.2
TV11-S4	33.7	0.5	0.0	5.2	696.2	3600.3	3600.2	3600
	11.4	6.6	0.0	0.0	3600.2	3600.2	3600.2	3600.1
TV13-S1	7.8	8.6	0.0	0.0	3600.1	3600.3	3600.2	3600.1
	12.1	1.4	0.0	0.0	1252.2	3600	3600.2	3600.1
TV13-S2	0.0	0.0	0.0	0.9	3600.1	3600.2	3600.3	31.9
	2.0	4.4	0.0	0.0	1636.2	3600.4	315	385.7
TV13-S3	9.8	1.0	0.0	0.0	3600.1	3600.3	3600.3	3600
	1.2	0.0	0.0	0.0	3600.1	3600.5	3600.4	3600.1
TV13-S4	10.9	0.0	0.0	9.5	673.2	3600.2	3600.2	3600.1
	24.0	1.6	0.0	4.0	480.1	3600.3	3600.4	3600.1
TV14-S1	3.2	1.9	0.0	0.5	3600.1	3600.3	3600.3	3600.1
	14.8	1.4	0.0	0.0	536.2	2450.3	3600.2	3600.1
TV14-S2	4.4	0.0	0.0	0.0	529.8	3600.3	3600.2	170.9
	5.3	0.9	0.0	0.0	918.6	3600.5	103.3	225.8
TV14-S3	4.0	4.3	2.4	0.0	3600.1	3600.4	3600.3	3600.1
	13.5	0.0	0.0	0.0	620	3600.4	3600.3	3600.1
TV14-S4	14.2	3.5	2.6	0.0	3600	3600.3	3600.2	3600.1
	24.8	2.4	0.0	0.0	558.4	3600.4	3600.1	3600.1
Average	9.9	1.8	0.5	1.3	2858.3	3600.3	3600.2	2722.5
	14.0	2.0	0.0	0.3	1266.5	3504.5	2775.7	2769.9

Table 2.3: Summary of results for MP models for the triangular instance set. The first row in each model corresponds to the original formulation and the second to the FL reformulation.

	Problem type	15-10-0.6-50	15-10-0.6-100	15-10-0.8-50	15-10-0.8-100	15-15-0.6-50	15-15-0.6-100	15-15-0.8-50	15-15-0.8-100	25-10-0.6-50	25-10-0.6-100	25-10-0.8-50	25-10-0.8-100	25-15-0.6-50	25-15-0.6-100	25-15-0.8-50	25-15-0.8-100	Average
Deviation [%]	SDR	0.0	2.9	0.3	3.5	0.2	4.8	0.4	5.4	0.3	2.5	0.3	3.6	0.1	inf	1.0	inf	1.9
		0.0	0.3	0.0	0.9	0.2	1.0	0.2	1.2	0.0	0.4	0.1	0.6	0.4	0.8	0.5	0.9	0.5
	H	0.1	1.5	0.0	2.8	0.4	4.5	0.4	4.3	0.2	2.2	0.4	2.9	0.4	3.6	0.9	inf	1.5
		0.0	0.1	0.0	0.7	0.0	0.7	0.3	1.6	0.1	0.4	0.1	0.6	0.1	0.5	0.3	0.6	0.4
	AL1	0.0	1.5	0.1	3.0	0.1	1.5	0.1	2.9	0.3	2.3	0.3	2.4	3.7	9.8	2.8	18.9	3.1
		0.0	0.1	0.0	0.4	0.0	0.3	0.0	0.4	0.0	0.2	0.0	0.3	0.1	0.2	0.1	0.5	0.2
	AL2	0.0	inf	0.0	inf	0.0	inf	inf	inf	0.0	inf	0.0	inf	inf	inf	inf	inf	0.0
		0.0	0.0	0.0	inf	0.0	inf	0.0	inf	0.0	0.0	0.0	inf	0.0	inf	0.0	inf	0.0
	BW	0.0	0.0	0.0	inf	0.0	inf	0.0	inf	0.0	inf	0.0	inf	0.0	inf	0.0	inf	0.0
		0.0	0.0	0.0	0.0	0.0	inf	0.0	inf	0.0	0.0	0.0	inf	0.0	inf	0.0	inf	0.0
	MCAL	0.0	0.0	0.0	inf	0.0	inf	0.0	inf	0.4	0.0	0.0	4.1	0.0	6.8	1.0	5.2	0.6
		0.0	0.2	0.0	1.4	0.0	1.3	0.1	2.9	0.0	0.0	0.1	1.0	0.8	inf	0.7	inf	0.3
	SCF	0.0	0.1	0.0	0.2	0.0	0.1	0.0	0.3	0.0	0.2	0.0	0.2	0.0	0.4	0.0	0.6	0.1
		0.0	0.0	0.0	0.2	0.0	0.2	0.0	0.3	0.0	0.3	0.0	0.2	0.0	0.2	0.0	0.3	0.1
	MCF	0.0	1.3	0.0	1.9	0.0	2.3	0.2	3.3	1.7	inf	0.0	inf	3.3	inf	inf	inf	1.2
		0.0	0.4	0.0	1.2	0.0	1.2	0.1	1.8	0.1	0.6	0.1	inf	0.1	inf	0.2	inf	0.5
	HK	3.9	4.9	4.9	5.6	4.5	5.3	5.6	5.9	7.7	7.4	9.0	9.6	8.7	10.2	10.7	11.1	6.9
		0.1	1.2	0.3	2.6	0.1	2.8	0.3	2.9	0.3	2.4	0.3	3.2	0.9	3.8	0.9	4.6	1.7
	GKAL	5.5	6.2	6.0	6.9	7.5	6.7	7.3	8.6	8.6	9.2	9.7	10.3	11.2	11.2	11.8	11.2	8.1
		0.7	2.6	0.8	3.4	0.6	3.1	1.1	4.3	1.1	3.3	1.3	4.7	1.1	4.6	1.4	5.1	2.4
Time [s]	SDR	3037.8	1976.2	3433.7	2744.7	3600.1	3114.7	3600.1	3369.6	3138.6	3154.0	2883.1	2927.8	3157.9	3568.9	3437.4	3454.1	3162.4
		887.5	3401.1	2531.9	3324.6	2839.4	3242.7	2758.7	3404.3	3441.2	3600.5	2941.4	3600.5	2876.0	3600.9	2772.4	3600.8	3051.5
	H	2790.5	2909.4	3587.5	2789.8	3600.0	3232.8	3600.1	3600.1	3495.2	3263.7	3352.3	3329.6	3600.2	2932.5	3131.4	2584.4	3237.5
		165.4	3071.4	2029.9	3485.6	2386.8	3408.5	2578.3	3170.6	3347.6	3529.3	3022.4	3600.5	3278.4	3600.7	3516.0	3600.6	2987.0
	AL1	2444.6	2724.3	3349.8	2517.6	3600.1	3291.3	3600.1	3444.1	3483.1	3439.1	3475.6	3417.4	3398.1	3087.8	3368.2	3005.2	3227.1
		117.5	2650.0	499.3	3600.2	1183.8	3600.3	2695.6	3600.2	2232.3	3423.1	2774.5	3600.5	3085.6	3600.9	3600.5	3487.6	2734.5
	AL2	507.0	3600.5	2079.7	3600.6	3392.2	3600.4	3600.3	3600.3	2089.9	3600.5	2775.1	3600.6	3600.5	3600.7	3600.5	3600.7	3153.1
		37.1	2582.6	248.5	3600.3	269.6	3600.3	1984.1	3600.5	718.3	3515.3	865.0	3600.5	2299.1	3600.6	2856.9	3600.5	2311.2
	BW	79.4	3580.6	142.8	3600.3	409.8	3600.5	1338.3	3600.4	621.2	3600.3	1112.8	3600.4	2545.1	3600.5	2661.0	3600.5	2355.9
		18.4	1966.4	89.0	3552.2	85.7	3600.4	738.3	3601.3	487.3	3454.9	525.0	3602.0	1749.1	3601.7	2225.1	3600.5	2056.1
	MCAL	108.1	2558.1	327.0	3600.1	810.4	3600.2	2366.7	3600.2	1799.8	3568.0	2150.1	3502.2	3098.1	3351.2	3387.7	3560.7	2576.6
		143.6	2365.4	402.1	3261.5	1719.7	3335.4	1982.8	3380.8	2657.6	3574.3	2993.9	3564.1	3503.0	3600.6	3325.7	3600.2	2713.2
	SCF	54.0	2708.6	135.4	3600.4	259.6	3600.6	1151.4	3600.6	742.7	3601.2	1188.2	3600.9	2165.1	3600.7	2972.3	3600.6	2286.4
		43.0	2489.5	140.6	3600.3	251.9	3600.6	1066.6	3600.7	731.5	3601.1	922.4	3600.9	2127.1	3600.5	2896.4	3600.6	2242.1
	MCF	441.2	3600.0	2111.6	3600.0	1297.1	3600.1	3497.4	3600.1	3385.8	3600.4	3546.5	3600.3	3600.4	3600.4	3600.5	3600.4	3142.6
		184.7	3600.0	2340.6	3600.0	820.8	3600.1	3327.7	3600.1	2996.1	3600.2	3208.8	3601.4	3437.9	3600.3	3602.1	3600.3	3045.1
	HK	66.0	3600.1	81.9	2823.1	1063.8	3600.2	1186.0	2888.8	380.8	3223.5	328.5	1758.3	1520.8	3600.3	1002.0	2217.7	1833.9
		88.7	2512.8	144.3	3209.1	426.0	3600.1	1162.2	3600.2	1706.3	3600.1	1544.8	3600.1	3540.4	3608.3	3593.9	3610.1	2471.7
	GKAL	160.6	2447.7	177.2	1456.9	2443.1	3243.6	1784.5	2531.8	833.4	3246.3	454.6	1569.4	2475.7	2935.5	1146.1	1091.9	1741.0
		81.9	1754.7	153.6	2816.6	897.4	3600.0	1823.7	3600.0	1775.5	3426.8	2057.1	3600.0	3288.3	3600.0	3227.2	3600.0	2456.4
Feasible [# of instances]	SDR	10	10	9	8	9	9	7	5	9	10	6	7	1	0	2	0	102
		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	160
	H	10	10	9	10	9	10	6	2	8	9	5	6	6	1	1	0	102
		10	10	10	10	10	10	10	10	10	10	10	10	9	10	10	10	159
	AL1	10	10	10	10	10	8	10	10	10	10	10	10	10	10	10	10	158
		10	10	10	10	10	10	10	10	10	9	10	10	9	10	9	6	153
	AL2	10	0	8	0	1	0	0	0	6	0	3	0	0	0	0	0	28
		10	4	10	0	10	0	8	0	9	1	10	0	4	0	5	0	71
	BW	10	1	10	0	10	0	9	0	9	0	9	0	5	0	4	0	67
		10	6	10	1	10	0	10	0	9	1	10	0	8	0	5	0	80
	MCAL	10	4	10	0	9	0	5	0	9	1	6	2	3	3	1	1	64
		10	6	10	4	7	3	8	2	6	1	5	1	1	0	3	0	67
	SCF	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	160
		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	160
	MCF	10	10	10	10	10	10	10	10	2	0	1	0	1	0	0	0	84
		10	10	10	10	10	10	10	10	10	1	10	0	6	0	3	0	110
	HK	10	10	9	9	10	10	8	8	10	10	8	6	10	10	4	5	137
		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	160
	GKAL	10	9	8	7	9	9	8	7	7	8	5	4	7	7	3	2	110
		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	160

Table 2.4: Summary of results for MP models for the non-triangular instance set. The first row in each model corresponds to the original formulation and the second to the FL reformulation.

Problem type		15-10-0.6-50	15-10-0.6-100	15-10-0.8-50	15-10-0.8-100	15-15-0.6-50	15-15-0.6-100	15-15-0.8-50	15-15-0.8-100	25-10-0.6-50	25-10-0.6-100	25-10-0.8-50	25-10-0.8-100	25-15-0.6-50	25-15-0.6-100	25-15-0.8-50	25-15-0.8-100	Average
Deviation [%]	SDR	2.0	3.3	inf	inf	19.1	4.6	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	7.2
		inf	4.2	1.4	9.5	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	4.8
	H	inf	2.8	inf	inf	inf	5.7	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	3.3
		inf	2.9	3.4	5.6	inf	2.6	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	3.5
	AL1	2.5	3.7	4.1	6.3	4.9	5.8	5.8	8.0	10.7	12.6	11.2	15.7	17.0	20.9	18.0	26.0	10.7
		3.3	5.2	4.1	6.1	9.4	7.7	5.3	8.4	13.2	15.9	15.2	25.0	17.8	35.9	18.6	42.5	14.6
	AL2	1.7	inf	1.8	inf	2.0	inf	inf	inf	1.8	inf	1.4	inf	inf	inf	inf	inf	1.7
		1.7	1.0	1.8	inf	2.0	inf	1.6	inf	1.8	inf	1.6	inf	1.5	inf	inf	inf	1.7
	BW	0.0	0.0	0.0	0.0	0.0	inf	0.0	inf	0.0	inf	0.0	inf	0.0	inf	0.0	inf	0.0
		0.0	0.0	0.0	0.0	0.0	inf	0.0	inf	0.0	0.0	0.0	0.0	0.0	inf	0.0	inf	0.0
	MCAL	0.0	0.0	0.0	inf	0.0	inf	0.0	inf	0.0	inf	0.0	inf	0.0	inf	inf	inf	0.0
		0.0	0.0	0.0	inf	0.0	inf	0.0	inf	0.0	inf	0.0	inf	0.0	inf	inf	inf	0.0
	SCF	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.1	0.0	0.2	0.0	0.1	0.0	0.1	0.0	0.1	0.1
		0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0
	MCF	0.0	0.5	0.0	0.8	0.0	1.8	0.1	1.8	0.1	1.8	0.3	1.9	2.0	18.4	3.4	17.3	1.6
		0.0	0.0	0.0	0.2	0.0	0.3	0.0	0.7	0.0	0.5	0.0	1.8	0.6	3.4	0.9	5.7	0.8
	HK	9.6	5.7	16.2	8.1	14.3	8.5	11.9	9.8	16.9	11.1	12.8	17.7	19.4	15.4	16.4	inf	12.2
		1.8	2.7	2.0	4.1	2.1	4.3	2.0	4.7	1.9	3.8	1.8	4.8	2.1	5.5	2.4	6.3	3.3
	GKAL	14.3	10.8	32.7	13.9	21.9	13.3	28.9	18.3	29.6	16.8	47.0	26.5	31.9	23.0	50.4	37.5	22.6
		1.3	3.7	2.4	5.6	1.2	4.3	1.8	6.0	1.8	4.7	2.6	6.8	3.2	7.7	4.0	10.3	4.2
Time [s]	SDR	3600.0	3600.0	3600.0	3600.0	3335.3	3600.0	3600.0	3600.0	3600.1	3600.0	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1	3583.3
		3600.0	2184.6	3600.1	1936.8	3428.4	2236.4	3113.0	1938.8	3359.8	2336.4	3237.9	2695.4	3321.8	2937.0	3052.0	2721.3	2856.2
	H	3600.0	3600.0	3600.0	3600.0	3600.0	3600.1	3600.0	3600.0	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1
		3600.1	2173.4	3600.1	2308.7	3600.1	2153.8	3395.1	1856.6	3082.2	2164.6	3299.5	2407.3	2627.9	2374.7	3145.1	2675.3	2779.0
	AL1	3600.1	3011.2	3600.0	3600.0	3600.1	3600.1	3600.0	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1	3600.2	3563.0
		3522.7	1758.5	3103.6	2922.4	2452.2	2324.6	3399.2	3387.7	3317.4	2867.4	3108.7	2821.8	3350.5	2970.7	3207.2	2767.2	2955.1
	AL2	187.9	3600.4	1281.8	3600.4	3304.2	3600.4	3600.3	3600.4	1372.9	3600.6	2695.5	3600.7	3600.8	3600.8	3600.5	3600.7	3024.4
		33.9	1969.7	169.4	3600.2	537.0	3600.3	2512.9	3600.6	489.1	3600.3	816.5	3600.7	1680.2	3600.8	3600.4	3600.8	2305.2
	BW	60.5	2902.2	456.5	3495.3	943.2	3600.4	2226.4	3600.5	248.9	3600.6	308.4	3600.5	1892.7	3600.8	3302.2	3600.6	2340.0
		36.9	1430.2	199.8	1851.8	611.1	3600.2	1720.7	3600.4	90.1	2844.3	165.9	3421.3	1530.2	3600.5	2162.5	3620.9	1905.4
	MCAL	281.5	3513.8	1753.4	3600.2	1441.3	3600.5	2591.1	3600.6	1737.3	3600.4	1739.0	3601.0	3045.2	3602.2	3600.5	3601.5	2806.8
		175.2	3514.8	1583.0	3600.4	1273.6	3600.5	2467.7	3601.1	1306.7	3600.2	1615.2	3600.6	2269.6	3602.6	3600.5	3602.4	2688.4
	SCF	40.8	2433.3	125.2	3199.7	286.2	3600.2	1254.0	3600.5	235.3	3600.5	268.3	3600.9	1525.2	3600.6	2327.8	3600.5	2081.2
		25.3	2074.1	103.8	2471.7	192.4	3600.3	1292.0	3600.8	148.9	3570.6	285.0	3600.6	1634.1	3600.3	2050.7	3600.3	1990.7
	MCF	268.9	3340.8	1232.6	3600.0	1241.7	3600.1	2655.5	3600.1	3303.5	3600.2	3600.3	3600.3	3600.4	3600.3	3600.8	3600.3	3002.9
		67.0	2586.0	648.9	3576.2	455.6	3600.1	2137.7	3600.1	1821.5	3600.2	3146.9	3600.3	3492.8	3601.7	3470.5	3601.0	2687.9
	HK	74.5	2946.6	102.3	1251.5	1383.1	3600.2	633.1	1128.3	991.0	2252.1	731.7	666.7	2210.0	3037.1	975.2	1053.3	1439.8
		79.5	1485.8	131.0	2014.3	530.5	3600.1	984.0	3600.1	1642.1	3600.1	1735.0	3600.0	3353.4	3607.8	3291.2	3607.7	2303.9
	GKAL	201.4	1052.8	115.8	797.4	2170.7	3600.0	1040.7	1123.0	967.4	2095.0	651.1	992.0	2860.7	2676.7	586.1	1145.6	1379.8
		115.8	753.6	266.7	1268.8	975.0	3600.0	2246.0	3600.0	2163.2	3600.0	2953.9	3571.0	3604.8	3602.7	3605.1	3606.7	2470.8
Feasible [# of instances]	SDR	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	4
		0	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	4
	H	0	5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	6
		0	6	1	2	0	1	0	0	0	0	0	0	0	0	0	0	10
	AL1	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	9	158
		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	160
	AL2	10	0	10	0	1	0	0	0	8	0	4	0	0	0	0	0	33
		10	6	10	0	9	0	5	0	9	0	10	0	7	0	0	0	66
	BW	10	4	10	2	9	0	5	0	10	0	10	0	6	0	1	0	67
		10	8	10	8	10	0	7	0	10	5	10	1	7	0	7	0	93
	MCAL	10	2	7	0	8	0	4	0	7	0	7	0	4	0	0	0	49
		10	1	7	0	8	0	4	0	9	0	7	0	6	0	0	0	52
	SCF	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	160
		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	160
	MCF	10	10	10	10	10	10	10	10	10	10	10	4	10	4	10	1	139
		10	10	10	10	10	10	10	10	10	10	10	9	10	10	10	7	156
	HK	9	9	4	6	10	10	4	3	9	9	1	1	7	8	1	0	91
		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	160
	GKAL	10	10	7	7	9	10	4	3	9	9	5	2	8	7	1	2	103
		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	160

CHAPTER 3

An innovative solution approach for lot sizing and scheduling problems with sequence dependent setups

Pricing, relaxing and fixing under lot sizing and scheduling

Luis Guimarães* · **Diego Klabjan†** · **Bernardo Almada-Lobo***

Accepted for publication in *European Journal of Operations Research*, 2013 (Minor Revision)

Abstract We present a novel mathematical model and a mathematical programming based approach to deliver superior quality solutions for the single machine capacitated lot sizing and scheduling problem with sequence-dependent setup times and costs. The formulation explores the idea of scheduling products based on the selection of known production sequences. The model is the basis of a matheuristic, which embeds pricing principles within construction and improvement MIP-based heuristics. A partial exploration of distinct neighborhood structures avoids local entrapment and is conducted on a rule-based neighbor selection principle. We compare the performance of this approach to other heuristics proposed in the literature. The computational study carried out on different sets of benchmark instances shows the ability of the matheuristic to cope with several model extensions while maintaining a very effective search. Although the techniques described were developed in the context of the problem studied, the method is applicable to other lot sizing problems or even to problems outside this domain.

Keywords Lot sizing and scheduling · Sequence-dependent setups · Non-triangular setups · Column generation · MIP-based heuristics

*INESC TEC, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal

†Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA

3.1. Introduction

In many production environments, production planning problems involve the determination of production lot sizes and sequence of different products on a single capacitated machine. Production lot sizes are driven by deterministic demand over the planning horizon. Switching between production runs of two different products triggers operations, such as machine adjustments and cleaning procedures, which consume scarce production time and can cause costs due, for example, to losses in materials. Under these conditions, production sequencing must explicitly take into account for these sequence–dependent setup times and costs. In this context, the need for simultaneous lot sizing and scheduling decisions arises. Production plans are created with the objective of minimizing the overall costs consisting mainly of holding and setup costs, while satisfying the available capacity in each time period from which the expenditure in setup times is deducted. Examples of industries where these decisions must be taken concurrently are chemicals, drugs and pharmaceuticals, pulp and paper, textiles, foundries, glass container, and food and beverage, among many others (see Clark et al. (2011)).

Tackling real world problems requires to address special cases that may occur by introducing additional features into mathematical models. Among these realistic features are changeovers that do not respect the triangle inequality. When setups obey the triangle inequality with respect to both the setup time and costs, i.e. it is more efficient to change directly between two products than via a third product, at most one setup for each product per time period occurs. In some industries, contamination occurs when changing from one product to another implying additional cleaning operations. If a 'cleaning' or shortcut product can absorb contamination while being produced, replacing the cleaning operations, non-triangular setups appear. In their presence, models have to allow for more than one production run of each product per time period as it potentially reduces setup times and costs. Many examples of this type are known in the chemical, pharmaceutical, food and dyeing industries.

Mixed integer programming (MIP) models are unable to solve relevant size instances of the problem, suffering from its computational intractability (they are NP-hard by Bitran and Yanasse (1992)). State-of-the-art optimization engines either fail to generate feasible solutions to this problem or take a prohibitively large amount of computational time, as the computational experiments presented herein attest. Therefore, solving this class of problems requires the use of efficient solution approaches. Mathematical programming-based heuristics (Ball (2011)), also known as matheuristics (Maniezzo et al. (2010)), are algorithms which integrate exact and heuristic search techniques. Exact algorithms probably achieve optimal or quasi-optimal solutions, yet the size of tractable problems is limited. On the other hand, metaheuristics (heuristic search) are tailored to solve large-scale combinatorial optimization problems exploring large size neighborhoods efficiently. The underlaying idea of matheuritics is to seek the best trade-off between the efficacy of exact approaches and the efficiency of metaheuristics. Furthermore, in general, these algorithms are flexible enough to cope with different model extensions and new features.

The motivation for this work is the development of a flexible solution methodology integrating exact and approximate methods able to solve lot sizing and scheduling problems of relevant sizes and features present in real world applications. We introduce a new MIP model for the single machine capacitated lot sizing and scheduling problem (CLSD) that accommodates non-triangular settings. The model schedules production based on the selection of feasible production sequences. We develop a pricing heuristic (*SeqSearch*) to generate the sequences to be incorporated in the model since including all possible sequences is intractable as its number grows exponentially with the number of products. To obtain superior quality solutions to the CLSD, we develop a construction and improvement heuristics combining *SeqSearch* with mathematical programming-based heuristics. The construction heuristic (*Relax-Price-Fix*) uses a rolling horizon approach to sequentially construct a solution to the problem, while the improvement heuristic (*Fix-Price-Optimize*) attempts to partially optimize a feasible solution by solving small MIP subproblems. Different neighborhood structures are explored during the local search to avoid local entrapment. The two driving principles of neighborhood structures definition are to consider subproblems having a small number of consecutive time periods with all products or a small set of products over a larger portion of the planning horizon.

Our contributions are as follows. To the best of our knowledge, the new MIP model is the first to capture non-triangular settings based on the selection of a single sequence from a pre-determined set in each time period. This is a non-trivial extension since products can repeat. An important ingredient of our solution methodology is a formulation that combines a compact and an extended formulation within a single model. This formulation trades-off accuracy and computational complexity. On the algorithmic front, we create a new MIP-based construction heuristic using this hybrid formulation. Another very important contribution concerns our novel ideas to use column generation for local search within lot sizing problems. The methodology exposed in this paper can be generalized to different lot sizing problems or even to problems outside this research field.

The remainder of this paper has the following structure. In Section 3.2 we overview the most relevant literature in the context of this work. Section 3.3 presents the new formulation for the CLSD. Section 3.4 describes our solution approach to solve the CLSD and its main building blocks. A series of computational experiments with different problem sets having distinct features are shown in Section 3.5. Finally, Section 3.6 is devoted to final remarks, conclusions from this work and some future research directions are pinpointed.

3.2. Literature review

The field of lot sizing and scheduling has received an increased attention from the research community due to its inherit applicability to real world problems as shown in the reviews by Drexl and Kimms (1997), Zhu and Wilhelm (2006), Jans and Degraeve (2008) and, recently, by the special issue Clark et al. (2011). This applicability can only be achieved with adequate solution approaches, most of which are based on mathematical representations of the problem. Mathematical formulations for lot sizing and scheduling assume a planning horizon divided into a finite number of time buckets. These discrete time formulations can be grouped into two types: large and small bucket models.

Large bucket models allow for more than one setup per time period. Sequencing decisions within each time period use decision variables similar to those of routing problems formulations and require sub-tour elimination constraints to correctly represent production sequences. Almada-Lobo et al. (2007) present an exact formulation for the CLSD when setups obey the triangle inequality, which was extended by Menezes et al. (2011) to the non-triangular case using an exponential number of constraints. Sarin et al. (2011) present a formulation with a polynomial number of sub-tours elimination constraints through multi-commodity-flow-type constraints. All these works deal with compact formulations, while we develop an extended formulation.

On the other hand, in small bucket models the production sequence comes for free directly from the assumption of allowing at most one setup per period. These models do not impose any restriction on the setup configuration and neither require sub-tour elimination constraints. The general lot sizing and scheduling problem (GLSP) model described by Fleischmann and Meyr (1997) and Meyr (2000) is the most flexible of such models. In the GLSP, time periods are divided into micro-periods using an *a priori* defined parameter. The number of micro-periods may account for the maximum number of setup operations allowed in each period, or divide each time period (e.g. weeks) into many shorter periods (e.g. days, hours or shifts). Hence, the model size is dramatically increased and/or multiple optimization runs with different parameter choices must the conducted to achieve optimality. Furthermore, Wolsey (2002) shows that the linear relaxation of small bucket models results in much weaker lower bounds in comparison to large bucket models.

The aforementioned models can be called compact or product related formulations, as sequencing decisions are taken from decision variables indexed by product. An alternative model may select the production sequence from a set of available production sequences, which are acceptable in each time period. We call these models extended or sequence related formulations. Examples are given in Haase and Kimms (2000) and Kovács et al. (2009) for big bucket formulations, and Kang et al. (1999) for a small bucket model. Sequence related formulations usually result in simpler models as sub-tour elimination constraints and auxiliary decision variables used to ordinate products are not required. However, as the number of products increases, the number of sequences grows exponentially. The mathematical formulation presented in this paper is, to the best of our knowledge, the first sequence related formulation considering a large bucket model for non-triangular setups.

Most solution procedures for the CLSD combine heuristics with exact methods. In Meyr (2000) the small bucket mathematical model is solved by embedding a dual network flow algorithm into threshold accepting and simulated annealing. These procedures were later extended for the case of parallel machines in Meyr (2002).

With the main purpose of solving specific instances Kang et al. (1999) present a branch-and-price algorithm for a small bucket sequence related formulation of the CLSD. It consists in dividing the entire production schedule into smaller production sequences, which the authors call split-sequences. For each period t the production sequence is composed of L_t split-sequences, resembling subperiods in product related formulations. To address the large number of split-sequences arising they propose a column generation based heuristic, where in each iteration the new split-sequences are obtained by an enumeration algorithm with an additional parameter $maxBR$, the maximum number of products in the split-sequence. To generate upper bounds, two different algorithms apply mathematical programming methods heuristically: one truncates the branch-and-bound search with respect to the number of fractional variables, while the other iteratively executes local search to improve the incumbent solution. A major disadvantage of this methodology is that to solve a given problem multiple runs are needed with different values of L_t and $maxBR$.

Progressive interval heuristics (Federgruen et al. (2007)) are MIP-based heuristics which solve a series of partially relaxed MIP subproblems to construct an initial feasible solution to the original MIP. Relax-and-fix heuristic starts from the first period in the planning horizon and progressively moves forward fixing the setup variables at their optimal value obtained in previous iterations. It is applied in Ferreira et al. (2009) to a practical case in the beverage industry on a small bucket model. Similarly to relax-and-fix, the 'exchange' improvement heuristic of Pochet and Wolsey (2006) and the fix-and-optimize version of Sahling et al. (2009) decompose the set of integer variables in the original MIP to create MIP subproblems to re-optimize. At each iteration, integer setup variables are fixed to their previous best value, apart from a small subset in which they are required to take any integer value, defining the subproblem to be optimized. Based on a large bucket model for the CLSD with triangular setups, James and Almada-Lobo (2011) integrated fix-and-optimize in a stochastic local search algorithm to improve the initial solution obtained with the relax-and-fix heuristic, delivering solutions within a small deviation from theoretical lower bounds.

Our solution approach, based on a large bucket sequence related model, integrates column generation in relax-and-fix and fix-and-optimize schemes. The pricing heuristic developed, which deals with the exponential number of production sequences, replaces the application of a branch-and-price algorithm (Barnhart et al. (1996)) as the solution of the linear relaxation of our model is likely to be fractional. Moreover, as in Muter et al. (2010) and Alvelos et al. (2010), the purpose of column generation is to provide good partial solutions that can later be combined by solving the master model or via a metaheuristic. In our case, integer solutions are found by applying relax-and-fix to the master model and they are improved by using fix-and-optimize, which is a clear difference to the work of Kang et al. (1999) where column generation is embedded in the branch-and-bound tree to generate new integer solutions. Following the classification of Blum et al. (2011) our approach falls into the categories of the hybridization of metaheuristics with tree search techniques and problem relaxation.

3.3. New models for the CLSD with non–triangular setups

3.3.1 A sequence related model

In this section we introduce a new sequence related model for the CLSD with non-triangular setups. This model constitutes the basis of our heuristic procedures. Throughout the exposition, let us consider set $\mathcal{N}$ composed of N products indexed by $i, j = 1, \ldots, N$ to be produced on a single capacitated machine over a finite planning horizon of T periods, defining a set $\mathcal{T}$ indexed by $t, l = 1, \ldots, T$. The following data is associated with this problem:

d_{it}	demand of product i in period t (units),
h_i	holding cost of one stock unit of product i (cost/unit),
cap_t	capacity of the machine in period t (time),
p_i	processing time of product i (time/unit),
b_{it}	upper bound on the production quantity of product i in period t (units),
sc_{ij}	cost incurred to set up the machine from product i to product j (cost),
st_{ij}	time needed to set up the machine from product i to product j (time).

The mathematical model stated next is a big bucket sequence related model. The setup state is carried over among adjacent periods, i.e. the setup state is preserved even over idle time. Moreover, setup crossovers are not allowed, which force setup operations to be performed within the time period, without spanning to the following period. The validity of this assumption relies on the fact that we are dealing with a big-bucket model. As several products can be produced per period (e.g. week), interesting production plans should not be excluded. Stockouts are not accepted, which is a common setting in deterministic demand environments, and no initial inventory is considered. However, such extensions are relatively straightforward. Finally, more than one setup may be performed to each product within a time period to address instances where setups do not obey the triangle inequality.

The first set of decision variables captures lot sizing decisions. To this end, let variables X_{itl} define the quantity of product i produced in period t to satisfy demand in period l. A model using such variables is usually referred to as a facility location model (FLM), originally proposed by Krarup and Bilde (1977) for the single-item problem. The FLM is known to be strong for lot sizing problems, giving tight lower bounds. To determine scheduling decisions, let $\mathcal{S}_t$ denote the set of all S_t feasible production sequences to schedule products on the machine in period t, indexed by $s = 1, \ldots, S_t$. Associated with each sequence we define the following parameters:

$\widehat{sc}_s$	setup cost incurred if sequence s is selected,
$\widehat{st}_s$	setup time incurred if sequence s is selected,
f_{is}	(=1) if product i is first in sequence s,
l_{is}	(=1) if product i is last in sequence s,
e_{is}	(=1) if the machine is ever set up for product i in sequence s,
a_{is}	number of setups performed to product i in sequence s.

Product sequencing can be modeled by the following decision variables:

W_{ts}	(=1) if sequence s is chosen in period t,
U_{it}	(=1) if at least one setup is performed to product i in period t,
Y_{it}	number of setups performed to product i in period t,
Z_{it}	(=1) if the machine is set up for product i at the beginning of period t.

Our sequence-related MIP model for the CLSD reads:

$$(FS) \quad \text{Min} \sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{S}_t} \widehat{sc}_s \cdot W_{ts} + \sum_{t\in\mathcal{T}} \sum_{\substack{l\in\mathcal{T} \\ l>t}} \sum_{i\in\mathcal{N}} (l-t)\cdot h_i \cdot X_{itl} \tag{3.1}$$

$$\text{s.t.} \sum_{\substack{t\in\mathcal{T} \\ t\leq l}} X_{itl} = d_{il} \qquad \forall i\in\mathcal{N}, l\in\mathcal{T} \tag{3.2}$$

$$\sum_{i\in\mathcal{N}} \sum_{\substack{l\in\mathcal{T} \\ l\geq t}} p_i \cdot X_{itl} + \sum_{s\in\mathcal{S}_t} \widehat{st}_s \cdot W_{ts} \leq cap_t \qquad \forall t\in\mathcal{T} \qquad (\lambda_t) \tag{3.3}$$

$$X_{itl} - d_{il}\cdot(U_{it}+Z_{it}) \leq 0 \qquad \forall i\in\mathcal{N}, t,l\in\mathcal{T}, l\geq t \tag{3.4}$$

$$\sum_{s\in\mathcal{S}_t} f_{is}\cdot W_{ts} = Z_{it} \qquad \forall i\in\mathcal{N}, t\in\mathcal{T} \qquad (\theta^f_{it}) \tag{3.5}$$

$$\sum_{s\in\mathcal{S}_t} l_{is}\cdot W_{ts} = Z_{i,t+1} \qquad \forall i\in\mathcal{N}, t\in\mathcal{T} \qquad (\theta^l_{it}) \tag{3.6}$$

$$\sum_{i\in\mathcal{N}} Z_{it} = 1 \qquad \forall t\in\mathcal{T} \tag{3.7}$$

$$\sum_{s\in\mathcal{S}_t} e_{is}\cdot W_{ts} = U_{it} \qquad \forall i\in\mathcal{N}, t\in\mathcal{T} \qquad (\alpha_{it}) \tag{3.8}$$

$$\sum_{s\in\mathcal{S}_t} a_{is}\cdot W_{ts} = Y_{it} \qquad \forall i\in\mathcal{N}, t\in\mathcal{T} \qquad (\pi_{it}) \tag{3.9}$$

$$(X_{itl}, W_{ts}) \geq 0, (U_{it}, Z_{it}) \in \{0,1\}, Y_{it}\in\mathbb{N} \quad \forall i\in\mathcal{N}, t,l\in\mathcal{T}, s\in\mathcal{S}_t \tag{3.10}$$

The objective function (3.1) minimizes the total holding and setup costs. Demand fulfillment is expressed in constraints (3.2). Constraints (3.3) guarantee that the total production and setup times in each period do not exceed available capacity. Requirements (3.4) ensure for each period that a product is only produced in case the machine is properly configurated. Such a configuration might have been carried over from the previous period or resulted from a setup in that period. Constraints (3.5) and (3.6) link the machine initial configuration in each period with the first and last product in the selected sequence, implying that if a given product is the first of the sequence in the current period, then it has to be the last in previous period (setup carry-over). Constraints (3.7) state that the machine is set up for exactly one product at the beginning of each time period. Product setup decisions are linked with sequence selection through constraints (3.8) and (3.9). Variable domains are defined in (3.10). The model extension to capture minimum and maximum lot sizes is shown in Appendix 3.A.

Remark 1. *Any model capable of tackling non-triangular setups can also address triangular setups. In the presence of setups that obey the triangle inequality, decision variables Y_{it} and constraints (3.9) are not required. In fact, when the triangle inequality holds, any optimal solution for the CLSD contains at most one setup for each product in each time period, turning Y_{it} redundant in the presence of U_{it}. Nevertheless, the consideration of maximum lot sizes may result into more than one production run in an optimal solution,* *requiring once again variables Y_{it} and constraints* (3.9)*.*

Remark 2. *Integrality of variables W_{st} is relaxed as the integrality of variables U_{it}, Y_{it}, Z_{it}, constraints* (3.5)-(3.9) *and the minimization of the setup cost imply the selection of a single sequence in the pool.*

3.3.2 A mixed product and sequence related model

The sequence related model FS just presented has an exponential number of variables W_{st} making a full implementation impracticable. Therefore, to achieve an efficient model implementation sequence assignment variables W_{st} have to be dynamically generated. The utilization of a column generation algorithm to provide the required variables introduces an additional computational effort which can compromise the efficiency of model FS. We introduce a hybrid formulation combining the sequence related formulation and a product related formulation, listed in Appendix 3.B, to relief the effort spent in generating new columns.

Consider a partition of the set of planning periods $\mathcal{T}$ into two disjoint subsets $\mathcal{T}_s$ and $\mathcal{T}_p$. Model FS is applied to subset $\mathcal{T}_s$ and sequencing decisions are obtained through variables Y_{it}, U_{it}, Z_{it} and W_{st}. In the remaining portion of the planning horizon, subset $\mathcal{T}_p$, the product related model of Appendix 3.B determines production sequences using variables T_{ijt}, G_{it} and Z_{it}. Note that Z_{it} ensure the proper linking between the two formulations as they appear in both. We omit the hybrid formulation since it is a straightforward combination of the two formulations.

Despite this effort to manage computational intractability, large-scale mathematical models arising in real-world problems still require additional measures. We have developed a solution approach to the CLSD with non-triangular setups based on the two described models. The next section describes the proposed heuristic in which mathematical programming techniques are combined with metaheuristics, aiming to achieve a flexible method able to tackle different features of this problem, while delivering superior quality solutions.

3.4. Solution approach

This section is devoted to our solution approach to solve the CLSD, which we call *Price-and-MIP* (*P&MIP*). The method is composed of three main building blocks, as depicted in Figure 3.1.

SeqSearch A pricing heuristic which deals with the large number of variables W_{st} present in our model formulation. It identifies a subset of production sequences to be kept in the model at each step of the solution approach.

Relax–Price–Fix An MIP-based construction heuristic to build an initial feasible integer solution to the CLSD. It essentially results from combining the relax–and–fix framework with *SeqSearch*.

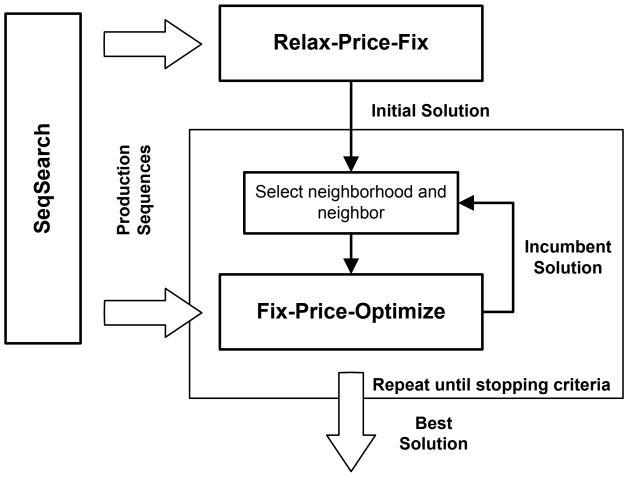


Figure 3.1: *P&MIP* Flowchart

Fix–Price–Optimize An improvement heuristic which attempts to improve a feasible solution by decomposing the original MIP problem into smaller subproblems to be solved. It also combines mathematical programming and *SeqSearch*.

SeqSearch is embedded into the construction and improvement heuristics. It is responsible for generating, updating and managing the pool of sequences preserved in FS. The overview of the various stages of the approach is given in Figure 3.2. A feasible initial solution to the CLSD is obtained through *Relax-Price-Fix* by progressively fixing integer variables in model FS in a rolling horizon fashion. The construction heuristic is described in more detail in Section 3.4.2. To improve the incumbent feasible integer solution we use *Fix-Price-Optimize* (see Section 3.4.3), which re-optimizes parts of a feasible solution. As shown in Figure 3.2 we rely on a systematic exploration of different neighborhoods to escape from local entrapment when applying the improvement heuristic.

In the following subsections we detail the main features of these building blocks.

3.4.1 *SeqSearch*: Pricing production sequences

The purpose of *SeqSearch* is to identify the set of production sequences (related to variables W_{st}) to include in model FS and iteratively finding an integer solution. In Figure 3.3 the outline of the heuristic is presented. The overall procedure is composed of two nested loops, an inner and an outer loop. In the inner loop a column generation algorithm manages and updates sets of period production sequence. The outer loop guides the search of production sequences towards integer solutions. It corresponds to an LP-driven diving heuristic (see Pochet and Wolsey (2006)), which is in fact a way to perform a depth-first search strategy in the branch-and-bound tree. Iteratively, the information from the incumbent LP solution is used to fix integer variables to an integer value, until all variables are fixed (or the problem becomes infeasible).

The procedure starts with the definition of model FS using an initial subset $\mathcal{S}'_t$ of feasi-

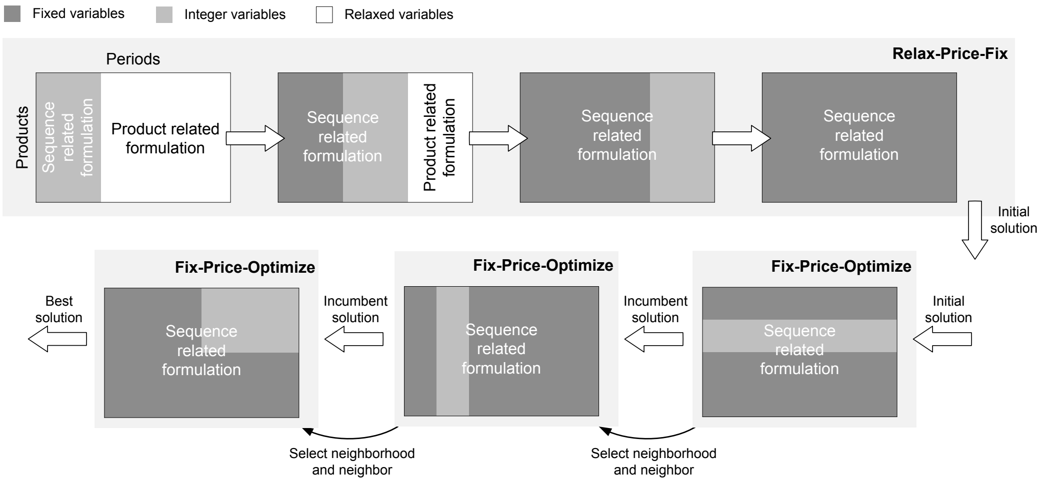


Figure 3.2: $P\&MIP$ overview

ble production schedules in each period $\mathcal{S}_t^{'} \subset \mathcal{S}_t$ (restricted problem - FS_r). In each iteration of the inner loop the linear relaxation of FS_r is solved and the dual information obtained is used to update and manage the sequence pool in each period. The pool is trimmed if the maximum number of sequences is exceeded. Proving optimality of the relaxed FS_r can imply a strong computational effort due to the tailing-off effect presented by the column generation method and the difficulty of the subproblems to solve. Therefore, a lower bound is calculated based on the reduced costs to invoke an early termination of the inner loop.[*] The loop is stopped if the percentage difference between the upper bound provided by the current solution of FS_r and the lower bound is less than a predefined threshold. Other stopping criterion for the inner loop can be: (1) no more negative reduced cost sequences, (2) iteration limit and (3) time limit.

The objective of generating 'good' production sequences used to obtain superior quality integer solutions for the CLSD may not be achieved only by the inner loop. The column generation algorithm is mainly concerned in solving the linear relaxation of FS_r. Therefore, although some of the production sequences generated may contribute to the final purpose of the heuristic, other may only be useful to find the LP-optimum. Hence, after the termination of the inner loop, the outer loop obtains a primal solution. By rounding integer variables of FS, it guides the inner loop to generate sequences useful for integer solutions. The diving scheme rounds the least fractional variables hierarchically first on set U, then on set Y and finally on set Z.

The search for production sequences ends when a feasible integer solution is found or the model becomes infeasible after fixing some of the integer variables. The final output of the heuristic is an updated pool of production sequences for the time periods considered

[*]Let $z(FS_r^k)$ be the objective value of FS_r at iteration k of our column generation algorithm, rc_t^k be the minimum reduced cost associated with the solution of the pricing subproblem in time period t and $z(FS)$ the optimal value of the LP relaxation of FS. A lower bound on $z(FS)$ can be calculated by the following expression: $z(FS_r^k) \geq z(FS) \geq z(FS_r^k) + \sum_{t \in \mathcal{T}} rc_t^k$

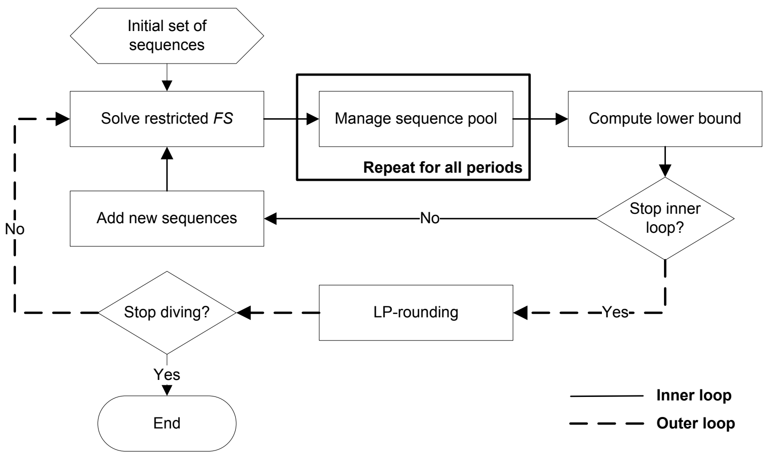


Figure 3.3: Outline of *SeqSearch*

together with an integer solution if one is found.

Next we present in more detail the subproblem solved to generate new production sequences.

3.4.1.1 Subproblems

Consider λ_t, θ^f_{it}, θ^l_{it}, α_{it} and π_{it} to be the dual variables associated with constraints (3.3), (3.5), (3.6), (3.8) and (3.9), respectively. For a specific time period t the subproblem objective function, which represents the reduced cost associated with variable W_{ts}, becomes:

$$(sub_t) \quad \underset{\chi_{ij}}{\text{Min}} \quad \widehat{sc}_s - \widehat{st}_s \cdot \lambda_t - \sum_{i \in \mathcal{N}} \left(f_{is} \cdot \theta^f_{it} - l_{is} \cdot \theta^l_{it} - e_{is} \cdot \alpha_{it} - a_{is} \cdot \pi_{it} \right). \tag{3.11}$$

The subproblem arising in each time period is a generalization of the prize collecting traveling salesman problem introduced by Balas (1989) as nodes can be visited more than once. Network $G = (V, A)$ consists of node set $V = \mathcal{N} \cup \{0, N+1\}$ and arc set A. Node 0 is the source and node $N+1$ the sink while the remaining nodes represent products (see Figure 3.4). The source and the sink are used to identify the starting and ending products of the production sequence, hence an arc connecting the source to a product means a carry over from the previous period and, similarly, an arc connecting a product to the sink represents a carry over to the next period. There is a prize ρ_i for visiting node i, as well as travel costs c_{ij} for traversing arcs (i, j). A node is considered to be visited if it follows another node other than the source. In addition, no penalties are considered for excluding nodes from the walk. The objective is to find the minimum cost walk through the network from the source to the sink. To mathematically state the subproblem, we introduce integer decision variables χ_{ij} representing the number of times arc (i, j) is traversed and y_i which equals 1 if node i is visited at least once or 0 otherwise. The MIP model formulation for the subproblem is presented in Appendix 3.C.

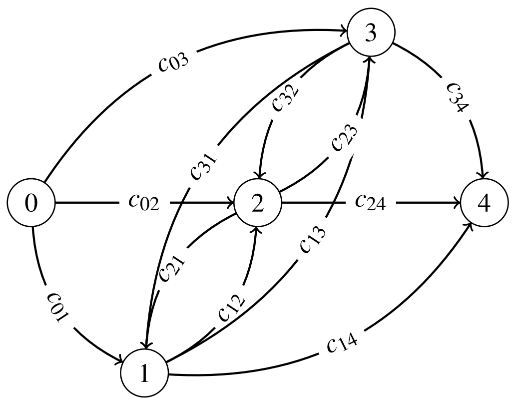


Figure 3.4: Network of the subproblem for $N = 3$

3.4.2 *Relax–Price–Fix*: Constructing an initial solution

To create a feasible integer solution to the CLSD, we have developed a construction heuristic based on the relax-and-fix scheme (Pochet and Wolsey (2006)) and the formulations discussed in Section 3.3. Integer variables of the original MIP problem are partitioned into subsets. Then by sequentially solving a collection of partially relaxed MIP subproblems an integer solution is found to the original MIP. At each iteration of the heuristic, integer variables can be grouped into three different subsets: (1) variables whose values have been fixed in previous iterations, (2) variables required to be integer in the current stage and (3) relaxed variables. As the heuristic progresses these three subsets are being updated. The heuristic finishes when a feasible integer solution is found to the entire problem, or when a subproblem results infeasible. The partitioning strategy of the integer variables of the original MIP determines both the solution quality and computational effort. The larger the subsets, the better the solution quality, however a more complex MIP subproblem has to be solved in each iteration.

Our strategy relies on time partitioning of the original MIP, where the planning horizon is divided into time intervals containing a subset of in-time adjacent time periods. This partition creates a rolling horizon approach, as the heuristic starts by solving subproblems corresponding to the first time periods and progressively moves towards the end of the planning horizon. Let k be the current relax–and–fix heuristic iteration and let t_s^k and t_f^k denote the starting and ending periods of the current subset. The subproblem to be solved in iteration k, labeled as $subMIP^k$, corresponds to the model FS where equations (3.10) are replaced by:

$$(X_{itl}, W_{ts}) \geq 0 \qquad \forall i \in \mathcal{N}, t, l \in \mathcal{T}, s \in \mathcal{S}_t \tag{3.12}$$

$$U_{it} = \overline{U}_{it}, Z_{it} = \overline{Z}_{it}, Y_{it} = \overline{Y}_{it} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, t < t_s^k \tag{3.13}$$

$$(U_{it}, Z_{it}) \in \{0,1\}, \quad Y_{it} \in \mathbb{N} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, t_s^k \leq t \leq t_f^k \tag{3.14}$$

$$(U_{it}, Z_{it}, Y_{it}) \geq 0 \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, t > t_f^k. \tag{3.15}$$

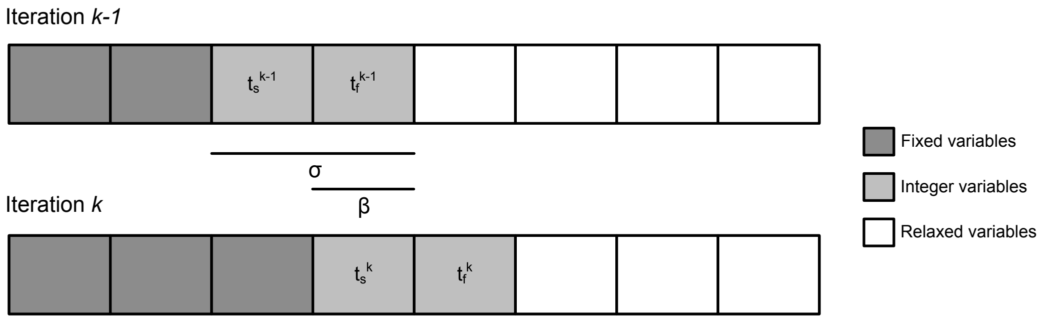


Figure 3.5: Successive iterations of *Relax–Price–Fix*

Figure 3.5 depicts two successive iterations of the heuristic. Time periods colored in dark gray are those in which the value of integer variables are fixed to the solution obtained in previous iterations (equations (3.13)). The subset of integer variables belonging to period t_s^k up to period t_f^k (periods in light gray) are restricted to assume integer values (equations (3.14)). Finally, the integer variables of later periods (filled in white) are relaxed to take fractional values (equations (3.15)). Consider σ to be the number of time periods in the subset of in-time adjacent periods and β to be the number of overlapping time periods between iterations. At the end of each iteration, integer variables from period t_s^k up to period $t_s^k + \sigma - \beta - 1$ are fixed to their respective value in the solution obtained by solving *subMIP*^{k}. The heuristic proceeds by moving t_s^k and t_f^k, according to $t_s^k = t_f^{k-1} - \beta + 1$ and $t_f^k = \min\{t_f^{k-1} + \sigma - \beta, T\}$, where σ is the number of time periods in each time partition and β is the number of time periods overlapping between iterations (in Figure 3.5 we have $\sigma = 2$ and $\beta = 1$).

Due to the nature of the original MIP model used to run the relax-and-fix heuristic (exponential number of decision variables of type W_{st}), *SeqSearch* was embedded within the relax-and-fix framework. Furthermore, aiming for a more efficient method, during *Relax–Price–Fix* the hybrid model discussed in Section 3.3.2 is used. Time periods spanning from the beginning of the planning horizon up to t_f^k define $\mathcal{T}_s$ and scheduling decisions are made using model FS. For later time periods ($t \in \mathcal{T}_p$, $t > t_f^k$) the product related formulation provides a relaxed solution in order to estimate future costs of the schedule. In each iteration, we first solve the linear relaxation of $subMIP^k$ using *SeqSearch* identifying new production sequences to add to model FS in time periods colored in light gray (*SeqSearch* is called for each time period $t \in [t_s^k, t_f^k]$), considering their respective dual values. Therefore, new production sequences are only generated for periods requiring integrality for integer variables as in previous periods these decisions have already been fixed and in later periods sequences are estimated by the product related model. Restricting the generation of new sequences to a low number of time periods in each iteration relieves the computational burden of the construction heuristic. Between two consecutive iterations of the construction heuristic, the hybrid model is update by converting the product related formulation into the sequence related formulation for periods $t \in [t_f^{k-1} + 1, t_f^k]$.

3.4.3 *Fix–Price–Optimize*: Improving solution quality

Let $\mathcal{T}^{'}$ define a subset of periods and $\mathcal{N}^{'}$ a subset of products. The subproblem aiming to improve the current best solution corresponds to fixing the integer variables not present in these two sets to their incumbent value so that changes to the value of the integer variables are only allowed within the defined subsets. Before solving the subproblem, $SeqSearch$ heuristic is performed to identify new production sequences to add into the sequence pool of model FS for the subset of periods $\mathcal{T}^{'}$ to be re-optimized, based on the dual values of the variables related to the products and periods in the defined subsets. Naturally, new production sequences are created taking into account the setups that will remain unchanged.

We systematically explore changes in $\mathcal{N}^{'}$ and $\mathcal{T}^{'}$ in order to avoid local minima. Consider an ordered finite set of user-defined neighborhood structures N_n, $(n = 1, \ldots, n_{max})$, where n denotes the nth neighborhood structure. Each neighborhood structure contains several neighbors. After solving a subproblem from the current neighborhood structure the new solution objective value is compared with the previous best solution value. In case of an improvement, the search restarts at the first neighborhood structure ($n = 1$). Otherwise, the number of failed attempts within the current neighborhood structure is increased. We allow a limited number of failures before switching to the next neighborhood structure in the ordered set.

Neighborhood structures are defined by the number of products $N^{'}$ and the number of adjacent periods $T^{'}$ to be re-optimized. A neighbor corresponds to the selection of $\mathcal{N}^{'} \subseteq \mathcal{N}$ of cardinality $N^{'}$ and $\mathcal{T}^{'} \subseteq \mathcal{T}$ of cardinality $T^{'}$, defining the set of 'released' variables and the MIP subproblem to solve. Hence, neighborhoods contain all possible combinations of $\mathcal{N}^{'}$ and $\mathcal{T}^{'}$ of given cardinalities. Since our neighbor evaluation is a computational expensive process a full evaluation of the neighborhoods is unpractical. Therefore, a stochastic process controls neighbor selection to conduct a partial neighborhood search.

When starting the exploration of a given neighborhood structure, scores τ_i and ω_t are assigned to each product and period, respectively. Initially, at the beginning of a neighborhood phase, we set all of them to 1. As products and periods are selected their score is updated so that the more frequent (number of times selected during the neighborhood exploration) and recently (number of neighbors explored since last selected) a given product or period has been selected, the lower is its score (weighted average of both criteria). The neighbors scoring method used is similar to the one described in James and Almada-Lobo (2011).

Two alternatives were developed to select the next neighbor to explore. Both start with a biased selection of the subset of products and periods according to probabilities $p(i) = \frac{\tau_i}{\sum_{j \in \mathcal{N}} \tau_j}$ for all products and $p(t) = \frac{\omega_t}{\sum_{l \in \mathcal{T}} \omega_l}$ for all periods. In the first option, which we call $P\&MIP^{rnd}$, the products and periods subsets are selected just once defining the next neighbor to explore. In the second approach, $P\&MIP^{eval}$, the selection of $\mathcal{N}^{'}$ and $\mathcal{T}^{'}$ is repeated K times. For each one of the K neighbors a single iteration of the inner loop in *SeqSearch* is performed to estimate the potential improvement that the neighbor can yield, which is inferred based on the obtained objective value of FS_r. Neighbors are then sorted in ascending order according to their potential. Let $\eta(k)$ be the rank of neighbor k. The probability $\mu(r)$ of choosing a candidate neighbor is given by:

$$\mu(r) = \frac{\eta(r)}{(K+1)\cdot K/2}.$$

For both cases, rather than having a random rule, we try to guide the search for the most promising neighbors. *P&MIP* ends according to the following stopping criteria: (1) the maximum running time allowed has been achieved, or (2) the maximum number of neighbors without improvement has been achieved in all neighborhood structures.

3.5. Computational results

In this section we present the computational experiments performed to validate our solution approach. In the following subsections instances are divided into three families according to their features. The first group of benchmark instances considers problem data having setup matrices obeying the triangle inequality. In the second instance group setups can violate the triangle inequality and, maximum and minimum lot sizes are also introduced. Finally, the last family of test instances is a collection of real-world problems.

All computational tests were conducted on Intel @ 2.40 GHz processing units limited to 4 GB of random access memory using the Linux operating system. The algorithm was implemented in C++ and compiled using a *gcc* compiler. IBM ILOG Cplex 12.1 was used both as the mixed integer and linear programming solver.

In all benchmark sets we use two variants of our solution approach to compare its performance against state-of-the-art algorithms or commercial solvers. Both apply *Relax-Price-Fix* to construct an initial solution to the problem and *Fix-Price-Optimize* as the improvement heuristic. The variants only differ in the neighbor selection step. The $P\&MIP^{rnd}$ variant selects the neighbors to explore based on their score, while $P\&MIP^{eval}$ selects neighbors to explore according to their potential (as described in Section 3.4.3).

The main goal of these computational experiments is to validate the approach under different problem settings, showing flexibility and robustness of the heuristic. Parameters were tunned during pre-testing and also reflect the empirical knowledge about the problem.

The following parameter values were used throughout the computational experiments. In *SeqSearch* we limit the sequence pool to ten times the number of products and prune the master solution once the percentage gap from the lower bound is below 0.01%. The *Relax–Price–Fix* construction heuristic takes two arguments: (1) σ - the number of time periods in each time partition and (2) β - the number of overlapping time periods between iterations. Through these parameters the number of iterations is automatically defined as $K = \lceil (T-\sigma)/(\sigma-\beta) \rceil + 1$. Preliminary tests of the construction heuristic led to $\sigma = 2$ and $\beta = 1$, as these values represent the best trade-off between efficacy and efficiency.

Both variants of our solution approach control the neighbors to be evaluated by the *Fix–Price–Optimize* heuristic, which requires a subset of periods and products to be re-optimized. The neighborhood structure definition, i.e. the number of periods and products to be solved, depends on the size of the instance. Table 3.1 presents the neighborhood structures defined. The following rule for the neighborhood structure definition was applied for

Table 3.1: Neighborhoods used for *Fix–Price–Optimize*

	$T < 8$		$T \geq 8$	
Neighborhood	N'	T'	N'	T'
N_1	6	3	6	3
N_2	4	5	4	5
N_3	10	5	5	8

all of the test instances and is derived from empirical studies during pre-testing. We always use 3 neighborhood structures, starting with one having a large subset of products and a small subset of adjacent time periods. In the subsequent neighborhood structure the periods subset is increased and the number of products to be re-optimize reduced. This represents the underlaying trade-off between the efficiency and effectiveness of the search. With the first neighborhood structures local minima is achieved as re-optimization is conducted for a small subset of periods, despite allowing for faster neighbor evaluations. Increasing the number of periods greatly increases the computational burden of the neighbor evaluation, although potentially allowing for a greater improvement of the incumbent solution. Hence, the increase in the number of periods is followed by a reduction in the number of products to smooth this impact. Finally, the last neighborhood structure attempts to escape from close local minima by allowing changes in a large set of periods and products simultaneously. We allow up to 10 neighbors without improvement before moving to the next neighborhood structure.

3.5.1 Triangular setups

The first set of benchmark instances, available from James and Almada-Lobo (2011), assesses our solution approach under the presence of triangular setups. We use the data set related to single machine problems with capacity variation.

Problem instances are grouped into problems types defined by the quadruplet N, T, Cut, θ (representing: number of products, number of periods, average capacity utilization per period and cost of setup per time unit). A total of 240 instances were solved resulting from 10 different instances for each one of the 24 problem types created by combining the different values of the parameters: $N \in \{15, 25\}$, $T \in \{5, 10, 15\}$, $Cut \in \{0.6, 0.8\}$ and $\theta \in \{50, 100\}$. For details concerning the problem instance generator the reader is referred to James and Almada-Lobo (2011).

In this benchmark, we compare the two variants of our solution approach with the Iterative Neighborhood Search heuristic starting with a Relax-and-Fix construction heuristic (INSRF) described in James and Almada-Lobo (2011), the best known method for the CLSD with triangular setups. All approaches have a maximum running time of one hour. We also present results for the construction heuristic *Relax–Price–Fix* (RPF). To evaluate the performance of the heuristics we calculate the gap from the lower bound reported in James and Almada-Lobo (2011). A summary of the results is given in Table 3.2, which aggregates instances by each level chosen according to data parameters, e.g., row $N = 25$

aggregates all means obtained for instances with 25 products while the other parameters vary. We report the mean percentage gap from the lower bound, the average running times in seconds and the p-value resulting from the Student's t-test performed to validate the results (described later in this section).

Table 3.2: Summary of results for the CLSD with triangular setups and capacity variation

		Mean gap (%)				p-value		Mean running time (s)			
		RPF	$P\&MIP^{rnd}$	$P\&MIP^{eval}$	INSRF	(1)[a]	(2)[b]	RPF	$P\&MIP^{rnd}$	$P\&MIP^{eval}$	INSRF
N	15	1.76	1.34	**1.33**	**1.33**	0.480	0.499	85.2	246.0	315.1	2708.4
	25	2.10	0.94	**0.90**	0.99	0.113	*0.015*	601.1	1437.4	1854.4	1652.8
T	5	1.49	0.79	0.79	**0.67**	*≈ 0*	*≈ 0*	155.3	337.3	387.2	1140.6
	10	1.96	1.15	**1.14**	1.41	*≈ 0*	*≈ 0*	295.3	862.8	1054.9	3048.9
	15	2.34	1.47	1.42	**1.41**	*0.027*	0.364	578.9	1324.9	1812.0	2352.3
Cut	0.6	1.77	1.04	**1.01**	1.04	0.446	0.290	298.0	739.1	958.9	2004.6
	0.8	2.09	1.23	**1.22**	1.29	0.057	*0.021*	388.4	944.3	1210.5	2356.5
θ	50	0.97	0.35	**0.30**	0.31	*0.002*	0.361	190.2	436.5	796.4	1615.1
	100	2.89	**1.93**	1.93	2.02	*0.036*	*0.044*	496.2	1246.8	1373.0	2746.0
Overall mean		1.93	1.14	**1.12**	1.16	0.176	*0.041*	343.2	841.7	1084.7	2180.6

[a] Comparison of means between INSRF and $P\&MIP^{rnd}$
[b] Comparison of means between INSRF and $P\&MIP^{eval}$

These results validate the ability of our solution approach to successfully solve instances with triangular setups. As expected, the solution obtained by the construction heuristic is considerably improved by the neighborhood search. Generally, both variants have a lower mean gap than INSRF for harder problems, i.e. high number of products, high capacity utilization and high setup cost. The only exception occurs when the number of periods increases. For problems with fewer products, low capacity utilization and low setup cost the difference between the algorithms is less noteworthy, although for instances with a small number of periods INSRF is better than any of our solution approach variants. Clearly, the neighborhoods in INSRF are more effective for instances with a short planning horizon, while our solution approach is more effective to explore larger instances. Note that for $T = 15$, INSRF and $P\&MIP^{eval}$ produce almost the same gaps.

Neighbor selection also appears to play an important role in our solution approach. Guiding the partial neighborhood exploration process through the assessment of potential improvement of neighbors leads to lower mean gaps than by only combining neighbor scores and randomization.

To confirm these underlaying hypotheses of different performances of the tested heuristics, we carried out a paired Student's t-tests comparing the mean gaps of the two variants of $P\&MIP$ with INSRF for each one of the categories present in Table 3.2. The p-values reported refer to the alternative hypothesis that the method with lower mean gap has a better performance. In all cases $P\&MIP$ is considered, except for $T = 5$ and $T = 15$ where the alternative hypothesis is that the mean gap of INSRF is less than the mean gap of $P\&MIP$. Considering a significance level of 0.05, the statistical tests confirm $P\&MIP^{eval}$ as the best approach for hard problems ($N = 25$, $\theta = 100$, $Cut = 0.8$). For easier instances we cannot draw conclusions. With respect to the mean gap, tests point $P\&MIP^{eval}$ as the overall best performing method and no statistical evidence of different performances between

$P\&MIP^{rnd}$ and INSRF.

Finally, since the computational study of James and Almada-Lobo (2011) has been conducted on a similar computing architecture, using the same version of CPLEX, we assume that the running times are comparable. Both variants of our solution approach require considerable less computational time. The only exception are instances having a large number of products due to the increasing difficulty in solving the subproblem. Moreover, as expected, the neighbor evaluation step increases the solution time when compared to the selection based on a single sample.

3.5.2 Non-triangular setups

Next we present results concerning two benchmark sets in which setup matrices do not obey the triangle inequality. The first comes from the work of Kang et al. (1999) and enables us to validate the algorithm by benchmarking it against other solution procedures capable of tackling non-triangular setups. The second set is motivated by the small size of the instances in Kang et al. (1999). To create larger non-triangular instances, we modify the problem set of James and Almada-Lobo (2011) by introducing shortcut products in the setup matrices, so that the triangle inequality does not hold anymore.

3.5.2.1 Small problems

Kang et al. (1999) created modified instances based on CHES problem number 5 (Baker and Muckstadt (1989)). A total of nine instances are available among which six are single machine problems *Da*,..., *Df*. All problems have six products and a planning horizon of nine time periods. Moreover, setup times are zero, products setups present a clustered structure and requirements on the maximum and minimum lot size are imposed. Different combinations of machine utilization and lot size requirements are used to generate the problem set (see Table 3.3). The two variants of our solution approach are compared to:

- **Kang**: the branch-and-price based heuristic of Kang et al. (1999);
- **Meyr SAPL**: the simulated annealing algorithm of Meyr (2002);
- **Meyr TAPL**: the threshold acceptance algorithm of Meyr (2002);
- **CPLEX**: Branch-and-Cut performed by parallel CPLEX 12.1 on the compact model for the CLSD with non-triangular sequence-dependent setups presented in Appendix 3.B.

Besides the data features of the instances, Table 3.3 reports the upper bound provided by each method and the running times in seconds for the two variants of our solution approach and CPLEX. Both the upper bound and the running time presented in the case of our heuristic is the best run out of 20 different attempts (our heuristics embed a random component). We stress that Kang et al. (1999) and Meyr (2002) also report best values out of multiple runs.

Both variants of $P\&MIP$ were able to find the optimal solution for all problem instances (CPLEX 12.1 proves optimality in all instances). This validates the ability of the

Table 3.3: Summary of results for the problems in Kang et al. (1999)

Instance		*Da*	*Db*	*Dc*	*Dd*	*De*	*Df*
Data features	Utilization (%)	95	99	70	95	99	95
	Minimum lot size	20	20	20	20	20	0
	Maximum lot size	200	200	200	100	100	1000
Upper bound	Kang	856.81	865.29	816.61	1263.01	1360.87	832.95
	Meyr TAPL	846.97	859.97	766.58	1182.11	1248.26	812.66
	Meyr SAPL	844.62	869.3	760.08	1174.01	1260.33	812.66
	$P\&MIP^{rnd}$	842.64	852.32	758.49	1164.93	1202.33	812.66
	$P\&MIP^{eval}$	842.64	852.32	758.49	1164.93	1202.33	812.66
	CPLEX	842.64	852.32	758.49	1164.93	1202.33	812.66
Deviation from Kang, Meyr	$P\&MIP^{rnd}$	-0.2%	-0.9%	-0.2%	-0.8%	-3.7%	0.0%
	$P\&MIP^{eval}$	-0.2%	-0.9%	-0.2%	-0.8%	-3.7%	0.0%
	CPLEX	-0.2%	-0.9%	-0.2%	-0.8%	-3.7%	0.0%

heuristic to deliver superior quality solutions even for easy instances. The heuristic strictly outperforms the previous methods in terms of solution quality, except for problem D_f, for which the best solution previously reported is already optimal. The running times of our heuristic can not be compared to those of Kang, Meyr TAPL and Meyr SAPL since there are significant differences in hardware and software. However, we point out that CPLEX running times are shorter than the running times of our heuristic. In fact, the potential of our heuristic relies on solving bigger problems as exact methods are hard to beat for small instances like these ones. In the next section we report computational results on a set of larger instances to show this effect.

3.5.2.2 Modified triangular problems

Benchmark instances in the literature violating the triangle inequality are relatively scarce and small sized. The following benchmark set was designed in order to test the solution approach for harder instances of this type. Since hard instances for the triangular setup case are available in James and Almada-Lobo (2011), we chose to adapt them for the non-triangular setup case. To do so, we modify the setup time matrices to create a set $\mathcal{N}_{SC}$ of potentially shortcut products (products that lead to the violation of the triangle inequality). The number of shortcut products in the set is defined by $N_{SC} = \left\lceil \frac{N}{10} \right\rceil$. For each shortcut product $k \in \mathcal{N}_{SC}$ setup times st_{ik} and st_{ki} for each $i \notin \mathcal{N}_{SC}$ are generated from the uniform distribution between 2 and 4. The setup costs remain proportional to setup times using parameter θ.

Problems are classified as described in Section 3.5.1. We evaluate the performance of the two variants of $P\&MIP$ against branch-and-cut performed by parallel CPLEX 12.1 on the compact formulation for the CLSD with non-triangular sequence-dependent setups presented in Appendix 3.B. In order to evaluate the quality of the solutions generated by these procedures, we use the deviation from the best bound obtained by CPLEX during the tree search. Tables 3.4 and 3.5 present the results by problem type for low ($\theta = 50$) and high ($\theta = 100$) setup cost instances, respectively. Each instance is encoded under problem type as N-T-Cut-θ. Numbers in bold highlight the best average gaps. Problems types with five time periods were excluded, since they are too small.

Table 3.4: Summary of results for the CLSD with non-triangular setups and low setup cost ($\theta = 50$)

	Mean deviation (%)			Mean running time (secs)		
Problem Type	CPLEX	$P\&MIP^{eval}$	$P\&MIP^{rnd}$	CPLEX	$P\&MIP^{eval}$	$P\&MIP^{rnd}$
15-10-0.6-50	**0.00**	0.22	0.45	58.8	876.9	955.0
15-10-0.8-50	**0.00**	0.53	0.67	434.6	986.9	1066.8
15-15-0.6-50	**0.00**	0.39	0.31	496.5	1334.2	1151.1
15-15-0.8-50	**0.04**	0.60	0.47	2138.2	1265.2	1225.5
25-10-0.6-50	**0.02**	1.60	1.40	1803.2	3601.5	3601.7
25-10-0.8-50	**0.03**	1.57	1.81	3057.5	3602.5	3601.4
25-15-0.6-50	**0.46**	2.25	2.12	3523.1	3602.2	3602.2
25-15-0.8-50	**0.25**	2.30	2.26	3601.9	3602.3	3602.7

Table 3.5: Summary of results for the CLSD with non-triangular setups and high setup cost ($\theta = 100$)

	Mean deviation (%)			Mean running time (secs)		
Problem Type	CPLEX	$P\&MIP^{eval}$	$P\&MIP^{rnd}$	CPLEX	$P\&MIP^{eval}$	$P\&MIP^{rnd}$
15-10-0.6-100	**0.20**	1.06	1.36	2340.7	943.5	748.4
15-10-0.8-100	**0.76**	2.07	1.57	3600.3	1043.2	1086.1
15-15-0.6-100	**1.82**	2.20	2.07	3600.5	1214.9	1424.7
15-15-0.8-100	2.74	**2.39**	2.84	3600.4	1524.5	1186.2
25-10-0.6-100	**1.00**	2.15	2.34	3601.4	3601.6	3601.4
25-10-0.8-100	3.39	**2.35**	2.52	3601.4	3577.4	3601.3
25-15-0.6-100	6.82	2.67	**2.42**	3602.1	3602.1	3601.9
25-15-0.8-100[a]	9.79	**2.99**	3.24	3602.2	3602.2	3601.8

[a] CPLEX fails to achieve a feasible solution for 5 out of 10 instances in this problem type. Mean deviation were calculated based on the remaining instances.

These results indicate that instances with low setup cost ($\theta = 50$) are relatively easy for CPLEX, because it can often prove optimality or the final gaps are below 0.5%. Note that, in these settings, sequencing decisions are less important. For these instances our heuristic variants are not competitive. Nevertheless, the solution quality provided by the heuristics is never above 2.5% from the solution found by CPLEX (most of the times the optimal solution), which indicates good quality solutions. Since CPLEX can often prove optimality, running times are shorter when compared to those of the heuristic variants.

For problems with high setup cost ($\theta = 100$), the heuristics reveal their efficiency. As the problems become harder (more products and/or time periods) the heuristics outperform CPLEX. This is the case for 15 products and 15 time periods, and 25 products and 15 time periods. This effect is particularly pronounced in the last two problem types shown in Table 3.5, for which CPLEX solution quickly deteriorates. In the last problem type, CPLEX even fails to deliver a feasible solution for 5 out of 10 instances, while our heuristics allways find a feasible solution. Comparing the two variants, the guided neighbor selection has an advantage over a single sample selection in terms of the solution quality.

For problem types with 15 products running times of the heuristics are shorter when compared to CPLEX, indicating an opportunity for additional improvements in the solution quality of the heuristics. For problems with 25 products, the searches end by the maximum running time allowed in any approach.

3.5.3 Real world instances

The last set of instances corresponds to a collection of seven different real-world problems from the beverage industry. Problems have a planning horizon of 8 weeks, common in tactical production planning in the process industry, while the number of products varies from 8 to 33. Problems are quite diverse, ranging from scenarios with high demand of few standard products to production lines dedicated to several products with low demand and highly customized. Across all instances, capacity is constant throughout the planning horizon, however orders are highly unbalanced between time periods.

Table 3.6 reports the results for the two variants of the heuristic and the Branch-and-Cut performed by parallel CPLEX 12.1 on the compact model for the CLSD with non-triangular sequence-dependent setups presented in Appendix 3.B.

Table 3.6: Summary of results for real-world instances

Instance			Objective			Deviation (%)**			Time		
#	N	T	CPLEX	$P\&MIP^{eval}$	$P\&MIP^{rnd}$	$\frac{(b-a)}{a}$	$\frac{(c-a)}{a}$	$\frac{(c-b)}{b}$	CPLEX	$P\&MIP^{eval}$	$P\&MIP^{rnd}$
S2	8	8	**100915.6***	101470.1	101470.1	0.55	0.55	0.00	132.77	182.13	108.68
L3	10	8	**94010.79***	94191.8	94090.05	0.19	0.08	-0.11	142.11	208.59	221.71
L5	11	8	**77514.12***	77833.78	77582.06	0.41	0.09	-0.32	192.69	153.76	162.43
S1	15	8	267409.2	183722.9	**183068.8**	-31.30	-31.54	-0.36	3600.31	998.35	1062.6
R2	19	8	**159439.4**	167705.6	168977.9	5.18	5.98	0.76	3600.73	1928.87	1100.3
S7	20	8	-	**189556.7**	199323.2	-	-	5.15	3600.74	3603.16	3507.41
L6	33	8	-	**464409.8**	489310.4	-	-	5.36	3603.11	3659.48	3611.29

* Optimal solution values
** a - CPLEX, b - $P\&MIP^{eval}$, c - $P\&MIP^{rnd}$

Problems are sorted by the increasing number of products, which also increases the difficulty. For problems with less than 15 products CPLEX can obtain provably optimal solutions. Nevertheless, the percentage deviation of the solutions provided by our heuristics is quite small, attesting the ability of the heuristics in finding high quality solutions. For problems having a larger number of products, our heuristics outperforms the exact method. In these instances, CPLEX performance is not satisfactory as it even fails to deliver a feasible solution for two of the instances (S7 and L6, those with the largest number of products). Additionally, the solution found for instance S1 is 30% higher than the solutions of the heuristic variants.

Heuristic running times are competitive for easy problems compared to the exact method. In problems S1 and R2 we can observe that the neighborhood exploration process ends before the maximum running time limit, contrarily to CPLEX. For larger problems the search consumes the entire allowed running time in either case.

In terms of the two variants of the heuristic, there is not a clear difference in the first five problems. The benefits of the guided local search can only be seen in the last two instances, which have larger neighborhoods and, therefore, neighbor selection becomes more important. We observe an improvement of 5% in the solution quality for these cases.

3.6. Conclusions

In this paper we present a novel mixed integer programming formulation for the single machine capacitated lot sizing and scheduling problem. An important cause of computational intractability of large bucket models for lot sizing and sequencing often comes from the sequencing problem that has to be solved within each time period. The underlying idea of our model is to simplify the scheduling part by defining, for each time period, a limited pool of feasible production sequences. Afterwards, the model selects the most adequate one for each time period. The formulation handles non-triangular setups times and costs, as well as minimum and maximum lot sizes.

Based on the model we have developed a mathematical programming based solution approach that handles large size instances. It is composed of three main blocks: a pricing heuristic, a construction heuristic, and an improvement heuristic. The pricing heuristic, *SeqSearch*, is used to manage the sequence pool of each time period, since a direct implementation considering all possible production sequences would require the use of an exponential number of variables. Both the construction and improvement heuristics are the outcome of combining the pricing heuristic with mathematical programming based heuristics. The key principle behind these procedures is the selection of smaller subproblems which are easier to tackle. The construction heuristic, *Relax-Price-Fix*, generates a feasible solution by introducing column generation within the relax-and-fix framework. A series of partially relaxed MIPs are solved until a feasible solution is obtained. An important innovation is the combination of a compact and extended formulation in a single model during the construction phase to relief the computational burden of the procedure. *Fix-Price-Optimize*, our improvement heuristic, escapes from local minima by re-optimizing a feasible solution over a small subset of the original problem. These partitions are explored in a systematic framework and *SeqSearch* is embedded during re-optimization to discover new production sequences.

In our computational tests we show the flexibility of using mathematical programming based heuristics by tackling different features of the problem. The benchmark sets solved ranged from the case of triangular setups to non-triangular setups with minimum and maximum lot size restrictions. The approach outperformed the state-of-the-art algorithms for large problems. Using parallel CPLEX 12.1 on a compact formulation revealed to be the dominant method for small and low setup cost instances, but as the instances become bigger and harder both variants of *P&MIP* are a better approach. The results on the set of real-world problems solved have shown the practical application of the approach.

Our insights also indicate that to conduct a partial exploration of the neighborhood of an incumbent solution, devising a rule for neighbor selection has a positive effect regarding the quality of the solution.

One of the interesting topics for future research is the subproblem which arises in the pricing heuristic. In this paper, it is solved using exact approaches (branch-and-cut), however for instances having a large number of products to be scheduled, an approximate approach can save computational time. Even so, such an approach would slow the convergence of the column generation algorithm, hence the benefit is still to be studied and assessed. In addition, regarding the pricing heuristic, the diving scheme can be modified to improve the search for production sequences.

The techniques described in this paper aim to solve the problem under consideration, nevertheless most of the ideas can be applied to different lot sizing problems or to problems beyond this research subject.

Bibliography

B. Almada-Lobo, D. Klabjan, J. F. Oliveira, and M. A. Carravilla. Single machine multi-product capacitated lot sizing with sequence-dependent setups. *International Journal of Production Research*, 45(20):4873 – 4894, 2007.

F. Alvelos, A. de Sousa, and D. Santos. Searchcol: Metaheuristic search by column generation. In M. Blesa, C. Blum, G. Raidl, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 6373 of *Lecture Notes in Computer Science*, pages 190–205. Springer Berlin / Heidelberg, 2010.

T. Baker and J. Muckstadt. The CHES problems. Technical report, Chesapeake Decision Sciences, Inc., 1989.

E. Balas. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636, 1989.

M. O. Ball. Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science*, 16(1):21 – 38, 2011.

C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1996.

G. Bitran and H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28(10):1174–1186, 1992.

C. Blum, J. Puchinger, G. R. Raidl, and A. Roli. Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, 11(6):4135 – 4151, 2011.

A. Clark, B. Almada-Lobo, and C. Almeder. Lot sizing and scheduling: industrial extensions and research opportunities. *International Journal of Production Research*, 49(9): 2457–2461, 2011.

A. Drexl and A. Kimms. Lot sizing and scheduling - survey and extensions. *European Journal of Operational Research*, 99(2):221–235, 1997.

A. Federgruen, J. Meissner, and M. Tzur. Progressive interval heuristics for multi-item capacitated lot-sizing problems. *Operations Research*, 55(3):490–502, 2007.

D. Ferreira, R. Morabito, and S. Rangel. Solution approaches for the soft drink integrated production lot sizing and scheduling problem. *European Journal of Operational Research*, 196(2):697–706, 2009.

B. Fleischmann and H. Meyr. The general lotsizing and scheduling problem. *OR Spektrum*, 19(1):11–21, 1997.

K. Haase and A. Kimms. Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. *International Journal of Production Economics*, 66(2):159 – 169, 2000.

R. J. James and B. Almada-Lobo. Single and parallel machine capacitated lotsizing and scheduling: New iterative MIP-based neighborhood search heuristics. *Computers & Operations Research*, 38(12):1816 – 1825, 2011.

R. Jans and Z. Degraeve. Modeling industrial lot sizing problems: a review. *International Journal of Production Research*, 46(6):1619–1643, 2008.

S. Kang, K. Malik, and L. J. Thomas. Lotsizing and scheduling on parallel machines with sequence-dependent setup costs. *Management Science*, 45(2):273–289, 1999.

A. Kovács, K. N. Brown, and S. A. Tarim. An efficient MIP model for the capacitated lot-sizing and scheduling problem with sequence-dependent setups. *International Journal of Production Economics*, 118(1):282 – 291, 2009.

J. Krarup and O. Bilde. Plant location, set covering and economic lot size: An O(mn) algorithm for structured problems. In *Optimierung bei Graphentheoretischen und Ganzzahligen Probleme*, pages 155 – 180. BirkhauserVerlag, Berlin, 1977.

V. Maniezzo, T. Stützle, and S. Voß. *Matheuristics*, volume 10 of *Annals of Information Systems*. Springer US, 2010.

A. Menezes, A. Clark, and B. Almada-Lobo. Capacitated lot-sizing and scheduling with sequence-dependent, period-overlapping and non-triangular setups. *Journal of Scheduling*, 14(2):209–219, 2011.

H. Meyr. Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. *European Journal of Operational Research*, 120(2):311–326, 2000.

H. Meyr. Simultaneous lotsizing and scheduling on parallel machines. *European Journal of Operational Research*, 139(2):277 – 292, 2002.

I. Muter, Ş. İ. Birbil, and G. Şahin. Combination of metaheuristic and exact algorithms for solving set covering-type optimization problems. *INFORMS Journal on Computing*, 22 (4):603–619, 2010.

Y. Pochet and L. A. Wolsey. *Production Planning by Mixed Integer Programming (Springer Series in Operations Research and Financial Engineering)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

F. Sahling, L. Buschkühl, H. Tempelmeier, and S. Helber. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research*, 36(9):2546–2553, 2009.

S. Sarin, H. Sherali, and L. Yao. New formulation for the high multiplicity asymmetric traveling salesman problem with application to the Chesapeake problem. *Optimization Letters*, 5(2):259–272, 2011.

L. A. Wolsey. Solving multi-item lot-sizing problems with an MIP solver using classification and reformulation. *Management Science*, 48(12):1587–1602, 2002.

X. Zhu and W. E. Wilhelm. Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE Transactions*, 38(11):987–1007, 2006.

Appendix 3.A Imposing minimum and maximum lot sizes

Whenever non-triangular setups appear, minimum lot sizes must be imposed in the model to exclude solutions where 'empty' setups to shortcut or 'cleaning' are scheduled. Minimum lot sizes guarantee proper machine cleaning via the production of a minimum amount of a shortcut product. Furthermore, technological constraints may also impose a maximum lot size on each product run. Both of these features appear in the CHES instances (Baker and Muckstadt (1989)), which are a compilation of real world problems. To introduce lot size requirements in our model, let min_i^l, max_i^l be the minimum and maximum lot size of each production run of product i, respectively. We also introduce decision variables X_{it}^a, X_{it}^b to be the production quantities of product i manufactured after and before the first setup occurs in period t, respectively. The following constraints are adapted from Menezes et al. (2011) and must be added to model FS.

$$\sum_{l \geq t} X_{itl} = X_{it}^a + X_{it}^b \quad \forall\, i \in \mathcal{N}, t \in \mathcal{T} \tag{3.16}$$

Minimum lot sizes:

$$X_{it}^b \leq \frac{cap_t}{p_i} \cdot Z_{it} \qquad \forall\, i \in \mathcal{N}, t \in \mathcal{T} \tag{3.17}$$

$$X_{i1}^b \geq min_i^l \cdot Z_{i1} \qquad \forall\, i \in \mathcal{N} \tag{3.18}$$

$$X_{it}^a \geq min_i^l \cdot (Y_{it} - Z_{it}) \qquad \forall\, i \in \mathcal{N}, t \in \mathcal{T} \setminus \{T\} \tag{3.19}$$

$$X_{it}^a + X_{i,t+1}^b \geq min_i^l \cdot Y_{it} \qquad \forall\, i \in \mathcal{N}, t \in \mathcal{T} \setminus \{T\} \tag{3.20}$$

$$X_{iT}^a \geq min_i^l \cdot Y_{iT} \qquad \forall\, i \in \mathcal{N} \tag{3.21}$$

Maximum lot sizes:

$$X_{it}^b \leq max_{it}^l \cdot Z_{it} \qquad \forall\, i \in \mathcal{N}, t \in \mathcal{T} \tag{3.22}$$

$$X_{it}^a + X_{i,t+1}^b \leq max_i^l \cdot Y_{it} \qquad \forall\, i \in \mathcal{N}, t \in \mathcal{T} \setminus \{T\} \tag{3.23}$$

$$X_{iT}^a \leq max_i^l \cdot Y_{iT} \qquad \forall\, i \in \mathcal{N} \tag{3.24}$$

Constraints (3.16) are commonly used to model minimum and maximum lot size requirements and split the total period production into the amounts produced after and before the first setup is performed. Constraints (3.17)-(3.21) model minimum lot sizes. Constraints (3.17) impose that the production of a given product can only take place before the first setup if the product is carried over from the previous period. In (3.18) the minimum product lot is imposed for the initial setup configuration of the machine. Constraints (3.19) force production lots within the current period to respect the minimum lot size, while constraints (3.20) require that production within the current period plus the amount produced in the following period prior to the first setup must be at least proportional to the minimum lot size and the number of setups performed to that product. Finally, constraints (3.21) are a special case of constraints (3.20) for the final period of the planning horizon. To model maximum lot size constraints, (3.22)-(3.24) are necessary. Constraints (3.22) replace (3.17) with the same functionality and also restrict the amount produced before the first setup to the maximum of a production run. The maximum amount that can be produced considering the number of setups defined for the period is expressed by constraints (3.23), (3.24) for the first $T-1$ periods and for the final period, respectively. All these constraints rely on the assumption that at least a setup is performed in each time period. This implies that for some instances sub-optimal solutions are created. However, such cases are rare since we are modeling a big bucket problem, therefore is highly unlikely that production lots span more than one entire time period.

Appendix 3.B A compact formulation for the CLSD with non–triangular setups

In order to formulate a new product related MIP model to the CLSD considerer T_{ijt} to be the number of changeovers from product i to product j in period t and G_{it} a binary variable indicating if product i is part of the production sequence in period t or not. Variables f_{ijkt} contain the flow of commodity k from node i to node j in period t and are used to prevent sub-tours. The model reads:

$$(FP) \quad \text{Min} \quad \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} sc_{ij}\cdot T_{ijt} + \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{\substack{l\in\mathcal{T}\\ l>t}}(l-t)\cdot h_i\cdot X_{itl} \tag{3.25}$$

$$\text{s.t.} \sum_{\substack{l\in\mathcal{T}\\ l\leq t}} X_{itl} \geq d_{il} \qquad \forall\, i\in\mathcal{N}, l\in\mathcal{T} \tag{3.26}$$

$$\sum_{i\in\mathcal{N}}\sum_{\substack{l\in\mathcal{T}\\ l\geq t}} p_i\cdot X_{itl} + \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} st_{ij}\cdot T_{ijt} \leq cap_t \qquad \forall\, t\in\mathcal{T} \tag{3.27}$$

$$X_{itl} \leq d_{il}\cdot G_{it} \qquad \forall\, i\in\mathcal{N}, t,l\in\mathcal{T}, l\geq t \tag{3.28}$$

$$Z_{it} + \sum_{j\in\mathcal{N}} T_{jit} = Z_{i,t+1} + \sum_{j\in\mathcal{N}} T_{ijt} \qquad \forall\, i\in\mathcal{N}, t\in\mathcal{T} \tag{3.29}$$

$$\sum_{i\in\mathcal{N}} Z_{it} = 1 \qquad \forall\, t\in\mathcal{T} \tag{3.30}$$

$$\sum_{j\in\mathcal{N}} T_{jit} + Z_{it} \geq G_{it} \qquad \forall\, i\in\mathcal{N}, t\in\mathcal{T} \tag{3.31}$$

$$\sum_{j\in\mathcal{N}} T_{jit} + Z_{it} \leq setup_{it}^{max}\cdot G_{it} \qquad \forall\, i\in\mathcal{N}, t\in\mathcal{T} \tag{3.32}$$

$$f_{0jkt} \leq Z_{jt} \qquad \forall\, j,k\in\mathcal{N}, t\in\mathcal{T} \tag{3.33}$$

$$f_{ijkt} \leq T_{ijt} \qquad \forall\, i,j,k\in\mathcal{N}, t\in\mathcal{T} \tag{3.34}$$

$$\sum_{j\in\mathcal{N}} f_{0jkt} = G_{kt} \qquad \forall\, k\in\mathcal{N}, t\in\mathcal{T} \tag{3.35}$$

$$\sum_{j\in\mathcal{N}} f_{ijkt} = \sum_{j\in\mathcal{N}\setminus\{k\}} f_{jikt} + f_{0ikt} \qquad \forall\, k\in\mathcal{N}, i\in\mathcal{N}\setminus\{k\}, t\in\mathcal{T} \tag{3.36}$$

$$\sum_{j\in\mathcal{N}} f_{jkkt} + f_{0kkt} = G_{kt} \qquad \forall\, k\in\mathcal{N}, t\in\mathcal{T} \tag{3.37}$$

$$\left(X_{itl}, f_{ijkt}\right) \geq 0, (Z_{it}, G_{it}) \in \{0,1\}, T_{ijt} \in \mathbb{N} \quad \forall i, j, k, \in \mathcal{N}, t, l \in \mathcal{T}. \tag{3.38}$$

Objective function (3.25) minimizes the total expenditure in holding and setup costs. Constraints (3.26) guarantee demand fulfillment. The total production and setup time may not exceed the available machine capacity as ensured by constraints (3.27). Requirements (3.28) link products with the machine setup state. Constraints (3.29) balance the setup flow for each product. Equations (3.30) state that the machine is set up for exactly one product in the beginning of each time period. The relationship between the setup state and both the initial machine configuration and the changeovers are established by constraints (3.31) and (3.32). Constraints (3.32) also limit the number of changeovers to a given product in each time period by $setup_{it}^{max}$. Disconnected sub-tours are eliminated by constraints (3.33)-(3.37). Namely, constraints (3.33) and (3.34) ensure that the flows only traverse arcs in the period's sequence, while (3.35)-(3.37) impose that flow variables for each commodity k describe a path from the source (setup carry-over) to node k, if the node is present in the sequence of the respective time period.

Appendix 3.C Subproblem formulation

The MIP model for the subproblem in time period t is as follows.

$$(sub_t) \quad \text{Min} \quad \sum_{i \in V} \sum_{j \in V} c_{ij} \cdot \chi_{ij} - \sum_{i \in \mathcal{N}} \rho_i \cdot y_i \tag{3.39}$$

$$\text{s.t.} \quad \sum_{j \in V} \chi_{ji} = \sum_{j \in V} \chi_{ij} \qquad \forall i \in \mathcal{N} \tag{3.40}$$

$$\sum_{j \in \mathcal{N}} \chi_{0j} = 1 \tag{3.41}$$

$$\sum_{j \in \mathcal{N}} \chi_{j,N+1} = 1 \tag{3.42}$$

$$\sum_{j \in \mathcal{N}} \chi_{ji} \geq y_i \qquad \forall i \in \mathcal{N} \tag{3.43}$$

$$\sum_{j \in \mathcal{N}} \chi_{ji} \leq setup_{it}^{max} \cdot y_i \qquad \forall i \in \mathcal{N} \tag{3.44}$$

$$y_i \in \{0,1\} \quad \chi_{ij} \in \mathbb{N} \qquad \forall i, j \in \mathcal{N} \tag{3.45}$$

Here c_{ij} and ρ_i are derived from pricing equation (3.11):

$$c_{ij} = sc_{ij} - st_{ij} \cdot \lambda_t - \pi_{jt}, \quad c_{0i} = -\theta_{it}^{f}, \quad c_{i,N+1} = -\theta_{i,t+1}^{l}, \quad \rho_i = \alpha_{it} \qquad \forall i, j \in \mathcal{N}.$$

Objective function (3.39) minimizes the cost of the traversed arc minus the prizes collected from the scheduled products (visited nodes). Constraints (3.40) balance in- and out-flow of each product. The source and sink nodes must be connected to a product, guaranteed by requirements (3.41) and (3.42), representing the first and last products in the sequence. Constraints (3.43) and (3.44) enforce the logical relationship between the arcs traversed and the products visited. Parameter $setup_{it}^{max}$ is an upper bound on the number of setups for product i in period t.

The model for the subproblem is, however, still incomplete, as a solution for (3.39)-(3.45) permits disconnected sub-tours. To eliminate such sub-tours we use multi-commodity-flow type constraints. Consider decision variables f_{ijk} as the flow of commodity k from the source to node k traversing arc (i, j), which is constrained to be 0 or 1. Furthermore, the additional decision variables $y_i^{'}$ equal to 1 in case node i is ever traversed. The difference between y_i and $y_i^{'}$ relies on the fact that the latter equals to one also if the product is scheduled immediately after the source (first in the sequence, not representing an actual setup into it). The following constraints are added to sub_t to prohibit disconnected sub-tours (adapted from Sarin et al. (2011)):

$$f_{ijk} \leq \chi_{ij} \qquad \forall i \in V, j, k \in \mathcal{N} \tag{3.46}$$

$$\sum_{j \in \mathcal{N}} f_{0jk} = y_k^{'} \qquad \forall k \in \mathcal{N} \tag{3.47}$$

$$\sum_{j \in \mathcal{N}} f_{ijk} = \sum_{j \in \mathcal{N} \setminus \{k\}} f_{jik} \qquad \forall k \in \mathcal{N}, i \in \mathcal{N} \setminus \{k\} \tag{3.48}$$

$$\sum_{j \in V} f_{jkk} = y_k^{'} \qquad \forall k \in \mathcal{N} \tag{3.49}$$

$$y_i^{'} \leq y_i + \chi_{0i} \qquad \forall i \in \mathcal{N} \tag{3.50}$$

$$2 \cdot y_i^{'} \geq y_i + \chi_{0i} \qquad \forall i \in \mathcal{N}. \tag{3.51}$$

Constraints (3.46) ensure that flows only traverse the arcs in the solution. Constraints (3.47)-(3.49) require that flow variables for each commodity k describe a path from the source to node k, if node k is in the sequence defined by arc variables. In more detail, constraints (3.47) force a unit of flow to leave the source for each node in the sequence. Flow conservation constraints (3.48) are imposed for each node in the graph, and, finally, the flow of commodity k must reach node k by (3.49). The last two sets of constraints (3.50)-(3.51) represent the logical connections between node variables.

CHAPTER 4

Short-medium term production planning

Rolling horizon formulations for short-medium term production planning

Luis Guimarães* · **Diego Klabjan†** · **Bernardo Almada-Lobo***

Working paper

Abstract In their operational production planning several industries have to size and schedule production lots on a set of parallel machines to satisfy forecasted demand while facing sequence dependent changeover times and costs. Motivated by a case study in a beverage company, we exploit the practice of rolling basis planning to develop efficient approaches to the problem. The horizon is decomposed in two parts: the first periods explicitly consider the production sequences to obtain detail schedules, and in the remaining periods a rough plan is generated to give an estimation of future costs and capacity. Several simplification modeling alternatives proposed in the literature are reviewed and a new formulation that includes the setup loss in the future periods based on the loss witnessed in the detailed part of the horizon is proposed. An important contribution is an innovative iterative method to improve the accuracy of the approximate parameters used in the context of the simplified models. We assess the performance of the several alternatives by simulating the implementation of solutions on a rolling horizon first by using instances generated based on the features arising in the beverage industry and later on a small collection of real-world instances. The test show that in this context the new formulation is very successful and that applying the iterative approach on the approximate methods can significantly improve the solution quality.

Keywords Lot sizing and scheduling · Sequence-dependent setups · Rolling planning · Approximate formulations · Iterative method

---

*INESC TEC, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal

†Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA

4.1. Introduction

In this paper we discuss practical modeling techniques to integrate medium and short term decisions for production planning and scheduling in a single facility. We are motivated by a case study in a beverage company that produces mineral and sparkling water, beer and soft drinks in a set of different production centers. At each facility a series of filling lines, the production process bottleneck, is available to produce a wide range of final items. These resources are usually dedicated to produce certain type of final products (e.g. kegs, glass bottles, cans), however, nowadays advanced manufacturing technology has allowed for the appearance of more flexible filling lines. This flexibility and the existence of parallel production resources forces the simultaneous planning of all or at least most part of the available filling lines in order to achieve more efficient plans.

Filling lines can only produce one product at a time, being adjusted to fill a certain liquid, container type and size, and final package. Whenever a product changeover is necessary it may require several mechanical adjustments in the filling line and cleansing operations. The number and complexity of the operations triggered depend on the previous and following products, i.e. setup time and costs are dependent on the production sequence. Tackling accurately these sequence-dependent setups is vital to ensure the competitiveness of the company, as frequent and long setups can reduce substantially the operational efficiency of filling lines. Additionally, recent market dynamics led companies to increase the number of products and work with less stock, by delivering products more frequently, which further stresses the need for efficient plans due to the need for extra setups.

Production planning in the case study is a hierarchical process with several echelons, each containing different aims and planning horizons. In particular, the medium term planning focuses on defining a plan for the next 6/8 weeks and serves as an input to the short term planning that defines the schedule of operations for the next week. Mid term planning decisions consider the sizing of filling lines production lots and overtime utilization (whenever possible). Short term decisions look only at a filling line at a time and try to schedule the production lots defined in the previous level with the objective of minimizing the sequence-dependent setup times and the tardinesses.

The company uses SAP APO, an advanced planning system (APS), to perform its supply chain planning tasks. APSs are modular planning software which extract data from traditional enterprise resource planning (ERP) systems and support decision making by using pre-defined mathematical models, heuristics and other quantitative techniques, before feeding back the ERP with the final solutions for their execution (see Fleischmann and Meyr (2003)). Modules architecture is aligned with an hierarchical decision system and with the supply chain functions defined by the supply chain planning matrix (Fleischmann et al. (2008)). Specifically for the case in study, the two planning levels discussed above are tackled in two different modules. The medium term decisions are made in the Supply Network Planning (SNP) module, while the short term planning is carried out using the Production Planning and Detail Scheduling (PP/DS) module. We describe next the modules as installed in the company at the time this paper was written.

SNP delivers a weekly-bucket oriented plan to define the production lot sizes on the series of available filling lines. The algorithm in use is an adapted version of the Capable-

To-Match (CTM) heuristic of the SAP APO. It considers *a priori* a preferential filling line for every product to each the forecasted demand is assigned, usually the filling line with the fastest processing time. When the capacity of a filling line is exceeded at a given time period the allocated productions are moved according to pre-defined rules. These rules may imply moving production quantities to earlier periods or to another filling line capable of producing the product and with a capacity surplus. Overtime allocation is managed by exception. During the creation of the lot sizes the sequence-dependent setup times are not considered, being included in the average processing times used.

The PP/DS receives the lots planned in SNP as an input and tries to sequence them while minimizing the sequence-dependent setup times. This step is performed by a dispatching rule, which sequences products by grouping them according to their features and treating time as continuous. The heuristic is executed considering one filling line at a time and just for the first week of the medium term planning horizon. The detailed scheduled is then sent to the SAP ERP to be executed.

A planning step consists of the execution of the SNP followed by the PP/DS. The medium term plan is updated as new demand forecasts become available and the planning horizon is rolled forward one week. It is common for managers to manually change the SNP plans before sending them to PP/DS and changes to the PP/DS are also made before the plans implementation.

This type of hierarchical approach has been proved to give not only suboptimal solutions in the presence of sequence-dependent setups, but also to pose challenges in terms of solution feasibility. In Pinedo and Kreipl (2004) a similar framework is developed for a brewery company, but the authors highlight that "the results coming out of the detailed scheduling problem may be, for various reasons, not acceptable.". On one hand, the average processing times in the medium term planning lead to overestimation of the capacity, which results in unfeasible plans and forces the medium term planning to be run again. On the other hand, underestimation can result in low efficient plans. The same problem has been reported in Mateus et al. (2010) in an iterative approach to the integrated lot sizing and sequence dependent setup scheduling. Therefore, it is imperative to tackle these two levels of decision simultaneously to correctly evaluate capacity and determine the lot sizes. A model considering the lot sizing and sequencing decisions over the medium term horizon would address this problem. However, such type of models are known to pose hard optimization challenges and the fact that only the first period or periods are actually implemented questions the reasoning of expanding the sequencing decisions over such a longer horizon. Furthermore, forecasts for more distant periods are likely to change once the horizon is rolled forward and managers often revise the plans to cope with possible machine breakdowns, stock-outs, last minute imperative orders, and other disruptive events. This re-planning can occur more than once a day leaving few time for optimizing complex models and stressing the need for a computational efficient method of generating plans under these extreme conditions.

The objective of this paper is to respond to these requirements by developing new formulations and methods that link short and medium term production planning in a single facility/multi-machine environment, while being computational efficient for their use under rolling planning and event-based planning and re-planning. Figure 4.1 depicts the differ-

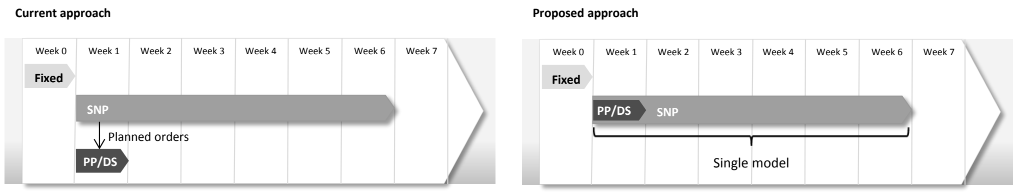


Figure 4.1: Medium and short term planning horizons

ences between the current approach followed by the company and the proposed approach.

To materialize our objectives the idea is to use approximate models that create highly detailed plans in the beginning of the planning horizon and make an estimation of future capacity utilization and costs in later periods. Such models are computationally less expensive than models integrating lot sizing and scheduling with sequencing for the entire medium term horizon enabling a fast generation and re-generation of plans as required in practice. Nonetheless, the quality of the solutions implemented in the detailed horizon is affected by the quality of the approximations made in the remaining horizon. We study the effect of different approximations and develop an iterative method aiming to improve the accuracy of the parameters used for the approximations.

Our contributions are as follows. We develop new formulations to use in rolling planning considering lot sizing and scheduling with sequencing decisions. These new formulations use a lot sizing and scheduling model with sequencing decisions that does not require the estimation of a maximum number of setups as the ones previously found in the literature. We also introduce a new approximate model that uses the information on setup time expenditure in detailed time periods to reduce the capacity in the simplified horizon. On the methodology front, an important contribution is the iterative method used to improve the accuracy of the parameters defined in the context of approximate models. The method is modular and can be applied in several distinct model formulations as shown here.

In the remainder of the paper we start by reviewing, in Section 4.2, the most important literature in the context of the current work. In Section 4.3 the complete lot sizing and scheduling model with sequencing decisions linking short and medium term is introduced. Based on this model, Section 4.4 presents the approximate models to be used in the rolling planning and Section 4.5 introduces the new iterative method to estimate the parameters required in the approximate models. In Section 4.6, the several alternatives discussed are tested on instances generated to simulate the problems arising in the case study. We finish by withdrawing conclusions from the conducted work and by setting potential future work.

4.2. Literature review

Much of the recent research dedicated to the integration of lot sizing and scheduling problems has focused on improving the plans detail. This has been shown by several authors (Gopalakrishnan et al. (2001); Porkka et al. (2003); Haase (1996)) to improve the utilization of the capacity and reduce costs. Many real-world problems inspired models extensions such as: parallel machines, setup carry-over, sequence-dependent setup times and costs, setup crossover, among others. On one hand, these extensions turn the mathematical models more realistic and increase their potential applicability by allowing for better and more realistic plans. On the other hand, models become much more complex due to the expansion in the number of variables and constraints to capture these features, leading to hard optimization challenges and solution times that may not be aligned with business practice. Formulations integrating sequencing decisions in lot sizing have been improved over the recent years and many solution approaches have been proposed. Among the papers introducing solution approaches we highlight Kang et al. (1999); Meyr (2002); Kovács et al. (2009); James and Almada-Lobo (2011); Guimarães et al. (2013b). Despite these improvements, problems with relevant size to real practice still pose difficulties in their solution.

As seen before, business practice establishes that these improved models and solutions are likely to be applied in a rolling horizon basis. Only the first part of the plan is actually implemented, corresponding to the initial time periods. The remaining part serves the purposes of estimating future costs and capacity shortages, in order to account for their impact on the nearer decisions. Having this premise in mind, some authors have incorporated the principles behind rolling planning on lot sizing and scheduling using two distinct approaches. One explores the idea of an *internal* rolling scheduling using the implicit time decomposition of the rolling planning to be able to efficiently handle large instances. The other focuses on the *external* rolling horizon defined by the successive planning steps to develop efficient mathematical formulations that trade-off the plans detail and the computational effort. These two approaches share a common time structure partitioning the planning horizon into three parts: fixed, detailed and simplified horizons. The fixed horizon is associated with previous iterations of the method or previous planning steps. The detailed horizon embeds at least all the time periods to be implemented in the next iteration and uses an accurate model to express the problem. On the other hand, the simplified horizon uses an approximation of the exact model. Figure 4.2 depicts the difference between the horizons in the two approaches.

Table 4.1 summarizes the main features of the most relevant work in the context of the current paper, all models consider a single facility. Several criteria are used to classify the papers, namely the production environment, the types of setups, the strategy considered to address feasibility issues of the plans, models objective function, the simplification strategy used in the simplified horizon and the type of rolling horizon approach.

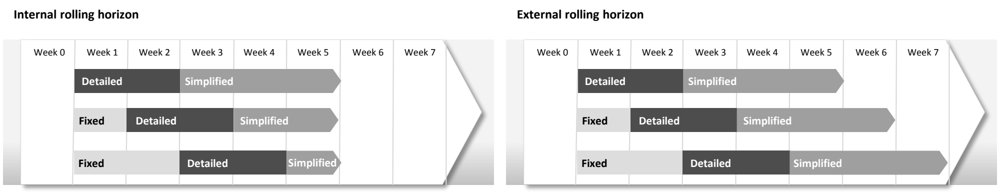


Figure 4.2: Internal and external rolling horizons

Table 4.1: Summury of related research

Reference	Planning environment: General*	Setups	Feasibility	Modelling: Objective Function**	Simplification Strategy	Rolling horizon type
Clark and Clark (2000)	SL, PM	Seq-Dep	Backlog	H, B	Increase processing times	Internal and External
Clark (2003)	ML, PM	Seq-Dep	Backlog	H, B	Increase processing times	Internal
Stadtler (2003)	ML, MR	Seq-Ind	Overtime	H, S, O	Capacity Reduction	Internal
Clark (2005)	SL, SM	Seq-Ind	Backlog	H, B	Increase processing times / Capacity reduction	Internal and External
Araujo et al. (2007)	SL, SM	Seq-Dep	Backlog	H, B, S	Sequence Independent Setups	Internal and External
Tiacci and Saetta (2012)	SL, SM	Seq-Dep	-	H, S	Sequence Independent Setups	External
This paper	SL, PM	Seq-Dep	Overtime	H, S, O	Increase processing times / Capacity reduction / Sequence Independent Setups	External

* SL - Single level, ML - Multi level, SM - Single machine, PM - Parallel machines, MR - Multi resource
** H - Holding costs, B - Backlog costs, S - Setup costs, O - Overtime costs

Clark and Clark (2000) propose a mathematical formulation to be used in an external rolling horizon approach to solve a parallel machine lot sizing and scheduling problem with sequencing decisions. The idea is to ignore binary setup decisions in the simplified horizon, either by relaxation or through an increase in the values of the processing times to incorporate the loss in setup times. Updating the processing times is non-trivial due to the simultaneous effect of the sequence-dependent setup times and lot sizes. Different approximations are proposed and compared by using a static instance as well as by means of a simulation of an external rolling horizon planning. Despite tackling a different problem, a subsequent paper (Clark (2003)) revises the method for estimating the corrected processing times by building an MIP model to estimate the increase. These approximations are also combined with an internal rolling scheduling based on a modified version of the relax-and-fix heuristic to solve the yet difficult detailed scheduling. However, their performance was not tested on an external rolling horizon environment. Clark (2005) further extends the approximation method, the rolling heuristics and conducts a deeper computational study using instances both with perfectly-known demand and including forecast errors during the rolling planning. Besides suggesting new methods to update the processing times, the author also investigates approximation schemes in which the processing times are kept, but capacity is reduced accordingly to the estimated loss in the setup times. In contrast to the two previous work and this paper, the setup times incurred are not dependent on the sequence of production.

An internal rolling horizon approach is designed in Stadtler (2003) for the multilevel lot sizing problem with setup times and multiple constrained resources. The idea is to limit the number of detailed periods to be able to use a tighten model formulation. The detailed window is then deployed (or partially deployed) and internally rolled forward until a complete solution is available for the original planning horizon.

The general lot sizing and scheduling problem (GLSP) is modified in Araujo et al. (2007) in the spirit of rolling planning. The idea is to schedule products in the simplified horizon, but ignoring both setup times and cost. The authors also develop local search heuristics to solve the problem of the detailed horizon with realistic data within acceptable computational time. Later, Tiacci and Saetta (2012) extended this approach by considering sequence-independent setup costs in the simplified horizon and proposing a formula for their estimation. The approach is validated by conducting experiments on data generated based on a case study of a company from the wood floors sector.

The formulations presented in this paper are, to the best of our knowledge, the first that apply rolling horizon principles using lot sizing and scheduling models which do not require the estimation of a maximum number of setups per period. We also propose an approximation model that reduces future capacity based on the setups witnessed in detailed time periods. A distinguishing element of our work is the idea to improve the accuracy of the parameters by incorporating additional information into the estimations on an iterative basis.

4.3. The complete lot sizing and scheduling model

The problem that considers the integration of sequencing decisions in the lot sizing and scheduling problem is known in the literature as the CLSD, which is an extension of the original Capacitated Lot Sizing Problem (CLSP). The objective is to minimize the total expenditure in inventory, setup and overtime costs over a finite planning horizon T. Plans define simultaneously for every time period the production quantities and sequences for N products on a set of parallel capacitated machines. This problem corresponds to tackle SNP and PP/DS decisions all together. Like in the APO modules, demand is assumed to be known from forecasts and is to be met without backlog, which is a common setting in the industry. Overtime can be used to face the potential capacity shortages. In the context of this work, overtime refers to the use of additional days or shifts that are still available, e.g. non-working days (saturdays, sundays or holidays) or a third shift in a production line operating on a two shift schedule. Hence, both the extra capacity and costs assume discrete values in opposition to most literature that considers these decisions as continuous (Özdamar and Birbil (1998); Stadtler (2003)). Since production lines have dedicated crews it is possible to schedule extra time independently.

Sequencing decisions are introduced since both the setup times and costs are dependent on the production sequence. In production lines the setup state is preserved over adjacent periods and also over idle periods of the machines. For operational reasons changeovers are not allowed to overlap between periods forcing them to start and end within a single time period. Due to technological constrains each production line can only produce a subset of all products.

To formulate the CLSD problem with parallel machines and overtime decisions, hereafter called (CSP), consider the following parameters:

Sets and indices

i, j products, $i, j = 1, \ldots, N$.

t time periods, $t = 1, \ldots, T$.

m machines (production lines), $m = 1, \ldots, M$.

o overtime types, $o = 1, \ldots, O$.

$\mathcal{A}_m$ set of products that can be produced on machine m

Data

d_{it} demand of product i in period t (units)

h_{it} holding cost of one unit of product i in period t

q_{imt} maximum number of production lots of product i on machine m in period t

r^o_{mt} cost of overtime type o on machine m in period t

cap_{mt} normal capacity of machine m in period t (time)

cap^o_{mt} overtime capacity of type o on machine m in period t (time)

p_{im} processing time of product i on machine m

b_{imt} upper bound on production quantity of product i on machine m in period t

st_{ijm} time required to perform a changeover from product i to product j on machine m

sc_{ijm} cost incurred when performing a changeover from product i to product j on machine m

The decision variables to be optimized are:

I_{it} stock of product i at the end of period t

B^o_{mt} (=1) if overtime capacity type o is used on machine m in period t

X_{imt} quantity of product i to be produced on machine m in period t

Z_{imt} (=1) if machine m is set up for product i at the beginning of period t

T_{ijmt} (=1) if a changeover from product i to product j is performed on machine m in period t

The mixed integer mathematical formulation (MIP) for the CSP reads:

$$\text{CSP} \quad \min \sum_{i,t} h_{it} \cdot I_{it} + \sum_{m,t,i,j \in \mathcal{A}_m} sc_{ijm} \cdot T_{ijmt} + \sum_{o,m,t} r^o_{mt} \cdot B^o_{mt} \tag{4.1}$$

$$\text{s.t.} \quad I_{i,t-1} + \sum_{m | i \in \mathcal{A}_m} X_{imt} = d_{it} + I_{it} \qquad \forall\, i, t, \tag{4.2}$$

$$\sum_{i \in \mathcal{A}_m} p_{im} \cdot X_{imt} + \sum_{i,j \in \mathcal{A}_m} st_{ijm} \cdot T_{ijmt} \leq cap_{mt} + \sum_{o} B^o_{mt} \cdot cap^o_{mt} \qquad \forall\, m, t, \tag{4.3}$$

$$X_{imt} \leq b_{imt} \cdot \left(\sum_{j \in \mathcal{A}_m} T_{jimt} + Z_{imt} \right) \qquad \forall m, i \in \mathcal{A}_m, t, \tag{4.4}$$

$$\sum_{i \in \mathcal{A}_m} Z_{imt} = 1 \qquad \forall m, t, \tag{4.5}$$

$$Z_{imt} + \sum_{j \in \mathcal{A}_m} T_{jimt} = \sum_{j \in \mathcal{A}_m} T_{ijmt} + Z_{im,t+1} \qquad \forall m, i \in \mathcal{A}_m, t, \tag{4.6}$$

$$\{(i,j) : T_{ijmt} > 0\} \text{does not include disconnected subtours} \qquad \forall m, t. \tag{4.7}$$

$$X, I \geq 0, \quad Z \in \{0,1\}, \quad T_{jimt} \in \{0, \ldots, q_{imt}\}, \quad B \in \{0,1\}. \tag{4.8}$$

The objective function (4.1) minimizes the sum of holding, setup and overtime costs. Without loss of generality we assume that production costs in the different machines are fixed and time independent. The demand balancing constraints are described by (4.2). Production time plus the time lost in setup operations should not exceed the available normal capacity incremented by the overtime decisions on each machine (4.3). Constraints (4.4) link production quantities with the machine setup state: production may only occur if a setup is carried over from the previous period or at least one setup is performed in the period. Constraints (4.5) ensure that the machine is set up for a single product in the beginning of each time period, while (4.6) keep trace of each machine configuration balancing the flow of setups as follows. If there are no setups in period t the machine configuration is carried to period $t+1$. On the other hand, for each product i three cases may appear: (i) more input than output setups, (ii) more output than input setups and (iii) equal number of input and output setups. In the first case the machine has to be set up for product i in the beginning of the next period $t+1$ ($Z_{im,t+1} = 1$). The opposite scenario, the second case, forces a setup for product i to be carried over from the previous period ($Z_{imt} = 1$). The third case happens when the product is neither the first nor the last in the sequence, or it is not part of the production sequence of the machine in the period.

Constraints (4.7) prevent disconnected subtours, i.e. sequences that start and end at the same setup state. Guimarães et al. (2013a) have shown in their study that model efficiency is directly linked to the selection of the proper subtour elimination constraints. We choose to use single commodity flow type constraints since as suggested by the results of that study, they give origin to a very computational efficient model which is able to provide feasible integer solution even for large instances of the single machine variant.

Let us introduce the binary setup state variables Y_{imt} which equal one, if machine m is prepared to produce product i in period t, or zero otherwise. We consider the machine to be prepared for product i, in case either a setup in the period exists or a setup is carried over from a previous period. The following constraints (4.9)-(4.10) link variables Y_{imt} to both Z_{imt} and T_{ijmt}, while (4.11) replace (4.4) to give a tighter formulation.

$$\sum_{j \in \mathcal{A}_m} T_{jimt} + Z_{imt} \geq Y_{imt} \qquad \forall m, i \in \mathcal{A}_m, t, \tag{4.9}$$

$$\sum_{j \in \mathcal{A}_m} T_{jimt} + Z_{imt} \leq q_{imt} \cdot Y_{imt} \qquad \forall m, i \in \mathcal{A}_m, t, \tag{4.10}$$

$$X_{imt} \leq b_{imt} \cdot Y_{imt} \qquad \forall m, i \in \mathcal{A}_m, t. \tag{4.11}$$

Additionally, we also introduce the continuous flow variables F_{ijmt}, which capture the commodity flow from product (node) i to product (node) j on machine m in period t. In order to model setup carryover an artificial node indexed by 0 is introduced and acts as the source of the flow. This formulation prevents disconnected subtours by imposing the connectivity of the graph induced by the setups selected through the changeover variables (T_{ijmt}). It forces the existence of a path connecting the artificial node to each one of the products in the sequence (unitary Y_{imt}). The constraints (4.7) are then defined as below:

$$\sum_{j \in \mathcal{A}_m} F_{0jmt} = \sum_{j \in \mathcal{A}_m} Y_{jmt} \qquad \forall m, t, \tag{4.12}$$

$$\sum_{j \in \mathcal{A}_m \cup \{0\}} F_{jimt} - \sum_{j \in \mathcal{A}_m} F_{ijmt} = Y_{imt} \qquad \forall m, i \in \mathcal{A}_m, t, \tag{4.13}$$

$$F_{0imt} \leq |\mathcal{A}_m| \cdot Z_{imt} \qquad \forall m, i \in \mathcal{A}_m, t, \tag{4.14}$$

$$F_{ijmt} \leq |\mathcal{A}_m| \cdot T_{ijmt} \qquad \forall m, i \in \mathcal{A}_m, j, t. \tag{4.15}$$

The amount of commodity flow forced to leave the source is defined by the number of paths needed, which is equivalent to the number of products produced in the time period (4.12). The flow balance constraints are expressed by (4.13) which ensure that a unitary flow is sent to every selected node, corresponding to a path from the source to every product being produced in the time period. A positive setup state acts as a unitary demand of the flow. Both (4.14) and (4.15) impose an upper bound on the amount of flow traversing the arcs. By (4.14), the flow can only leave the source to the first product in the sequence, while (4.15) guarantees that the flow only transverses arcs in the current solution.

As shown in Guimarães et al. (2013a), although the model has a good performance on instances of moderate large size, it is evident that in case the number of products and/or machines increases, its performance is expected to quickly deteriorate. Moreover, as discussed in the previous section, the need for an exact solution for later periods is questionable on the perspective of a rolling horizon planning. As such and aligned to the frequent re-planning performed by managers, the next sections discuss approximations of this model that can result in less computational expensive formulations solvable in a reasonable time for real-world practice.

4.4. The rolling horizon models

Aiming to reduce the complexity of model CSP, we consider a time-oriented partition of the planning horizon into two sections: scheduled and unscheduled. The scheduled horizon is composed by the first T_s periods of the original horizon $t = 1, \ldots, T_s$, while the

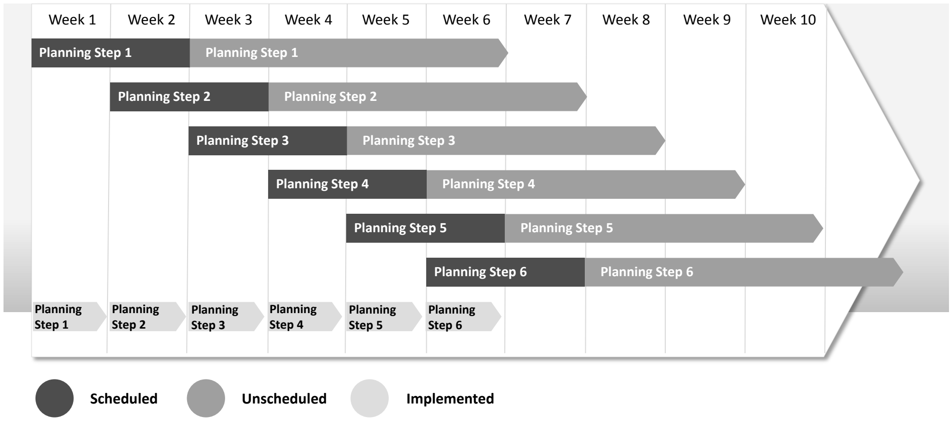


Figure 4.3: Rolling planning

remaining periods define the unscheduled horizon $t = T_s + 1, \ldots, T$. At each planning step the scheduled horizon consists of the model presented in Section 4.3 (CSP), while for the unscheduled horizon the model is replaced by a simplified version. The idea is to save computational time by creating less detailed plans for future periods, but still having an approximation of the future expenses in terms of cost and preventing future capacity bottlenecks. Hereafter, we will call rolling model the formulation containing CSP on the scheduled horizon and an approximation on the unscheduled horizon.

Considering that planning steps occur with a frequency Δ, only the first Δ periods of the plan are actually implemented. The size of the scheduled horizon is, therefore, forced to be $T_s \geq \Delta$. Defining T_s greater than Δ gives origin to an overlap ($\Phi = T_s - \Delta$) of the consecutive scheduled horizons, in which decisions are reconsidered. This can translate into better plans for the short term as the approximation improves, since the decisions in the Δ horizon consider additional information given by the more detailed solutions in the overlapping time periods. However, extending too much the overlap can have a significant negative impact on the efficiency of the rolling model. Figure 4.3 depicts an example of the application of the rolling model in 6 consecutive planning steps. The planning horizon is divided into weeks with a total of 6 weeks to be considered at each planning step ($T = 6$). The scheduled horizon is composed by the first 2 weeks (T_s=2) and only the first week is implemented at each planning step ($\Delta = 1$), hence we have 1 week of overlap in the detailed horizon.

Next we detail the different approximations that can be used in the unscheduled horizon. In this context, we also revise some of the approaches followed by the works highlighted in Section 4.2.

4.4.1 Linear relaxation

The most straightforward model to apply in the unscheduled horizon is the linear relaxation of CSP over $t = T_s + 1, \ldots, T$. Hence, variables Z_{imt}, T_{ijmt} and Y_{imt} are redefined as:

$$Z \in \{0,1\}, \quad T_{jimt} \in \{0,\ldots,q_{imt}\}, \quad Y \in \{0,1\}, \quad t = 1,\ldots,T_s, \tag{4.16}$$

$$0 \le Z \le 1, \quad 0 \le T_{jimt} \le q_{imt}, \quad 0 \le Y \le 1, \quad t = T_s+1,\ldots,T. \tag{4.17}$$

The resulting rolling model is denoted as CSP_{rel}. Note that the relaxation over the unscheduled horizon still approximates the future costs and capacity constraints. This approximation has already been suggested in Clark and Clark (2000).

Applying this approach on a successive series of planning steps is similar to the relax-and-fix heuristic, also known as fix-and-relax (Dillenberger et al. (1994); Pochet and Wolsey (2006); Federgruen et al. (2007)). Likewise, this constructive heuristic solves a series of partially relaxed MIP subproblems to construct an initial feasible solution to an original MIP. It starts from the first period of the planning horizon and progressively moves forward fixing the integer variables at their optimal value obtained in previous iterations.

The main difference to the heuristic lies on the planning horizon definition. Successive planning steps always define a horizon of T periods: as the first Δ periods are fixed the horizon is expanded by appending periods $t = T+1,\ldots,t+\Delta$. On the other end, in relax-and-fix the initial horizon is never expanded, hence after the fix step the planning horizon is reduced (in case all decisions are fixed in Δ) corresponding to a reduction of the problem size.

Thus, the relax-and-fix heuristic can be seen as an internal rolling horizon approach to solve the original model CSP at each planning step. Hence, it is also valid that the following approximations can be applied in an internal rolling horizon approach to solve instances of CSP (by replacing CSP_{rel}), widening the applicability of the new results proposed here.

4.4.2 Increase processing times

Although model CSP_{rel} results in a less computational demanding model, its size can be further reduced. Clark and Clark (2000); Clark (2003) and Clark (2005) explore the idea of completely ignoring setup decisions in the unscheduled horizon, by increasing processing times to include the loss in setup times. Consider $\widehat{p}_{im}$ to be the increased processing times, the following constraints replace (4.3) for $t = T_s+1,\ldots,T$:

$$\sum_{i \in \mathcal{A}_m} \widehat{p}_{im} \cdot X_{imt} \le cap_{mt} + \sum_{o} B^o_{mt} \cdot cap^o_{mt} \qquad \forall m, t = T_s+1,\ldots,T. \tag{4.18}$$

Note that (4.3)-(4.15) only apply to the scheduled horizon and the objective function (4.1) also needs to be rewritten to consider setup costs only up to T_s.

The question raised by this approximation is how to increase the processing times to include setup times. A simple technique is to do nothing and set $\widehat{p}_{im}$ equal to p_{im} (Clark (2005)), this will be referred as model CSP_p. Alternatively, the loss in setups can be included assuming a lot-for-lot production policy (Clark and Clark (2000)) via:

$$\widehat{p}_{im} = \frac{\overline{st}_{im} + p_{im} \cdot \overline{d}_i}{\overline{d}_i}, \qquad \forall\, i, m \tag{4.19}$$

where $\overline{d}_i$ is the average period demand for product i and $\overline{st}_{im}$ is the average time to set up machine m to product i. Average setup times are calculated considering setup times incurred when changing to product i and from product i to account for a non-symmetric setup matrix. This approximation to the unscheduled horizon originates rolling model CSP$_p^f$.

CSP$_p$ overestimates capacity in the unscheduled horizon by completely ignoring future losses due to setups. On the other hand, CSP$_p^f$ may overestimate and/or underestimate capacity. Overestimations are caused by less frequent production lots in comparison to lot-for-lot and also by changeovers below average, while underestimations result from the production of smaller quantities and/or larger setups when compared to the average.

4.4.3 Reduce available capacity

An alternative to model capacity loss due to setup times when sequencing decisions are eliminated in the unscheduled horizon is to explicitly reduce the capacity by introducing the parameter ST_{mt} defined as the estimated setup time expenditure on machine m in period t. The new capacity constraints are introduced in the previous model with no setup time loss and no increasing processing times (CSP$_p$) as follows:

$$\sum_{i \in \mathcal{A}_m} p_{im} \cdot X_{imt} + ST_{mt} \leq cap_t + \sum_{o} B_{mt}^{o} \cdot cap_{mt}^{o} \qquad \forall\, m, t \geq T_s + 1. \tag{4.20}$$

Again the onus of this model is the estimation of ST_{mt}. Setting it to zero would result in CSP$_p$ again. In his work, Stadtler (2003) proposes to estimate future capacity loss based on the losses observed in the setup decisions fixed in previous periods by his internal rolling heuristic. It is suggested to take the setup time loss mean or to increase the mean of setup time by the absolute deviation average multiplied by a safety factor. To follow a similar reasoning in an external rolling planning, let us transform ST_{mt} into a decision variable. The following constraints are therefore possible estimations for future capacity loss:

$$ST_{mt} \geq \begin{cases} CSP_{st}^{mean}: & \sum_{l \leq T_s} \frac{\sum_{i,j \in \mathcal{A}_m} st_{ijm} \cdot T_{ijml}}{T_s} \qquad \forall\, m, t > T_s, & \text{(4.21a)} \\ CSP_{st}^{max}: & \sum_{i,j \in \mathcal{A}_m} st_{ijm} \cdot T_{ijml} \qquad \forall\, m, l \leq T_s, t > T_s. & \text{(4.21b)} \end{cases}$$

Model CSP$_{st}^{mean}$ estimates future capacity loss by computing the average setup time per machine in the scheduled time periods (4.21a), while model CSP$_{st}^{max}$ uses the maximum setup loss in the scheduled horizon to reduce capacity in future time periods (4.21b).

4.4.4 Sequence independent setups

A downturn of models CSP_p, CSP_p^f, CSP_{st}^{mean} and CSP_{st}^{max} is that setup costs are neglected over the scheduled horizon and there is no penalty neither for a high number of setups on a given machine and time period, nor for small sized lots. A modeling technique with potential to overcome these problems is to consider sequence independent setups in the unscheduled horizon, by using the setup binary variables previously defined (Y_{imt}).

$$\sum_{i \in \mathcal{A}_m} p_{im} \cdot X_{imt} + \sum_{i \in \mathcal{A}_m} \widehat{st}_{im} \cdot Y_{imt} \leq cap_t + \sum_{o} B^o_{mt} \cdot cap^o_{mt} \qquad \forall m, t. \tag{4.22}$$

Constraints (4.22) replace the capacity constraints of model CSP_p and account for future setup times. Constraints (4.11) are now applied to the whole planning horizon and the term $\sum_{t,m,i \in \mathcal{A}_m} \widehat{sc}_{im} \cdot Y_{imt}$ is added to the objective function to estimate future setup costs. This rolling model is denoted as CSP_{ind} and is closely related to the models presented by Araujo et al. (2007) and Tiacci and Saetta (2012), although the first did not account for setup costs nor the latter for setup times. Values for the sequence-independent parameter $\widehat{st}_{im}$ can be estimated using mean setup time $\overline{st}_{im}$ introduced to incorporate setup times in processing times. The same formula is valid for $\widehat{sc}_{im}$. An alternative approximation can be obtained by the following expression (Tiacci and Saetta (2012)):

$$\widehat{sc}_{im} = \frac{2/(|\mathcal{A}_m| - 1)}{\sum_j (1/sc_{ijm}) + \sum_j (1/sc_{jim})} \tag{4.23}$$

which tries to factor the fact that setup costs are part of the objective function and, therefore, low values of sc_{ijm} are more likely to be used than large ones. The model considering the use of sequence independent setup parameters estimated by (4.23) will be referred as CSP^f_{ind}.

4.5. An iterative method

The quality of the solutions obtained with the rolling models presented in Sections 4.4.2 to 4.4.4 is heavily dependent on the estimation accuracy of the approximate parameters. As pinpointed in Stadtler (2003) when incorporating losses in the capacity due to setups, overestimation can cause too many setups in the scheduled horizon, while underestimation might result in finding no feasible solution at all. Dealing with sequence dependencies further stresses the difficulty of this assessment.

The previous methods presented have in common the fact that the estimation considers average values or is based on values observed in the past which implies that information available in the unscheduled horizon is totally or partially neglected. We propose to improve the estimations accuracy by adapting the approximation of the parameters through an iterative method. The idea is to use the rolling model solution for the unscheduled horizon to refine the estimations.

Suppose our rolling model requires the estimation of the approximate parameter u. Our iterative method considers the following steps at each planning step:

1. Initialize approximate parameters
2. Solve the rolling model
3. Compute the current values of the approximate parameters
4. Update the estimation of the approximate parameters
5. Repeat 2. to 4. until stopping criteria

Let r identify the current iteration of the algorithm and u^r the estimation for the approximate parameter available at the end of iteration r (u^0 represents the initial estimation). Our algorithm works as follows. At the beginning of each iteration r the rolling model is solved considering the estimation u^{r-1}. The current values of u^* are computed based on the current solution of the rolling model. Then they are used to update u^r the values for the next iteration. These steps are repeated until the stopping criteria is met.

Initialize approximate parameters. The initial values of u can be set by any of the techniques introduced in the previous section, e.g. for the estimation of the processing times factoring setup time loss both the approaches in CSP_p and CSP_p^f are possible initializations for our method. For models CSP_{st}^{mean} and CSP_{st}^{max} the first approximation is given by (4.21) which is latter updated as explained next.

Compute the current values of the approximate parameters To compute the values of the approximate parameters it is required to calculate the estimated loss in setups (time and cost) during the unscheduled horizon suggested by the current solution of the rolling model. For this purpose let $\dot{X}^r_{imt}$ define the production quantities defined by the rolling model in iteration r for product i on machine m in periods $t = T_s + 1, \ldots, T$. By definition the current setups in the unscheduled horizon are:

$$\dot{Y}^r_{imt} = \begin{cases} 1, & \text{if } \dot{X}^r_{imt} > 0 \quad \forall m, i \in \mathcal{A}_m, t > T_s, & (4.24a) \\ 0, & \text{if } \dot{X}^r_{imt} = 0 \quad \forall m, i \in \mathcal{A}_m, t > T_s. & (4.24b) \end{cases}$$

The non-zero $\dot{Y}^r_{imt}$ determine the products to be sequenced in each period of the unscheduled horizon. The potential sequences can be obtained by solving a Sequential Ordering Problem (SOP) for each machine (see Escudero (1988)). The SOP is a problem related to the Asymmetric Traveling Salesman Problem (ATSP), but where precedence relations between nodes exist, hence it is also called Precedence Constrained Asymmetric Traveling Salesman Problem (PCATSP). A solution to the SOP is a tour passing by all nodes and respecting the precedence constraints. In this context nodes correspond to the non-zero $\dot{Y}^r_{imt}$, arcs are the possible setups between the products to be sequenced and the precedence constraints define that all setups of period t must precede setups of the following period $t \in \{t+1, \dots, T\}$.

We can define the SOP arising in this context in graph theoretical terms as follows. A complete directed graph $D = (V, A)$ is given, being V the set of nodes and $A = (\bar{i}, \bar{j}) | \bar{i}, \bar{j} \in V$ the set of arcs. When traversing an arc $(\bar{i}, \bar{j}) \in A$ a cost $c_{\bar{i}\bar{j}} \geq 0$ is incurred. An artificial node 0 is introduced in V and is by definition the first and last node in the solution-tour. The precedences are given by a digraph $P = (V, R)$, defined on the same node set V but introducing an additional arc set R. An arc $(\bar{i}, \bar{j}) \in R$ represents a precedence relation, i.e. node $\bar{i}$ has to precede node $\bar{j}$ for a tour to be feasible. The precedence digraph P must be acyclic in order for a feasible solution to exist, and is also assumed to be transitively closed. The objective of the SOP is to find a feasible tour with the minimal total cost.

In our SOP the node set V can be decomposed into $T - T_s$ subsets, one for each period in the unscheduled horizon. Let V_t represent the subset of nodes to be scheduled in period $t \in \{T_s + 1, \dots, T\}$, composed by the products $\bar{i}$ such that $\dot{Y}^r_{imt} = 1$ and for the first and last periods the artificial node 0 is also appended. The arc set is defined by $A = \{(\bar{i}, \bar{j}) | \bar{i} \in V_t, \bar{j} \in V_t \cup V_{t+1} \; t = T_s + 1, \dots, T\}$, arcs between nodes in the same subset represent setups and arcs connecting nodes in adjacent subset model setup carryover. To model the problem as a MIP, let us define for each arc $(\bar{i}, \bar{j}) \in A$ a binary variable x_{ij} such that:

$$
x_{\bar{i}\bar{j}} = \begin{cases} 1, & \text{if arc } (\bar{i}, \bar{j}) \text{ is in the tour,} \quad (4.25a) \\ 0, & \text{otherwise.} \quad (4.25b) \end{cases}
$$

The MIP formulation for the SOP reads:

$$
\text{SOP} \quad \min \sum_{(\bar{i},\bar{j}) \in A} c_{\bar{i}\bar{j}} \cdot x_{\bar{i}\bar{j}} \tag{4.26}
$$

$$
\text{s.t.} \quad \sum_{\bar{j} \in V_t} x_{\bar{j}\bar{i}} = 1 \qquad \forall t = T_s + 1, \bar{i} \in V_t \tag{4.27}
$$

$$
\sum_{\bar{j} \in V_t \cup V_{t-1}} x_{\bar{j}\bar{i}} = 1 \qquad \forall t = T_s + 2, \dots, T, \bar{i} \in V_t \tag{4.28}
$$

$$
\sum_{\bar{j} \in V_t \cup V_{t+1}} x_{\bar{i}\bar{j}} = 1 \qquad \forall t = T_s + 1, \dots, T-1, \bar{i} \in V_t \tag{4.29}
$$

$$
\sum_{\bar{j} \in V_t} x_{\bar{i}\bar{j}} = 1 \qquad \forall t = T, \bar{i} \in V_t \tag{4.30}
$$

$$
\{(\bar{i}, \bar{j}) : x_{\bar{i}\bar{j}} = 1\} \text{does not contain subtours} \tag{4.31}
$$

$$
x_{\bar{i}\bar{j}} \in \{0, 1\} \qquad \forall (\bar{i}, \bar{j}) \in A \tag{4.32}
$$

Constraints (4.27)-(4.30) and (4.32) define the assignment problem relaxation of the SOP, and (4.31) can be written as the single commodity constraints proposed in Section 4.3.

Since the size of the SOP arising in the context of our study is considered small, we have chosen to solve it using a commercial solver of the MIP formulation just presented, although many heuristics are available if we want to speed up the solution process. The solution of the SOP together with $\dot{X}^r_{imt}$ and $\dot{Y}^r_{imt}$ can then be used to update the estimation of the parameters in the rolling models. Let $f(i,t)$ be a function that receives a product i of the original problem and transforms it to the corresponding node of V_t in our SOP problem. This allows us to capture a solution for the CSP as follows:

$$\dot{T}^r_{ijmt} = \begin{cases} x_{f(i,t),f(j,t)} & \text{if } \dot{Y}^r_{imt} \geq 1,\ \dot{Y}^r_{jmt} \geq 1 \text{ and } t > T_s, \quad (4.33a) \\ 0, & \text{otherwise.} \quad (4.33b) \end{cases}$$

$$\dot{Z}^r_{imt} = \begin{cases} \sum_{\bar{j} \in V_{t-1}} x_{\bar{j},f(i)} & \text{if } \dot{Y}^r_{imt} \geq 1 \text{ and } t > T_s + 1, \quad (4.34a) \\ 0, & \text{otherwise.} \quad (4.34b) \end{cases}$$

The current value for the approximate parameters is calculated as follows.

Increase processing times. To compute the current approximate processing times we distinguish between three cases: there is a setup on machine m for product i in period t (4.35a); there is at least one setup in the planning horizon on machine m for product i, but no setup in period t(4.35b); during the entire planning horizon machine m is never set up for product i. In the first case, the approximate processing time is given by the total time during period t machine m has been occupied to produce product i (setup time + total processing time) divided by the lot size. On the second case, the same idea applies but we use the total production times and quantities, and setup times over the entire planning horizon. Furthermore, in contrast to the previous case, we also impose that at least ϕ% of product's i total demand has to be produced on machine m. By doing so, we aim to prevent that small lot sizes in a given period have an effect in all the approximate processing times in the machine. Finally, if in the current solution no setup for product i occurs the previous estimation is carried on (4.35c).

$$\widehat{p}^*_{imt} = \begin{cases} \left(\sum_j \dot{T}^r_{jimt} \cdot st_{jimt} + p_{im}\dot{X}^r_{imt}\right) / \dot{X}^r_{imt}, & \text{if } \dot{Y}^r_{imt} \geq 1, \quad (4.35a) \\ \left(\sum_{t,j} \dot{T}^r_{jimt} \cdot st_{jimt} + p_{im} \cdot \sum_t \dot{X}^r_i\right) / \sum_t \dot{X}^r_i, & \text{if } \dot{Y}^r_{imt} = 0 \text{ and } \sum_t \dot{X}^r_{imt} \geq \phi \cdot \bar{d}_i, \quad (4.35b) \\ \widehat{p}^{r-1}_{imt}, & \text{otherwise.} \quad (4.35c) \end{cases}$$

Reduce available capacity. The current losses in setup time at each machine and period are given by:

$$ST^*_{mt} = \sum_{i,j} \dot{T}^r_{ijmt} \cdot st_{ijmt} \quad \forall m, t = T_s + 1, \ldots, T \tag{4.36}$$

Sequence independent setups. When computing the sequence independent parameters we also distinguish between the same three cases that appear for computing the approximate processing times. If a setup for product i on machine m in period t exists the current sequence independent setup times and costs are set by the solution of the SOP (4.37a). On the other hand, if no setup exists in the period, but at least one occurs during the planning horizon, we compute the sequence independent parameters as the average of the sequence dependent setups over the planning horizon (4.37b). When no setup takes place on machine m for product i in the entire planning horizon, then the previous estimations correspond to the current values to be considered (4.37c).

$$\widehat{st}^*_{im}(\widehat{sc}^*_{im}) = \begin{cases} \sum_j \dot{T}^r_{ijmt} \cdot st_{ijmt}(sc_{ijmt}), & \text{if } \sum_j \dot{T}^r_{ijmt} \geq 1, \quad (4.37a) \\ \dfrac{\sum_{t,j} \dot{T}^r_{ijmt} \cdot st_{ijmt}(sc_{ijmt})}{\sum_{t,j} \dot{T}^r_{ijmt}}, & \text{if } \sum_j \dot{T}^r_{ijmt} = 0 \text{ and} \\ & \sum_{t,j} \dot{T}^r_{ijmt} \geq 1, \quad (4.37b) \\ \widehat{st}^{r-1}_{im}(\widehat{sc}^{r-1}_{im}), & \text{otherwise.} \quad (4.37c) \end{cases}$$

Update the estimation of the approximate parameters. To update the estimations of the parameters to be used in the next iteration we apply an exponential smoothing technique. The idea is to reduce the effect of nervousness along iterations. Let us explain what we consider nervousness by giving an example. Consider that we were using model CSP$_{ind}$ and our initial estimation was given by (4.23). In the first iteration after solving CSP$_{ind}$ and when computing the current values for the sequence independent parameters imagine that the following happens $\widehat{st}^*_{im} \gg \widehat{st}^0_{im}$. This can point in two directions: either the initial estimation is of poor quality, or the best sequence of the products scheduled in the period corresponds to a high setup time for product i. In the first case we may want to quickly update our estimation, however the high sequence dependent setups may not occur when sequencing the products in the future, and are the result of a myopic schedule using sequence independent setups. The smoothing parameter α^{upd} weighs these two cases according to the following formula:

$$u^r = \alpha^{upd} \cdot u^* + (1 - \alpha^{upd}) \cdot u^{r-1}. \tag{4.38}$$

High values of α^{upd} give origin to a more responsive update, whereas low values of α^{upd} update slowly the estimations.

Stopping criteria. The stopping criteria of the algorithm is defined by calculating the solution's stability in consecutive iterations as a measure of the variation of the estimations. We assume that a stable solution for the rolling model is an indication of the stability of the estimations for the approximate parameters. Following the stability measures suggested in Kimms (1998), we introduce $\dot{q}^r_{im}$ as the weighted production quantities associated with product i on machine m in the solution of iteration r:

$$\dot{q}^r_{im} = \sum_t \xi_{it} \cdot X_{imt} \qquad \forall\, m, i \in \mathcal{A}, t. \tag{4.39}$$

The scores of ξ_{it} are defined to weigh more periods in the scheduled horizon since these correspond to the solutions which are to be implemented. A geometric decay function can serve this purpose ($\xi_{it} = t^{-\beta}$). In order to measure the stability over the iterations, we first measure the stability (sm^r_i) of the production plan defined to each product as follows:

$$sm^r_i = \frac{\sum_{m|i\in\mathcal{A}} \left|\dot{q}^r_{im} - \ddot{q}^{r-1}_{im}\right|}{\max\{\sum_{m|i\in\mathcal{A}} \dot{q}^r_{im}, 1\}} \qquad \forall\, i. \tag{4.40}$$

We compare the current weighted production quantities with the previous by using $\ddot{q}^r_{im}$ which incorporates the past solutions also relying on exponential smoothing method:

$$\ddot{q}^r_{im} = \alpha^{stab} \cdot \dot{q}^r_{im} + (1 - \alpha^{stab}) \cdot \ddot{q}^{r-1}_{im} \tag{4.41}$$

The overall solution stability corresponds to the average among all products: $SM^r = \frac{1}{N} \cdot \sum_i sm^r_i$. The algorithm stops iterating if $SM^r \leq \varepsilon$ and the current solution of the rolling model is implemented.

4.6. Computational Experiments

In this section we assess the performance of the rolling models and of the iterative approach. Two data sets are used to perform the comparison: (1) the first data set is composed of instances randomly generated, but designed to simulate the problem features arising in the case study described in Section 4.1; and (2) the second set corresponds to a collection of real-world instances available from the case study.

The motivation for creating random instances lies on the small number of real-world instances that we were able to collect, but also on the need to extend our analysis to problems with different production environments. Nevertheless, all the tests were designed assuming the same planning system described in Figure 4.3. Let the planning step be denoted by $k \in \{1, \ldots, K\}$, that when used as a superscript in the decision variables introduced before represents their value at the end of the planning step. The quality of the approximations is measured by evaluating the cost incurred from the partial solutions implemented on a rolling horizon basis. The evaluation function (F_1) is then given by:

$$F_1 = \sum_k \left[\sum_{t=1}^{\Delta} \left(\sum_i h_{it} \cdot I_{it}^k + \sum_{m,i,j \in \mathcal{A}_m} sc_{ijm} \cdot T_{ijmt}^k + \sum_{o,m} r_{mt}^o \cdot B_{mt}^{ok} \right) \right] \tag{4.42}$$

In this context each planning step k covers 6 weeks and there is a weekly re-planning frequency ($\Delta = 1$). Therefore, at the end of our simulation we will be comparing the solutions implemented for 6 consecutive weeks. We assume that at the end of each planning step new data becomes available for $t = T+1, \ldots, T+\Delta$ and that the data between $t = \Delta+1, \ldots, T$ does not suffer any changes, i.e. we assume zero forecast error. Although this is a strong assumption to be made for real cases, Clark (2005) pointed out that the relative performance of methods under the static and rolling horizon versions of the problem are comparable.

Table 4.2: Summary of the rolling horizon methods

Simplification Strategy	Acronym	Description
Linear relaxation	CSP_{rel}	linear relaxation of model CSP
Increase processing times	CSP_p	no setup times included
	CSP_p^f	estimation of $\widehat{p}_{im}$ by (4.19)
Reduce available capacity	CSP_{st}^{mean}	previous max loss (4.21a)
	CSP_{st}^{max}	previous mean loss (4.21b)
Sequence independent setups	CSP_{ind}^f	weighted setup times (costs) (4.23)
Iterative method	CSP_p^{iter}	Initial estimation given by CSP_p
	CSP_{st}^{iter}	Initial estimation given by CSP_{st}^{mean}
	CSP_{ind}^{iter}	Initial estimation given by the minimum setup time (cost) to each product

Table 4.2 summarizes the rolling approaches that will be compared. We exclude the use of the original CSP as in the preliminary tests conducted, it turned out to be extremely difficult to solve using a commercial solver. Most often there was not a feasible solution in the maximum running time allowed and when such solution existed the gap to optimality was too high suggesting a poor quality. All the computational experiments were conducted on Intel @ 2.40 gigahertz processing units limited to 4 gigabytes of random access memory and using the Linux operating system. The formulations and the iterative method were implemented in C++ using the ILOG Concert Technology and compiled with a gcc compiler. IBM ILOG Cplex 12.4 was used to solve the mixed integer programming models. A limit of 10 minutes is imposed to each planning step, which is considered to be reasonable by the case study managers.

4.6.1 Generated problem set

The random instances generated reflect the data properties found in the case study, which are common to many semi-continuous process industries. The instance design intends to test the influence of the following problem features in the performance of the different rolling models (Table 4.3): demand profile, setup variability and tightness of the capacity.

Table 4.3: Levels for the problem features

Problem feature	
Demand profile	Stationary ABC
Setup times variability	Low High
Tightness of capacity	Tight Very Tight

Demand Pattern. The 6 planning steps of 6 time periods each require the generation of demand forecasts for 11 time periods. To simulate the demand pattern in the beverage industry, we have created two profiles: *Stationary* and *ABC*. The realism is preserved by considering demand seasonality in both profiles. Demand is generated according to the following formula:

$$d_{it} = \mu_i \cdot s(t - \left\lfloor \frac{t}{c} \right\rfloor) + \sigma_i \cdot \delta_t \quad \forall i, t, \tag{4.43}$$

where μ_i is the average demand of product i, $s(t - \left\lfloor \frac{t}{c} \right\rfloor)$ is the seasonal effect on demand (c is the length of the seasonal cycle), σ_i is the standard deviation of product's i demand and δ_t is a normally distributed random variable. Note that $\lfloor\rfloor$ denotes the lower rounding value. Since the planning horizon is relatively short no trend is considered when computing the demand. Seasonality accounts for the end of month effect, which is defined by an increase of sales in the final weeks of each month, therefore in our tests $c = 4$. Because this effect is more evident in some products than others we define three functions for s: no effect $\{1.0, 1.0, 1.0, 1.0\}$, moderate effect $\{0.8, 0.8, 1.1, 1.3\}$ and high effect $\{0.6, 0.8, 1.2, 1.4\}$. The demand profiles affect the mean and standard deviation of products as shown in Table 4.4. In the stationary demand profile all products have similar values for the mean and standard deviation (in this case only medium volume products were considered), while ABC profile presents high variability in the demand pattern of products (all four product types are considered). Furthermore, we also introduce in ABC products with periodic demand, being time between orders (TBO) the average number of weeks between periods with non-null demand.

Setup times variability. We introduce different types of setup matrices to test its influence in the rolling models. As the objective is to try to reproduce as close as possible the features of the case study, setup times can be grouped into two kinds: major and minor setups. Major setups occur when changing between products of different product families and minor setups occur within products of the same family. In instances with low setup

Table 4.4: Demand generation

Product type	Demand features		% of the total products	
			Stationary	ABC
High volume	μ_i σ_i	U[200,400] U[10,20]	0%	20%
Medium volume	μ_i σ_i	U[60,140] U[6,14]	100%	60%
Low volume	μ_i σ_i	U[5,20] U[1,2]	0%	15%
Periodic	μ_i σ_i TBO	100 10 2	0%	5%

time variability minor setups are taken at random from {60, 70, 90}, while major setups from {100, 120, 140}. For the case of high variability, setups between products can take values in {45, 50, ..., 85, 90} (same family) and in {120, 150, ..., 210, 240} (different families). Thus, the second case penalizes more heavily the changeover among the different families.

Tightness of capacity. A feature that has proved to have a strong impact on the performance of solution methods is the tightness of the capacity. Again, to shape random instances similar to the case study, demand is constant throughout the planning horizon corresponding to the number of production hours available in regular time. Its value is determined for each machine according to:

$$Cap_{mt} = \sum_{i \in \mathcal{A}} D_i \cdot p_{im} \cdot \frac{p_{im}}{\sum_l p_{il}} \cdot \frac{1}{T \cdot Cut} \quad \forall m, t \tag{4.44}$$

where D_i is the total demand for product i in the planning horizon, partially assigned to each machine according to a weighted average considering the processing times. The tightness is defined by Cut and is held at two levels: 0.8 (tight) and 0.9 (very tight). The existence of setup times and the seasonal effect can result in insufficient capacity. This can be faced by producing in advance or by using overtime. Random instances have two types of overtime $o \in \{1, 2\}$, each corresponding to 15% of the regular capacity.

For each product, a machine independent processing time $\overline{p}_i$ is generated using an uniformly distributed function between 2 and 7. Then, the machine dependent processing times are obtained according to the efficiency of each machine, eff_m, taken from a uniform distribution U[0.7,1], $p_{im} = \overline{p}_i / eff_m$. Holding costs are selected from the interval U[2,10]. To set the overtime and setup cost we follow a similar approach to Özdamar and Birbil

(1998). Let θ be the average cost per unit of time calculated as follows:

$$\theta = \sum_i \frac{h_i}{\sum_m \left(st_{im}/\bar{d}_i + p_{im} \right) \cdot \frac{p_{im}}{\sum_l p_{il}}} \cdot \frac{D_i}{\sum_j D_j} \cdot ratio. \tag{4.45}$$

This expression estimates the cost saving of having an additional unit of time to reduce the inventory cost. In the random instances *ratio* is set to 3, setup costs are defined by: $sc_{ijm} = \theta \cdot st_{ijm}$ and overtime costs by: $r^o_{mt} = \theta \cdot (1-o) \cdot (1.1) \cdot cap^o_{mt}$. Thus overtime type 2 costs 10% more than type 1.

For each combination of demand profile, setup variability and tightness of the capacity we create instances with 2 and 3 machines, and 15 and 25 products, as the problem size can also have an important effect on the performance of models. Products are associated to machines according to a certain probability τ_{avg} limited to an upper and lower bound. Each machine has a τ_{avg}=40% probability of producing each product and can not produce less than $\lceil 0.2 \cdot N \rceil$ or over $\lceil 0.8 \cdot N \rceil$ products.

We generate 10 instances for each combination of demand profile (two levels), setup times variability (two levels), tightness of capacity (two levels), number of machines (two levels) and number of products (two levels) giving origin to a total of 320 instances.

Figure 4.4 and 4.5 summarizes the solution quality over all the generated instances. Figure 4.4 shows for each of the rolling methods the distribution of the percentage deviation from the best solution (lowest value of F_1 among all methods) by using a box plot. The mean percentage deviation for each method is also depicted with a diamond. Figure 4.5 reproduces the estimated commutative distribution of the deviation given by the percentage of the total instances with a solution within a given deviation.

From Figure 4.4 we observe that using the linear relaxation on the unscheduled horizon yields lower deviations than incorporating in the processing times the losses from setup times. Model CSP^f_p has the worst performance in terms of deviation which can be explained by the fact that the assumption of a lot-for-lot basis and the mean average setup times are far from being observed in practice. As a consequence this model often underestimates the future capacity by considering to much setup time in each period leading to solutions that have higher holding costs when compared to the best solution. On the other hand, the performance of CSP_p is not far from the performance of CSP_{rel}. In theory such a result is not a surprise since relaxing the integrality constraints on setup variables leads to a model that can produce any item by assigning the smallest possible setup time/cost possible. A common feature to both models is the overestimation of the future capacity which leads to solutions with higher overtime costs and lower holding costs when compared to the best solution. Reducing the future capacity based on the setups witnessed in the scheduled horizon outperforms the use of the relaxation, with the use of the maximum loss being better than the use of the average loss. In these models either overestimation or underestimation occur less frequently leading to smaller deviations in terms of overtime and holding costs. An interesting fact is that the use of a more detailed model, CSP^f_{ind}, does not seem to constitute an advantage over CSP^{mean}_{st} and CSP^{max}_{st}. Indeed the worst case

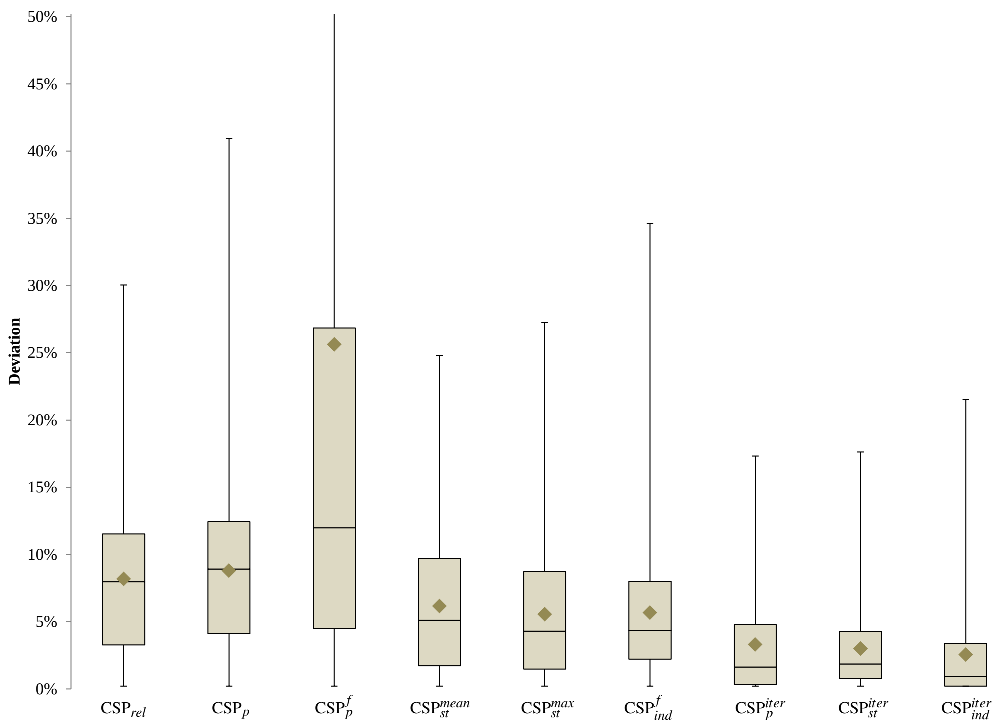


Figure 4.4: Distribution of the percentage deviation from the best solution for each method

performance deteriorates, although the results are very similar to the previous two models, with a slight reduction of setup costs and increase of both overtime and holding costs. The iterative method when applied to either CSP$_p$, CSP$_{st}$ or CSP$_{ind}$ achieves very interesting results. CSP$_p^{iter}$ clearly outperforms the single point estimations of CSP$_p$ and CSP$_p^f$, as well as CSP$_{st}^{iter}$ outperforms CSP$_{st}^{mean}$ and CSP$_{st}^{max}$. And CSP$_{ind}^{iter}$ is generally better than CSP$_{ind}^f$.

In Figure 4.5 in the horizontal axis we have the percentage deviation from the best solution and in the vertical axis the percentage of the total problems, thus each line approximately describes the cumulative distribution of the solutions within a given percentage deviation for each method. We observe that the relative performance of the models is relatively stable with respect to the deviation from the best solution. Let us focus our attention on the percentage of solutions within up to 3% of the best solution. Models CSP$_p$, CSP$_p^f$ and CSP$_{rel}$ have similar performances at this point with the relaxation being slightly better. The main difference between CSP$_p^f$ and the other two models is the fact that few solutions have high quality and the convergence as the deviation is increased is much slower. The second group of models with similar performance are CSP$_{st}^{mean}$, CSP$_{st}^{max}$ and CSP$_{ind}^f$. The model CSP$_{st}^{max}$ is the best among these three. Interesting is the fact that CSP$_{ind}^f$ performance can be explained by its variability. Low values are found within smaller deviation ranges and there are also some instances in which the model performance is poor. Finally, the three methods using the iterative approach form the third group with similar performance. The chart also points to the fact that both CSP$_{st}^{iter}$ and CSP$_{ind}^{iter}$ are better than CSP$_p^{iter}$. Only

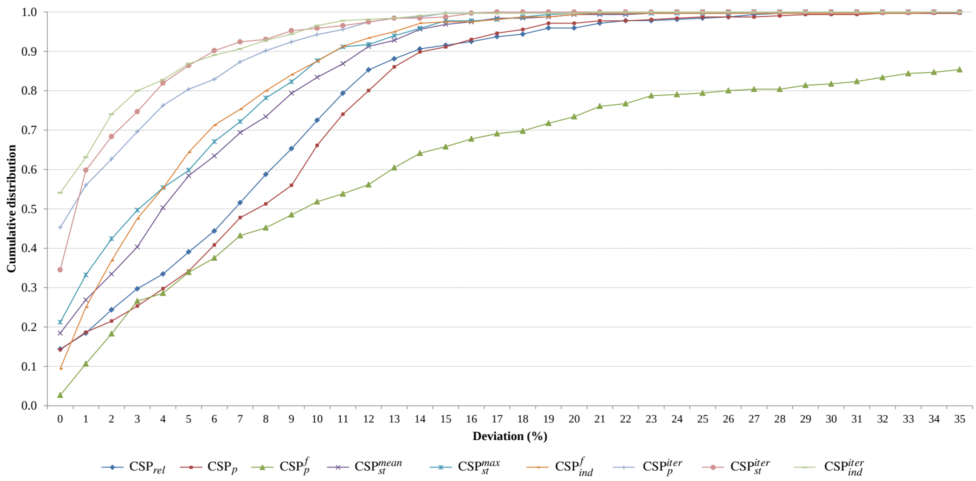


Figure 4.5: Commulative distribution of the percentage deviation from the best solution for each method

approximately 70% of the solutions of CSP_p^{iter} are below 3% deviation in comparison to 75% of CSP_{st}^{iter} and 80% of $\mathrm{CSP}_{ind}^{iter}$.

All these results were obtained with the limit of 10 minutes for each planning step, but most models finished before this threshold. Hence, it is also important to look into the efficiency of the approaches. The iterative approach is expected to take longer running times as the base models have to be solved (re-solved) more than once in each planning step, in comparison to the models that only require one solution of the MIP. Figure 4.6 depicts the distribution of the running times for each planning step using again a box plot with the diamond denoting the average. As expected the iterative approach increases running times. The most computational efficient methods are CSP_p, CSP_p^f, CSP_{st}^{mean} and CSP_{st}^{max}, with a neglectable difference among them. Applying the iterative approach to these models increases the average running time by about 3 times per planning step. Due to the high number of decisions variables in the formulation used in the unscheduled horizon on model CSP_{rel}, this method takes longer than most of the others with the exception of the ones using the sequence independent setups. These are by far the least computational efficient methods, which is not a surprise as the MIP model behind them is substantially harder to solve as a result of the increased number of binary variables. Interestingly applying the iterative approach on CSP_{ind} only slightly increases the average running time of each planning step. This is explained by the fact that after having the optimal solution for the MIP in the first iteration the following re-optimization is carried rather quickly.

Crossing the deviation with the running times, the results indicate that CSP_p trades-off better the solution quality versus running time than CSP_{rel}. As suggested by the results, reducing the capacity in the unscheduled periods is preferable to increase processing times, since for similar running times deviations are lower. In the case of sequence dependent setups, the iterative approach appears as an interesting solution to improve deviation with limited impact on the running times. For the case of CSP_p and CSP_{st}, applying the iterative approach is clearly a question of compromise between solution quality and running time.

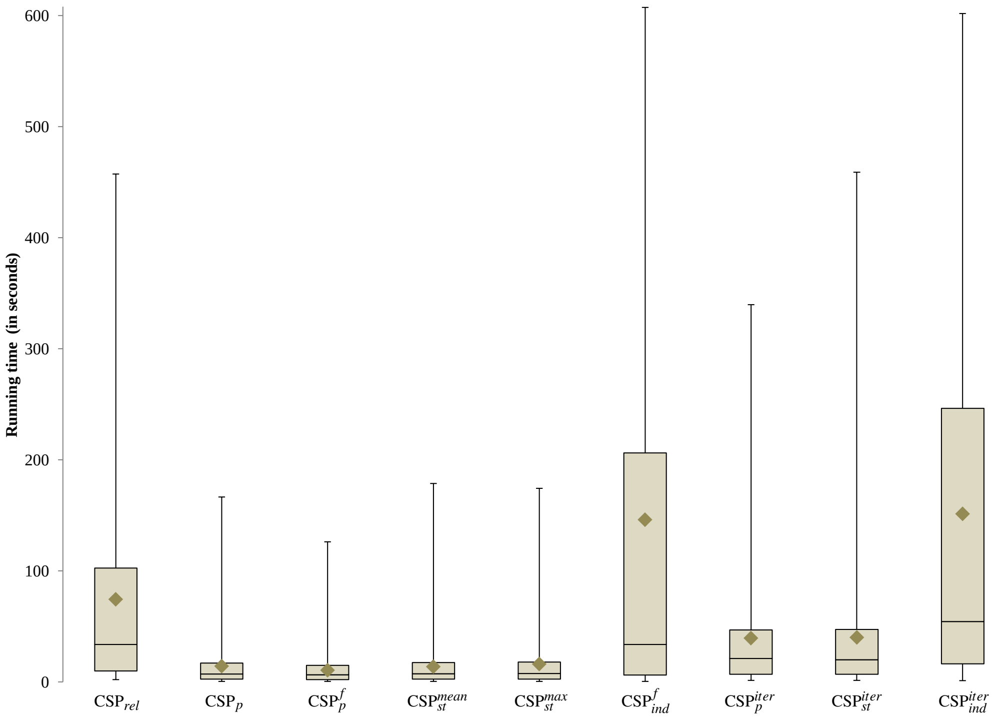


Figure 4.6: Distribution of the running time for each method

Overall, and considering that for each planning step a limit of 10 minutes is given CSP_{st}^{iter} provided the most cost-time efficient solutions.

4.6.2 Real-world problem set

This instance set corresponds to a collection of three problem instances. Table 4.5 summarizes the main features of each instance. All the instances have 11 time periods of available demand forecast collected over 6 consecutive planning steps of 6 time periods. As for the random instances F_1 is used to evaluate the performance of the several methods keeping $T = 6$, $T_s = 2$ and $\Delta = 1$. We also study different planning assumptions that can influence the performance of the methods. Each instance is solved either considering all the available capacity without overtime costs, or considering overtime costs whenever overtime is used. Two type of overtime are available, type I considers Saturdays and Holidays, and type II the use of Sundays. For the instances with or without overtime costs we also test the efficiency of the methods in the presence or not of setup costs. Setup costs are introduced by multiplying setup times by 0.1 in order to turn this a secondary objective. The results are given in Tables 4.6 and 4.7. In bold are the best solution values for each instance.

When overtime costs are not consider all the models, except CSP_p^f, have very similar performances in terms of cost function. Factoring processing times using (4.19) reveals again a very poor performance and for instance R_2 no feasible solution could be achieved (denote in the table as n.a.). A cost zero in the instances without overtime and setup costs

Table 4.5: Summary of real-world instances data

Instance	N	$\frac{N}{M}$	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$
R_1	20	12.5	10	15		
R_2	35	22	11	33		
R_3	40	12.25	16	8	13	12

Table 4.6: Results for the real-world instances without overtime costs

	Solution						Running time					
	without setup cost			with setup cost			without setup cost			with setup cost		
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
CSP_{rel}	**0**	**0**	**0**	**757**	1179	**2730**	28	128	37	98	519	167
CSP_p	**0**	**0**	**0**	**757**	1176	2805	31	103	36	40	189	75
CSP_p^f	12219	n.a.	4	13011	n.a.	2818	17	n.a.	34	14	n.a.	66
CSP_{st}^{mean}	**0**	93	211	**757**	1238	2953	29	147	64	41	196	80
CSP_{st}^{max}	**0**	10	49	**757**	1181	2848	31	178	95	53	349	89
CSP_{ind}^f	**0**	**0**	**0**	**757**	1204	2805	20	104	33	136	3697	1230
CSP_p^{iter}	**0**	**0**	**0**	**757**	1202	2805	61	327	106	108	491	184
CSP_{st}^{iter}	**0**	**0**	**0**	**757**	1202	2796	77	349	116	112	467	204
CSP_{ind}^{iter}	**0**	**0**	**0**	758.61	**1166**	2824	70	294	93	124	681	217

means that all the demand could be supplied without the need for inventory build up. In this sense, models CSP_{st}^{mean} and CSP_{st}^{max} tend to slightly underestimate future capacity giving origin to some stock creation. If setup costs are introduced the scenario remains very similar. Running times are considerably larger for the methods based on the iterative approach if no setup cost is considered. On the other hand, in the presence of setup times their computational burden is much less significant.

Table 4.7: Results for the real-world instances with overtime costs

	Solution						Running time					
	without setup cost			with setup cost			without setup cost			with setup cost		
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
CSP_{rel}	24887	49917	14689	25644	52460	14560	77	2726	475	107	2808	354
CSP_p	26946	52541	14691	27627	51239	15107	39	1211	257	36	1307	136
CSP_p^f	54416	n.a.	17865	48383	n.a.	16440	27	n.a.	157	31	n.a.	101
CSP_{st}^{mean}	26541	50562	8203	27358	50675	10766	52	1418	304	48	655	169
CSP_{st}^{max}	28366	50216	8720	28921	50503	11049	51	767	106	47	577	89
CSP_{ind}^f	**23609**	48395	8061	26605	46763	7408	827	3623	3612	2189	3620	3610
CSP_p^{iter}	24134	**43114**	7439	**24835**	**43420**	9891	123	2169	368	113	2618	900
CSP_{st}^{iter}	24580	48350	7900	25261	47593	10658	117	2532	268	130	2296	269
CSP_{ind}^{iter}	23717	45681	**6720**	25579	45735	**6975**	273	3650	1339	746	3622	1221

Tackling overtime decisions reveals significant differences between the methods. The grouping present in the results of the generated instances appears to be repeated again, either if we consider or not setup costs. The first group is composed by CSP_p^f which can be isolated as the worst performing method. The second group contains CSP_{rel} and CSP_p, in the third group are CSP_{st}^{mean} and CSP_{st}^{max}, and in the best performing group are CSP_{ind}^f

plus the methods based on the iterative approach. Interesting is the fact that in the real-world instances CSP^f_{ind} improves its performance, however the effect on the running time of the underlying complex problem is still visible, especially for the larger instances (R_1 and R_2). We also highlight that the iterative version of CSP_{ind} not only delivers better quality solution than CSP^f_{ind}, but also has better running times. This can be explained by the fact that the initial estimation leads to simpler problems. Overall, from the results we can observe that applying the iterative method improves the solution quality of the rolling models also in the real-world instances, leading to superior quality solutions.

4.7. Conclusions

In this paper we investigate a rolling horizon approach to the parallel machine lot sizing and scheduling problem with sequence dependent setups. Our study is motivated by the planning problems arising at a case study in the beverage industry. In the company, the sequence of production lots is only needed for the short term to be implemented in practice, while their sizing is done for a longer horizon to estimate future capacity shortages and costs. The current approach follows a hierarchic decomposition of the problem, by first solving the lot sizing problem and at the second level sequencing the lots defined for the first planning period. However, this approach is known to deliver poor performance especially in the presence of sequence dependent setups. As such, the objective of this work is to render mathematical formulations and methods that can be used in practice to solve the aggregate problem, bearing in mind the rolling planning approach and the constant event-based re-planning.

We follow a line of research that explores the rolling planning features to reduce the complexity of lot sizing and scheduling MIP models by simplifying the decisions beyond the implementation horizon. This reduces the solving time, enhancing their potential applicability in practice. We focus our attention on the determination of the simplification strategies that produce a better estimation of the future capacity and costs, thus leading to the most cost-efficient solutions to be implemented in a rolling basis. We study the proposed simplifications strategies and also introduce a new approximate model that uses the information on setup time expenditure in detailed time periods to reduce the capacity in the simplified horizon. Moreover, these simplifications are appended to a very flexible and computational efficient model to integrate sequencing decisions in lot sizing and scheduling problems.

An important innovation of our work is the iterative method used to improve the accuracy of the parameters defined for the simplified horizon in the context of approximate models. The new method builds on the idea that the solutions for the simplified horizon contain valuable information to refine these estimations. To the best of our knowledge, prior research has just focused on deriving one shot attempt to calculate them. The method is modular and can be applied in several distinct model formulations as shown here.

In our computational tests we focus on analyzing the performance of the MIP models resulting from the various simplification strategies on a large set of instances that mimic the problems faced by the beverage and similar industries. Instances have a high capacity utilization and often require the use of overtime to face the demand peaks. A small collection of data from the case study also validates our comparisons. Overall, we conclude that reducing the capacity of future periods according to the setups witnessed in the detailed part of the planning horizon leads to a very efficient rolling model and that the use of the iterative approach improves the quality of the solutions for all rolling methods. The use of sequence independent setups may also be a very interesting approach mainly for two reasons. First, it results in more reasonable rolling solutions as the setup cost and time incurred when scheduling the production of item limits the number of lots per period. Second, it gives the basis to include some operational constraints such as minimum lot sizes due to the presence of binary decision variables in the simplified horizon. Thus, rolling solutions emerging from this approximation can have more value from the managers perspective. Nevertheless, an important concern regarding this approximation model is its computational tractability as shown in the computational results, pointing out that efficient solution approaches are required in this context.

Another important topic for future research is the use of the simplification strategies and the iterative method to create efficient solution algorithm to the static version of the problem as pointed in Section 4.4.1 and following the works of Clark (2005) and Araujo et al. (2007).

Bibliography

S. A. Araujo, M. N. Arenales, and A. R. Clark. Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. *Journal of Heuristics*, 13(11): 337–358, 2007.

A. R. Clark. Optimization approximations for capacity constrained material requirements planning. *International Journal of Production Economics*, 84(2):115 – 131, 2003.

A. R. Clark. Rolling horizon heuristics for production planning and set-up scheduling with backlogs and error-prone demand forecasts. *Production Planning & Control*, 16 (1):81–97, 2005.

A. R. Clark and S. J. Clark. Rolling-horizon lot-sizing when set-up times are sequence-dependent. *International Journal of Production Research*, 38(10):2287–2307, 2000.

C. Dillenberger, L. F. Escudero, A. Wollensak, and W. Zhang. On practical resource allocation for production planning and scheduling with period overlapping setups. *European Journal of Operational Research*, 75(2):275 – 286, 1994.

L. Escudero. An inexact algorithm for the sequential ordering problem. *European Journal of Operational Research*, 37(2):236 – 249, 1988.

A. Federgruen, J. Meissner, and M. Tzur. Progressive interval heuristics for multi-item capacitated lot-sizing problems. *Operations Research*, 55(3):490–502, 2007.

B. Fleischmann and H. Meyr. Planning hierarchy, modeling and advanced planning systems. In S. Graves and A. de Kok, editors, *Supply Chain Management: Design, Coordination and Operation*, volume 11 of *Handbooks in Operations Research and Management Science*, pages 455 – 523. Elsevier, 2003.

B. Fleischmann, H. Meyr, and M. Wagner. Advanced planning. In H. Stadtler and C. Kilger, editors, *Supply Chain Management and Advanced Planning*, pages 81–106. Springer Berlin Heidelberg, 2008.

M. Gopalakrishnan, K. Ding, J. M. Bourjolly, and S. Mohan. A tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover. *Management Science*, 47(6): 851–863, 2001.

L. Guimarães, D. Klabjan, and B. Almada-Lobo. Modeling lotsizing and scheduling problems with sequence dependent setups. Submmited to the European Journal of Operational Research, 2013a.

L. Guimarães, D. Klabjan, and B. Almada-Lobo. Pricing, relaxing and fixing under lot sizing and scheduling. *European Journal of Operational Research*, (0):–, 2013b. doi: 10.1016/j.ejor.2013.04.030. Available in Press.

K. Haase. Capacitated lot-sizing with sequence dependent setup costs. *OR Spectrum*, 18 (1):51–59, 1996.

R. J. James and B. Almada-Lobo. Single and parallel machine capacitated lotsizing and scheduling: New iterative MIP-based neighborhood search heuristics. *Computers & Operations Research*, 38(12):1816 – 1825, 2011.

S. Kang, K. Malik, and L. J. Thomas. Lotsizing and scheduling on parallel machines with sequence-dependent setup costs. *Management Science*, 45(2):273–289, 1999.

A. Kimms. Stability measures for rolling schedules with applications to capacity expansion planning, master production scheduling, and lot sizing. *Omega*, 26(3):355 – 366, 1998.

A. Kovács, K. N. Brown, and S. A. Tarim. An efficient MIP model for the capacitated lot-sizing and scheduling problem with sequence-dependent setups. *International Journal of Production Economics*, 118(1):282 – 291, 2009.

G. Mateus, M. Ravetti, M. Souza, and T. Valeriano. Capacitated lot sizing and sequence dependent setup scheduling: an iterative approach for integration. *Journal of Scheduling*, 13(3):245–259, 2010.

H. Meyr. Simultaneous lotsizing and scheduling on parallel machines. *European Journal of Operational Research*, 139(2):277 – 292, 2002.

L. Özdamar and S. I. Birbil. Hybrid heuristics for the capacitated lot sizing and loading problem with setup times and overtime decisions. *European Journal of Operational Research*, 110(3):525 – 547, 1998.

M. Pinedo and S. Kreipl. Planning and scheduling in supply chains: An overview of issues in practice. *Production and Operations Management*, 13(1):77–92, 2004.

Y. Pochet and L. A. Wolsey. *Production Planning by Mixed Integer Programming (Springer Series in Operations Research and Financial Engineering)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

P. Porkka, A. P. J. Vepsäläinen, and M. Kuula. Multiperiod production planning carrying over set-up time. *International Journal of Production Research*, 41(6):1133 – 1148, 2003.

H. Stadtler. Multilevel lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows. *Operations Research*, 51(3):487–502, 2003.

L. Tiacci and S. Saetta. Demand forecasting, lot sizing and scheduling on a rolling horizon basis. *International Journal of Production Economics*, 140(2):803 – 814, 2012.

Chapter 5

Long term production planning

Annual Production Budget in the Beverage Industry

Luis Guimarães* · **Diego Klabjan**† · **Bernardo Almada-Lobo***

Published in *Engineering Applications of Artificial Intelligence*, 2012
http://dx.doi.org/10.1016/j.engappai.2011.05.011

Abstract Driven by a real-world application in the beverage industry, this paper provides a design of a new VNS variant to tackle the annual production budget problem. The problem consists of assigning and scheduling production lots in a multi-plant environment, where each plant has a set of filling lines that bottle and pack drinks. Plans also consider final product transfers between the plants. Our algorithm fixes setup variables for family of products and determines production, inventory and transfer decisions by solving a linear programming (LP) model. As we are dealing with very large problem instances, it is inefficient and unpractical to search the entire neighborhood of the incumbent solution at each iteration of the algorithm. We explore the sensitivity analysis of the LP to guide the partial neighborhood search. Dual re-optimization is also used to speed-up the solution procedure. Tests with instances from our case study have shown that the algorithm can substantially improve the current business practice, and it is more competitive than state-of-the-art commercial solvers and other VNS variants.

Keywords Long-term production planning · Beverage industry · Large neighborhood search · Mathematical programming

5.1. Introduction

The beverage industry is a sub-sector of the food industry, the second largest sector in the European manufacturing industry in terms of value added. It supplies a variety of products from wine, beer and spirits to mineral and sparkling water and soft drinks. Markets worldwide are strongly affected by cultural differences, especially in Europe. This effect creates the environment for the appearance of small to medium size companies that are specialized

*INESC TEC, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal
†Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA

in local products and/or local brands. Nevertheless, there are a number of large multinational companies able to compete in markets across the globe offering a wide variety of products, such as soft drinks. Today's competition in this sector leads companies to expand their product portfolio, which combined with the advanced technology present in modern production sites, raises the need for efficient production planning. Moreover, production sites in this industry tend to be geographically disperse allowing companies to satisfy local demands at lower costs. Production planning is often conducted considering only one plant at time, ignoring the potential benefits of coordination. This paper is inspired by a real industrial case from a company competing in the beer and soft drink industries. The focus is to define a long-term production plan to a series of production (filling) lines located in different plants. The scheduling of product families at each filling line is the basis for production, inventory, and transfer decisions. Transfer decisions represent movements of finished products and come from the fact that demand observed at a geographical area around each plant can be satisfied by other production sites to cope with under capacity of a given plant. Under these conditions, plants act both as production and distribution centers since warehouses are located near them and have individual demand. Decisions are traditionally made for a rolling planning horizon of 12 to 18 months with a monthly bucket.

Real-world production planning problems often result in intractable models, and even simplified versions result in NP-hard problems. However, only realistic modeling of the problem features can help managers in their decisions, which was already pointed out as a field of future research of two previous literature reviews on production planning problems Karimi et al. (2003); Jans and Degraeve (2006). Furthermore, to deal with the complexity of industrial applications, Jans and Degraeve (2006) encourage the use of metaheuristics. Naturally, the large scale instances that arise in our application demand their use. Metaheuristics are frameworks used to solve combinatorial optimization problems, guiding other simple heuristics to search for high quality solutions. Local search (or neighborhood search) is among these heuristics. They attempt to iteratively improve an incumbent solution by replacing it with a better solution found in its neighborhood leading to a local optimum. Several schemes have been developed to overcome the entrapment in local optima. Variable Neighborhood Search (VNS) is a local search framework based on systematic change of neighborhoods both to find local optimum and to perturb the solution to emerge from entrapment Hansen et al. (2008). We make use of VNS principles and of our problem formulation to develop a heuristic for the problem. Our local search attempts to find a better assignment of families to filling lines, and the subsequent decisions on production, inventory and transfers are achieved through linear programming (LP). This neighbor evaluation methodology can be expensive as we are dealing with very large neighborhoods, therefore we test different techniques to speed-up the local search. The final tableau of the LP simplex algorithm provides valuable information that we use to guide the search. Other, speed-up techniques involve the use of dual-reoptimization to quickly identify and get rid of low quality solutions. Tests performed on a set of randomly generated instances attested to the algorithm's superiority over commercial solvers and other VNS variants for medium and large size instances. Later the benchmark against industry current practice revealed its potential cost saving capability.

The remainder of the paper is as follows. In Section 5.2 we start by describing the production process of the two different types of products (beer and soft drinks) tackled here. Planning process and major planning constraints are introduced in Section 5.3. We also integrate long-term planning in the industry planning framework and pinpoint industry practices. Section 5.4 presents the real-world case study and related work in the literature. Section 5.5 is dedicated to the problem modeling and solution methodology. Section 5.6 reports numerical experiments, first on a collection of randomly generated instances to assess the quality and robustness of the algorithm, and later on real-world instances, together with a benchmark against the current practice at a company. Finally, the paper ends with a short summary and outlook.

5.2. The beer and soft-drink production process

Beer and soft drinks industries share some common features in their production process. Both encompass two main production stages: liquid production (stage I) and liquid bottling (stage II).

Stage I of the beer production process, also known as brewing, it has the purpose of converting the sugars present in the starch source into alcohol through a reaction of a yeast. Different beers have different recipes that determine their production process. Yet, generally there are three main processes in beer production: wort preparation, fermentation and maturation, and filtering. Wort preparation consists of the extraction of the fermentable sugars from usually barley malt and, in addition, of hop. Fermentation follows next and its goal is to transform the sugars into ethanol through the action of fermenting yeasts. Undesirable substances from the censorial point of view, are removed in maturation in a series of chemical, biological and physical steps. Fermentation and maturation processes have the longest processing times and depending on the beer recipe, they can last from 4 days to 3 weeks. The beer resulting from the previous process is turbid, therefore a filtration process is conducted. During this step for some flavoured beers, syrups or concentrates are also added. Non-alcoholic beers pass through a stripping process to remove the ethanol. For more detail about the beer production process the reader is referred to Eskin (1990); Kent and Evers (1994); Kourtis and Arvanitoyannis (2001).

On the other hand, soft drinks are beverages consisting primarily of carbonated water, sugar, and flavourings. Stage I of soft drink production starts with water clarification. Liquid flavour preparation follows next, and is conducted in specialized mixing tanks. Sugar, flavour concentrates and water are pumped in a specific sequence and then carefully mixed. Sophisticated machines control the flow of the ingredients to ensure the perfect recipe. Carbonation is generally the last step in soft drink production, normally performed just before liquid bottling. For more information on soft drink production see Matthews (2003).

In the second stage, different sized cans, glass bottles (disposable and reusable), kegs and plastic (PET) bottles (less common in the beer case) are filled with beer and soft drinks. A filling line consists on a series of conveyor belts and machines that wash, fill, seal, label and pack the bottles, cans or kegs Cooke et al. (2005); Tsarouhas and Arvanitoyannis (2010). The first step involves washing and disinfection of containers, which afterwards pass through an inspection to guarantee the absence of potential hazards. The next machine performs the filling and capping of containers. To ensure product shelf life over a determined period, a pasteurization step follows container filling. For soft drinks, the pasteurization step may take place in a mixing tank instead. Its duration depends on the product features. Labelling is carried out next. Filled containers inspection certificates that the specified volume has been introduced and no defects occurred during the process. Packing containers into paper-boxes, packs or other selling units precedes palletization and storage. Since these processes are done in series from hereafter, we will refer to the set of machines that compose a filling line as a whole.

5.3. Planning production in the beverage industry

Planning production in the beverage industry, specifically in the beer industry, is a complex process. Not only are there several processes involved, but also increasing competitiveness of the market forces companies to enlarge their product portfolio posing new challenges and raising the need for decision support tools to help managers.

5.3.1 Main planning constraints

One of the main planning constraints is related to the sales profile of these products. Sales of beer and soft drinks have high seasonality and variability. Beer and soft drinks consumption peaks at Easter and Christmas, but summer is by far the highest point in terms of sales. Moreover, there is a clear increase of sales in the second half of each month. On the other hand, capacity remains almost constant throughout the year and it can be evaluated by the number of production hours available. Product demand is also affected by other sources of variability, such as brand management and clients commercial policy. Some of the most important customers of these companies are large retailers with extremely aggressive marketing strategies that require almost instantaneous response from suppliers. These sales characteristics stress production and lead the industry to work on a make-to-stock basis. But, diversity of the product range makes sales hard to forecast.

Looking at the industry supply chain, a typical beverage industry company has one or more plants relatively close to the geography of demand, in order to avoid transfer costs, which otherwise would have assumed an important percentage of the total cost. Within each plant, stages I and II of the production process are most of the time divided and buffers may exist between stages, with typically a single unit of stage I supplying a series of parallel filling lines. It is a common practice in industry to consider the filling stage (II) as the bottleneck of the entire production process, due to several reasons. For the beer case, buffers between the different main processes of stage I allow it to be more flexible. Moreover, the high number of different products that have to be manufactured in stage II correspond to a few different types of beer (or syrups for the soft drinks) in stage I, since SKU (Stock Keeping Unit) differences may rely on a different container, label or package affecting only stage II.

Filling lines are usually divided according to their technological aspects (e.g. filling lines for kegs are unable to fill bottles or cans). Furthermore, an important distinction is made between filling lines for disposable and reusable bottles, since an extra step and machine are needed to conduct an additional washing procedure in the latter case. Hence, disposable bottles filling lines can not fill reusable bottles, but no restrictions are present the other way around. Even so, filling lines are relatively flexible and often a certain product can be assigned to several alternative lines, even within the same plant, but with throughput rates (measured in terms of litres per minute (l/min)), that might be substantially different.

Each filling line can only produce a single product at any time, being adjusted to fill a certain liquid, container type and size, and final package. A product changeover may involve several changes in the filling line and possibly a cleaning step. Liquid type switchovers always involve the cleaning of the filling line and sometimes the setup of the pasteurization machine. On the other hand, switches on the container type and/or size and final packaging trigger mechanical adjustments in most machine settings. These operations consume scarce production time (capacity) and can cause loss of material, that depend on the production sequence. Therefore we have the presence of sequence-dependent changeover times and costs. The increase of the number of products that took place in recent years has reduced the operational times of filling lines as more setups are needed. In addition, market pressure to work with less stock and to deliver products more frequently has also increased the number of production batches, reducing their size and consequently leading to the appearance of additional extra setups.

Filling lines operate on a shift basis and their capacity can be translated into the number of hours available for production. Some of the filling lines operate around the clock, therefore overtime is not always an option. Investing in new lines is also problematic as it greatly increases fixed costs. Some investments can be made in order to make filling lines more flexible, but they have to be carefully studied since their cost can be significant. Yet, such changes would only produce effects in the long-term, and short-term capacity can be considered constant. All the aforementioned reasons raise the issue of efficient production planning as it can guarantee a better utilization of resources and, ultimately, the competitiveness of the company.

In the presence of a multi-plant environment further planning features appear. Some product specialization is possible, aiming to achieve better throughput rates or standard quality requirements in stage I due to larger production batches whose process is easier to control. Nevertheless, for standard products production near the consumption location should yield low cost production plans due to shorter transfer costs.

5.3.2 Production planning systems in the beverage industry

To face the constraints described above and the different nature of decisions and actions, production planning in the beverage industry is made by several company echelons with different aims and planning horizons. Although decisions are strongly dependent, it is virtually impossible to sustain a single decision model for the entire decision-making process as it would be extremely hard to maintain, solve and interpret. Moreover, market dynamics also determine that highly detailed plans for a distant future are in most occasions useless. Planning decisions are therefore made in a hierarchical process composed of three levels: strategic (long-term) planning, tactical (medium-term) planning and operational (short-term) planning. Long term planning assesses investments in the installed capacity, trying to balance capacity with demand for a planning horizon of 12 to 18 months. Concerning tactical planning, the focus is to derive plans for operations, essentially production and distribution, aiming at cost efficiency. Here planning horizons commonly span from 4 to 12 weeks. The lower level of the hierarchy schedules operations to the available resources looking at a very short planning period from 1 day up to 1 week. These levels operate in a rolling horizon approach, only a few periods in the begin of the planning horizon are actually executed, furthermore the output of an upper level constitutes an important input for the following level.

5.4. The case study

Our study is motivated by a Portuguese company that competes in the beverage industry with sales across the globe. The company holds many nationally very popular brands of beer, soft drinks, and mineral and sparkling water. Production sites are spread around the country, accounting for 8 plants and more than 20 filling lines. Mineral and sparkling water plants are located near a water source, while other production sites are responsible for beer and soft drink production. Only planning of beer and soft drinks has to be done simultaneously as both product types share common production resources and this will be the scope of our study. The aim is to create the annual production budget (PB).

PB is part of the company's annual budgeting process. The budgeting is a vital tool to align company goals and translate the strategy defined into the next 12 months. Annual budgeting starts in mid September and lasts until late October. The first main task is the creation of an annual sales budget (SB). SB is driven by a monthly sales forecast for each product in the following year. In parallel, the production departments of each plant schedule the filling lines maintenance calendar and estimate throughput rates for each product. These throughput rates are approximations based on the previous years and also reflect expected gains or losses of efficiency. Embedded in these estimations are the sequence-dependent setups witnessed in the years before. The goal of the PB is not to obtain a detailed schedule for production lines, but rather an estimation of the adequacy of resource capacity to SB. Therefore, production sequencing is disregarded and capacity loss due to sequence-dependent setups is incorporated in throughput rates.

The PB is conducted by the planning department and aims at validating the SB from an industrial and economical point-of-view. Besides SB and throughput rates, the available capacity is an input determined from the filling lines maintenance calendar and the number of available days for production. Capacity is estimated per filling line and divided into three categories: normal workdays, Saturdays and holidays, and Sundays. This distinguishes normal capacity from overtime. SB is generally distributed among the plants according to the past years sales. However, technological constraints, production quality assurance or product specialization can imply a pre-determined plant. Technological constrains are related to production and filling equipments required to produce certain products. Production quality assurance deals with situations in which minimum batch sizes and/or production frequency may not be achieved if forecasts for family's demand are disaggregated.

PB only accounts for the filling stage, since this stage is considered the production process bottleneck. Therefore, the number of working shifts is decided only for the filling stage. PB decides on the assignment of products to the different filling lines in the planning horizon and the definition of production lotsizes. Through this step, plant inventory and inter-plant transfers are also determined. The objective is to satisfy SB while minimizing inventory holding costs, setup costs, inter-plant transfer costs and overtime costs.

PB conclusion triggers the creation of the transportation and materials procurement budget.

5.4.1 Company practice and opportunities

The creation of the PB is a hard time-consuming task. The planner is challenged with over 150 products divided by approximately 60 product families and 14 different filling lines, although technological constrains restrict the problem size. One of the strategies used by the company to overcome this problem is the choice of a preferential machine to supply the demand of each product at each plant. Frequently, more than one filling line of the same plant can produce a certain product gross requirement, but the definition of a preferential filling line rule automatically fixes allocation, turning the act of planning easier. Nevertheless, throughout the process is natural that some filling lines exceed their normal capacity. Requirements can be moved to another filling line of the same plant, can anticipate raising inventory holding costs or can be moved to filling lines of other plants originating transfers and/or inventory costs. Another possibility is to use overtime capacity, which is limited to a certain maximum. PB is done once a year and the main key performance indicators regard average filling line utilization, total inventory, transfer and overtime costs.

5.4.2 Related work

Most literature has focused on the operational and tactical levels and the integration of both in beverage and related industries (e.g. soft-drinks Ferreira et al. (2009), foundries Araujo et al. (2007), glass industry Almada-Lobo et al. (2007) and animal feed Clark et al. (2010)). Nevertheless, some work is also available on medium to long-term planning mostly in terms of mathematical formulations. These models often include production, distribution and inventory management with capacity investments. Chandra and Fisher (1994) show that the integration of production and distribution in a single decision model yields better results than optimizing separate models. In Jolayemi and Olorunniwo (2004) decisions on production, transportation, purchasing and warehouse capacity extension are made for a multi-plant and multi-warehouse environment. Martin et al. (1993) study a real world case in the flat glass business. Production, distribution, and inventory operations are managed in a single model. An application in the chemical process industry is presented by Timpe and Kallrath (2000). Batch and campaign production in a multi-plant production system are decided along with distribution and marketing decisions. A real-world problem in steel manufacturing is approached by Sambasivan and Yahya (2005). Almada-Lobo et al. (2008) present a long term production planning model in the glass industry. A multi-plant production system where each plant has a set of production lines is considered but with common demand, i.e. demand is not attached to a specific location. The significant sequence-dependent setup costs and time that arise in this industry lead to the need of sequencing of product families. A VNS-based heuristic is used to solve the problem. Another industrial example at a metal item manufacturer is given by Dhaenens-Flipo and Finke (2001). Authors formulate an integrated production-distribution model considering a production system composed of multiple factories having several parallel production lines. Sequence-dependent setups are considerer at this level due to their magnitude and managed through predefined sequences. Distribution decisions relate to transfers from plants to warehouses and from warehouses to clients. Other examples occur in production environments where items are produced in a series of processes occurring in different plants Kaminsky and Simchi-Levi (2004).

5.5. Solution methodology

In this section we first present a mixed integer optimization model representing the problem that arises in the PB, hereafter called long-term production planning problem (LT3P). Based on this model a solution procedure is described aiming to achieve good quality solutions in limited computational time, which is not accomplished by exact methods.

5.5.1 Mathematical formulation

The model considers a multi-plant environment with P plants. Each plant has its own individual demand and storage capacity. As mentioned before, a certain plant can supply demand for another plant but additional transfer cost has to be accounted for. Common filling lines force us to simultaneously plan both beer and soft drink products. Filling lines are considered the production bottleneck of the production system, therefore decisions are taken only for this step. Products sharing common production features, the same container type and final package, are grouped into product families. The model considers a planning horizon divided into T periods, usually months. To formulate the model the following nomenclature is used:

Indices

i	product: $i \in \mathcal{N} = \{1,...,N\}$
f	family: $f \in \mathcal{F} = \{1,...,F\}$
j,k	plant: $j,k \in \mathcal{P} = \{1,...,P\}$
m	filling line: $m \in \mathcal{M} = \{1,...,M\}$
t	period: $t \in \mathcal{T} = \{1,...,T\}$

Sets

$\mathcal{M}_j$	set of filling lines belonging to plant j
$\mathcal{F}_m$	set of families that can be produced on filling line m

$\mathcal{N}_m^{line}$	set of products that can be produced on filling line m
$\mathcal{N}_f^{fam}$	set of products belonging to family f

Parameters

cap_{mt}	available capacity at filling line m in period t (in time units)
j_m	plant of filling line m
f_i	family of product i
d_{ijt}	demand of product i at plant j at the end of period t
h_{ijt}	unitary holding cost of product i at plant j at the end of period t
r_{ijkt}	unitary transfer cost of product i from plant j to plant k in period t
c_{fmt}	setup cost of family f on filling line m in period t
p_{fmt}	throughput rate of family f on filling line m in period t
b_{imt}	upper bound on production quantity of product i on filling line m in period t

To capture decision making the following variables are defined:

X_{imt}	production quantity of product i on filling line m in period t
I_{ijt}	stock of product i at plant j at the end of period t
W_{ijkt}	transfer quantity of product i from plant j to plant k in period t
Y_{fmt}	(=1) if a setup occurs to family f on filling line m in period t, (=0) otherwise.

The model is stated as follows:

$$\min \quad Obj_1 = \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{P}}\sum_{t\in\mathcal{T}}\left(h_{ijt}\cdot I_{ijt} + \sum_{k\in\mathcal{P}} r_{ijkt}\cdot W_{ijkt}\right) + \sum_{f\in\mathcal{F}_m}\sum_{m\in\mathcal{M}}\sum_{t\in\mathcal{T}} c_{fmt}\cdot Y_{fmt} \tag{5.1}$$

$$I_{ij,t-1} + \sum_{m\in\mathcal{M}_j} X_{imt} + \sum_{k\in\mathcal{P}\setminus\{j\}} W_{ikjt} = I_{ijt} + d_{ijt} + \sum_{k\in\mathcal{P}\setminus\{j\}} W_{ijkt}, \quad \forall i\in\mathcal{N}, j\in\mathcal{P}, t\in\mathcal{T} \tag{5.2}$$

$$\sum_{i\in\mathcal{N}_m^{line}} \frac{X_{imt}}{p_{f_i mt}} \le cap_{mt}, \quad \forall m\in\mathcal{M}, t\in\mathcal{T} \tag{5.3}$$

$$X_{imt} - b_{imt}\cdot Y_{f_i mt} \le 0, \quad \forall i\in\mathcal{N}_m^{line}, m\in\mathcal{M}, t\in\mathcal{T} \tag{5.4}$$

$$(I_{ijt}, W_{ijkt}, X_{imt}) \ge 0,\ Y_{fmt}\in\{0,1\} \tag{5.5}$$

The objective function (5.1) minimizes the sum of the holding, transfer and setup costs. Inventory balance constraints (5.2) control product flow in each plant. Demand for product i at plant j in period t is either met by available stock, production within the plant or from transfers from other plants, not considering backlogging or sales lost. Transfers among plants occur within the same time period. Without loss of generality, we assume that transfer cost r_{ijkt} satisfies the triangular inequality, $r_{ijkt} \leq r_{ijlt} + r_{ilkt}$ for all $i \in \mathcal{N}$, $(j,k,l) \in \mathcal{P}$, $t \in \mathcal{T}$. Links between setup and production variables are guaranteed in (5.4). Production of product i on filling line m in period t can only occur if the filling line has been set up for the respective family f_i ($Y_{f_i mt} = 1$). Additionally, production is limited to $b_{imt} = min\{cap_{mt} \cdot p_{f_i mt}, \sum_{j\in\mathcal{P}} \sum_{u=t}^{T} d_{iju}\}$.

The problem described is similar to the single stage, multi-plant, multi-item and multi-period capacitated lot sizing problem (MPCLSP) described in Sambasivan and Schimidt (2002); Sambasivan and Yahya (2005); Nascimento et al. (2010). Few papers address this variant of the standard capacitated lot sizing problem (CLSP). In Sambasivan and Schimidt (2002) the authors describe a heuristic to solve the problem based on transfers of production lots. The paper Sambasivan and Yahya (2005) presents a heuristic based on Lagrangian relaxation. The authors dualize capacity constraints and solve the N uncapacitated subproblems via reformulation into a set of shortest path problems with common fixed-charge constraints. Computational experiments are conducted with instances of up to 15 products, 6 periods and 4 plants. Nascimento et al. (2010) propose a greedy randomized adaptive search procedure (GRASP) combined with path-relinking. Results are compared to the method described in Sambasivan and Yahya (2005) and the authors claim to achieve a better performance in terms of the mean gap of the linear relaxation of the problem. In addition, the proposed heuristic was also tested in the parallel machine lot sizing problem, which in fact is a special case of MPCLSP, when transfers among plants are discarded. In this scenario each plant corresponds to a machine.

Although similar to the MPCLSP, our model has different assumptions. As we can not agregate machine resources of the same plant due to technological constraints, each plant may have one or more machines, contrarily to the MPCLSP that assumes a single machine. Moreover, setup times are not considered here (contrarily to Sambasivan and Yahya (2005); Nascimento et al. (2010)) since throughput rates used by the company already include them considering an average lotsize. Still, such a generalization could be easily made considering s_{fmt}, the time to set up family f on machine m in period t, and replacing constraints (5.3) by:

$$\sum_{i\in\mathcal{N}_m^{line}} \frac{X_{imt}}{p_{f_i mt}} + \sum_{f\in\mathcal{F}_m} s_{fmt} \cdot Y_{fmt} \leq cap_{mt}, \quad \forall m \in \mathcal{M}, t \in \mathcal{T}.$$

In addition, production costs are neglected, which could be overcome by introducing the parameter v_{imt} defining the unitary production cost of product i on machine m in period t, and adding production cost into the objective function (5.1):

$$\min \quad Obj_2 = Obj_1 + \sum_{m\in\mathcal{M}} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{N}_m^{line}} v_{imt} \cdot X_{imt}.$$

Finally, setups are not considered in terms of products but rather of product families. Such assumption relies on the purpose of our model. The family setup costs aim to minimize the number of production lines producing the same family as an indicator of future capacity losses at the operational level when sequence dependent setup times are considered. Products within each family are strongly related representing minor setups among them as few characteristics vary from one to another. In other words, sequence-dependent setups among products of different families are much more costly in terms of time and cost. Hence, by minimizing the number of times a family is produced we are also faced with a reduced number of major setups between families in detailed plans. Regardless these observations, if each family is only composed by a single product we end up with setups defined by product.

Further modifications to the model incorporate other important planning decisions. One of them is the use of overtime (Özdamar and Bozyel (2000)). Overtime can be used to face lack of production capacity and is especially important during peak seasons. Distinction is made between overtime on Saturdays and holidays (type I) and overtime on Sundays (type II). Type I overtime is less costly than type II. To introduce these decisions in the model we first need to define the respective parameters and decision variables.

Parameters

$co^{I}_{mt}\,(co^{II}_{mt})$ unitary cost of an extra time unit of type I (type II) overtime on filling line m in period t

$mo^{I}_{mt}\,(mo^{II}_{mt})$ maximum available overtime capacity of type I (type II) on filling line m in period t

Variables

$O^{I}_{mt}\,(O^{II}_{mt})$ overtime of type I (type II) used on filling line m in period t

To integrate overtime decisions in the model, objective function (5.1) must be transformed into:

$$\min \quad Obj_3 = Obj_1 + \sum_{m\in\mathcal{M}}\sum_{t\in\mathcal{T}}\left(co^{I}_{mt}\cdot O^{I}_{mt} + co^{II}_{mt}\cdot O^{II}_{mt}\right), \tag{5.6}$$

and constraints (5.3) become:

$$\sum_{i\in\mathcal{N}^{line}_{m}} \frac{X_{imt}}{p_{f_i mt}} \leq cap_{mt} + O^{I}_{mt} + O^{II}_{mt}, \quad \forall m\in\mathcal{M}, t\in\mathcal{T}. \tag{5.7}$$

Finally, the following constraints impose limits on overtime utilization:

$$0 \leq O^{I}_{mt} \leq mo^{I}_{mt},\ 0 \leq O^{II}_{mt} \leq mo^{II}_{mt}, \quad \forall m\in\mathcal{M}, t\in\mathcal{T}. \tag{5.8}$$

Hence, the overall LT3P model reads:

$$\begin{aligned}
&\min Obj_3\\
&\text{satisfying } (5.2), (5.4), (5.7)-(5.8),\\
&(I_{ijt}, W_{ijkt}, X_{imt}, O^{I}_{mt}, O^{II}_{mt}) \geq 0,\ Y_{fmt}\in\{0,1\}.
\end{aligned}$$

5.5.2 Solution procedure

Standard single-item CLSP has been proven to be NP-hard Bitran and Yanasse (1992), so are the respective multi-item and multi-plant versions. In this paper, we present results that confirm the difficulty of solving to optimality moderate and large size instances, thus motivating heuristics to find approximate solutions to the problem. We propose heuristics inspired by the VNS principles. As briefly mentioned before, VNS systematically exploits the change of neighborhood both in descent to local optima and in escape from them Hansen et al. (2008). VNS relies on local search heuristics that starting from an initial solution x attempt to find an improvement within a neighborhood $N(x)$. Until such an improvement is possible, the heuristic iterates, otherwise it stops. To create a VNS scheme, one must provide a set of pre-defined neighborhoods structures $N_k(k = 1, \dots, k_{max})$ and an initial solution x. The initial solution can be obtained from any simple construction heuristic. A basic scheme of VNS (see Mladenovic and Hansen (1997)) combines stochastic and deterministic changes of neighborhoods using the following three steps that are repeated until the stopping criteria is reached.

Shaking: The stochastic component of the method, where a point x' is randomly generated from $N_k(x)$ in order to avoid cycling.

Local Search: The local search heuristic is applied to x' until a local optimum (x'') is achieved.

Move and Neighborhood Change: If the local optimum x'' found during the search is better than the incumbent best solution x, then x'' is accepted and replaces x, setting $k = 1$. Otherwise, the algorithm proceeds to the next neighborhood structure $k = k + 1$ (if $k > k_{max}$, then $k = 1$).

Several VNS variants have been developed since it first appeared to solve many combinatorial optimization problems. One of the best known variants is the variable neighborhood descent (VND) method, which is a deterministic version of VNS where several neighborhood structures in sequence within the Local Search phase are searched, but no Shaking step is performed. Furthermore, VND can replace the Local Search phase in VNS giving origin to General VNS (GVNS). When local search is costly in computational terms, these methods can suffer from efficiency problems. The Reduced VNS (RVNS) is useful in such cases, as it is a pure stochastic method where random points are generated from N_k and the incumbent solution is updated in case of an improvement. Naturally, this variant reduces the effectiveness of the search. VNS design and consequently its efficiency and effectiveness are closely related to the selection of neighborhoods and their order. VNS conducts the search through different neighborhoods usually in increasing distances, evaluated by some metric (or quasi-metric). In 0-1 mixed integer problems like ours, the distance between two solutions can be based on the Hamming distance (Δ_H) that states the number of elementary changes in 0-1 variables to turn one solution into another. Most applications rank the neighborhoods in increasing order of their complexity, which usually corresponds to a bigger Hamming distance. Moreover, the use of nested neighborhoods, i.e., $N_1(x) \subset N_2(x) \subset \ldots \subset N_{k_{max}}(x)$, is often a common choice. Still, the understanding of the problem structure can be crucial to a suitable choice of neighborhoods and their sequence. VNS also represents a trade-off between intensification of the search (Local Search) and diversification (Shaking), which is important to balance.

5.5.2.1 Solution representation

We use an incomplete representation of the solution, considering the different permutations of the binary family setup variables Y_{fmt}, which are controlled by our algorithm. Given a fixed set of binary values (setup pattern Y'_{fmt}), by replacing them in our MIP model, the remaining problem can be solved optimally as an LP. Hence, production, inventory, transfer and overtime decisions are dependent on the setup pattern. The neighborhood structures are induced from the changes in the setup pattern.

5.5.2.2 Initial solution

Finding a feasible solution for the LT3P is difficult, specially in tight capacity scenarios. In our case, it implies selecting a setup pattern, which can be translated into a plan that verifies demand and capacity constraints. We overcome this problem by introducing artificial decision variables defined as the initial stock of product i at plant j (I^0_{ij}). This initial stock is heavily penalized in the objective function using h^0_{ij}. Doing so, we allow any setup pattern to be feasible, which is also important during the execution of the algorithm. Under these conditions, the term $\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{P}} h^0_{ij}\cdot I^0_{ij}$ becomes a measure of infeasibility. Nevertheless, for the case study, if real-world initial stock is considered, it is not penalized in the objective function.

Even though solutions with empty setups or with setups for every family in each period are now feasible, we want to test the impact of the initial solution on the search efficiency and efficacy. For that purpose we have developed three procedures to define an initial setup pattern.

1. *LotForLot* is inspired in a lot-for-lot policy. The procedure works period-by-period and plant-by-plant identifying the total gross requirements for a certain product family $R_{fjt} = \sum_{i\in\mathcal{N}^{fam}_f} d_{ijt}$. If $R_{fjt} > 0$, a setup will be triggered on machine $m \in M_j$ having the highest throughput rate p_{fmt}.

2. The second and third procedures are both inspired by the work of Nascimento et al. (2010). Ignoring capacity constraints and/or inter-plant transfers, the problem can be compared to F uncapacitated lot sizing problems on parallel machines, which are solvable through the optimal algorithm of Sung (1986).

   a) *Uncap* works plant by plant and a minimum production schedule is found for each family satisfying the demand for the incumbent plant (j) having as potential sources the filling lines belonging to that same plant ($\mathcal{M}_j$), thus only ignoring capacity constraints.

   b) *UncapNoTransf* attempts to find a production schedule for each family satisfying the demand for all plants having as potential sources the set of available filling lines ($\mathcal{M}$), therefore ignoring both capacity constraints and transfer costs.

Next, we describe the general procedure for *UncapNoTransf* as it is a generalization of *Uncap*. For each family f let ϕ_{fmut} be the production cost of family on production line m in period u to meet requirements D_{iut} for all products $i \in \mathcal{N}_f^{fam}$ from periods u to t for all plants:

$$D_{iut} = \sum_{j \in \mathcal{P}} \sum_{s=u}^{t} d_{ijs},$$

$$\phi_{fmut} = c_{fmu} + \sum_{i \in \mathcal{N}_f^{fam}} \left(v_{imu} \cdot D_{iut} + \sum_{s=u+1}^{t} h_{ij_m s} \cdot D_{ist} \right),$$

Moreover, let φ_{ft} be the minimum production cost from period 1 up to period t ($\varphi_{f0} = 0$). Quantity φ_{ft} can be obtained recursively using:

$$\varphi_{ft} = min \left\{ \varphi_{f,u-1} + \phi_{fmut} \right\} \quad u \in 1 \ldots t, \, t \in \mathcal{T}.$$

A dynamic programming forward recursion algorithm has been used to solve each subproblem and thus fixing the setup pattern. The assumptions made during this procedure allow us to have a rough approximation of a possible interesting setup pattern.

5.5.2.3 Neighborhood structures

Neighbors of an incumbent solution x are obtained by slightly changing the setup pattern and solving afterwards the resulting LP. The set of all possible minor changes, also called moves, constitute the neighborhood $N(x)$. We have defined three different type of moves:

a) *insertion*(f,m,t) consists in changing the setup state of family f on machine m at period t from 0 to 1, therefore neighborhood $N_I(x)$ include all possible changes of the variables Y'_{fmt} from 0 to 1,

b) *remove*(f,m,t) is the inverse move of *insertion*, thus $N_R(x)$ are the potential changes of variables Y'_{fmt} from 1 to 0,

c) *transfer*(f,m_o,t_o,m_d,t_d) reallocates a production lot by means of moving the setup of family f from its origin (machine m_o at period t_o) to a new destination (machine m_d at period t_d). The neighborhood $N_T(x)$ corresponds to all possible moves where $Y'_{fm_ot_o} = 1$ and $Y'_{fm_dt_d} = 0$.

All neighbor solutions of $N_I(x)$ and $N_R(x)$ have a $\Delta_H = 1$, while neighbor solutions of $N_T(x)$ have $\Delta_H = 2$. The size of the neighborhood $N_T(x)$ can be controlled by setting limits to $t_d = [t_o - \delta_b, t_o + \delta_f]$, when δ_b and δ_f control the backward and forward searching ranges, respectively. Note that when $t_o \neq t_d$ and only in this case m_o may equal to m_d.

These three types of neighborhoods try to explore different ideas. The *insertion* move attempts to find a new family allocation such that the additional setup cost incurred is shorter than the savings resulting from production, holding, transfer and overtime costs.

On the other hand, its inverse *remove* tries the opposite allowing an increase of the other costs through the mitigation of setup costs. Moves resulting from *transfer* are more difficult to interpret. When changing a setup within the same machine we attempt to introduce or eliminate inventory, whether we try backward or forward movements, and possibly decrease overtime costs. Moving setups to other machines in the same period tries to save setup costs that may benefit potential transfers. Moving to other machines in different periods can cause or eliminate inventory, transfers and overtime and eventually reduce setup costs. *Transfer* moves also seek for machine load balancing in tight capacity problems. Nevertheless, when evaluating a neighbor, the LP mathematical model is optimized over the entire planning horizon, therefore performing a change in the setup pattern can have multiple effects on the production, inventory, transfer and overtime decisions, and the above description is myopic in those cases. In fact, this constitutes the reason for the partial solution representation, since heuristically determining production, inventory, transfer and overtime quantities may wrongly reject interesting setup patterns. The price to pay for such decisions is a more costly local search in terms of computational times.

5.5.2.4 Algorithm design

In theory, the larger the neighborhood, the better is the quality of locally optimal solutions, and the greater the accuracy of the final solution obtained. Standard VNS examines the entire neighborhood during local search. For large problem instances it is impractical to search the neighborhood exhaustively as it can be too time consuming. In practice, strategic/tactical decisions can be taken in a relatively wide time window, however this is only true if a single plan is to be created. Frequently, these plans are obtained by studying different scenarios varying data inputs substantially reducing the available time for response. Therefore, it is required to partially search the neighborhood in an efficient manner.

To speed-up the algorithm, the evaluation of each neighbor can incorporate rules to quickly identify expensive neighbors and save time in the LP optimization. Since moves are performed based on a known solution, plenty of information is available. In addition, changes in the setup pattern are usually rather small. Solving the LP from scratch can be very time-consuming, thus the previous best found solution constitutes the initial basis in the new LP and then it is just re-optimized. To early discard expensive neighbors, let z^t_{best} denote the best solution found to date and z^s_{best} the respective setup pattern cost. The remaining costs related to production, inventory and transfers calculated by the LP are expressed as $z^o_{best} = z^t_{best} - z^s_{best}$. After generating a new neighbor, $z^s_{neighbor}$ can be easily computed based on z^s_{best}, therefore we can reject a neighbor whose $z^s_{neighbor} > z^t_{best}$ without making any iteration in the LP. Besides, $z^t_{best} - z^s_{neighbor}$ is the maximum value that $z^o_{neighbor}$ can take before being refused. By the solving the LP using a dual simplex method, at each iteration the dual solution corresponds to a lower bound on $z^o_{neighbor}$. Thus the method can be stopped as soon as the lower bound exceeds $z^t_{best} - z^s_{neighbor}$, potentially saving precious computational time. If the method does not stop the LP optimization in course, it means that we have found a new best solution. This technique has already been explored by Meyr (2002).

Although these rules can save valuable computational time, for large size instances the result of exhaustively exploring the entire neighborhood still remains unsatisfactory. The ultimate goal is to somehow explore only a portion of each large neighborhood and still find the local optima, or at least find an improvement move, if such a move exists. Traditional techniques to improve efficiency of VNS such as the aforementioned RVNS often compromise efficacy, especially because neighbors are selected randomly. Hung et al. (2003) propose in the context of Tabu Search the usage of ranking heuristics based on the information provided by the LP to prune the search of the neighborhood. Their techniques were able to reduce running time through one of two strategies: explore a portion of the ranked neighbors according to the heuristics or to evaluate sequentially the ranked neighbors until an improvement is found. They have developed heuristics to rank neighbors that are obtained either by *insertion* or *remove* moves.

RVNS can be a solution to explore large neighborhoods due to its smaller CPU effort vital in the case study. Still, the randomness of the Shaking Phase can lead the algorithm to randomly suggest expensive neighbors too often, despite that the dual reoptimization process may perform an early rejection. Inspired by the work Hung et al. (2003), we have designed new rules to improve the standard RVNS. The idea is to associate a probability to each neighbor according to its potential cost savings. For that purpose we make use of the information available after solving the LP. Let us define β_{mt} and π_{imt} as the shadow prices (dual variables) of constraints (5.7) and (5.4), respectively. Additionally, we define SL_{mt} as the surplus of capacity on machine m in period t in the current best solution. An *insertion*(f,m,t) move can be evaluated through the criteria presented in Algorithm 1. Initially the potential improvement of an *insertion*(f,m,t) move is the cost of the extra setup that has to be performed (line 1). Then the maximum production quantity of family f is determined considering the surplus of capacity on machine m in period t (line 2). The procedure then iterates through the products belonging to family f ($\mathcal{N}_f^{fam}$) selecting the one with maximum value of π_{imt} (line 5). Let Θ be the set of products selected previously. The potential improvement is increased by the term $\pi_{i_{min}mt} \cdot a$, where a may equal the total demand of the selected product at the plant of machine in the incumbent period (line 7) or the maximum of the remaining surplus of capacity (line 10). In both situations, the remaining surplus of capacity, and Θ are updated (lines 8, 11 and 13). The algorithm loops until no remaining surplus is available or all products belonging to the family have been selected ($\Theta = \mathcal{N}_f^{fam}$). Move *remove*$(f,m,t)$ is evaluated according to Algorithm 2. Its initial potential improvement is the saving coming from removing the existent setup (line 1). The potential improvement is then updated (line 3) using β_{mt} over the total production quantity of family f determined in line 2. Evaluating a *transfer*(f,m_o,t_o,m_d,t_d) move is hard because it introduces more changes in the model and therefore the available information is less reliable, yet it can be seen as a combination of an *insertion*(f,m_d,t_d) and a *remove*(f,m_o,t_o), thus summing both potential improvements. Shadow prices can be seen as the marginal utility of the resources. Move *insertion*(f,m,t) explores the marginal utility of an additional setup of family f on machine m in period t assuming that it remains valid for the maximum between the surplus of capacity and the total demand of the family. The same principle is behind the evaluation of *remove*(f,m,t), but this time making use of the marginal utility of “freeing" capacity. The potential improvements are only estimations of the real improvement on the objective function, thus we should not restrict too much

Algorithm 1 Potential improvement of an *insert*(f,m,t) move

Potential improvement: $Imp_{fmt} := -c_{fmt}$
Maximum family production: $MaxProd = \frac{SL_{mt}}{p_{fmt}}$
Current produced products: $\Theta \leftarrow O$
while $MaxProd > 0$ **and** $\Theta \subset \mathcal{N}_f^{fam}$ **do**
  $i_{min} = argmax\{\pi_{imt} \,|\, i \in \mathcal{N}_f^{fam} \setminus \Theta\}$
  if $MaxProd > d_{i_{min} j_m t}$ **then**
    $Imp_{fmt} = Imp_{fmt} + \pi_{i_{min} mt} \cdot d_{i_{min}, j_m, t}$
    $MaxProd = MaxProd - d_{i_{min} j_m t}$
  else
    $Imp_{fmt} = Imp_{fmt} + \pi_{i_{min} mt} \cdot MaxProd$
    $MaxProd = 0$
  $\Theta \leftarrow \Theta \cup \{i_{min}\}$
return Imp_{fmt}

Algorithm 2 Potential improvement of a *remove*(f,m,t) move

Potential improvement: $Imp_{fmt} := c_{fmt}$
Actual family production: $X_{mt}^f = \sum_{i \in \mathcal{N}_f^{fam}} X_{imt}$
$Imp_{fmt} := Imp_{fmt} + \beta_{mt} \cdot \frac{X_{mt}^f}{p_{fmt}}$
return Imp_{fmt}

the search based on them. After calculations are made for all neighbors in the incumbent neighborhood, the candidate neighbors $r \in N_k(x)$, are sorted according to their potential improvement (Imp_{fmt}). Let $\sigma(r)$ be the rank of neighbor r. The probability $\mu(r)$ of choosing a candidate neighbor is given by:

$$\mu(r) = \frac{bias(r)}{\sum_{r' \in N_k(x)} bias(r')}$$

where $bias(r)$ is called the bias function. Pure RVNS makes use of a random bias, i.e. $bias(r) = 1$. Since our idea is to prioritize candidates at the top of the list, any of the following bias functions can be used: linear bias $bias(r) = 1/\sigma(r)$, log bias $bias(r) = log^{-1}(\sigma(r) + 1)$ and exponential bias $bias(r) = e^{-\sigma(r)}$. Exponential bias is the most extreme case were mostly the top candidates are chosen, liner bias is less extreme than exponential and log bias is the least differentiator function. To select the candidate neighbor to explore a random number is generated according to an uniform distribution and compared to the probabilities calculated. We call this enhancement to standard RVNS as Adaptive Reduced Variable Neighborhood Search (ARVNS).

5.6. Production plans validation and comparison

In this section we present computational experiments divided in two sections. First we validate our solution procedures on a randomly generated set of small to medium sized instances (*Tests I*). Afterwards, we use the algorithm with the best performance to solve two real-world instances based on the annual production budget of the case study for the years of 2010 and 2011 (*Tests II*). All heuristics were implemented in C++, compiled using Microsoft Visual Studio 2008 and run on an Intel Core i7 Q720 1.60 GHz processing unit with 6 GB of random access memory, using a single core. IBM ILOG Cplex 12.1 was used both as mixed integer and liner programming solver and was limited to one thread to have a fair comparison.

5.6.1 Tests I

This set of tests is designed to validate the proposed algorithm and prove its superiority against other variants reported in the literature. Nevertheless, the features from the case study instances are kept, such as the absence of setup times and production costs, and the use of overtime. All parameters with the exception of demand are considered to be time independent, for example $p_{fmt} = p_{fm}$, $\forall t \in \mathcal{T}$. This applies only for the generated instances since our heuristics can manage time dependent parameters. Input parameters of each problem instance were generated based on the uniform distribution. The ranges used for the parameters are given in Table 5.1.

Table 5.1: Parameter ranges

d_{ijt}	U[40,180]	p_{fmt}	U[1,5]
h_{ijt}	U[0.2,0.4]	co^{I}_{mt}	U[1,2]
r_{ijkt}	U[0.2,0.4]	co^{II}_{mt}	U[3,6]
c_{fmt}	U[50,950]		

Available capacity of all machines of the same plant is calculated according to:

$$cap_{mt} = \sum_{m \in \mathcal{M}_{jm}} \frac{\sum_{i \in \mathcal{N}_f^{fam}} d_{ij_mt}}{p_{fmt}} \cdot (\alpha|\mathcal{M}_j|)^{-1},$$

with $\alpha = 1.25$. The maximum amount of both types (mo^{I}_{mt}, mo^{II}_{mt}) of overtime is set to 10% of the available capacity. Test classes are defined by the quintuplet (F, N, P, M, T). The number of families F and the number of plants P are always less or equal to the number of products N and the number of machines M, respectively. The process to assign product to families and machines to plant is the same. For example, if 5 products have to be assigned to 3 families, the first 3 are assigned each one to a different family and the remaining 2 are randomly allocated to a family. Tests were conducted using $F = \{5, 10, 15\}$, $N = 2F$, $P = \{3\}$, $M = \{4, 6\}$ and $T = \{6, 9, 12\}$ and for each combination 10 different instances were generated, corresponding to a total of 180 instances.

We have run each test instance using Cplex 12.1 on the mathematical formulation LT3P presented in Section 5.5 with a maximum running time of 600s (time required by the company to have a solution for a new scenario). Thus, at the end of each run we potentially have an upper bound (the current best integer solution found by the branch-and-cut algorithm) and a lower bound also provided by the same algorithm. The mean gap obtained through Cplex 12.1 is our solution evaluation metric. The percentage Gap to the best known lower bound is then computed as:

$$Gap = \frac{z_h - z_{lb}}{z_{lb}} \cdot 100,$$

where z_h is the solution obtained by the method under evaluation and z_{lb} is the best lower bound known provided by Cplex 12.1. All instances were feasible without considering initial inventory. Furthermore, for some problems the optimal solution was found. Table 5.2 reports the number of probabily optimal solutions (out of 10) found in each test class by Cplex 12.1. Naturally, as the number of families, machines and periods increase, the number of instances solved until optimality decreases sharply. We then tested our solution approach for two variants: RVNS and ARVNS. RVNS relies on a random bias function, while ARVNS makes use of a linear bias function. Neighborhoods were ordered according to *insertion*, *transfer* and *remove*, as it was proved during pre-testing to be the most promising sequence. The maximum number of successive iterations without improving was used as stopping criterion and set to 1000. Both methods were run for the three different types of initial solutions (*LotForLot*, *Uncap* and *UncapNoTransf*). Ten runs were executed for each instance using the diferent configurations: initial solution and solution approach variant.

Table 5.2: Number of optimal solutions found by Cplex 12.1 for the different test classes

		T		
F	N	6	9	12
$P = 3$, $M = 4$				
5	10	10	8	4
10	20	9	2	0
15	30	2	0	0
$P = 3$, $M = 6$				
5	10	10	7	4
10	20	6	1	0
15	30	5	0	0

Tables 5.3 and 5.4 report the average solution gap and Tables 5.5 and 5.6 present the average running times for the three methods under evaluation in the different test classes. The performance of our solution approach clearly depends on the initial solution. *UncapNoTransf* yields the overall best mean gap and *Uncap* generally outperforms *LotForLot*, in particular when using ARVNS. Running times increase as the problem size increases, specially when using *LotForLot* as initial solution. Regarding solution quality, for small sized instances exact methods have, as expected, the best mean gaps. Nevertheless, for medium

Table 5.3: Results for the average (minimum;maximum) Gap (%) for the different test classes with $P = 3$ and $M = 4$ for the methods under evaluation. Best average gaps are in boldface.

	LotForLot		Uncap		UncapNoTransf		
{T,F,N}	*RVNS*	*ARNVS*	*RVNS*	*ARNVS*	*RVNS*	*ARNVS*	*CPLEX*
{6,5,10}	1.24 (0.22 ; 2.73)	1.13 (0.06 ; 4.02)	0.96 (0.06 ; 1.93)	0.89 (0.05 ; 2.26)	1.21 (0.26 ; 2.62)	0.67 (0.02 ; 1.92)	**0.00** (0.00 ; 0.00)
{6,10,20}	0.83 (0.13 ; 1.31)	0.40 (0.03 ; 0.93)	0.92 (0.28 ; 1.42)	0.42 (0.11 ; 1.29)	0.86 (0.34 ; 1.35)	0.27 (0.05 ; 0.74)	**0.04** (0.00 ; 0.39)
{6,15,30}	1.91 (0.72 ; 3.43)	1.28 (0.27 ; 2.62)	1.92 (1.05 ; 3.34)	1.31 (0.38 ; 2.79)	1.72 (0.70 ; 3.49)	1.12 (0.09 ; 2.94)	**0.77** (0.00 ; 2.48)
Mean	*1.33*	*0.93*	*1.27*	*0.87*	*1.26*	*0.69*	***0.27***
{9,5,10}	1.57 (0.30 ; 3.85)	0.97 (0.28 ; 2.16)	1.67 (0.31 ; 3.93)	0.89 (0.24 ; 2.11)	1.39 (0.37 ; 2.83)	0.86 (0.16 ; 2.17)	**0.16** (0.00 ; 1.22)
{9,10,20}	2.75 (0.58 ; 5.36)	2.42 (0.24 ; 5.29)	2.89 (0.52 ; 5.85)	2.48 (0.15 ; 5.24)	2.64 (0.61 ; 5.09)	2.20 (0.03 ; 4.54)	**2.14** (0.00 ; 4.91)
{9,15,30}	4.92 (2.47 ; 6.56)	4.41 (1.95 ; 5.93)	4.83 (2.69 ; 6.70)	4.43 (1.97 ; 5.94)	4.89 (2.65 ; 6.52)	**4.33** (1.84 ; 6.17)	4.70 (1.74 ; 6.93)
Mean	*3.08*	*2.60*	*3.13*	*2.60*	*2.98*	*2.46*	***2.33***
{12,5,10}	3.19 (0.79 ; 7.69)	2.57 (0.26 ; 8.91)	2.55 (0.79 ; 4.35)	2.11 (0.18 ; 4.28)	2.88 (0.46 ; 6.11)	2.61 (0.01 ; 8.40)	**1.20** (0.00 ; 3.16)
{12,10,20}	5.67 (1.90 ; 7.61)	5.12 (1.36 ; 6.83)	5.60 (2.43 ; 7.43)	5.26 (1.48 ; 7.14)	5.62 (2.19 ; 7.47)	**5.07** (1.34 ; 7.13)	5.84 (1.51 ; 8.33)
{12,15,30}	5.71 (3.14 ; 6.94)	5.10 (2.69 ; 6.43)	5.79 (3.34 ; 6.95)	5.15 (2.84 ; 6.42)	5.49 (3.21 ; 6.71)	**4.88** (2.57 ; 6.31)	7.90 (3.00 ; 16.10)
Mean	*4.86*	*4.26*	*4.65*	***4.17***	*4.66*	*4.19*	*4.98*

Table 5.4: Results for the average (minimum;maximum) Gap (%) for the different test classes with $P = 4$ and $M = 6$ for the methods under evaluation. Best average gaps are in boldface.

	LotForLot		Uncap		UncapNoTransf		
{T,F,N}	*RVNS*	*ARNVS*	*RVNS*	*ARNVS*	*RVNS*	*ARNVS*	*CPLEX*
{6,5,10}	1.20 (0.43 ; 1.92)	0.39 (0.06 ; 1.08)	1.02 (0.28 ; 1.79)	0.48 (0.11 ; 1.62)	0.92 (0.41 ; 1.80)	0.49 (0.00 ; 1.96)	**0.00** (0.00 ; 0.00)
{6,10,20}	1.49 (0.77 ; 3.10)	0.63 (0.08 ; 2.23)	1.68 (1.03 ; 3.17)	0.71 (0.21 ; 2.68)	1.26 (0.57 ; 2.49)	0.60 (0.04 ; 2.11)	**0.34** (0.00 ; 2.12)
{6,15,30}	1.91 (0.74 ; 3.20)	0.81 (0.06 ; 1.86)	1.96 (1.08 ; 3.02)	0.91 (0.09 ; 2.11)	1.76 (0.72 ; 2.74)	0.69 (0.01 ; 1.85)	**0.63** (0.00 ; 2.37)
Mean	*1.53*	*0.61*	*1.55*	*0.70*	*1.32*	*0.59*	***0.32***
{9,5,10}	1.76 (0.66 ; 3.59)	0.68 (0.03 ; 2.32)	1.60 (0.22 ; 4.07)	0.73 (0.03 ; 2.48)	1.75 (0.84 ; 4.16)	0.75 (0.03 ; 2.34)	**0.29** (0.00 ; 1.67)
{9,10,20}	3.74 (0.88 ; 6.73)	2.68 (0.15 ; 5.44)	3.54 (0.68 ; 6.90)	2.74 (0.11 ; 5.33)	3.44 (0.98 ; 5.92)	**2.56** (0.15 ; 5.37)	2.75 (0.00 ; 6.71)
{9,15,30}	3.81 (1.05 ; 5.55)	2.90 (0.46 ; 4.77)	3.75 (1.47 ; 5.28)	2.98 (0.55 ; 4.83)	3.86 (1.92 ; 5.20)	**2.79** (0.29 ; 4.47)	2.85 (0.19 ; 4.74)
Mean	*3.10*	*2.09*	*2.96*	*2.15*	*3.02*	*2.03*	***1.96***
{12,5,10}	2.92 (0.57 ; 5.07)	2.28 (0.08 ; 5.13)	3.09 (0.95 ; 5.44)	2.29 (0.04 ; 4.87)	2.86 (0.34 ; 5.57)	2.12 (0.05 ; 4.75)	**1.64** (0.00 ; 3.97)
{12,10,20}	4.66 (1.88 ; 6.77)	3.70 (0.47 ; 6.37)	4.43 (0.98 ; 6.78)	3.70 (0.44 ; 6.26)	4.71 (2.29 ; 6.73)	**3.57** (0.45 ; 6.25)	4.87 (0.12 ; 14.03)
{12,15,30}	4.40 (3.23 ; 5.31)	3.34 (2.34 ; 4.24)	4.27 (3.48 ; 5.34)	3.43 (2.35 ; 4.28)	4.23 (3.05 ; 4.97)	**3.16** (2.27 ; 4.05)	5.39 (2.11 ; 13.14)
Mean	*3.99*	*3.11*	*3.93*	*3.14*	*3.93*	***2.95***	*3.97*

Table 5.5: Results for the average (minimum;maximum) running times (s) for the different test classes with $P = 3$ and $M = 4$ for the methods under evaluation. Best average running times are in boldface.

	LotForLot		Uncap		UncapNoTransf		
{T,F,N}	*RVNS*	*ARNVS*	*RVNS*	*ARNVS*	*RVNS*	*ARNVS*	*CPLEX*
{6,5,10}	1.2 (1.0 ; 1.5)	1.1 (0.8 ; 1.6)	1.0 (0.8 ; 1.2)	**0.9** (0.7 ; 1.3)	1.1 (0.9 ; 1.3)	0.9 (0.7 ; 1.3)	12.7 (1.3 ; 40.2)
{6,10,20}	3.1 (2.6 ; 3.7)	2.8 (2.4 ; 3.5)	2.62 (2.2 ; 3.1)	2.1 (1.7 ; 2.5)	2.6 (2.2 ; 3.0)	**1.6** (1.8 ; 1.8)	107.0 (21.0 ; 605.5)
{6,15,30}	6.7 (5.9 ; 7.2)	6.4 (4.7 ; 10.6)	5.34 (4.0 ; 6.7)	5.0 (3.8 ; 6.5)	5.3 (4.8 ; 5.7)	**3.9** (2.1 ; 5.6)	532.4 (84.3 ; 605.2)
Mean	*3.6*	*3.4*	*3.0*	*2.7*	*3.0*	***2.1***	*217.4*
{9,5,10}	2.1 (1.6 ; 2.6)	1.9 (1.3 ; 2.5)	1.7 (1.5 ; 1.9)	1.7 (1.2 ; 2.3)	2.1 (1.5 ; 3.5)	**1.4** (0.9 ; 1.9)	279.0 (20.6 ; 606.5)
{9,10,20}	6.4 (5.5 ; 7.1)	6.2 (4.6 ; 11.4)	5.1 (4.5 ; 6.1)	4.4 (3.3 ; 5.3)	4.9 (4.1 ; 6.4)	**3.0** (1.9 ; 3.8)	552.4 (85.82 ; 611.3)
{9,15,30}	13.2 (11.0 ; 15.7)	13.5 (9.9 ; 16.9)	11.0 (9.1 ; 13.0)	9.8 (7.9 ; 12.4)	9.8 (8.5 ; 11.2)	**7.5** (3.9 ; 12.5)	624.9 (602.9 ; 642.5)
Mean	*7.2*	*7.2*	*5.9*	*5.3*	*5.6*	***4.0***	*485.4*
{12,5,10}	3.2 (2.7 ; 3.6)	3.2 (2.0 ; 4.1)	2.7 (2.3 ; 3.1)	3.2 (1.6 ; 4.8)	2.6 (2.1 ; 2.9)	**2.0** (1.0 ; 2.8)	421.8 (19.4 ; 605.5)
{12,10,20}	10.1 (8.9 ; 11.4)	10.4 (7.5 ; 14.2)	8.2 (6.9 ; 11.0)	7.3 (5.5 ; 9.0)	7.6 (6.8 ; 9.4)	**5.5** (2.4 ; 8.5)	603.0 (602.7 ; 603.4)
{12,15,30}	21.8 (18.5 ; 23.6)	22.9 (19.2 ; 25.8)	16.9 (14.6 ; 19.5)	17.0 (13.2 ; 20.4)	15.4 (11.1 ; 17.8)	**11.4** (8.2 ; 16.0)	617.8 (615.9 ; 621.0)
Mean	*11.7*	*12.2*	*9.3*	*9.2*	*8.5*	***6.3***	*547.5*

Table 5.6: Results for the average (minimum;maximum) running times (s) for the different test classes with $P = 4$ and $M = 6$ for the methods under evaluation. Best average running times are in boldface.

	LotForLot		Uncap		UncapNoTransf		
{T,F,N}	*RVNS*	*ARNVS*	*RVNS*	*ARNVS*	*RVNS*	*ARNVS*	*CPLEX*
{6,5,10}	1.7 (1.4 ; 2.2)	1.5 (1.2 ; 2.0)	1.5 (1.0 ; 2.7)	1.2 (0.9 ; 1.9)	1.4 (1.1 ; 1.9)	**1.0** (0.7 ; 1.5)	20.2 (1.5 ; 47.9)
{6,10,20}	5.1 (4.4 ; 5.6)	4.3 (3.6 ; 5.2)	3.9 (3.4 ; 4.3)	3.3 (2.7 ; 3.9)	4.1 (3.7 ; 4.6)	**2.6** (1.8 ; 3.4)	332.6 (29.2 ; 617.1)
{6,15,30}	10.3 (9.5 ; 11.5)	8.5 (6.5 ; 11.8)	7.4 (6.8 ; 8.6)	**5.4** (4.5 ; 6.6)	8.1 (6.9 ; 9.3)	5.5 (3.5 ; 8.6)	357.1 (36.6 ; 604.7)
Mean	*5.7*	*4.8*	*4.3*	*3.3*	*4.5*	***3.1***	*236.6*
{9,5,10}	3.1 (2.7 ; 3.6)	2.9 (1.6 ; 3.5)	2.5 (2.0 ; 3.3)	**2.4** (1.2 ; 3.0)	2.6 (2.1 ; 3.1)	2.7 (1.5 ; 5.4)	254.4 (1.7 ; 638.6)
{9,10,20}	10.1 (8.8 ; 11.1)	9.1 (6.9 ; 17.4)	8.1 (6.8 ; 10.8)	6.5 (5.3 ; 8.9)	7.7 (6.7 ; 9.6)	**6.1** (4.0 ; 8.3)	578.1 (270.3 ; 636.3)
{9,15,30}	20.1 (17.5 ; 23.8)	19.2 (14.1 ; 22.1)	14.8 (12.5 ; 18.5)	**11.5** (9.4 ; 13.0)	14.4 (11.6 ; 17.0)	11.5 (7.9 ; 16.3)	603.2 (602.2 ; 606.3)
Mean	*11.1*	*10.4*	*8.5*	*6.8*	*8.2*	***6.8***	*478.6*
{12,5,10}	5.2 (4.4 ; 6.5)	4.8 (3.4 ; 7.0)	4.0 (3.2 ; 5.6)	**3.4** (2.2 ; 4.6)	4.2 (3.2 ; 5.1)	3.5 (1.5 ; 6.9)	419.1 (7.7 ; 619.1)
{12,10,20}	17.5 (14.6 ; 20.0)	14.8 (11.3 ; 18.2)	12.4 (10.1 ; 15.5)	9.0 (7.1 ; 11.5)	12.1 (9.4 ; 17.0)	**8.9** (3.9 ; 17.4)	606.91 (603.0 ; 616.8)
{12,15,30}	33.0 (29.5 ; 38.2)	31.3 (25.3 ; 39.4)	24.0 (20.5 ; 28.1)	18.2 (13.3 ; 23.5)	22.4 (18.7 ; 29.7)	**13.7** (5.9 ; 22.6)	602.7 (602.2 ; 603.6)
Mean	*18.5*	*17.0*	*13.5*	*10.2*	*12.9*	***8.7***	*542.9*

size instances, such as test class (15,30,3,4,12) and (15,30,3,6,12) our solution approaches are more competitive than exact methods both in solution quality, running time and robustness (minimum and maximum gaps). Thus, tests show that our solution approaches are more competitive for large-scale problems. Exact methods are less attractive as problems grow in size, specially as the number of periods increases. Enhanced neighbor selection present in ARVNS proved to be profitable. ARVNS is always superior to standard RVNS (with a statistically significant p-value < 0.01) and generally takes less running time. Iterations in ARVNS take longer to perform because both evaluation of potential improvement and sorting of neighbors have to be done to properly calculate $\mu(r)$. Yet, this type of neighbor selection allows the search to converge faster and to a better local optimum. Hence, ARVNS seams to be a very promising tool to effectively explore large neighborhoods and therefore to be used in real-world problems.

5.6.2 Tests II

The second set of instances are based on real data from the case study. There are two instances corresponding to the annual production budget of 2010 and 2011, respectively. Both only consider the planning of beer and soft drinks plants. The instance related to the year of 2009 comprises data from 3 plants, each one having a set of 1 to 5 filling lines, totaling 10 filling lines. Sales budget forecasts are available over the next 12 months for a total of 125 products, which can be aggregated into 62 different product families. Technological restrictions limit family assignments to filling lines, nevertheless more than 100 family-filling line allocations are possible in each time period. In the year of 2011, again the total number of plants is 3, but the number of filling lines increased to a total of 14, ranging between 4 to 5 in each plant. The number of products also increased to over 160, which are now aggregated into 68 different product families. As a result, the number of possible family-filling line allocations is now over 120. For both instances, data related to family throughput rates, product holding and transfer costs, and overtime costs are estimations made by the company based on previous years.

The benchmark was conducted on the following PB scenarios:

U1: Company's PB transformed into a solution of our optimization model (LT3P), thus allowing to compute the objective function.

U2: Fixed family allocation (setup pattern Y'_{fmt}) from the company's PB solving the subsequent problem optimally through the LT3P LP model.

U3: PB obtained using ARVNS with a maximum number of iterations without improvement of 1000, *LotForLot* as the initial solution strategy and the same neighborhood order from the previous tests (configuration with best overall performance in *Tests I*). The final solution is the best among 5 runs.

Nowadays, PB is done using spreadsheets, but is mainly a manual process. Previous experience in PB creation constitutes the pillar of the planning process as it follows implicitly cost based decisions. A comparison with company planning is not always straightforward since a manual planning solution does not always strictly obeys all restrictions. Scenario

U2 tries to reduce this gap by creating the best possible scenario with the current family allocation and also shows the drawback of pre-defined family-filling line allocatios. Table

Table 5.7: Results of the different scenarios for the two real-world instances

	U1	U2	Savings (U2)	U3	Savings (U3)
			2010		
Objective Function	1.962.720	1.509.115	453.605 (23%)	1.275.769	686.951 (35%)
Holding Costs	108.589	185.748	-77.159 (-71%)	134.426	-25.837 (-24%)
Tranfer Costs	204.129	162.247	41.883 (21%)	148.772	55.357 (27%)
Setup Costs	42.550	42.550	0 (0%)	41.650	900 (2%)
Total Number of Setups	851	851	0 (0%)	833	18 (2%)
Overtime Costs	1.607.451	1.118.570	488.881 (30%)	950.921	656.530 (41%)
Objective Function (without Setup Costs)	1.920.170	1.466.565	453.605 (24%)	1.234.119	686.051 (36%)
			2011		
Objective Function	3.259.777	2.163.237	1.096.540 (34%)	1.976.865	1.282.912 (39%)
Holding Costs	450.926	171.429	279.496 (62%)	163.060	287.866 (64%)
Tranfer Costs	317.965	312.157	5.808 (2%)	365.503	-47.538 (-15%)
Setup Costs	48.350	48.350	0 (0%)	46.050	2.300 (5%)
Total Number of Setups	967	967	0 (0%)	921	46 (5%)
Overtime Costs	2.442.536	1.630.334	812.202 (33%)	1.402.252	1.040.283 (43%)
Objective Function (without Setup Costs)	3.211.427	2.114.887	1.096.540 (34%)	1.930.815	1.280.612 (40%)

5.7 reports results for the three scenarios for the two real-world instances. All costs are measured in terms of monetary units (m. u.). Not surprisingly, optimizing production, inventory, transfer and overtime decisions, for the company's family allocation (scenario *U2*) has a strong impact. Manually performing these decisions will likely lead to sub-optimality. Creating PB with ARVNS by relaxing family allocations can further improve these results. Scenario *U2* achieves a total cost saving of 24% and 34% in 2010 and 2011, respectively. A large portion of cost savings cames for overtime reduction, an interesting result since the company is obsessed with the holding costs. Our heuristic obtained the best plans, yielding 35% cost reduction in 2010 and 39% in 2011. Plans clearly show the existing trade-offs among costs. For example, in 2010 both *U2* and *U3* yield higher holding costs than those in the company's plan, while in 2011 transfer costs suggested by *U3* increase as this can lessen overtime. Moreover, scenario *U3* always reduces the total number of setups. Excluding setup costs, all other costs are relatively easy to quantify and, therefore, very accurate. The direct potential cost savings from inventory, transfer and overtime costs in both years are significant, representing 36% and 40%, respectively for 2010 and 2011. Savings in 2011 are bigger because we are considering more products and filling lines. Figure 5.1 and 5.2 help to understand the obtained results. In the 2011 instance, scenario *U1* comprehensibly shows the difficulty of dealing with peak demand that occurs during Summer. Inventory is built up early in the year to face seasonality, in addition during the summer season both transfers and overtime requirements increase. The two optimized scenarios can deal with this effect more smoothly. Scenario *U3* in the last part of the year uses notably less overtime, but it uses more inventory and transfers quantities compared to scenario *U2* as a means to achieve a more cost-efficient plan. The average running times

Figure 5.1: Comparison of inventory, transfer and overtime decisions of the different scenarios for the 2010 PB

of our heuristic were 100s and 180s for the 2010 and 2011 instances, respectively. This confirms the ability of the heuristic of effectively solving large problems.

5.7. Discussion

This paper is motivated by a real-world production planning problem in the beverage industry. The goal is to produce a long-term plan assigning and scheduling product family production lots in a multi-plant environment, having each plant one or more production lines. Total setup, inventory, transfers and overtime costs constitute the objective to minimize. We first formulate the problem as a mixed integer program. Based on our mathematical formulation we have developed a heuristic suitable for the large size instances present in industrial applications. A partial solution representation of product family setup decisions (binary variables) was used and the production, inventory, transfer and overtime quantities (continuous variables) are determined by solving a linear program. We make use of the information provided by sensitivity analysis of the linear program to guide the local search. Neighbors are evaluated and sorted according to their potential improvement and neighbor selection is done according to this rank. We are dealing with very large problem instances from the case study and as tests have proved, the heuristic is able to efficiently explore

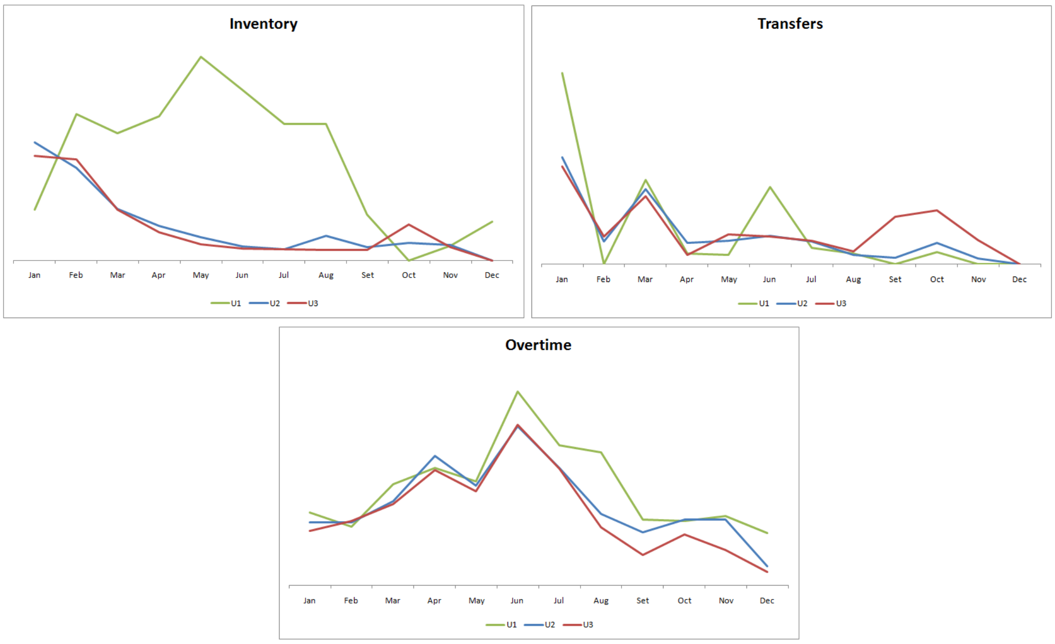


Figure 5.2: Comparison of inventory, transfer and overtime decisions of the different scenarios for the 2011 PB

wide solution spaces. Another important feature of our heuristic is its flexibility, due to the partial solution representation. One can add different requirements to the model, for example production costs, family setup times, minimum family production batches, that were not considered in this application, without having to change the procedure. These adjustments are only needed in the mixed integer problem, which constitutes the base of the linear program.

Tests on real-world instances validated our approach, as we are able to notably improve current company practice. Therefore, this study can constitute the basis for the implementation of a decision support tool for long-term production planning within the company. The test of different planning scenarios and the introduction of a rolling horizon procedure for long-term planning can be features of the planning tool with great capability of enhancing current planning decisions.

It would be interesting to test the new heuristic in other large scale problems to further validate its potential.

Bibliography

B. Almada-Lobo, D. Klabjan, M. A. Carravilla, and J. F. Oliveira. Single machine multi-product capacitated lot sizing with sequence-dependent setups. *International Journal of Production Research*, 45(20):4873–4894, 2007.

B. Almada-Lobo, J. F. Oliveira, and M. A. Carravilla. Production planning and scheduling in the glass container industry: A vns approach. *International Journal of Production Economics*, 114(1):363–375, 2008.

S. A. Araujo, M. N. Arenales, and A. R. Clark. Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. *Journal of Heuristics*, 13(11): 337–358, 2007.

G. Bitran and H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28(10):1174–1186, 1992.

P. Chandra and M. L. Fisher. Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503 – 517, 1994.

A. Clark, R. Morabito, and E. Toso. Production setup-sequencing and lot-sizing at an animal nutrition plant through atsp subtour elimination and patching. *Journal of Scheduling*, 13(2):111–121, 2010.

R. M. Cooke, A. Bosma, and F. Härte. A practical model of heineken's bottle filling line with dependent failures. *European Journal of Operational Research*, 164(2):491 – 504, 2005.

C. Dhaenens-Flipo and G. Finke. An integrated model for an industrial production–distribution problem. *IIE Transactions*, 33(9):705–715, 2001.

N. A. Eskin. *Biochemistry of Foods, Second Edition.* Academic Press, Inc., London, 1990.

D. Ferreira, R. Morabito, and S. Rangel. Solution approaches for the soft drink integrated production lot sizing and scheduling problem. *European Journal of Operational Research*, 196(2):697–706, 2009.

P. Hansen, N. Mladenović, and J. Moreno Pérez. Variable neighbourhood search: methods and applications. *4OR: A Quarterly Journal of Operations Research*, 6(4):319–360, 2008.

Y.-F. Hung, C.-P. Chen, C.-C. Shih, and M.-H. Hung. Using tabu search with ranking candidate list to solve production planning problems with setups. *Computers & Industrial Engineering*, 45(4):615 – 634, 2003.

R. Jans and Z. Degraeve. Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research*, 177(3):1855–1875, 2006.

J. K. Jolayemi and F. O. Olorunniwo. A deterministic model for planning production quantities in a multi-plant, multi-warehouse environment with extensible capacities. *International Journal of Production Economics*, 87(2):99 – 113, 2004.

P. Kaminsky and D. Simchi-Levi. Production and distribution lot sizing in a two stage supply chain. *IIE Transactions*, 35(11):1065 – 1075, 2004.

B. Karimi, S. F. Ghomi, and J. Wilson. The capacitated lot sizing problem: a review of models and algorithms. *Omega*, 31(5):365–378(14), October 2003.

N. L. Kent and A. D. Evers. *Technology of Cereals. An Introduction for Students of Food Science and Agriculture, Fourth Edition.* Elsevier Science, Ltd., Kingston, Oxford., 1994.

L. K. Kourtis and I. S. Arvanitoyannis. Implementation of haccp system to the alcoholic beverages industry. *Food Reviews International*, 17(1):1–44, 2001.

C. H. Martin, D. C. Dent, and J. C. Eckhart. Integrated Production, Distribution, and Inventory Planning at Libbey-Owens-Ford. *Interfaces*, 23(3):68–78, 1993.

C. Matthews. Soft drinks | production. In B. Caballero, editor, *Encyclopedia of Food Sciences and Nutrition*, pages 5352 – 5358. Academic Press, Oxford, 2003.

H. Meyr. Simultaneous lotsizing and scheduling on parallel machines. *European Journal of Operational Research*, 139(2):277 – 292, 2002.

N. Mladenovic and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097 – 1100, 1997.

M. C. Nascimento, M. G. Resende, and F. M. Toledo. Grasp heuristic with path-relinking for the multi-plant capacitated lot sizing problem. *European Journal of Operational Research*, 200(3):747 – 754, 2010.

L. Özdamar and M. A. Bozyel. The capacitated lot sizing problem with overtime decisions and setup times. *IIE Transactions*, 32(11):1043–1057, 2000.

M. Sambasivan and C. P. Schimidt. A heuristic procedure for solving multi-plant, multi-item, multi-period capacited lot-sizing problems. *Asia Pacific Journal of Operational Research*, 19:87–105, 2002.

M. Sambasivan and S. Yahya. A lagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers. *Computers and Operations Research*, 32(3):537–555, 2005.

C. S. Sung. A single-product parallel-facilities production-planning model. *International Journal of Systems Science*, 17(7):983 – 989, 1986.

C. H. Timpe and J. Kallrath. Optimal planning in large multi-site production networks. *European Journal of Operational Research*, 126(2):422 – 435, 2000.

P. H. Tsarouhas and I. S. Arvanitoyannis. Assessment of operation management for beer packaging line based on field failure data: A case study. *Journal of Food Engineering*, 98(1):51 – 59, 2010.

CHAPTER 6

Long term distribution planning

Annual Distribution Budget in the Beverage Industry: a case study

Luis Guimarães* · **Pedro Amorim*** · **Fabrício Sperandio*** · **Fábio Moreira†** · **Bernardo Almada-Lobo***

Submitted to *Interfaces*, 2013 (under revision)

Abstract Nowadays Unicer can improve its tactical distribution planning decisions and study several alternative scenarios for its supply strategies and network configuration thanks to an operations research driven process. In this paper we present the decision support system responsible for this new methodology in the major Portuguese beverage company. At the core of this system there is a mathematical programming-based heuristic that has decision variables related to transportation and inventory management problems. The company runs a set of production and distribution platforms with different characteristics to fulfill customers demand. The main challenge of this work was to render a tactical distribution plan, also known in the company by annual distribution budget, as realistic as possible without jeopardizing the nature of the strategic/tactical tool. The company presents a very complex tactical distribution planning due to the increasing variety of stock-keeping-units and to the need of a very flexible distribution network to satisfy customers, who demand a very fragmented product basket. One of the main causes of this complexity is the existence of uncommon flows of finished products from the distribution centers to the production platforms. These movements yield an intricate supply chain that needs to be properly handled.

The quality of the solutions provided and the implementation of a user friendly interface and very readable and editable inputs/outputs for the decision support system gave the necessary motivation for its wide use by the company practitioners. The corollary of the utilization of this tool translates on a potential cost reduction of about 2M euro per year, on the quality of information made available to decision makers and on their engagement in looking to operations from a different perspective having a more operations research reasoning.

Keywords Tactical Distribution Planning · Mixed-Integer Programming · Decision Support System · Beverage Industry · Supply Chain Management

*INESC TEC, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal

†INEGI, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal

6.1. Introduction

Beverage and food industries have a huge impact on the European Union (EU) economy. In fact, EU is the largest producer of food and beverages, ranking first in terms of sales and exports (in value). In Portugal, the scenario is not different and the food industry generates a market value of over 10.6 billion of euros, which is approximately 7.3 percent of the total gross domestic product (Instituto Nacional de Estatítica, 2011). The beverage industry faces ever increasing competition and companies that want to strive have to excel in terms of price, quality and customer service. To that end, distribution efficiency and efficacy has become a major point. Indeed, many authors claim that in order to build up a sustainable competitive advantage there is just about supply chain management to achieve it (e.g. Fearne and Hughes, 1999).

Generally, beverage companies supply a variety of products that range from wine to beer or water. In fact, these companies pertain to the more general fast moving consumer goods industry and are affected by the same issues. Hence, the beverage market is becoming more and more demanding. This fact translates into an increasing variety of stock-keeping-units as well as on the need for a very flexible distribution network to fulfill customers' demand, which relates to a very fragmented product basket. The combination of a large products portfolio, complex distribution networks and demanding customers give rise to very intricate supply chains that need to be properly handled. The problems that these companies face can be found on different levels. On a strategic level it can be important to know where to locate production or distribution platforms; on a tactical level, a company may be faced upon the decision of choosing which logistics providers to select and with which kind of contract, or in which days the clients should be visited; on a more operational level, there exists the daily problem of the design as well as the consolidation of routes to serve customers previously assigned to that day based on their demand orders.

This paper describes the work done closely with Unicer, the major Portuguese beverage company part of the Carlsberg group, with a revenue of around 500 million euros. Unicer's operations include the production, commercialization and distribution of beer, plain and sparkling water, soft drinks and wine. The project aim was to design, develop and implement an operations research (OR)-based approach to support managers to take their tactical decisions concerning *distribution planning*. However, the operational complexity is not completely put aside. Unicer has used and still runs the tool that emerged from this project not only to perform its annual budgeting of the distribution operation at a tactical level, but also to model and test alternative strategies for the supply chain.

Next we detail the problem addressed, the solution approach developed and discuss the impact of our work in the company. We also discuss important aspects of applying OR to practical problems. We finish by withdrawing some concluding remarks and future project developments.

6.2. The Challenge

Unicer holds some of the Portuguese most popular brands of beer, soft drinks, plain and sparkling water. Today the company sells more than 380 SKUs to over 19,000 different clients across the globe.

Tactical distribution planning is a vital step at Unicer's planning tasks as transport costs correspond to a significant share of the total product cost. This process is under the responsibility of the logistics department director, the main stakeholder of the plans. The logistics director reports the achieved results directly to the company's board, namely the Chief Operations Office - COO. Tactical distribution planning is important at two phases of the company planning process.

The first concerns the creation of the *annual distribution budget* (DB). DB is part of the company's annual budgeting process. The budgeting is a vital tool to align company goals and translate the strategy defined into the next 12 months. This process starts in mid September and lasts until late October. The first main task is the creation of an *annual sales budget* (SB). SB is responsibility of the sales department and defines a monthly sales forecast for each product for the following year. With this input the production planning department works on the *annual production budget* (PB) which defines the total production quantities of each product at each production platform for the entire planning horizon of the 12 months. The results of these two steps define the input required for the DB.

The second important phase of application of the tactical distribution planning involves the validation of strategy changes in the supply chain configuration, product portfolio, clients supply mode and negotiations with the outsourced companies responsible for the transports. Whenever these situations happen, the company studies their impact in terms of future distribution costs. The process is similar to the creation of the DB, however the planning horizon usually increases and data is often more aggregated.

The challenge present in this project is therefore twofold. First, we aim to create a tactical distribution plan for the next 12 months, the so-called DB. The plan details the flow of the finished products among the different locations of the supply chain, by knowing the production plan for a set of production facilities and the customers demand for the next year. Simultaneously, it defines the supply chain configuration by deciding which platforms are operating and their respective activity level. Second, we intent to build an approach capable of modeling various scenarios for the supply chain network to provide a flexible tool. To better understand the problem at hand, in the following subsections we describe the main entities and movements in the supply chain (*Platforms*, *Clients* and *Transports*) associating the expected outputs in each area both at tactical and strategic level.

6.2.1 Platforms

The company has nine production platforms spread across the country and specialized in producing different types of product families. The production facilities dedicated to beer and soft drinks are strategically located close to the geographical center of demand, while, mineral and sparkling water plants are restricted to be placed near a water source often distant from the final consumer.

Distribution platforms are used to storage purposes, consolidate shipments and perform picking operations. The company holds two major distribution platforms, one is located near the Oporto region and the other in the region of Lisbon, the two main consumption areas in Portugal. Other distribution platforms are available across the country, but are much smaller.

Additionally, some of the production platforms also have available areas to store products and supply clients' orders. Therefore, they act as both production and distribution sites. Such feature is uncommon and introduces an additional level of complexity to the supply chain management.

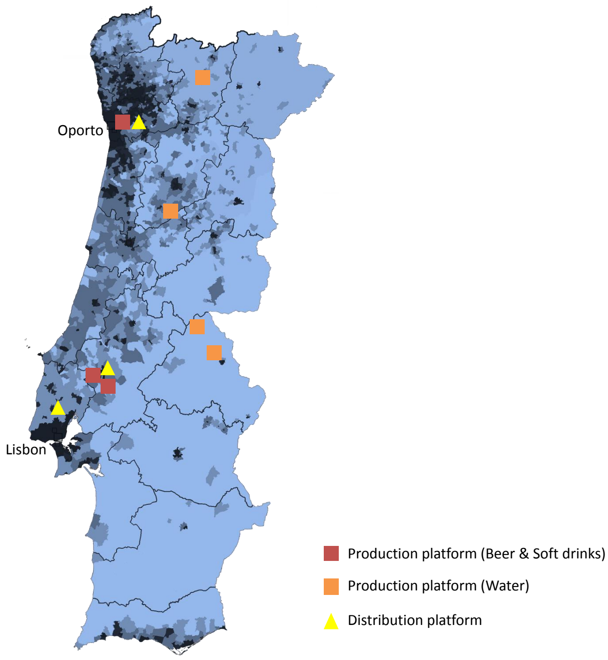
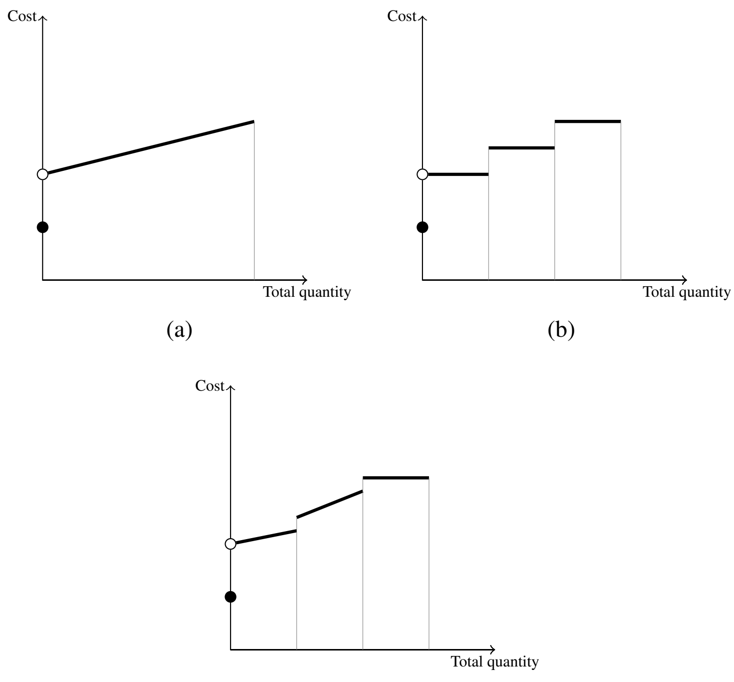


Figure 6.1: Geographic location of Unicer's main production and distribution platforms. The darker the area in the map the higher the population density of the region. In addition there are other smaller platforms not shown in this figure.

Platforms have limited capacity for storage, pallet movement, picking operations and loading shipping containers. Storage capacity is detailed into three different types of pallet storage: (1) drive-in, (2) racks and (3) floor stacking. In drive-in and rack pallet storage, the amount of slots available for pallets is strictly defined. Whereas, for the floor stacking capacity one has to take into account the number of stacking levels that a given product pallet allows for. The available capacities are determined by the activity level selected. Figure 6.2 depicts possible cost curves for operations at different activity levels. Furthermore, some platforms may only operate during some months remaining idle for the rest of the year. This corresponds to the filled dot depicted in every plot of Figure 6.2 and the cost corresponds to the fixed cost without activity. Figures 6.2a and 6.2b may represent an outsourced platform. In the first case, the same unitary cost is paid for any quantity moved / stored, whereas, in the second case, there are contracted levels of activity for which fixed and different unitary costs have to be paid. The cost structure depicted in Figure 6.2c is more likely to occur for platforms managed by Unicer.



(a) (b) (c)

Figure 6.2: Examples of possible cost functions for platform activities. (a) initial fixed cost and linear variable costs; (b) piecewise fix cost activity costs; (c) initial fix cost and variable piecewise activity costs.

Both the definition of the operating platforms and the adjustment of platforms capacity are particularly important in the beverage industry due to the high seasonality of sales. The sales profile of these products presents peaks of demand at Easter, Christmas, and especially summer. On the other hand, production capacity remains almost constant throughout the year. This fact forces the industry to work on a make-to-stock basis as the capacity in the peak of sales is insufficient to match the demand, thus stressing the supply chain. Traditionally, during the low season (December to March) the company makes use of the idle platforms to store the seasonal stock, and adjusts the activity level of the remaining platforms to increase their operational capacity during summer.

Concerning the platforms, the tactical distribution plans should define for each platform at each month:

- If it should be operating or not;

- If operating, what is the optimal activity level;
- The utilization of each type of storage;
- The total number of pallets handled;
- The total amount of picking operations;
- The total number of containers shipped;
- The activity costs. Depending on the cost function associated with the platform it may be proportional to the number of products stored, pallets handled, picking operations and containers shipped.

On a simulation perspective it is important that the approach can model new scenarios for the opening and closing of production and distribution platforms.

6.2.2 Customers

Customers can be categorized into four sales groups: "Capilar", Retailers, Strategic and Exports. This distinction is important as the different customer groups present distinct relationships in the supply chain.

Customers belonging to the "Capilar" group are located in the regions of Oporto and Lisbon and the company supplies them with a door-to-door delivery system. They range from small to large restaurants, coffees shops, bakeries, bars and related establishments which serve food and beverages. Their orders are rather small, generally less than a pallet, but of a very diverse product basket requiring a complex picking operation.

Retailers are companies with special commercial contracts with Unicer that perform their own door-to-door delivery, especially in the regions outside Oporto and Lisbon. Orders of these clients are restricted in size to 33 pallets or 25.5 tons, which ensures the full use of a large truck. Furthermore, picking operations are not allowed to this type of customers.

Strategic clients consist of modern retail chains, wholesalers and chains of restaurants, hotels and other businesses dedicated to commercialize food and beverage. All have several stores spread all over the country. This sales group is the most heterogeneous, thus there is not a typical order, both in terms of quantities and product mix. Nevertheless, these clients are particularly important as stockouts in their stores have a huge impact in the brands visibility and recognition.

Finally, Exports clients are located outside Portugal. This segment represents over 40% of the total sales volumes with Spain and Angola as the main destinations, although Unicer sells its products to over 40 different countries. Most of the orders of these customers are large amounts of a single or two products and are shipped in containers.

In relation to the customers, tactical plans define for each one:

- The total quantity of each product sent from each platform;
- The total supply costs.

From a simulation point of view, it is interesting to study the effect of adding a new set of customers to the network or to assess the impact of changing the order policy of a given group of customers.

6.2.3 Transports

Production platforms, distribution platforms and customers form a three echelon distribution network. The first echelon is composed by the production sites and the second one by distribution platforms. Customers in the downstream echelon can be supplied by both upstream echelons. Figure 6.3 sketches an example to better understand the dynamics of Unicer's supply chain. This example considers two production platforms (PP_1 and PP_2), two distribution platforms (PD_1 and PD_2) and three customers (C_1, C_2 and C_3). In this representation of the supply chain nodes are locations and arcs represent the flow of finished products. Only the arcs used to supply client C_2 are shown. We distinguish between two type of flows: direct supply transportation movements which aim to supply client orders (depicted as solid arcs) and transportation movements intended to reallocate the stock among the facilities (depicted as dashed arcs and called inverse movements hereafter).

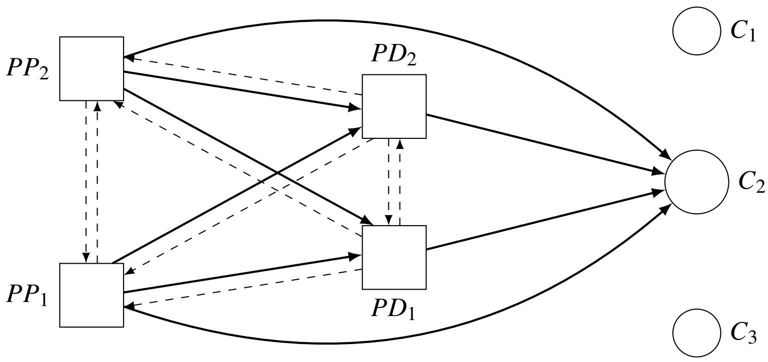


Figure 6.3: Schematic representation of the distribution network.

Usually, inverse movements are not considered in distribution planning as often supply chains are acyclic networks, where production platforms can only send its products either to a distribution platform or directly to a client and distribution platforms only deliver to clients. The situation present in this case study is far more complex as the finished products can flow among production platforms and distribution platforms, and distribution platforms can also send products back to production platforms. These inverse movements aim to deliver clients orders more efficiently. Before introducing the different supply strategies to the different customer groups, we first introduce some more general aspects of their supply.

Apart from "Capilar" customers which are supplying using the company's fleet, Unicer subcontracts the services of trucking companies or third party logistics providers (3PLs) to deliver its products. These companies use trucks able to carry up to 33 pallets or a maximum weight of about 25.5 tons. Hereafter the term truck will be used to refer to these large vehicles.

We use the term full pallet to refer a pallet loaded with a single product, in opposition to the term picking that is used to mention units (boxes) of products or pallets with several products. A picking operation (only possible at distribution platforms) takes full pallets which originally came from the production lines and coverts them into separate product units or rearranges them to form pallets with several products (mix pallets).

The supply of “Capilar” clients starts by sending full pallets from production platforms to distribution platforms. At the distribution platforms the small sized heterogeneous orders are picked and loaded into the company vehicles to which are assigned routes visiting several customers.

Retailers receive their large orders in trucks completely loaded with full pallets directly from a production platform. All the products in the order are commonly produced in just one production center from which the shipping is made. If so no further transport movements occur. Exceptionally, these orders can also include very small amounts of products produced in other platforms or have some picking units. In case this happens the following cases may occur. If some products are not produced at the platform, full pallets are sent from a distribution platform or from the production platform responsible for their production. In the case of the presence of picking in the order, these operations are performed at a distribution platform which sends the units of picking back to the production platform to deliver the order.

Due to the orders diversity coming from Strategic clients, their supply triggers the most complex movements in the supply chain. This is explained by different inventory management strategies adopted by these clients, which can be classified into: centralized and decentralized. Clients with a centralized strategy have their own central distribution platforms. Stores send their requirements to these depots which are responsible for sending orders to Unicer and for receiving the products to later send to the stores. This creates large orders which may be adequate to be supplied directly from production platforms, like in Retailers. However, if the product mix is too wide and no particular production platform produces the majority of the products or the amount of picking operations required is high, the orders are served from a distribution platform. In decentralized strategies the stores send orders directly to Unicer that is responsible for shipping products directly to them, resulting in much smaller orders. Therefore, these orders must be consolidated at a distribution platform to achieve an efficient use of the truck capacity, which afterwards performs a route over several stores.

Shipments to Exports customers can travel by land or by sea, but in both cases they are performed using containers. These deliveries are made from the production platforms and are mostly composed of full pallets. Exceptionally in the case of orders from these clients, the production platform can also perform limited picking operations.

Table 6.1 summarizes the different delivery modes for the different customer groups.

In resume, to fulfill customers’ orders a transportation movement has to initiate at a production platform afterwards many options are available. We can distinguish between three different general paths to fulfill a customer order: (1) these movements serve a given customer directly from the production platform; (2) these movements start by sending full pallets products to a distribution platform that are then partially picked and sent back to a production platform so they are, finally, shipped to the customer; (3) in these movements

Table 6.1: Main features of the different supply strategies of the customer groups.

	Customer Group			
	"Capilar"	**Retailers**	**Strategic**	**Exports**
Supply mode	door-to-door from distribution platform	directly from production platforms	Both from production and distribution platforms	directly from production platforms
Truck Utilization	LTL[2]	FTL[1]	FTL[1]/LTL[2]	Containers
Full Pallets	No	Yes	Yes/No	Yes
Picking	Product units	Rarely	Product units and mix pallets	At the production platform
Product Mix	Complex	Simple	Complex	Simple

[1]FTL - Full Truck Load, [2]LTL - Less than full Truck Load.

a distribution platform receives products from different production platforms, consolidates the orders and send them to the final customers.

With respect to transports we are particularly interested in capturing the flow among the different platforms, including the inverse movements. The tactical distribution plans created should define:

- The quantities to be sent through the different platforms (full pallets and picking);
- The total inter-platform costs.

Transportation management also rises important simulation questions, such as what happens in case of a more flexible distribution network achieved by adding more transportation lines linking the existing points (i.e. production platforms, distribution platforms and clients) as by definition of the problem not all possible transportation lines are used by the company. It is also important to understand the net effect of negotiating some present transportation tariffs.

6.2.4 Initial Situation

Up to the beginning of this project, the main purpose of the DB was to estimate the cost that the logistic department would incur in the following year. It was a common practice at the department to further divide the DB into the platforms budget and transports budget. Platforms budget estimated the expenditures related to the operation of the platforms, namely its activity costs, and the transportation budget projected the total costs in terms of inter-platform movements and client supply. Both these processes were performed manually with the help of spreadsheets and were strongly dependent on the managers' experience. These estimates were based on past data, since no plan for the distribution was created. Managers looked at total volumes per month as a detailed sales budget per client or client category was not available, and tried to detect discrepancies in comparison to the previous year. If such differences were found or the managers acknowledged some significant change in past assumptions, for example new important clients or products, or changes in the delivery mode for a set of customers, they would adapt last year's budget to reflect the new reality.

To validate the different strategical choices, the procedure embedded the same idea of looking to the past and compared it to the new scenario estimating the future costs.

The company was aware that its approach had some drawbacks:

1. It was heavily dependent on the experience of the managers assigned to the tasks.

2. A significant deal of manual effort was required to conduct the spreadsheet-based planning, leading to a hard time-consuming task. This limited the number of scenario analysis performed to evaluate the performance of different solutions or to assess the sensitivity of the plans to the input data.

3. The current plans did not ensure the capacity constraints identified at the platform level. Furthermore, as the plans did not detail the flows of finished products it was hard to identify and evaluate the bottleneck activities.

4. Finally, no optimization of costs used to take place, both when choosing the platforms activity level and when selecting the supply mode of clients. These decisions enclose trade-offs which were not being explicitly considered, such as increasing the level of activity of one platform compared to opening for a short period a platform previously inactive or defining the platform to supply a given client.

This project seeks to develop a new process to breach the deficiencies detected at the current methodology. Furthermore, this new process aims at increasing the detail of the decisions and the quality of information made available to decision makers to support better managerial decisions.

6.3. Distribution Planning in the Fast Moving Consumer Goods Industry

In the fast moving consumer goods industry the transportation process is usually managed by third party logistics providers (3PL). It is widely acknowledged that by outsourcing the transportation services, an increase in the truck utilization is achieved since the 3PL can consolidate several shipments from different clients. Moreover, the efficient utilization of the truck capacities results in a means of reducing freight costs (Stank and Goldsby, 2000).

Our case study is not an exception and the company has contracts with several 3PLs. Nevertheless, the transportation planning is completely under their control. Crainic and Laporte (1997) distinguish between three different levels of transportation planning:

- Strategic transportation planning that encompasses a long planning horizon and is responsible for defining the distribution network structure and for defining the customer service levels.

- Tactical transportation planning that still uses aggregate information to define the best affectation of resources. Hence, having the distribution network fixed, this level aims at introducing activities on the fixed facilities.

- Operational transportation planning that deals with the detailed planning of vehicle loads, routing and platform management. This planning level has to be very agile in adapting to a very dynamic setting.

The decision support system of this case study helps in making decisions both at the strategic and tactical levels. However, its underlying mathematical model has a clear tactical scope. Many of the mathematical models available in the literature for tactical transportation planning in a multi-echelon network also include production decisions as shown in the review by Mula et al. (2010). Although capturing more upstream decisions, they have a lower level of detail in the downstream echelons when compared to formulation presented in this paper. For example, an application in the chemical process industry is presented by Timpe and Kallrath (2000). Distribution and marketing decisions are planned simultaneously with batch and campaign production in a multi-plant production system. In their case study, the network is composed of 4 plants and 4 sales points which is significantly smaller than ours. We can find a distribution network similar to the one in this paper in the work of Bassett and Gardner (2010). In their study authors formulated two mathematical models, the first for a three echelon distribution network and the second adding an extra production echelon to form a four echelon network. Even though the structure of the network in this problem resembles ours, both the decision level of the model (strategic vs. tactic) and distance in the location of the facilities (long vs. short) are clearly different in comparison to the present case study.

To the best of our knowledge, the work by Kreipl and Pinedo (2004) is the one sharing more common features with ours. It also covers a tactical problem faced by a beverage company, namely Carlsberg A/S beerbrewer in Denmark, and the model considers a three echelon distribution network in which customers can be supplied from both upstream echelons. However, contrarily to our work, it defines production decisions and no inverse movements can occur. Furthermore, a single distribution platform composes the second echelon and inventory can not be kept at production platforms. Finally, the activity level of the platforms remains constant throughout the planning horizon.

For the aforementioned reasons, besides its relevance in the context of the case study, we consider that the mathematical formulation developed in this project is an important contribution. The main innovations rely on the consideration of the inverse movements in the network, the several activity levels of the platforms and the introduction of operational insight at a tactical level.

6.4. The Solution Approach

Our solution strategy to the tactical distribution planning problem relies on a heuristic solution based on the mathematical formulation of the problem. This is due to the large scale of the instances resulting from the case study which prohibited the use of a commercial solver on the complete mathematical formulation to achieve an optimal or quasi-optimal solution. In our first experiments we had problems loading the complete model into the solver for memory reasons or the solver took a prohibitively large amount of computational time to provide the first feasible solution. Below we start by describing in general lines the mixed integer programming (MIP) model for the tactical distribution planning problem, given in Appendix 6.A, and later we specify how we solve it heuristically to find good solutions to the problem.

6.4.1 A Model for the Tactical Distribution Planning Problem

The aforementioned three echelon distribution network can be described as a graph $G = (V, A)$. The vertex set is composed by the reunion of the set $\mathcal{P}$ of available production platforms (echelon 1), the set $\mathcal{D}$ of distribution platforms (echelon 2) and the set $\mathcal{C}$ of customers (echelon 3). The arc set explicitly defines the possible paths among vertices corresponding in practice to the transportation lines used by the company. These lines can be split into connections between platforms ($\mathcal{I}$) and the paths linking platforms to customers ($\mathcal{A}$).

The problem is to define the flow of finished products $\mathcal{K}$ from the production platforms to the customers over the planning horizon $\mathcal{T}$, in order to satisfy the customers demand at minimum cost. It can be understood as the integration of two subproblems: a transportation and an inventory management subproblem. The first subproblem resembles a multi-commodity, multi-echelon, multi-period transportation problem. In this scope, the model has to decide about the quantity shipped in each period from each platform to another and from each platform to the set of final customers. The second subproblem handles all the activities within a platform subject to capacity constraints. Hence, the model decides about handled pallets, units of picking and shipping containers, as well as it controls the inventory and allocation of products to different storage types. Of course, these subproblems are deeply intertwined because the transportation quantities decided will have a direct impact on the amount of products handled/stored.

Next we present the main decisions taken at each entity of the supply chain.

6.4.2 Platforms

Platforms can work on different activity levels ($\mathcal{N}$) that allow for different capacity restrictions. The definition of these activity levels is crucial to attain a realistic representation of the functioning of a platform. The different activity levels have a set of related costs and capacities that incorporate directly the possibility of hiring additional employees to picking or container loading operations, extra forklifts to increase pallet movement or even the creation of new working shifts. In the original case study formulation, we made further distinctions on the abilities of each platform, such as the ability of loading or not maritime containers. Nevertheless, for the sake of clarity we will not present these details.

Regarding platforms the main decision involves setting the platform activity level in each period. To do so we use a binary variable associated to each activity level, a_{it}^{n} which takes the value 1, if the (production or distribution) platform i is at activity level n in period t. We model the inactivity of a platform using an artificial activity level 0. At this level of activity a fixed cost can be incurred but other costs and capacities are set to zero.

To capture the stock level at the end of each period, we define s_{ikt}^{e} as the number of pallets of product k stored in platform i at storage of type e in period t.

6.4.3 Customers

In our mathematical formulation we grouped customers belonging to the set C according to transportation type into maritime ($\mathcal{M}$) or terrestrial ($\mathcal{R}$) to capture the requirements in terms of containers. Customers are divided by a second criteria into national ($\mathcal{F}$) or international ($\mathcal{E}$). We assume to know $\overline{D}_{jkbt}$ representing the demand of customer j for product k in the palletization type b in period t. Palletization types correspond to the previously defined full pallets ($b = 1$) and picking ($b = 2$). This immediately suggests the following decision variables $\overline{x}_{ijkbt}$ defined as the number of pallets of product k with palletization type b transported from platform i to customer j in period t to catch the supply decisions. However, these variables are insufficient to translate the reality into the model as they do not capture the real operational move in the tactical model, especially the inverse moves.

To overcome this, both the parameter $\overline{D}_{jkbt}$ and the decision variable $\overline{x}_{ijkbt}$ were refined. For this purpose we rely on the historical data for customer's demand orders. The demand of customer j for each product k in a given period t is split into *types of orders* where it is inserted. The orders are classified into types according to: order size (total order weight measured in tons), production platform producing the majority of products that appear in this order and the magnitude of this majority (a percentage of the total order weight). We define a finite set $q \in Q$ to classify orders according to their size and called it tonnages. Similarly, the magnitude of the majority of products belonging to a single production platform is also classified by intervals $p \in \mathcal{G}$ denoted as percentages. In Figure 6.4 an example of the demand conversion for a customer j for 10,000 pallets of Product k in March is given.

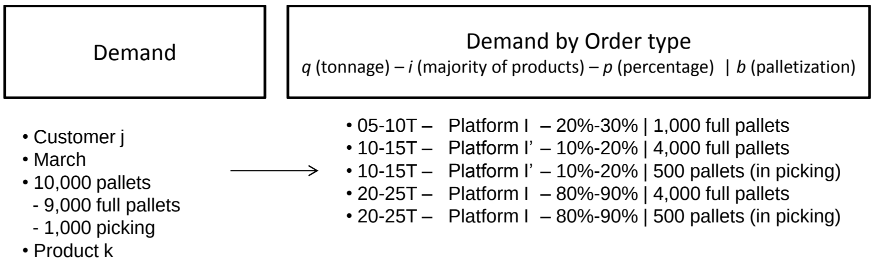


Figure 6.4: Example of the demand transformation.

Hence, we also need to detail $\overline{x}$ and introduce a new decision variable f:

x_{ijkbwt} number of pallets of product k with palletization b transported from platform i to customer j in period t to supply an order with the majority of products from production platform w

f_{jit}^{qp} binary variable which takes the value 1, if demand orders of customer j in period t with a majority of products from production platform i having a percentage p and a tonnage q are satisfied directly from i, or 0 if these orders are satisfied through a distribution platform from the set $\mathcal{D}$

Ensuring the link between x and f allows us to capture the operational behavior of

Table 6.2: Example of the selection of a supply strategy.

		$q \in \mathcal{Q}$				
		0-5t	5-10t	10-15t	15-20t	20-25t
	100-80%	$i \in \mathcal{D}$	$i \in \mathcal{D}$	$\mathbf{i} \in \mathcal{P}$	$\mathbf{i} \in \mathcal{P}$	$\mathbf{i} \in \mathcal{P}$
$p \in \mathcal{G}$	80-60%	$i \in \mathcal{D}$	$i \in \mathcal{D}$	$\mathbf{i} \in \mathcal{P}$	$\mathbf{i} \in \mathcal{P}$	$\mathbf{i} \in \mathcal{P}$
	60-40%	$i \in \mathcal{D}$	$i \in \mathcal{D}$	$i \in \mathcal{D}$	$i \in \mathcal{D}$	$i \in \mathcal{D}$

deliveries in our tactical model.

6.4.4 Transports

The last decision variable details the inter-platform movements. z_{iwkbt} represents the number of pallets of product k with palletization b transported from platform i to platform w in period t.

Having presented the main entities of our model, we can now describe the objective function and the main constraints.

Objective function

The objective function minimizes the total distribution costs over the whole planning horizon. These total costs correspond to: platform fixed activity costs, platform storage costs, pallet moving costs, picking moving costs, shipping container loading costs, transportation costs between platforms and transportation costs to deliver orders to customers.

Notice that all the transportation costs take into account the possibility of dealing with returnable products in routes that may be subject to such accounting. Furthermore, for FTL cost calculation one has to take into consideration the transportation mode used in a given transportation line from platform i to client j, i.e. either trucks or containers.

Constraints

Demand fulfillment constraints: All orders from customers should be delivered without any delays, i.e in same period as they have occurred. In other words, backlogging is not allowed.

Demand supply strategy constraints: For each client the model assigns demand of orders types (i, q, p) either to the corresponding production platform i or to the set of distribution platforms $\mathcal{D}$. To give coherence to the plans and comply with the planners reasoning we also enforce that as soon as a given order type (i, q, p) is fulfilled through the corresponding production platform i, then all demand orders having either a heavier tonnage q or a higher percentage p to be also fulfilled from i. Thus, this corresponds to the selection of a cut-off point both in terms of tonnage and percentage above which orders are consider properly to be supplied from a production platform.

As an example, Table 6.2 presents the impact on the distribution paths for a given customer in a given time period after fixing the supply strategy of order type $(i, 10-15t, 80-60\%)$ to be fulfilled directly from the production platform.

Inventory balance constraints (production platforms): The inventory balance constraints related to the production platforms characterize the movements that are allowed on these platforms. We distinguish between the inventory balance constraints for products produced in the respective production platform or not. We also have into consideration that, due to custom duties, at the production platforms picking can only be done to satisfy international customers. Finally, every unit of picking entering the platform is forced to leave the platform in the same period in order to satisfy national customers demand.

Inventory balance constraints (distribution platforms): The inventory balance constraints at distribution platforms show their flexibility. In fact, contrarily to the production platforms they can process any entering pallet into picking and dispatch in all palletization forms to the connected customers.

Activity levels constraints: Each platform at each period can only operate at one activity level.

Platform activity cost constraints: Platforms costs depend on the activity level selected, thus we have to link them.

Platform capacity constraints: Similar to the platform costs, the amount of activity performed in each platform depends on the decided activity level. Hence, we have to impose the corresponding limits to the number of shipping containers loaded, pallets stored, number of pallets moved and amount of picking performed in the platforms.

In this section (together with Appendix 6.A) the general mathematical formulation for our problem was introduced. It is important to clarify that we presented here the simplified version of the model on top of which the heuristic embedded in our optimization tool is built and, therefore, only focuses on the key modeling characteristics that can be replied in similar situations.

6.4.5 The Heuristics

The MIP model is not solvable for the large size instances of the case study. It suffers from its computational intractability especially because of the large number of demand and flow variables. Therefore, solving this problem requires the use of efficient solution approaches. Mathematical programming-based heuristics, also known as matheuristics (James and Almada-Lobo (2011); Ball (2011); Maniezzo et al. (2010)), are algorithms which seek the best trade-off between the effectiveness of exact approaches and the efficacy of metaheuristics. We based our solution strategy in MIP-based heuristics which are a class of matheuristics relying on the heuristic solution of the mathematical formulation.

The MIP-based heuristic designed has two phases: construction and improvement. Each phase uses a decomposition of the original mathematical formulation by time period. At each iteration of the construction phase we solve a single-period version of our MIP model. We start by solving the subproblem corresponding to the first time period, then we fix the solution of this period and set the final stock decisions as an input to the following subproblem, in this case the second period. We repeat this process and progressively move towards the end of the planning horizon. Once the solution of the last period is finished a feasible solution to the problem has been achieved.

The improvement phase of the heuristic seeks to increase the solution quality of the feasible solution at hand. Solving a single-period version of our model turns the final inventory decisions at each platform myopic since no further information on the demand is considered at the model. To overcome this and other potential limitations of the single-period version model, at each iteration of the improvement phase a two-period model is solved. These two time periods must be adjacent on time. Once again we start at the beginning of the planning horizon and re-optimize the solution corresponding to the first two time periods obtained in the construction phase. In next iteration, we fix the solution for the first time period and re-optimize time periods two and three together, and so forth until the final period in reached again. By keeping some overlap among successive iterations we guarantee a less myopic heuristic and potentially reduce the solution cost. Figure 6.5 depicts a visual interpretation of the heuristics.

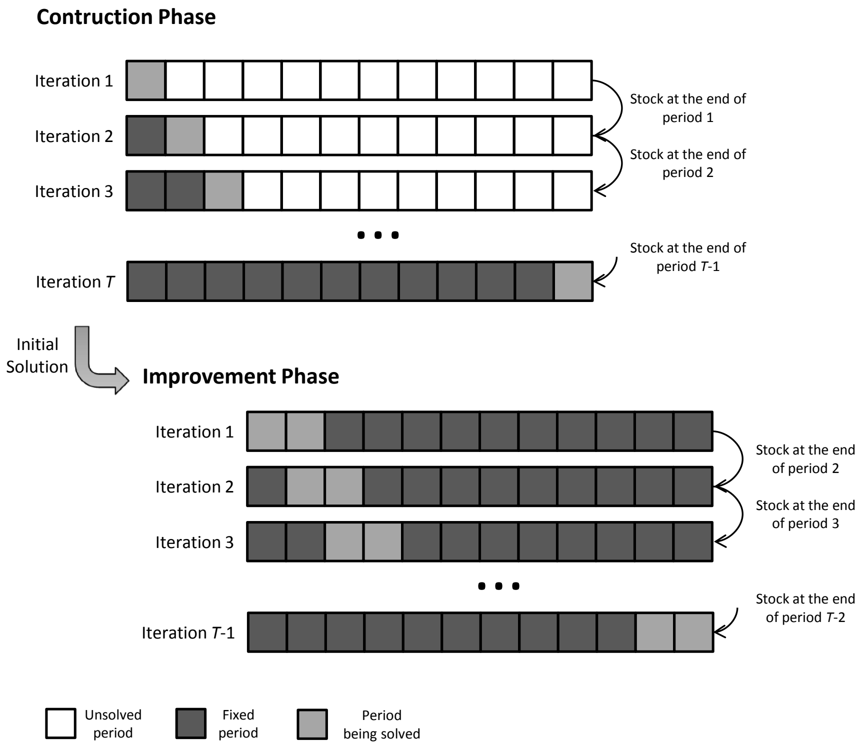


Figure 6.5: Solution strategy outline.

The use of MIP-base heuristics for a practical case study offers several advantages:

- With some expertise and compared to the traditional heuristic approaches, these heuristics are more easily implemented (i.e. require less parameters, less effort in tuning parameters and validating solutions). Moreover, they are rather problem-independent.

- They take advantage of the computational efficiency of modern commercial solvers.

- Despite being based on the model, they can cope with models extensions such as new constraints or even new decisions variables with limited or none changes in the heuristic.

- It has been proved in the literature that their performance often achieves quasi-optimal solution for a variety of different problems, which for the majority of the companies is more than enough.

However, it should be noticed that these heuristics rely on a decomposition of a larger problem, expecting that the resulting subproblems are easier to solve than the main one. When this is not the case, these heuristics can either lose their efficiency or fail to deliver a feasible solution to the problem. This is an important risk to manage.

Summing its pro and cons we believe that the use of MIP-based heuristics pays-off its use by appearing as a more flexible approach to cope with future changes of the problem.

6.5. Decision Support System

This section describes the decision support system that wraps around the optimization tool using the solution strategy of the core problem described in the previous section. In Figure 6.6 the relation between the building blocks of this decision support system (detailed in the next sections) and other software owned by the company are presented.

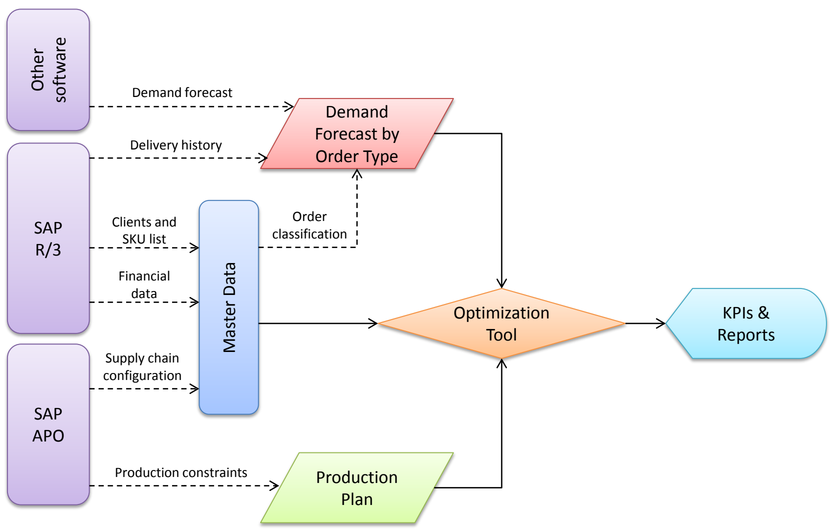


Figure 6.6: Framework of the decision support system.

This decision support system works through an on-line platform that can be accessed by any computer connected to the Internet. To develop this tool several programming languages were used. The browser interface is coded in JavaScript and the communication with the dedicated server is established through C#. The core optimization tool uses C++ to read the data, execute the solution strategy and output the solution. The mathematical models are solved with the help of a commercial mathematical programming solver. Finally, an add-in coded in Visual Basic for Applications (VBA) uses the raw output data to build user-friendly reports and extract information from the output solution.

6.5.1 Master Data

The Master Data input corresponds to a spreadsheet which organizes most of the parameters of the model. Hence, the user feeds a list of products with all the required characteristics, a list of clients and their information, a list of platforms and their abilities, the allowed activity levels for each platform and the possible transportation arcs with the respective costs. As shown in Figure 6.6 a major part of the information is gathered from Unicer's ERP system SAP R/3 and loaded into a spreadsheet for further validation and modification. SAP APO system can also be useful for loading the current configuration of the supply chain, namely the production and distribution platforms and the used transportation lines.

This considerable amount of data is permanently stored at the decision support system's server and it can be changed on an incremental fashion. Due to the strong interaction between these different data fields, this input is very prone to yield consistency errors as reported by other authors such as Farasyn et al. (2008). To circumvent this fact, we have implemented a VBA add-in that identifies all potential errors and missing data. Moreover, it points towards intelligent suggestions to rectify the incoherent and missing values. This add-in may for example indicate that a client has no transportation lines from any of the platforms and suggest some possible corrections. This type of information revealed itself to be crucial for the good usage of the tool and it saved considerable time to the analyst setting up the decision support system.

Finally, the use of spreadsheets to input the data required constitutes an easy and inexpensive way for defining new planning scenarios, one of the goals of the project. Inserting new products is a straightforward operation. Adding new customers or platforms, or even changing their location can be more time demanding due to the number of transportation lines affected, but the process is as simple as adding or editing lines in the master data template.

6.5.2 Demand forecast by order type

This block is responsible for creating the demand parameter. Two distinct inputs are needed to calculate such parameter. The demand forecast per client and per time period of the planning horizon and the deliveries history of a similar horizon in the past which is available from SAP R/3. For the DB demand forecast corresponds to monthly estimates for the next year and the delivery history of the current year. With the demand forecast we obtain total demand per client, product and period. The deliveries history allows us, with the necessary preprocessing calculations, to achieve the disaggregation detail of demand by order types based on past customer demand orders.

6.5.3 Production budget

The production plan that feeds the distribution planning defines products' production quantities at each production platform in each time period. The core problem consists of assigning and scheduling production lots in a multi-plant environment, where each plant has a set of filling lines that bottle and pack drinks. The work of Guimarães et al. (2012) proposes a method to create these plans. The output of this method or other solution approaches enter directly as parameter in the distribution planning.

6.5.4 Optimization tool

The optimization tool is responsible for feeding the information to our solution strategy, applying the heuristics and feeding forward the output to be decoded by the VBA add-in. Meanwhile, during the execution, it also sends feedback to the user on the current solution status. The optimization tool is triggered by the user through the on-line interface after setting up the input. Afterwards the heuristic starts with its linkage to the mixed-integer programming.

6.5.5 KPIs and Reports

After processing the raw output by the developed add-in, the decision makers have available graphical Key Performance Indicators (KPI) and extensive reports to perform their analysis and make informed decisions. We have implemented 7 different KPIs that cover the main areas of influence of the decision support system as follows:

- **KPI 1: Aggregated Costs: Platforms vs. Transportation** - This KPI reports the main partition of total costs between platform and transportation costs.

- **KPI 2: Platform Costs by Process** - This KPI disaggregates the overall platform costs by activity cost, storage costs, pallet moving cost, picking moving cost and shipping container loading cost.

- **KPI 3: Transportation Costs by Process** - This KPI decomposes the overall transportation costs in its main components: transportation costs to serve customers and between platforms.

- **KPI 4: Platform Utilization** - Based on the amount of pallets distributed by each platform, this KPI shows their relative importance to satisfy customers demand among all platforms and allows to identify the major bottlenecks in the distribution network.

- **KPI 5: Platform Costs by Platform** - This KPI separates the overall platform costs by platform. This allows the decision maker to have a quick view over the platforms yielding higher costs.

- **KPI 6: Transportation Costs by Type of Client** - Unlike KPI 3, the disaggregation of the transportation costs in this case is done by type of client.

- **KPI 7: Expeditions per Platform** - The last KPI has a more operational character. For each platform we assess the amount of pallets sent, which are split by those shipped to clients and to other platforms.

Beyond the general information that the KPIs display, the decision maker has the possibility of digging further into the results through the seven implemented reports. The following description gives a hint on the type of information made available within our decision support system.

- **Report 1: Total Costs** - This report gives the same information of KPI 1, 2 and 3 but, moreover, it is possible to see the period (monthly) evolution of these costs.

- **Report 2: Platform Costs by Process** - This report breaks down KPI 2 and, therefore, the process costs are split by platform and by month.

- **Report 3: Transportation Costs by Type of Client** - Similarly, in this report, KPI 6 is broken down per month.

- **Report 4: Transportation Costs by Client** - This report goes further in detail than Report 3 and details costs for each client independently and, moreover, the costs are also split by product transported.

- **Report 5: Movements Report** - This is the most important report that summarizes the activity of all platforms. For each platform we have the monthly evolution of the activity levels, stock, entries and deliveries. Moreover, utilization rates are also available for each resource. Figure 6.7 shows the details of this report for one platform.

- **Report 6: Activity Levels** - Activity levels for all platforms throughout the months are given in this report.

- **Report 7: Stock Report** - This report shows the amount of stock at each platform and in each type of storage for all months.

6.5.6 Interface

The final block of our decision support system is the interface. The on-line interface has three main areas: (1) data files upload, (2) tool execution and (3) history of solutions. Figures 6.8 and 6.9 show the graphical interface and the expected interactions with it, respectively. The left column in the graphical interface is responsible to manage the data files of the run, in the central column the user can launch new runs of the tool and has access to the log of the incumbent and previous runs. Finally, in the last column the solution files are available to be downloaded.

Platform	Type		Jan	Fev	Mar	Apr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Total
Activity															
Platform X	Level		2	2	2	2	2	2	2	2	2	2	2	2	
Stock															
Platform X	Drive In	Capacity													
Platform X	Drive In	Stock													
Platform X	Drive In	Utilization													
Platform X	Racks	Capacity	511	511	511	511	511	511	511	511	511	511	511	511	
Platform X	Racks	Stock	409	409	409	409	409	409	409	409	409	409	409	409	4 906
Platform X	Racks	Utilization	80%	80%	80%	80%	80%	80%	80%	80%	80%	80%	80%	80%	80%
Platform X	Floor	Capacity	20 000	20 000	20 000	20 000	20 000	20 000	20 000	20 000	20 000	20 000	20 000	20 000	
Platform X	Floor	Stock	6 041	6 041	6 041	6 041	6 882	8 215	11 743	14 131	16 522	15 930	16 818	14 739	129 144
Platform X	Floor	Stock (Pal)	16 697	16 818	16 705	16 303	18 732	22 300	31 212	37 196	43 230	41 801	44 058	38 955	344 007
Platform X	Floor	Utilization	30%	30%	30%	30%	34%	41%	59%	71%	83%	80%	84%	74%	54%
Platform X	Total	Total Space	6 450	6 450	6 450	6 450	7 290	8 624	12 152	14 540	16 931	16 339	17 227	15 147	134 049
Entries															
Platform X	PAL	Total Platforms	345	514	523	436	825	647	662	763	618	691	593	687	7 304
Platform X	Picking	Total Platforms	1 530	171 445	16 565	13 426	29 711	36 511	41 003	38 661	20 537	20 685	14 404	16 205	420 683
Platform X	Picking	Total Platforms (PAL)	19	2 126	187	144	426	527	595	587	230	242	167	163	5 413
Platform X	PAL	Total Production	48 195	42 173	51 172	56 921	72 139	66 392	69 500	73 885	66 921	64 512	54 150	50 070	716 029
Platform X	PAL	Initial Stock	13 444	17 105	17 227	17 114	16 712	19 141	22 708	31 621	37 604	43 639	42 210	44 467	322 993
Platform X	Total	Total Entries	62 004	61 918	69 108	74 614	90 102	86 708	93 466	106 855	105 374	109 083	97 120	95 387	1 051 740
Expeditions															
Platform X	PAL	Total Platforms	25 619	16 237	12 461	22 598	24 720	24 577	19 906	20 885	19 275	20 241	15 676	21 059	243 254
Platform X	PAL	Total Clients	13 395	21 793	23 911	26 349	36 516	30 867	36 686	40 758	29 657	34 623	30 311	27 158	352 024
Platform X	PAL	Capacity	70 000	70 000	70 000	70 000	70 000	70 000	70 000	70 000	70 000	70 000	70 000	70 000	
Platform X	PAL	Utilization	56%	54%	52%	70%	87%	79%	81%	88%	70%	78%	66%	69%	71%
Platform X	Picking	Total Platforms													
Platform X	Picking	Total Platforms (PAL)													
Platform X	Picking	Total Clients	447 903	680 541	1 392 261	873 367	804 146	736 656	568 342	916 367	1 117 337	911 346	512 528	624 854	9 585 650
Platform X	Picking	Total Clients (PAL)	5 898	8 487	17 338	10 673	10 578	10 142	7 572	11 839	14 677	12 870	6 677	8 033	124 784
Platform X	Picking	Capacity	1 200 000	1 200 000	1 200 000	1 200 000	1 200 000	1 200 000	1 200 000	1 200 000	1 200 000	1 200 000	1 200 000	1 200 000	14 400 000
Platform X	Picking	Utilization	37%	42%	115%	72%	65%	58%	44%	73%	91%	74%	42%	51%	64%
Platform X	Contentores	Total Clients	648	661	1 110	868	1 019	704	755	1 031	964	1 218	984	843	10 805
Platform X	Contentores	Total Clients (PAL)	16 294	16 678	28 197	22 017	25 901	17 860	19 087	26 156	24 522	30 924	25 055	21 408	274 100
Platform X	Contentores	Capacity	1 100	1 100	1 100	1 100	1 100	1 100	1 100	1 100	1 100	1 100	1 100	1 100	
Platform X	Contentores	Utilization	59%	60%	101%	79%	93%	64%	69%	94%	88%	111%	89%	77%	82%
Platform X	Total	Total Expeditions	44 912	46 517	53 710	59 620	71 813	65 586	64 164	73 483	63 609	67 734	52 664	56 251	720 063

Figure 6.7: Example of Report 5: Movements Report.

6.6. Validation

Together with Unicer's planning team we validated our approach in October 2012 during the creation of the DB. This was also a phase of intensive training of the decision support system future users. We helped them in defining the master data spreadsheet and ensuring a proper set up to the production plans and demand forecast required. In this master data the original 380 SKUs sold by the company were clustered into around 120 product clusters by merging products with similar physical and demand properties. Similarly, the customers were also clustered into client clusters by using the aforementioned customers categories and the district in which they are located. This reduces the original 19,000 clients to about 200 client clusters. Clustering guarantees the tractability of the MIP models in our solution strategy.

In the 2012 DB, a total of twenty one platforms were to be planned: nine production platforms, two major distribution platforms and the remaining locations are auxiliary distribution platforms. Over 200 transportation lines were available among platforms and more than 1300 lines connecting platforms and client clusters were defined as supply alternatives. Concerning activity levels, larger platforms could operate in three to four levels, while the auxiliary platforms usually presented only two possible levels of activity (active or inactive).

Our decision support system converted the SB per client into a detailed forecast by order typology considering a total of 180 possible types defined by the nine product platforms, five tonnages and four percentages. This process took 300 seconds since over 30,000 demand orders have to be analyzed in every month. At this point the main goal was to evaluate the plans created by the tool from the business perspective. This was done in several

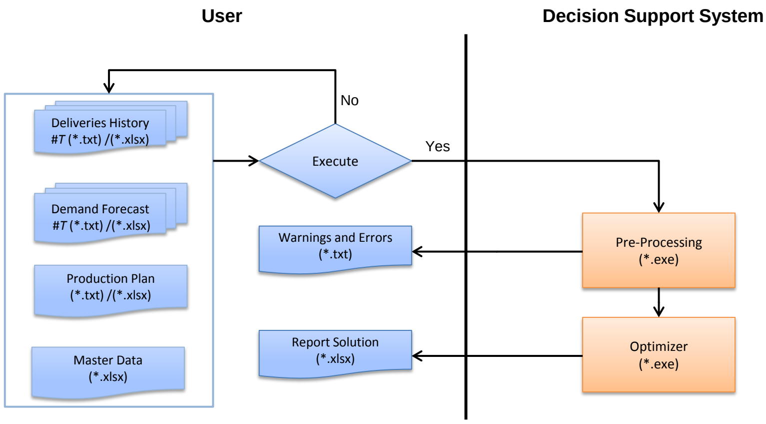


Figure 6.8: Interactions, inputs and outputs of the decision support system.

Table 6.3: Comparison of the MIP models.

	Variables (Binary)	Constraints	Non-Zeros	Avg. Solution Time (secs)
Complete Model	830,000 (30,000)	335,000	3,900,000	3600[1]
Single Period Model	66,000 (2,400)	26,000	300,000	30
Two Period Model	135,000 (4,900)	51,000	595,000	115

[1] maximum time limit

meetings and the costs analyzed had a similar order of magnitude to the estimates and business sense of the managers. Moreover, a deeper analysis of the plans established full confidence in the tool as these suggestions embedded important insights on how to operate the distribution process.

We have repeated this process at the beginning of 2013 to compare the yearly plan defined by Unicer to the year (without any use of the new tool) against the potential solution provided by the decision support system from a cost-efficiency point of view. The real plan defined by the operations over the year was evaluated according to the costs defined in the master data file and set the base total cost of the operation. We compared this plan with the ones obtained by:

- Solving the complete mathematical model formulation present in Appendix 6.A;
- The construction heuristic;
- The construction heuristic followed by the improvement heuristic.

We used an Intel i7-3630QM processor with 16,0 GB of RAM in our tests. All the approaches had their running times limited to one hour. Solving the complete model revealed to be inadequate as the solution provided was of very poor quality. The solver stopped its search at a very early stage after achieving the maximum running time (3600s) and the solution had no value in practice. Table 6.3 presents the average model size and solution

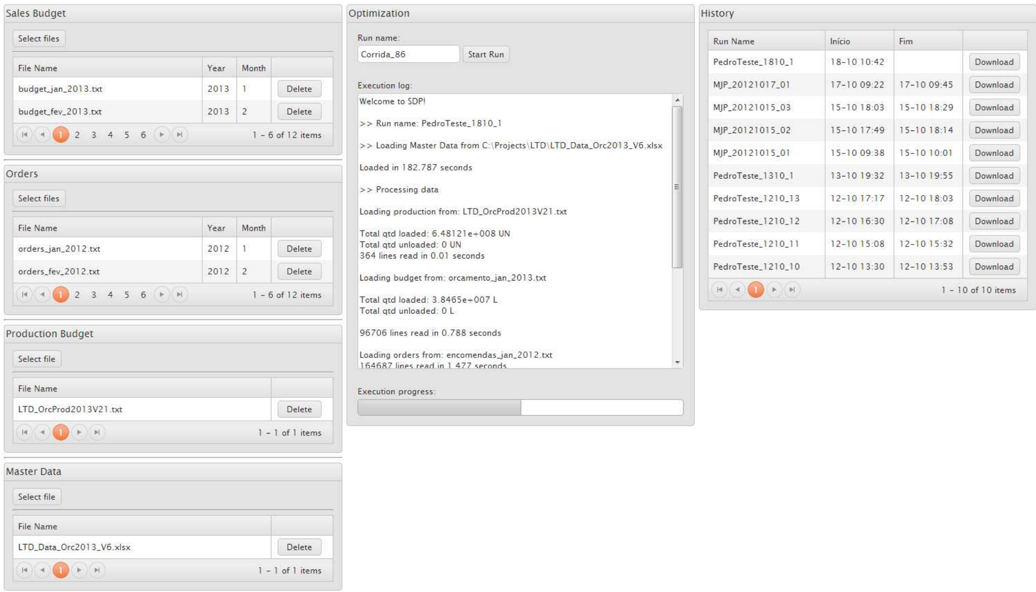


Figure 6.9: On-line interface of the decision support system.

time for the different models. The single period model corresponds to the model used in the construction phase and the two period model to the one used in the improvement phase.

Both versions of our heuristic procedures finished long before the time limit and delivered better solutions than the one by the company. The plan obtained by the constructive heuristic was obtained in 354s and reduced the cost of the company's plan by 5.25%. Applying the improving heuristic on top of the initial solution further reduced the total cost up to 6.8% below the original company's plan. These improvements in the total cost correspond to reductions of approximately 1.7M and 2.2M euros, respectively. Figures 6.10 and 6.11 depict in more detail the differences among the alternative plans and Figure 6.12 highlights the benefits of the improvement phase. Both heuristic procedures trade-off the costs categories involved differently from the company perspective: they increase the transportation costs among platforms and the storage costs in order to induce a significant reduction in the customers' supply cost. Moreover, Figure 6.11 shows that the cost difference between the plans was mostly explained by the behavior in the peak months of the Summer, when the capacity of the supply chain is more taken and, therefore, decisions have a higher impact which persists until the end of the year. Another important aspect to emphasize is that both plans obtained using the heuristics resulted in less transgression of both the storage and the movement capacities. We had to allow these violations in the model, otherwise it would be infeasible due to the low capacities defined for these resources in the master data file. These low capacities are explained by the fact that managers often underestimate them to ensure feasibility when performing the operational planning of the distribution as shown by their level of over utilization in the real plan.

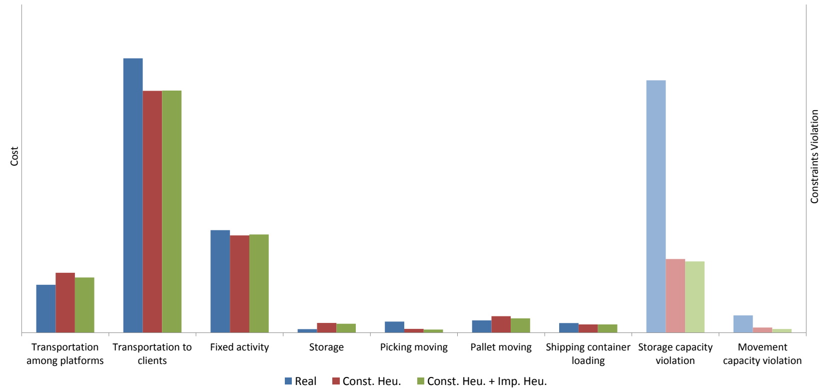


Figure 6.10: Comparison of the two heuristic plans and the original company plan in the different cost categories. Both the storage capacity violation and the movement capacity violation are depicted considering a secondary axis to highlight the differences between plans.

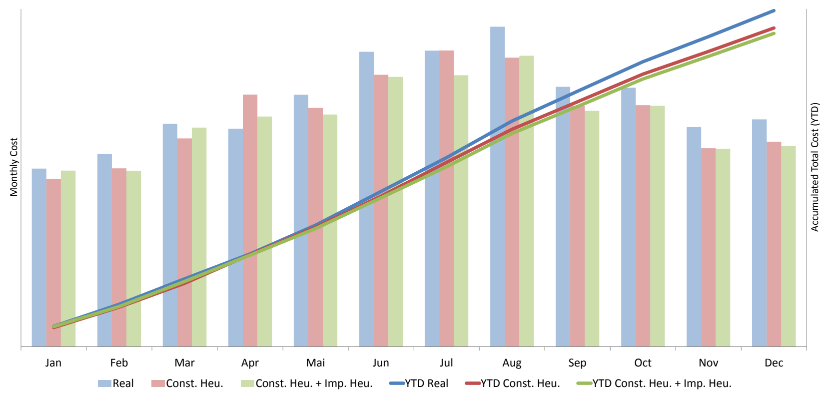
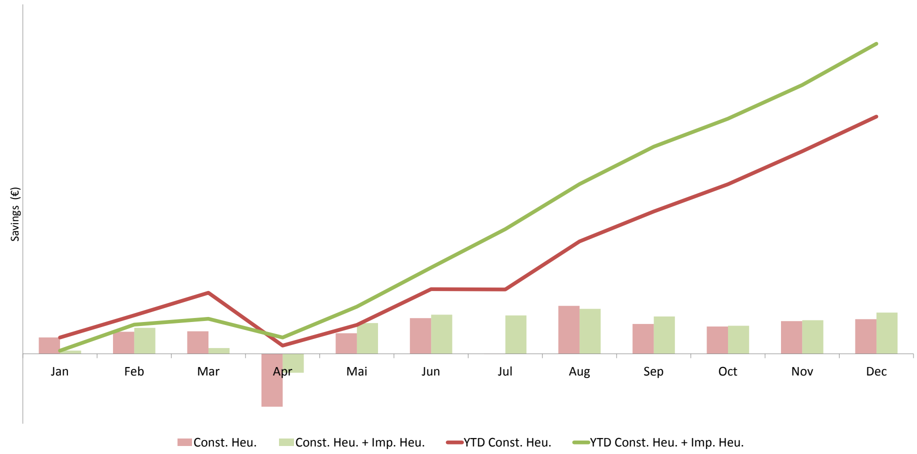


Figure 6.11: Year-to-date total cost and total cost in each month of the different plans.

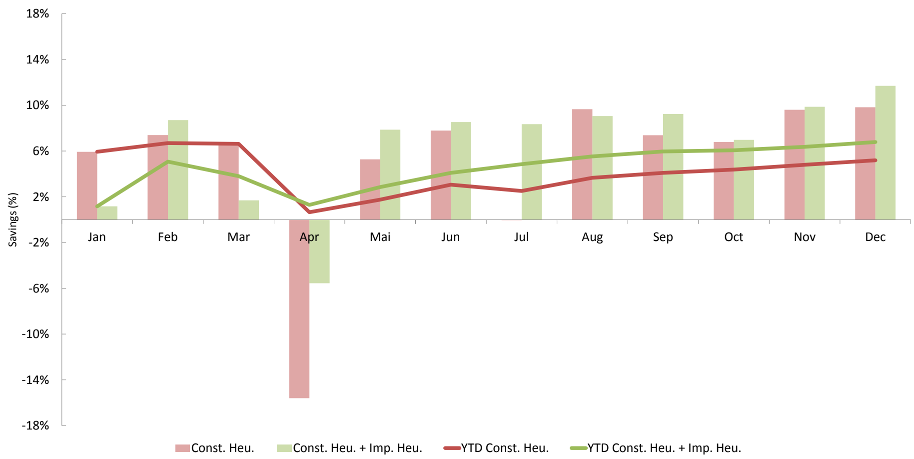
6.7. Conclusions and Future Work

In this paper we describe the real-world tactical distribution problem faced by Unicer, the major Portuguese beverage production company. The literature tackling tactical distribution problems with the features of this real-world application is sparse. However, we built on existing concepts from transportation and inventory a new mixed-integer programming model having as a key feature the insights of operational practice at a tactical level. The model is the basis of the solution strategy designed and implemented in a decision support system which is being used by the company. We scrutinized what is behind the main building blocks in order to increase the awareness of important factors which can give interested readers a basis to build something similar and shown its potential cost reduction impact.

Today Unicer uses OR in their tactical distribution decisions, which are now based on automated, detailed and accurate tactical distribution plans improving this planning step



(a)



(b)

Figure 6.12: Comparison of the heuristics in terms of savings versus the company plan. (a) absolute value; (b) percentage value.

at the company. The Decision Support System is being used to evaluate different logistic scenarios and to help in preparing the annual budget. The budget for 2013 was already validated using this new tool. The attained benefits of using the Decision Support System are evident not only by its potential of cost reduction but also by the easiness of simulation of multiple logistic scenarios and by the time saved in preparing the annual budget. Today Unicer can analyze virtually all possible distribution scenarios. This is of great value to a company that needs to challenge its practices very frequently. Moreover, the new planning methodology makes the process more transparent and the lead time to deliver the plans has decreased enormously. Analysts recognized that the Decision Support System has an underlying optimization model that retrieves solutions that were hard to grasp with the previous empirical methods, dotting the decision maker with information and perspectives that he did not have prior to the project.

Since the decision support system is built on a modular basis with very tunable blocks, it has the potential to be rolled-out to other facing similar real-world problems. Of course, the most straightforward step would be to adapt this approach to other beverage companies having similar distribution problems. However, other fast moving consumer goods companies seem also a natural extension as they also handle a vast product portfolio, many clients and a dynamic distribution network.

Future work could be devoted to integrate distribution and production tactical planning as they are intrinsically correlated. It is also interesting to extend the decision support tool to accommodate customer service levels and give more empowerment to the decision maker about the sense of the solution. Letting, for example, the possibility to adding some ceilings on key costs, such as transportation costs between platforms.

Bibliography

M. O. Ball. Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science*, 16(1):21 – 38, 2011.

M. Bassett and L. Gardner. Optimizing the design of global supply chains at Dow AgroSciences. *Computers & Chemical Engineering*, 34(2):254 – 265, 2010.

T. G. Crainic and G. Laporte. Planning models for freight transportation. *European Journal of Operational Research*, 97(3):409 – 438, 1997.

I. Farasyn, K. Perkoz, and W. Van de Velde. Spreadsheet models for inventory target setting at Procter & Gamble. *Interfaces*, 38(4):241–250, 2008.

A. Fearne and D. Hughes. Success factors in the fresh produce supply chain: insights from the UK. *Supply Chain Management: An International Journal*, 4(3):120–131, 1999.

L. Guimarães, D. Klabjan, and B. Almada-Lobo. Annual production budget in the beverage industry. *Engineering Applications of Artificial Intelligence*, 25(2):229 – 241, 2012.

Instituto Nacional de Estatítica. Estatísticas da produção industrial. http://www.ine.pt, 2011.

R. J. James and B. Almada-Lobo. Single and parallel machine capacitated lotsizing and scheduling: New iterative MIP-based neighborhood search heuristics. *Computers & Operations Research*, 38(12):1816 – 1825, 2011.

S. Kreipl and M. Pinedo. Planning and scheduling in supply chains: An overview of issues in practice. *Production and Operations Management*, 13(1):77–92, 2004.

V. Maniezzo, T. Stützle, and S. Voß. *Matheuristics*, volume 10 of *Annals of Information Systems*. Springer US, 2010.

J. Mula, D. Peidro, M. DíÂaz-Madroñero, and E. Vicens. Mathematical programming models for supply chain production and transport planning. *European Journal of Operational Research*, 204(3):377 – 390, 2010.

P. Stank and T. Goldsby. A framework for transportation decision making in an integrated supply chain. *Supply Chain Management: An International Journal*, 5(2):71–78, 2000.

C. H. Timpe and J. Kallrath. Optimal planning in large multi-site production networks. *European Journal of Operational Research*, 126(2):422 – 435, 2000.

Appendix 6.A The MIP Formulation

The parameters related to the platforms needed to formulate the problem are as follows:

Costs

F_i^n	fixed activity cost in platform $i \in \mathcal{P} \cup \mathcal{D}$ for activity level n
uH_i^n	unitary storage cost in platform $i \in \mathcal{P} \cup \mathcal{D}$ for activity level n
uM_i^n	unitary pallet moving cost in platform $i \in \mathcal{P} \cup \mathcal{D}$ for activity level n
uP_i^n	unitary picking moving cost in platform $i \in \mathcal{P} \cup \mathcal{D}$ for activity level n
uC_i^n	unitary shipping container loading cost in platform $i \in \mathcal{P} \cup \mathcal{D}$ for activity level n

Capacities

rH_{in}^e	storage capacity in number of stack positions in platform $i \in \mathcal{P} \cup \mathcal{D}$ for activity level n and storage type e
rM_i^n	capacity for pallet movements in platform $i \in \mathcal{P} \cup \mathcal{D}$ for activity level n
rP_i^n	capacity for picking movements in platform $i \in \mathcal{P} \cup \mathcal{D}$ for activity level n
rC_i^n	capacity for loading shipping containers in platform $i \in \mathcal{P} \cup \mathcal{D}$ for activity level n

To correctly assess the available storage capacity we further have to take into consideration the number of pallets that we can stack as this depends on the product and storage type. The parameter v_k^e sets the maximum number of pallet stacking levels of product k at storage type e. For production platforms the planned production quantities are also an input to the model. Let P_{jkt} define the quantity of product k produced at platform $j \in \mathcal{P}$ in period t.

The demand parameter used in the mathematical model is as follows:

D_{jkbit}^{qp}	demand of client j in period t for product k with palletization b within an order with the majority of products from $i \in \mathcal{P}$ having a percentage p and a tonnage q

The following parameters are necessary to capture the transportation costs among the different locations of the supply chain:

fT_{ij}	full truck load (FTL) cost for traveling from i to j
vT_{ij}	less than full truck load (LTL) cost for traveling from i to j

To describe the MIP model formulated for the tactical distribution problems the following additional parameters are required:

α_k	cost factor to account for the inverse logistics of product k
β_{ij}	cost factor to account for the inverse logistics of passing in arc $(i, j) \in \mathcal{A} \cup \mathcal{I}$
γ_k^R	land container capacity if only product k is transported
γ_k^M	maritime container capacity if only product k is transported
ζ_k	weight of each pallet of product k
δ_k	number of product units in a pallet of product k
μ_k	factor for converting pallets of product k into full pallets of product k

We also have to introduce auxiliary decision variables to linearize the piecewise cost functions at platforms which are defined as follows:

cH_{it}	storage cost in platform $i \in \mathcal{P} \cup \mathcal{D}$ in period t
cM_{it}	pallet moving cost in platform $i \in \mathcal{P} \cup \mathcal{D}$ in period t
cP_{it}	picking moving cost in platform $i \in \mathcal{P} \cup \mathcal{D}$ in period t
cC_{it}	shipping container loading cost in platform $i \in \mathcal{P} \cup \mathcal{D}$ in period t

The overall MIP model reads:

$$\begin{aligned} \min \sum_{i \in \mathcal{P} \cup \mathcal{D}, n, t} F_i^n a_{it}^n + \sum_{i \in \mathcal{P} \cup \mathcal{D}, t} (cH_{it} + cM_{it} + cP_{it} + cC_{it}) \\ \sum_{(i,j) \in \mathcal{I}, k, b, t} fT_{ij} / \gamma_k^R (1 + \alpha_k \beta_{ij}) z_{ijkbt} + \\ \sum_{(i,j) \in \mathcal{I}, k, b, t} vT_{ij} \delta_k \zeta_k (1 + \alpha_k \beta_{ij}) z_{ijkbt} + \\ \sum_{(i,j) \in \mathcal{A}: j \in \mathcal{R}, k, b, w, t} fT_{ij} / \gamma_k^R (1 + \alpha_k \beta_{ij}) x_{ijkbwt} + \\ \sum_{(i,j) \in \mathcal{A}: j \in \mathcal{M}, k, b, w, t} fT_{ij} / \gamma_k^M (1 + \alpha_k \beta_{ij}) x_{ijkbwt} + \\ \sum_{(i,j) \in \mathcal{A}, k, b, w, t} vT_{ij} \delta_k \zeta_k (1 + \alpha_k \beta_{ij}) x_{ijkbwt} \end{aligned} \tag{6.1}$$

The following auxiliary constraints to quantify the different platform costs that depend on the platform activity level. Constraints (6.2) quantify the storage cost of each platform in each period. This cost is incurred for every full pallet stored and depends on the platform activity level. Note that M denotes a big number.

$$cH_{it} \geq \sum_{k,e} uH_i^n \frac{s_{ikt}^e}{\mu_k} + M(a_{it}^n - 1) \quad \forall i \in \mathcal{P} \cup \mathcal{D}, n \in \mathcal{N}, t \in \mathcal{T} \tag{6.2}$$

Constraints (6.3) account for the full pallet moving costs. These costs have to consider all pallets handled either when receiving or sending products.

$$\begin{aligned} cM_{it} \geq \sum_{(i,j) \in \mathcal{A}, k, b, w} uM_i^n \frac{x_{ijkbwt}}{\mu_k} + \sum_{(i,j) \in \mathcal{I}, k, b} uM_i^n \frac{z_{ijkbt}}{\mu_k} + \sum_{(j,i) \in \mathcal{I}, k, b} uM_i^n \frac{z_{jikbt}}{\mu_k} + \\ M(a_{it}^n - 1) \quad \forall i \in \mathcal{P} \cup \mathcal{D}, n \in \mathcal{N}, t \in \mathcal{T} \end{aligned} \tag{6.3}$$

On the other hand, to obtain the picking costs (constraints (6.4)) it is only valued the amount of units of picking exiting the platform.

$$\begin{aligned} cP_{it} \geq \sum_{(i,j) \in \mathcal{A}, k, w} uP_i^n x_{ijk2wt} \varsigma_k + \sum_{(i,j) \in \mathcal{I}, k} uP_i^n z_{ijk2t} \varsigma_k - \sum_{(j,i) \in \mathcal{I}, k} uP_i^n z_{jik2t} \varsigma_k + \\ M(a_{it}^n - 1) \quad \forall i \in \mathcal{P} \cup \mathcal{D}, n \in \mathcal{N}, t \in \mathcal{T} \end{aligned} \tag{6.4}$$

The final cost constraints refers to the loading shipping containers cost that is obtained through constraints (6.5).

$$cC_{it} \geq \sum_{(i,j)\in\mathcal{A}:\, j\in\mathcal{M},k,b,w} uC_i^n \frac{x_{ijkbwt}}{\gamma_k^M} + M(a_{it}^n - 1) \quad \forall i \in \mathcal{P}\cup\mathcal{D}, n \in \mathcal{N}, t \in \mathcal{T} \tag{6.5}$$

Next we introduce demand fulfillment constraints. The first constraints of this group (6.6) state that the customer's demand has to be completely satisfied.

$$\sum_{(i,j)\in\mathcal{A}} x_{ijkbwt} = \sum_{q,p} D_{jkbwt}^{qp} \quad \forall j \in \mathcal{C}, k \in \mathcal{K}, b \in \mathcal{B}, w \in \mathcal{P}, t \in \mathcal{T} \tag{6.6}$$

Constraints (6.7) and (6.8) make use of decision variables f_{jit}^{qp} to assign demand order typologies to a certain distribution echelon (production or distribution platforms).

$$x_{ijkbit} = \sum_{q,p} D_{jkbit}^{qp} f_{jit}^{qp} \quad \forall (i,j) \in \mathcal{A} : i \in \mathcal{P}, j \in \mathcal{C}, k \in \mathcal{K}, b \in \mathcal{B}, t \in \mathcal{T} \tag{6.7}$$

$$\sum_{(i,j)\in\mathcal{A}:\, i\in\mathcal{D}} x_{ijkbwt} = \sum_{q,p} D_{jkbit}^{qp}(1 - f_{jit}^{qp}) \quad \forall j \in \mathcal{C}, k \in \mathcal{K}, b \in \mathcal{B}, w \in \mathcal{P}, t \in \mathcal{T} \tag{6.8}$$

Constraints (6.9) and (6.10) define the cut-off point for supplying orders from the production platforms.

$$f_{jit}^{q'p} - f_{jit}^{qp} \geq 0 \quad \forall j \in \mathcal{C}, i \in \mathcal{P}, q, q' \in \mathcal{Q} : q' \geq q, p \in \mathcal{G}, t \in \mathcal{T} \tag{6.9}$$

$$f_{jit}^{qp'} - f_{jit}^{qp} \geq 0 \quad \forall j \in \mathcal{C}, i \in \mathcal{P}, q \in \mathcal{Q}, p, p' \in \mathcal{G} : p' \geq p, t \in \mathcal{T} \tag{6.10}$$

The inventory balance constraints related to the production platforms are expressed in (6.11)-(6.13). Constraints (6.11) and (6.12) distinguish between the inventory balance constraints for products produced in the respective production platform or not (making use of set $\mathcal{K}^j$ that stands for the set of products belonging to platform $j \in \mathcal{P}$), respectively. These equations show that picking at the production platforms can only be done to satisfy international customers.

$$\begin{aligned} P_{jkt} + \sum_e s_{jk,t-1}^e + \sum_{(i,j)\in\mathcal{I}} z_{ijk1t} = \sum_e s_{jkt}^e + \sum_{(j,i)\in\mathcal{I}} z_{jikbt} + \\ \sum_{(j,i)\in\mathcal{A}:\, i\in\mathcal{N},k,j} x_{jik1jt} + \sum_{(j,i)\in\mathcal{A}:\, i\in\mathcal{E},k,b,j} x_{jikbjt} \quad \forall j \in \mathcal{P}, k \in \mathcal{K}^j, t \in \mathcal{T} \end{aligned} \tag{6.11}$$

$$\begin{aligned} \sum_e s_{jk,t-1}^e + \sum_{(i,j)\in\mathcal{I},b} z_{ijk1t} = \sum_e s_{jkt}^e + \sum_{(j,i)\in\mathcal{I}} z_{jikbt} + \\ \sum_{(j,i)\in\mathcal{A}:\, i\in\mathcal{N},k,j} x_{jik1jt} + \sum_{(j,i)\in\mathcal{A}:\, i\in\mathcal{E},k,b,j} x_{jikbjt} \quad \forall j \in \mathcal{P}, k \in \mathcal{K} \setminus \mathcal{K}^j, t \in \mathcal{T} \end{aligned} \tag{6.12}$$

Constraints (6.13) force every unit of picking entering the platform to leave the platform in the same period in order to satisfy national customers demand.

$$\sum_{(i,j)\in\mathcal{I}:i\in\mathcal{D}} z_{ijk2t} = \sum_{(j,i)\in\mathcal{A}:i\in\mathcal{N},j} x_{jik2jt} \quad \forall j\in\mathcal{P}, k\in\mathcal{K}, t\in\mathcal{T} \tag{6.13}$$

The inventory balance constraints at distribution platforms is given in (6.14).

$$\sum_e s^e_{jk,t-1} + \sum_{(i,j)\in\mathcal{I}:i\in\mathcal{D},b} z_{ijkbt} + \sum_{(i,j)\in\mathcal{I}:i\in\mathcal{P}} z_{ijk1t} =$$
$$\sum_e s^e_{jkt} + \sum_{(j,i)\in\mathcal{I}} z_{jikbt} + \sum_{(j,i)\in\mathcal{A},b,j} x_{jikbjt} \quad \forall j\in\mathcal{D}, k\in\mathcal{K}, t\in\mathcal{T} \tag{6.14}$$

The following capacity constraints limit the amount of activity performed in each platform depending on the decided activity level: shipping containers loaded (6.15), pallets stored (6.16), pallets moved (6.17), picking performed in production platforms (6.18) and picking performed in distribution platforms (6.19).

$$\sum_{(j,i)\in\mathcal{A}:i\in\mathcal{M}\cap\mathcal{Y}\mathcal{D},k,b,w} x_{jikbwt}/\gamma^M_k \le \sum_n rC_{jn}a^n_{jt} \quad \forall j\in\mathcal{P}\cup\mathcal{D}, t\in\mathcal{T} \tag{6.15}$$

$$\sum_{k\in\mathcal{K}} s^e_{jkt}/\mu_k/\nu^e_k \le \sum_n rH^e_{jn}a^n_{jt} \quad \forall j\in\mathcal{P}\cup\mathcal{D}, t\in\mathcal{T} \tag{6.16}$$

$$\sum_{(i,j)\in\mathcal{A},k,w} x_{jik1wt}/\mu_k + \sum_{(i,j)\in\mathcal{I},k} z_{jik1t}/\mu_k \le \sum_n rM_{jn}a^n_{jt} \quad \forall j\in\mathcal{P}\cup\mathcal{D}, t\in\mathcal{T} \tag{6.17}$$

$$\sum_{(i,j)\in\mathcal{A}:i\in\mathcal{E},k,w} x_{jik2wt}\varsigma_k \le \sum_n rP_{jn}a^n_{jt} \quad \forall j\in\mathcal{P}, t\in\mathcal{T} \tag{6.18}$$

$$\sum_{(i,j)\in\mathcal{A}:i\in\mathcal{N},k,w} x_{jik2wt}\varsigma_k + \sum_{(i,j)\in\mathcal{I},k} z_{jik2t}\varsigma_k - \sum_{(j,i)\in\mathcal{I},k} z_{ijk2t}\varsigma_k \le \sum_n rP_{jn}a^n_{jt} \quad \forall j\in\mathcal{D}, t\in\mathcal{T} \tag{6.19}$$

Finally, equations (6.20) ensure that each platform only operates at a single activity level in each period.

$$\sum_n a^n_{jt} = 1 \quad \forall j\in\mathcal{P}\cup\mathcal{D}, t\in\mathcal{T} \tag{6.20}$$

CHAPTER 7

Conclusions and future work

This thesis tackles the major production planning challenges faced by the beverage industry. On the short-medium term production planning we investigate how to approach the sequence dependent changeover times and costs observed in the production lines. Recently, these setups are becoming increasingly more important as the beverage companies expand their product portfolios leading to a substantial reduction of the production lines operational efficiency. Chapters 2, 3 and 4 are dedicated to the short-medium term planning proposing new formulations and solution approaches. In Chapter 2 we review and compare the computational efficiency of the existing mathematical formulations for the capacitated lot sizing and scheduling problem with sequencing decisions (CLSD) and propose a very efficient new formulation. Chapter 3 is dedicated to an innovative solution approach to the CLSD and in Chapter 4 the focus is on how to adapt CLSD mathematical formulations to be used on a rolling planning basis. In the context of the long term production planning, the main challenges consist in the strong demand seasonality and in the coordination of the multi-plant supply chain. Due to the influence of distribution decisions in multi-plant lot sizing models we also cover them at this level ensuring a comprehensive view of the supply chain. Chapter 5 explores the coordination of geographical disperse production plants to reduce the operational costs and in Chapter 6 we address the client's supply strategy and its effect on the supply chain configuration and demand allocation to plants. We highlight that most or at least part of the challenges faced by the beverage industry are common to other fast moving consumer good industries, thus the formulations and solutions approaches developed in this thesis can be applied to other problems with a limited effort. Next we overview the main contributions and pinpoint ongoing and future research.

We start by focusing on the short-medium term planning challenges. First, an in depth study of the several existing modeling techniques to incorporate sequencing decisions in the original capacitated lot sing and scheduling problem is conducted. From this study emerged a new two-dimensional framework used to classify the different modeling approaches to the CLSD by grouping them into classes. Meanwhile, we also introduce a new formulation that uses commodity flow based constraints to eliminated disconnect subtours. Extensive computational experiments evaluate all the formulations and the results present an evaluation of the pros and cons of the different modeling techniques, comparing models which, to the best of our knowledge, had never been compared. This allowed us to indicate the probably most efficient models in several contexts, namely in the presence or not of setups disobeying the triangle inequality.

Second, on the solution approach front for the CLSD, we design a new MIP-based construction heuristic using a hybrid formulation and explore in an MIP-based improvement heuristic, novel ideas to use column generation for local search within lot sizing problems with sequencing decisions. The hybrid formulation combines a new MIP model, the first to capture non-triangular settings based on the selection of a single sequence from a pre-determined set in each time period, with a compact model to create a single model that trades-off accuracy and computational complexity.

More than proposing valid formulations and solution approaches to the short-medium term production planning in the beverage industry, we believe to have enriched the line of research dedicated to include sequencing decisions in lot sizing models. The results of the aforementioned contributions are two research papers:

- L. Guimarães, D. Klabjan, and B. Almada-Lobo. Modeling lotsizing and scheduling problems with sequence dependent setups. *Submmited to the European Journal of Operational Research*, 2013.
- L. Guimarães, D. Klabjan, and B. Almada-Lobo. Pricing, relaxing and fixing under lot sizing and scheduling. *European Journal of Operational Research*, 2013. Available in Press. http://dx.doi.org/10.1016/j.ejor.2013.04.030

Inspired by the current case study practice, we then explore how mathematical formulations to the CLSD can be adapted to be used on a rolling horizon planning. The main idea is that the planning horizon can be decomposed in two parts: a first set of initial periods in which production sequences are explicitly considered to obtain detailed schedules; and a second set composed by the final periods, where a rough plan is enough for giving an estimation of future costs and potential capacity shortages. Based on this decomposition, several simplification modeling alternatives present in the literature are reviewed and a new formulation is proposed. This formulation includes the setup loss in the future periods based on the loss witnessed in the detailed part of the horizon . Building on the idea that the solutions of the approximate models for the undetailed horizon can enclose important information, we develop an iterative method to improve the accuracy of the approximate parameters used in the simplified formulations. The method is modular and can be used to refine the estimation of distinct parameters arising in the different models. From this study resulted a research paper:

- L. Guimarães, D. Klabjan, and B. Almada-Lobo. Rolling horizon formulations for short-medium term production planning. *Working paper*, 2013.

Regarding the long term planning, we start by introducing a new mathematical formulation that assigns and schedules production lots in a multi-plant environment, where each plant has a set of filling lines. This formulation also considers the distribution decisions related to the transfers of final products between the plants. To allow the creation of good quality solutions in reasonable computational time, we develop a novel hybrid algorithm that explores sensitivity analysis to guide a partial neighborhood search embedded in a Variable Neighborhood Search scheme. We show the applicability of the new algorithm by solving real-world instances from our case study and the results proved that the current business practice could be significantly improved with an estimated cost reduction of up to 40% of the total cost, about 1.2M euros in 2011. We next study the impact of downstream distribution decisions in the configuration of the supply chain and consequently, on the allocation of the demand to the plants. Driven by the discussion of the design, development and implementation of an operations research (OR)-based approach to tackle distribution planning, we propose an innovative model to grasp the operational complexity in a higher planning level and a mathematical programming-based heuristic to achieve good quality solutions in acceptable running times. We show how to engage the main issues of implementing OR-based solutions in practice and report a potential cost reduction of up to 2M euros per year compared to the company's plan. The following two research papers report our findings:

- L. Guimarães, D. Klabjan, and B. Almada-Lobo. Annual production budget in the beverage industry. *Engineering Applications of Artificial Intelligence*, 25(2):229 – 241, 2012. http://dx.doi.org/10.1016/j.engappai.2011.05.011

- L. Guimarães, P. Amorim, F. Sperandio, F. Moreira and B. Almada-Lobo. Annual Distribution Budget in the Beverage Industry: a case study. *Submmited to Interfaces*, 2013. (under revision)

Additionally, two decision support systems were implemented and are currently in function in the case study company as a result of the two works mentioned above. The first - LTP, is used by managers to build long term production plans allowing to assess the production capacity based on periodic demand forecast revisions for the following months and to evaluate different scenarios for the production system configuration. Managers have been using LTP independently since 2011 as part of their planning system. Moreover, this heavily utilization has already motivated additional improvements, namely its extension to include distribution decisions to the main company warehouses. During 2012, it helped the managers to validate a significant supply chain re-configuration by evaluating from an operational point of view different strategic changes. The second decision support system - LTD, looks at the distribution process by considering all the echelons of the supply chain. It generates plans determining the flow of products between plants, distribution centers and clients. LTD validates the production plans by considering additional distribution details and also allows manager to realize the best operational level of the logistics sites over the planning horizon.

Ongoing research in the short-medium term production planning considers that in some other beverage companies the production process bottleneck may shift between stages I and II. Under these conditions, creating feasible production plans is extremely challenging, especially in the beer production process where long lead times exist in stage I. It is then vital to correctly synchronize the two stages. Even for our case study, in which the bottleneck is known to be on stage II, we aim to investigate if the total costs can benefit from a simultaneous planning of beer production and filling. We want to develop new mathematical formulations for the integration of these two production stages and explore the natural partitions of the problem in the two stages to derive efficient mathematical programming-based heuristics.

Future research on the CLSD can explore the literature related to the Asymmetric Traveling Salesman Problem to create more efficient models to the problem. Many classes of valid inequalities and reformulations are known to this widely studied problem and their interpretation at the light of the CLSD can originate substantial improvements. This may also lead to the creation of an efficient Branch-and-Cut or Branch-and-Price algorithm to this problem. Investigating this relationship has already originated some of the most recently relevant models, however we believe that there are still numerous opportunities to be explored.

Rolling models to solve real-world CLSD problems also constitute an interesting area for future research, especially due to their inherit applicability. Rolling horizon planning can be used to reduce the complexity of the problems at hand and also to accommodate uncertainty in future periods. Thus, it may be an important vehicle to implement lot sizing and scheduling in practice.

With respect to the long term production planning, it is important to observe a substantial difference when managers face these problems in comparison to short-medium term ones. While in short term planning managers can be satisfied with a single solution from an optimization algorithm as long as it makes sense for the business practice, long term decisions are mostly conducted on a what-if analysis basis. This is mainly motivated by two distinct reasons. On the one hand, models may not be covering all the possible decisions, such as opening or closing facilities, investing in the flexibility of the current production equipments, or moving production lines to new locations. In this sense models need to be extended creating a potential area for future research. On the other hand, long term planning data is usually less reliable, and therefore, future research in long term production planning problems would most likely benefit from integrating risk management analysis.

Naturally, the extension and continuous improvement of the decision support systems developed are a subject of future attention as well. Particularly worthwhile investigating is how to improve LTP-LTD integration. We would like to highlight that most often practice and research collide in both timing and goals. The application of models in practice may not guarantee enough material to be relevant to the scientific community and a scientific outcome can be misaligned with the business needs. However, as engineers, we are highly motivated by solving real problems and nothing is more rewarding than seeing analytics in practice.